

**YUMARC формат, радно окружење и команде
Библиотечког софтверског система БИСИС**

**YUMARC Format, Working Environment, and
Commands of the Library Software System
BISIS**

Едиција
Рачунарске науке

Уредници
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

025.31:004.4
004.42 BISIS

ПУПОВАЦ, Биљана

YUMARC формат, радно окружење и команде Библиотечког софтверског система БИСИС / Биљана Пуповац. – Нови Сад : Група за информационе технологије, 2004 (Нови Сад : АБМ Економик). – X, 112 стр. : илустр. ; 24 см. – (Едиција Рачунарске науке)

Тираж 300. – Библиографија.

ISBN 86-7444-008-8

а) Библиотечка каталогизација – Yumarc б) Апликативни програм "БИСИС"

Биљана Пуповац

**YUMARC формат, радно окружење и команде
Библиотечког софтверског система БИСИС**

Нови Сад 2004.

Едиција
Рачунарске науке

Назив публикације
YUMARC формат, радно окружење и команде
Библиотечког софтверског система БИСИС

Аутор
мр Биљана Пуповац, истраживач-сарадник
Природно-математички факултет, Нови Сад

Рецензенти
др Душан Сурла, ред. проф. Природно-математички факултет, Нови Сад
др Зора Коњовић, ред. проф. Факултет техничких наука, Нови Сад

Корице
Мирјана Исаков

Штампа
АБМ Економик

Тираж
300

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.

Foreword

This monograph belongs to the field of library information systems design. Modeling and implementation of the YUMARC format, librarian working environment and commands of the Library Information System BISIS ver. 3 are presented. Unified Modeling Language (UML) is used for modeling, while Java platform is used for implementation.

The monograph contains the following chapters:

1. Library Software System
2. YUMARC Format Modeling and Implementation
3. Librarian Working Environment Modeling and Implementation
4. Commands of the Library Software System

In the first chapter, the aims of the library network, which were used as a base for the realization of the BISIS software system, are described. Apart from that, the short overview of the BISIS software system development is given. In addition, the Project *Digital Library of the University of Novi Sad* is described and the results of the research in applying XML technology in BISIS system development are presented.

The second chapter presents class diagram of the YUMARC format. Also, the user interface for fields updating is described. The user interface for other concepts of the YUMARC format updating is given in the Appendix 2.1.

In the third chapter, the class diagram of the librarian working environment is presented. The user interface for publication type and publication type subfields updating is described. The user interface for other concepts of the librarian working environment updating is described in the Appendix 3.1.

In the fourth chapter, the object-oriented specification of the commands of the Library Software System BISIS ver.3 is presented. The usage of these commands is illustrated with examples in the Appendix 4.1.

I would like to express my gratitude to reviewers for the valuable comments and suggestions provided during the period of writing this monograph. Also, I would like to thank Tatjana Tosic, M.Sc. and Vera J. Sojadinovic, B.Sc. for careful and detailed reading of the text, which contributed to its better organization and coherence. Finally, I would like to express my gratitude to the European Commission, which provided financial support for printing this publication through Tempus project JEP 16114: *Web Based Interuniversity Library Network*.

Novi Sad, 2004

Author

Предговор

Монографија припада области пројектовања библиотечких информационих система. Приказано је моделирање и имплементација YUMARC формата, радног окружења библиотекара и команди библиотечког софтверског система БИСИС вер. 3. Моделирање је извршено у обједињеном језику моделирања (Unified Modeling Language – UML), а имплементација у Java окружењу.

Монографија садржи следећа поглавља:

1. Библиотечки софтверски систем,
2. Моделирање и имплементација YUMARC формата,
3. Моделирање и имплементација радног окружења библиотекара,
4. Команде библиотечког софтверског система.

У првом поглављу укратко су описани задаци библиотечке мреже на основу којих је реализован софтверски систем БИСИС. Сем тога, дат је кратак преглед развоја система БИСИС, описан пројекат Дигитална библиотека Универзитета у Новом Саду и наведени резултати истраживања у области примене XML технологије у развоју система БИСИС.

Дијаграм класа YUMARC формата приказан је у другом поглављу. Осим тога, приказан је прозор за ажурирање поља, док су у Прилогу 2.1 описани прозори за ажурирање осталих концепата YUMARC формата.

У трећем поглављу приказан је дијаграм класа радног окружења библиотекара. Описани су прозори за ажурирање типа публикације и потпоља типа публикације. У Прилогу 3.1 описани су прозори за ажурирање осталих концепата радног окружења библиотекара.

У четвртом поглављу приказана је објектно-оријентисана спецификација команди библиотечког софтверског система БИСИС вер. 3. У Прилогу 4.1 приказани су илустративни примери коришћења тих команди.

Захваљујем се рецензентима на корисним примедбама и сугестијама у току писања ове монографије. Такође, захваљујем се мр Татјани Тошић и Вери Ј. Стојадиновић које су детаљно прочитале текст и тиме допринеле да текст буде прегледнији и разумљивији. На крају, захваљујем се Европској комисији која је преко ТЕМПУС пројекта ЈЕР 16114 обезбедила трошкове за штампање ове монографије.

Нови Сад, 2004.

Аутор

Садржај

Foreword	v
Предговор	vii
1. Библиотечки софтверски систем.....	1
1.1 Задаци библиотечке мреже	1
1.2 Развој софтверског система БИСИС	3
1.2.1 БИСИС верзија 1	4
1.2.2. БИСИС верзија 2 и верзија 3	4
1.3. Дигитална библиотека Универзитета у Новом Саду	5
1.4. XML технологије и развој система БИСИС	7
2. Моделирање и имплементација YUMARC формата.....	9
2.1 Дијаграм класа YUMARC формата	13
2.2 Ажурирање YUMARC формата	16
2.2.1 Прозор за ажурирање поља	17
Прилог 2.1 Прозори за ажурирање YUMARC формата	21
3. Моделирање и имплементација радног окружења библиотекара.....	33
3.1 Дијаграм класа радног окружења библиотекара	34
3.2 Ажурирање радног окружења библиотекара	36
3.2.1 Прозор за ажурирање типа публикације	38
3.2.2 Прозор за ажурирање потпоља типа публикације	39
Прилог 3.1 Прозори за ажурирање радног окружења библиотекара	41
4. Команде библиотечког софтверског система.....	59
4.1 Дијаграм случајева коришћења команди	63
4.2 Дијаграм класа команди	67
4.2.1 Дијаграм класа команде <i>delete</i>	68
4.2.2 Дијаграм класа команде <i>display</i>	70
4.2.3 Дијаграм класа команде <i>new</i>	72
4.3 Динамички модел команди	74
4.3.1 Динамички модел команде <i>delete</i>	75
4.3.2 Динамички модел команде <i>display</i>	81
4.3.3 Динамички модел команде <i>new</i>	92
Прилог 4.1 Коришћење команди у апликацији за обраду библиографске грађе	101
Литература.....	109

Библиотечки софтверски систем

У Републици Србији 1991. године усвојена је *Концепција развоја Система научних и технолошких информација у Републици Србији* (СНТИС) [Laz91]. Овим документом предложено је да се развије више логичких мрежа за потребе СНТИС-а. Једна од тих мрежа је и библиотечка мрежа (БМ). Концепција БМ СНТИС дата је у пројекту [Sur92]. Министарство за науку и технологију Републике Србије у оквиру СНТИС-а финансирало је пројекат *Систем за формирање и претраживање информација* који се састојао од пет потпројеката, од којих су се два односила на библиотечку мрежу. Добијени резултати истраживања на овом пројекту приказани су у монографији [Laz96]. Пројекат је реализован у периоду од 1993. до 1996. године.

Као резултат истраживања на два потпројекта наведеног пројекта добијен је софтверски систем за библиотечко пословање, назван БИСИС вер. 1. Развој система БИСИС настављен је и после 1996. године. Резултат тог развоја су БИСИС вер. 2 и БИСИС вер. 3. Преглед развоја система БИСИС вер. 3 дат је у монографији [Сур04а], а детаљно упутство за коришћење овог система дато је у књизи [Сур03]. Део развоја система БИСИС реализује се и у оквиру ТЕМПУС пројекта под називом *Web Based Interuniversity Library Network* (JEP 16114).

У наставку овог поглавља укратко су описани задаци библиотечке мреже на основу којих је реализован софтверски систем БИСИС. Сем тога, дат је кратак преглед развоја система БИСИС, описан пројекат Дигитална библиотека Универзитета у Новом Саду и наведени резултати истраживања у области примене XML технологије у развоју система БИСИС.

1.1 Задаци библиотечке мреже

У пројекту [Sur92] дата је следећа дефиниција појма библиотечка мрежа: *Библиотечка мрежа, као инфраструктурни део СНТИС-а, јесте систем међусобно повезаних библиотечко информacionих и документацио-*

них институција са значајним фондовима научне и стручне литературе које, процесом узајамног рада, а уз подршку информационог сервиса и изграђене комуникационе мреже, обезбеђује корисницима, на непосредан, брз и лак начин библиотечке информације, као и примарне документе.

У том пројекту дефинисани су и задаци БМ СНТИС-а који се остварују на локалном нивоу, као и задаци који се остварују на нивоу информационог сервиса.

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

1. Локална подршка узајамне каталогизације,
2. *On-line* претраживање локалних база,
3. ОПАС,
4. Набавка,
5. Циркулација,
6. Библиотечка статистика,
7. Контрола комплетности серијских публикација,
8. Локална подршка међубиблиотечке позајмице,
9. Локално издавање секундарних публикација,
10. Локални сегмент координиране набавке.

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

1. Узајамна каталогизација примарних докумената,
2. Одржавање централних каталога,
3. *On-line* претраживање,
4. Међубиблиотечка позајмица,
5. Координација набавке примарних докумената,
6. Издавање секундарних публикација,
7. Дисеминација информација,
8. Инфо-сервис,
9. Приступ иностраним информационим изворима.

Развој система БИСИС базиран је на пројекту [Sur92] са циљем да се реализују наведени задаци БМ СНТИС. Спецификација информационих захтева система БИСИС остварена је у сарадњи са Библиотеком Матице српске из Новог Сада, Универзитетском библиотеком "Светозар Марковић" из Београда и Народном библиотеком Србије из Београда.

1.2 Развој софтверског система БИСИС

У монографији [Laz96] 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,
- Циркулација библиографских докумената.

Дефинисање стандарда за обраду и претраживање омогућава опис структуре појединих стандарда и њихово представљање у бази података преко којих ће се обављати основне функције БИСИС-а. 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 података библиографске грађе у систему БИСИС, усвојен је YUMARC формат базиран на UNIMARC формату [Uni94]. Дефинисање стандарда за одржавање библиографске грађе треба да омогући опис структуре YUMARC 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, библиотечких извештаја и каталога докумената, као и одговора на корисничке упите који су резултат претраживања базе података библиографске грађе на основу библиотечких 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 библиотечких извештаја и каталога о документима. Форми-

рање извештаја и каталога врши се на основу базе података библиографске грађе и стандарда за обраду библиографских докумената.

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

Циркулација библиографских докумената омогућава евидентирање корисника библиотеке, као и праћење коришћења фонда библиотеке, резервисање и задуживање докумената. Због тога је, поред базе података библиографске грађе, потребно обезбедити и базу података коришћења библиотечке грађе.

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 YUMARC формат. У овој верзији дефинисане су следеће базе података: база података YUMARC формата, база података радног окружења библиотекара, база података библиографске грађе и база података коришћења библиотечке грађе. Наведене базе података су се уз потребне измене користиле и у наредним верзијама система.

У овој верзији система подржане су следеће функције: Дефинисање стандарда за обраду и претраживање, Формирање библиографске грађе, Библиотечко извештавање и документовање и Корисничко претраживање.

Спецификација информационих захтева ове верзије урађена је по методологији описаној у Поглављу 4 монографије [Laz96], која се базира на Структурној систем анализи и Проширеном моделу објекти-везе.

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. Овом верзијом подржане су 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. Главне карактеристике овог сервера су: специјализованост која резултује бољим перформансама, трослојна архитектура, коришћење Java платформе и независност од коришћеног система за управљање релационом базом података. Подршка за Unicode стандард доследно је спроведена у целокупном систему БИСИС вер. 3. Ова верзија, поред функција које подржава прва верзија, подржава и функцију Циркулација библиографских докумената.

Моделирање система извршено је коришћењем обједињеног језика моделирања (*Unified Modeling Language* – UML), а имплементација је извршена у Java окружењу.

1.3. Дигитална библиотека Универзитета у Новом Саду

У раду [Сур04б] 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 (која су 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) [NDLTD].

БИСИС је, са својим текст сервером, искоришћен као језгро система. Део за претраживање овог система поседује све карактеристике претраге које су својствене БИСИС-у. Равноправан третман ћириличног и латиничног писма омогућава истовремено проналажење ћириличних и латиничних докумената. Вишејезичност, као једна од основних карактеристика система, потпуно је подржана текст сервером тако да је омогућено истовремено проналажење докумената писаних на различитим језицима. Детаљи руковања вишејезичним текстом у текст серверу БИСИС-а приказани су у тези [Мил99].

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. Систем је доступан путем Интернета преко Академске рачунарске мреже. Све функције система су веб оријентисане,

тако да је унос података и претраживање информација доступно са сваког клијента који је на адекватан начин повезан 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 релационом базом 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, нпр. Linux, Windows или MacOS.

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.

NDLTD мрежа базирана је на иницијативи отворених архива (*Open Archive Initiative – OAI, ОА иницијатива*) [OAI]. Ова иницијатива осмишљена је са циљем развоја и промовисања стандарда 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*) [PMH].

У раду [Зар04] описана је имплементација PMH протокола, у оном његовом делу који дефинише публикавање података у *Dublin Core* формату [DC], који је дефинисан под окриљем ОА 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 ОАИ институција прикључи у својству система који податке из своје архиве публикује за остале чланове ОА иницијативе (тј. да се појављује у улози *data provider-a*). ОА иницијатива даје смернице за имплементацију протокола у системима који у мрежи ОАИ институција имају улогу клијентских система.

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

1.4 XML технологије и развој система БИСИС

Примена XML технологија у развоју библиотечког софтверског система БИСИС била је предмет истраживања четири магистарске тезе и једне докторске дисертације.

Циљ истраживања у тези [Зер02] 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 и изврши валидација добијених XML записа. Доказано је да се помоћу XML Schema језика могу моделирати сви концепти UNIMARC формата. Такође, доказано је и да се сви концепти библиографског записа по UNIMARC формату могу моделирати помоћу XML Schema језика.

Добијени резултати у тези [Зер02] послужили су за дефинисање нових тема за истраживање. Циљ истраживања у тези [Миј03] био је да се моделира и имплементира XML едитор за UNIMARC формат. За остварење овог циља коришћена је XML шема UNIMARC формата предложена у [Зер02]. 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.

У тези [Вид03] извршено је моделирање библиотечких каталошких листића у облику XML докумената на основу XML библиографских записа. Концепти разних типова каталошких листића описани су XML Schema језиком, као и садржај конкретних каталошких листића. Резултат издвајања садржаја каталошких листића из записа такође је дат у облику XML документа. Визуелизација овог документа извршена је у програмском језику Java, а резултат визуелизације је HTML документ. Добијени резултати су основа за укључивање описаног система за генерисање библиотечких каталошких листића у библиотечке софтверске системе.

У тези [Буд04а] описан је систем за контролу квалитета библиографских записа базиран на моделирању XML Schema језиком и језицима XSLT и XPath. Главни научни резултати добијени у овој тези су:

- Дефинисање и систематизација контрола за утврђивање квалитета библиографских записа,
- Моделирање концепата контрола записа помоћу XML Schema језика,
- XSLT спецификација за опис контрола које се не могу описати XML Schema језиком и
- Имплементација система за контролу квалитета XML библиографских записа у Java окружењу.

Циљ истраживања дисертације [Мил03] био је развој модела проширеног 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).

Моделирање и имплементација YUMARC формата

Библиографски формат MARC (*Machine Readable Cataloguing*) [Mar03], 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 [Uni94]. UNIMARC формат представља универзални формат каталогизације библиографске грађе за бележење библиографских података у машински читљивом облику, који 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 [Буд04б] 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* 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.

Блок 0 је блок за идентификацију. Подаци који се уносе у одговарајућа поља, односно потпоља овог блока су, на пример, кодирани подаци који описују публикацију, међународни стандардни број књиге уколико се обрађују монографске публикације, међународни стандардни број серијске публикације уколико се обрађују серијске публикације и други.

Блок 1 је блок кодираних информација. У поља, односно потпоља овог блока уносе се општи подаци за обраду, као што су ознака године издавања, година издавања, језик каталогизације и други. Такође, уносе се подаци о језику публикације, земљи издавања, кодирани подаци о књизи (као што су кодови за илустрације, кодови за облик садржаја и слично), као и други подаци.

Блок 2 је блок главног описа. У овом блоку наводе се стварни наслов и подаци о одговорности, подаци о издању, подаци о издавању и дистрибуцији, подаци о материјалном опису и други.

Блок 3 је блок напомена. У овом блоку наводе се различите врсте напомена о публикацији која се каталогизира, као што су опште напомене (напомена о преводу, напомена о главном наслову, напомена о издавању, напомена о материјалном опису и друге), напомена о подацима о одговорности, кратак садржај или синопсис и друге напомене.

Блок 4 је блок за повезивање библиографских записа. Користи се при повезивању публикација са њиховим прилозима или прикљученим радовима који се каталогизирају као посебне библиографске јединице. Веза између записа остварује се уносом текстуалних или нумеричких података у одгова-

рајућа 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 5 је блок сродних наслова. У пољима, односно потпољима овог блока наводе се подаци о јединственом стварном наслову, заједничком стварном наслову, упоредном стварном наслову, омотном стварном наслову и други подаци о сродним насловима публикације која се каталогизира.

Блок 6 је блок садржајне анализе. У овом блоку наводе се подаци о предметним одредницама, као што су, лично име као предметна одредница, назив корпоративног тела као предметна одредница, породично име као предметна одредница, стварни наслов као предметна одредница, тематска предметна одредница и друге. Такође, у овом блоку наводе се подаци о универзалној децималној класификацији и другим класификацијама.

Блок 7 је блок података о одговорности. У овом блоку наводе се подаци о ауторима, било да аутори имају примарну, алтернативну или секундарну одговорност за публикацију која се каталогизира.

Блок 8 је блок за међународну употребу. У овом блоку би се уносили подаци од важности за међународну употребу записа, као што су подаци о установи која је креирала запис, установи која је мењала оригинални запис, библиографски, историјски и остали подаци о запису, подаци о локацији записа и о начину на који се може приступити запису.

Блок 9 дефинише се у националном оквиру. Наменен је за унос података који су од локалног значаја за библиотечку мрежу или поједину библиотеку (сигнатура, инвентарни број, начин набавке и други). У овом блоку постоје поља чија потпоља садрже 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*.

Свим пољима придружена су два индикатора која ближе одређују значење података у пољу, однос између одређеног поља и осталих поља у запису или приказ 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 разврстане 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 (објашњено у опису блока 4 и блока 9).

Следећи пример илуструје каталогизацију једне монографске публикације у складу са YUMARC форматом. Нека је дата публикација под називом “Структуре података и организација датотека” чији аутор је 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 М-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 пред-

стављају 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 >, а иза њих наводи се садржај потпотпоља.

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 а. Поља из блока 1 предвиђена су за кодирани информације о типу публикације, језику каталогизације, језику публикације и слично. Главни стварни наслов књиге смештен је у поље 200, потпоље а, а аутор у потпоље f. Подаци о аутору уписују се у поље 700, при чему се другим индикатором одређује да ли је у потпољу а 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 блока 3. Блок 6 резервисан је за унос података садржајне анализе публикације. У поље 992 потпоље b уносе се подаци о креатору и датуму креирања записа. У потпоља поља 996 уносе се подаци о сигнатури и инвентарном 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 и 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 библиотечно-информационој делатности.

2.1 Дијаграм класа YUMARC формата

У раду [Tri98] приказан је модел података UNIMARC формата у Oracle Designer/2000 на основу којег је формиран дијаграм класа YUMARC формата. Дијаграм класа YUMARC формата, моделиран коришћењем UML (*Unified Modeling Language*) нотације [Boo99, Rum99, Ста99], 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_kontrola*.

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

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

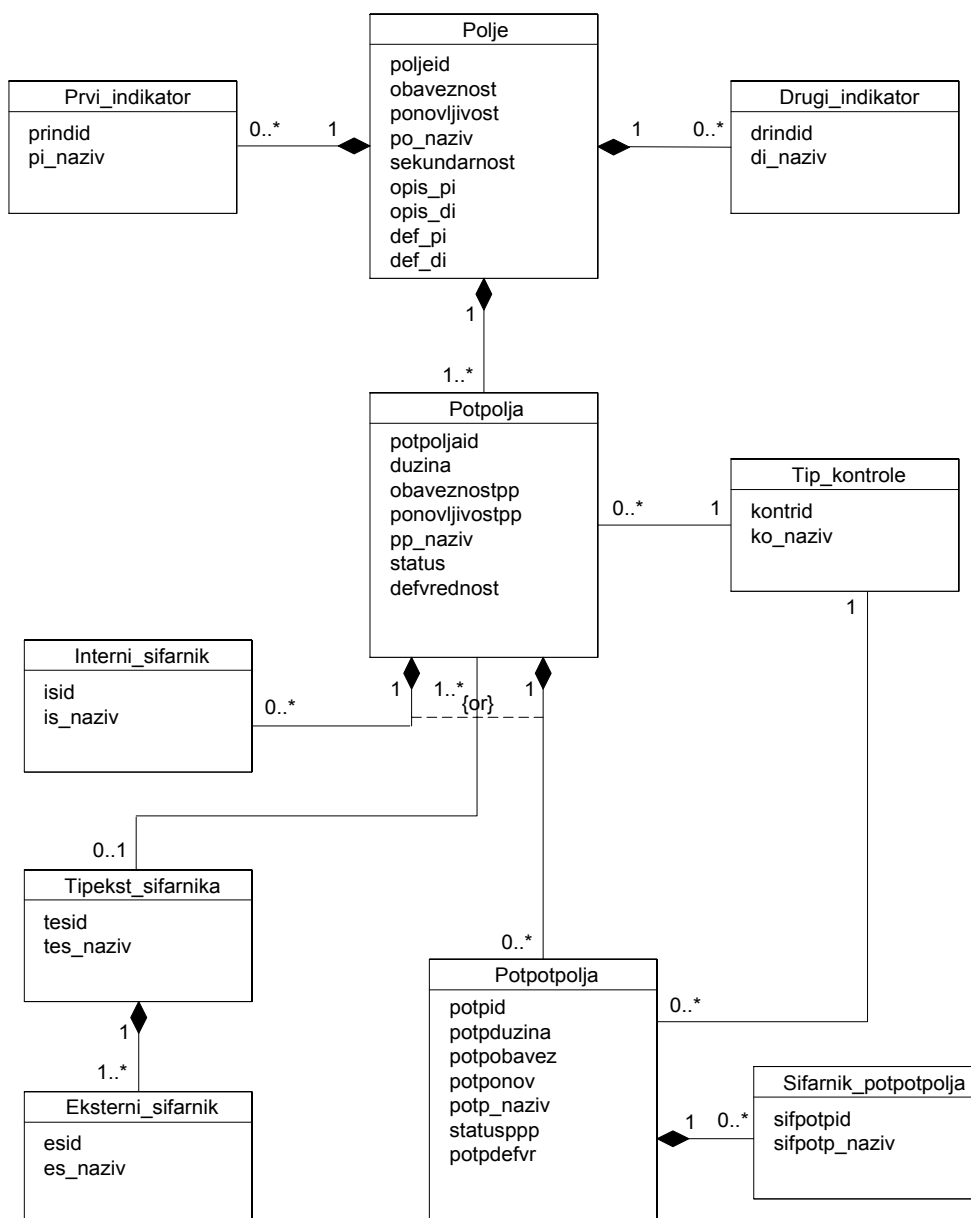
Осим тога, атрибут *opis_pi*, *opis_di*, *def_pi*, *def_di* дефинишу опис првог, односно другог индикатора за дато поље, и њихову подразумевану вредност (ако постоје).

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

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



Слика 2.2 Дијаграм класа YUMARC формата

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 (на пример, 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.

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 садржаја потпоља односно потпотпоља (на пример, контрола формата датума). Садржи атрибуте *kontrid* који дефинише јединствену вредност типа контроле и *ko_naziv* који описује значење одређене контроле.

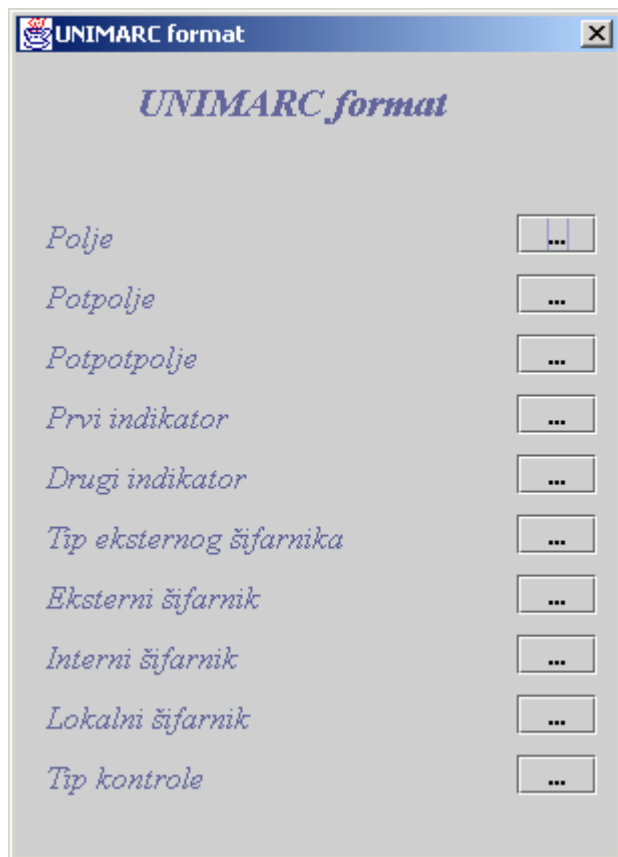
На основу дијаграма класа (слика 2.2) помоћу CASE алата *Rational Rose* [RaR02] може се генерисати шема базе података YUMARC формата. Тада, свакој класи са дијаграма класа одговара тачно једна табела (са истим именом) чији називи колона одговарају називима атрибута класе.

2.2 Ажурирање YUMARC формата

У оквиру софтверског система БИСИС вер. 3 имплементирана је апликација за ажурирање YUMARC формата [Hol98, Сур04а] у Јава окружењу [Eck00, J2SE].

Изглед почетног прозора за ажурирање YUMARC формата приказан је на слици 2.3. Овај прозор омогућава прелазак 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 слокове. У

наредном тексту приказан је прозор за ажурирање поља, док су у Прилогу 2.1 приказани прозори за ажурирање осталих концепата YUMARC формата. Приликом описивања ових прозора истакнута је веза графичких компонената прозора са колонама одговарајуће табеле базе података YUMARC формата.



Слика 2.3 Почетни прозор за ажурирање YUMARC формата

2.2.1 Прозор за ажурирање поља

Ажурирање поља YUMARC формата у софтверском систему БИСИС вер. 3 могуће је извршити помоћу прозора **Polje** који је приказан на слици 2.4. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Polje*, као и навигацију кроз постојеће слокове те табеле.

У једнолинијски едитор **Šifra** уноси/приказује се шифра поља (3 цифре). Садржај овог едитора је одговарајући садржај колоне *poljeid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив поља максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *po_naziv*.

Простори за потврду **Obaveznost**, **Ponovljenost** и **Sekundarnost** иницијално нису означени (односно, поље није обавезно, поновљиво и секундарно). Означивањем 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 колоне *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* придружених текућем пољу, као и навигацију кроз постојећа потпоља текућег поља (уколико постоје). Корисник обавезно мора да унесе бар једно потпоље за свако ново поље.

Слика 2.4 Прозор **Polje**

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 постојеће 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* за текуће поље.

Функционалност дела прозора **Polje** која се односи на други индикатор је иста као што је описано за први индикатор (само што 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. За ово поље опис првог индикатора је *Indikator vaznosti glavnog stvarnog naslova*, а подразумевана вредност му је 0.

Прилог 2.1 Прозори за ажурирање YUMARC формата

У Поглављу 2 приказан је прозор за ажурирање поља, као и веза графичких компонената овог прозора са колонама одговарајуће табеле базе података YUMARC формата. У овом прилогу приказани су прозори за ажурирање осталих концепата YUMARC формата.

2.1.1 Прозор за ажурирање потпоља

Ажурирање потпоља одређеног поља могуће је извршити помоћу прозора **Potpolja** који је приказан на слици 2.1.1. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Potpolja*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (поља) садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне *poljeid* табеле *Polje*. Одабрана шифра поља (на слици 2.1.1 шифра поља 200) из ове компоненте одговара садржају колоне табеле *Potpolja* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Polje*.

У једнолинијски едитор **Šifra** (потпоља) уноси/приказује се шифра потпоља (1 знак). Садржај овог едитора је одговарајући садржај колоне *potpoljaid*.

У вишеллинијски едитор **Naziv** уноси/приказује се назив потпоља максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *pp_naziv*.

Простори за потврду **Obaveznost** и **Ponovljenost** иницијално нису означени (односно, потпоље није обавезно и поновљиво). Означавањем ових простора за потврду постиже се да вредност у бази података буде постављена на 1 (односно, потпоље је обавезно и поновљиво). Ови простори за потврду одговарају садржајима колоне *obaveznostpp* и *ponovljivostpp* респективно.

Падајућа листа **Status**, као унапред дефинисану вредност, има одабрано *a* (активно). Корисник је у могућности да изабере једну од две вредности статуса: *a* или *p* (пасивно). Вредност ове компоненте одговара садржају колоне *status*.

Падајућа листа **Tip kontrole** садржи све унете вредности типа контроле. Садржај ове графичке компоненте су све вредности колоне *kontrid* табеле *Tip kontrole*. Одабрана вредност (на слици 2.1.1 вредност 0) из ове компонен-

те одговара садржају колоне табеле *Potpolja* (у којој се чувају вредности типа контроле), која настаје на основу везе те табеле са табелом *Tip kontrole*. Корисник је у могућности да изабере једну од понуђених вредности типа контроле. Ако је изабрани тип контроле различит од 1 (контрола преко екстерног шифарника), тада падајућа листа **Tip eksternog šifarnika** није доступна. Тип контроле као предефинисану вредност има вредност 0.

Слика 2.1.1 Прозор **Potpolja**

Падајућа листа **Tip eksternog šifarnika** садржи све унете шифре типа екстерног шифарника. Садржај ове графичке компоненте су све вредности колоне *tesid* табеле *Tipekst_sifarnika*. Одабрана вредност из ове компоненте одговара садржају колоне табеле *Potpolja* (у којој се чувају шифре типа екстерног шифарника), која настаје на основу везе те табеле са табелом *Tipekst_sifarnika*. Падајућа листа **Tip eksternog šifarnika** је недоступна ако је изабрани тип контроле различит од 1.

У једнолинијски едитор **Dužina** уноси/приказује се дужина садржаја потпоља (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *duzina*.

У једнолинијски едитор **Podrazumevana vrednost** уноси/приказује се подразумевана вредност потпоља максималне дужине 15 знакова. Садржај овог едитора је одговарајући садржај колоне *defvrednost*.

На слици 2.1.1 приказано је потпоље *a* поља 200. Назив потпоља *a* је *Glavni stvarni naslov*. Потпоље је обавезно и поновљиво. Статус потпоља је *a* (активно), тип контроле и дужина је 0.

2.1.2 Прозор за ажурирање потпотпоља

Ажурирање потпотпоља одређеног потпоља могуће је извршити помоћу прозора **Potpotpolja** који је приказан на слици 2.1.2. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Potpotpolja*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (поља) садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне табеле *Potpolja* у којој се чувају шифре поља. Одабрана шифра поља (на слици 2.1.2 шифра поља 996) из ове компоненте одговара садржају колоне табеле *Potpotpolja* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Potpolja*.

Падајућа листа **Šifra** (потпоља) садржи све унете шифре потпоља које се односе на одабрану шифру поља. Садржај ове графичке компоненте су све вредности колоне *potpoljaaid* табеле *Potpolja* придружене одабраној шифри поља. Одабрана шифра потпоља (на слици 2.1.2 шифра потпоља *d*) из ове компоненте одговара садржају колоне табеле *Potpotpolja* (у којој се чувају шифре потпоља), која настаје на основу везе те табеле са табелом *Potpolja*.

У једнолинијски едитор **Šifra** (потпотпоља) уноси/приказује се шифра потпотпоља (1 знак). Садржај овог едитора је одговарајући садржај колоне *potpid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив потпотпоља максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *potp_naziv*.

Простори за потврду **Obaveznost** и **Ponovljenost** иницијално нису означени (односно, потпотпоље није обавезно и поновљиво). Означавањем ових простора за потврду постиже се да вредност у бази података буде постављена на 1 (односно, потпотпоље је обавезно и поновљиво). Ови простори за потврду одговарају садржајима колоне *potpobavez* и *potponov* респективно.

Слика 2.1.2 Прозор **Potpotpolja**

Падајућа листа **Status**, као унапред дефинисану вредност, има одабрано *a* (активно). Корисник је у могућности да изабере једну од две вредности статуса: *a* или *p* (пасивно). Вредност ове компоненте одговара садржају колоне *statusppp*.

Падајућа листа **Tip kontrole** садржи све унете вредности типа контроле. Садржај ове графичке компоненте су све вредности колоне *kontrid* табеле *Tip_kontrole*. Одабрана вредност (на слици 2.1.2 вредност 2) ове компоненте одговара садржају колоне табеле *Potpotpolja* (у којој се чувају вредности типа контроле), која настаје на основу везе те табеле са табелом *Tip_kontrole*.

У једнолинијски едитор **Dužina** уноси/приказује се дужина садржаја потпотпоља (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *potpduzina*.

У једнолинијски едитор **Podrazumevana vrednost** уноси/приказује се подразумевана вредност потпотпоља максималне дужине 15 знакова. Садржај овог едитора је одговарајући садржај колоне *potpdefvr*.

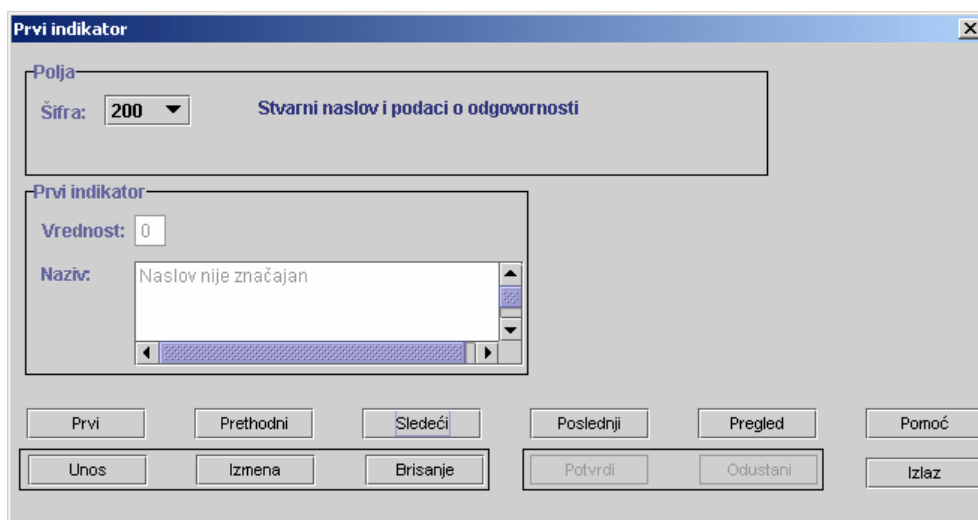
На слици 2.1.2 приказано је потпотпоље *l* потпоља *d* поља 996. Назив потпотпоља *l* је *Oznaka podlokacije*. Статус потпотпоља је *a* (активно), тип контроле 2, а дужина је 10.

2.1.3 Прозор за ажурирање првог индикатора

Ажурирање вредности првог индикатора одабраног поља могуће је извршити помоћу прозора **Prvi indikator** који је приказан на слици 2.1.3. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Prvi_indikator*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (поља) садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне *poljeid* табеле *Polje*. Одабрана шифра поља (на слици 2.1.3 шифра поља 200) из ове компоненте одговара садржају колоне табеле *Prvi_indikator* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Polje*.

У једнолинијски едитор **Vrednost** уноси/приказује се вредност првог индикатора (1 цифра). Садржај овог едитора је одговарајући садржај колоне *prindid*.



Слика 2.1.3 Прозор **Prvi indikator**

У вишелинијски едитор **Naziv** уноси/приказује се назив првог индикатора максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *pi_naziv*.

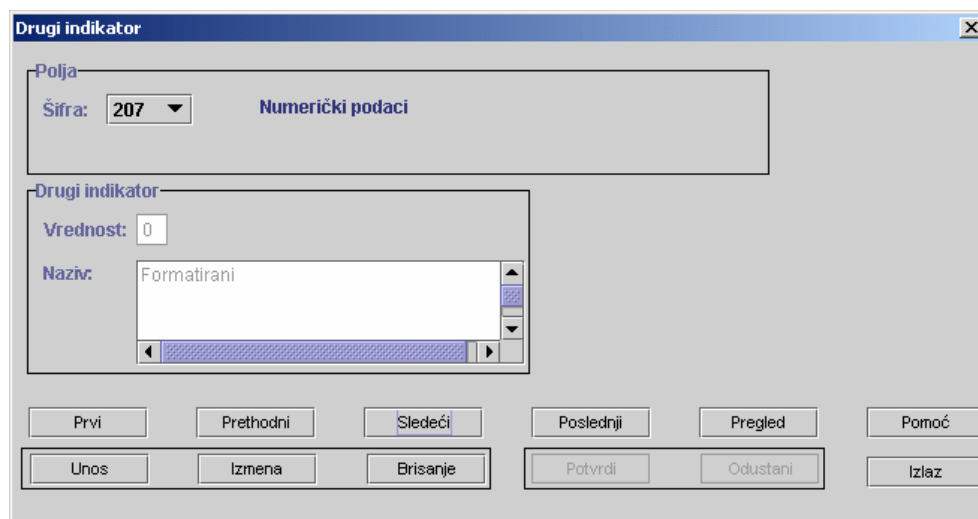
На слици 2.1.3 приказано је поље 200 са вредношћу првог индикатора 0, чији назив је *Naslov nije značajan*.

2.1.4 Прозор за ажурирање другог индикатора

Ажурирање вредности другог индикатора одабраног поља могуће је извршити помоћу прозора **Drugi indikator** који је приказан на слици 2.1.4. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Drugi_indikator*, као и навигацију кроз постојеће слоге те табеле.

Падајућа листа **Šifra** (поља) садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне *poljeid* табеле *Polje*. Одабрана шифра поља (на слици 2.1.4 шифра поља 207) из ове компоненте одговара садржају колоне табеле *Drugi_indikator* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Polje*.

У једнолинијски едитор **Vrednost** уноси/приказује се вредност другог индикатора (1 цифра). Садржај овог едитора је одговарајући садржај колоне *drindid*.



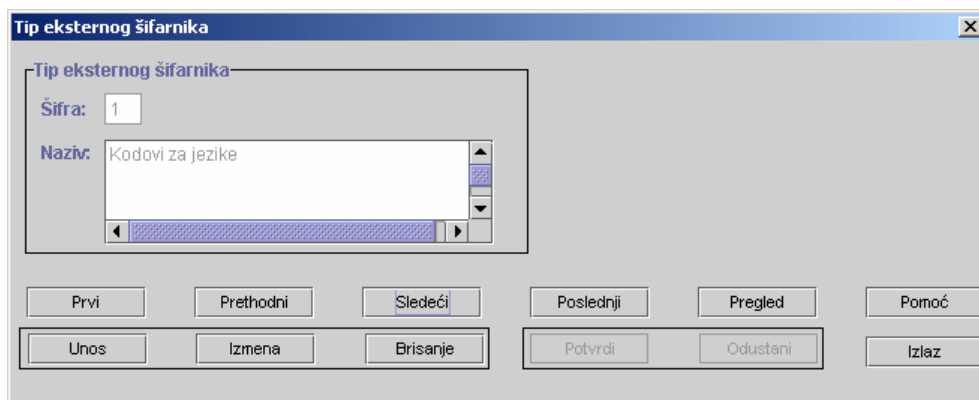
Слика 2.1.4 Прозор **Drugi indikator**

У вишелинијски едитор **Naziv** уноси/приказује се назив другог индикатора максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *di_naziv*.

На слици 2.1.4 приказано је поље 207 са вредношћу другог индикатора 0, чији назив је *Formatirani*.

2.1.5 Прозор за ажурирање типа екстерног шифарника

Ажурирање шифри типа екстерног шифарника могуће је извршити помоћу прозора **Tip eksternog šifarnika** који је приказан на слици 2.1.5. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Tipekst_sifarnika*, као и навигацију кроз постојеће слокове те табеле.



Слика 2.1.5 Прозор **Tip eksternog šifarnika**

У једнолинијски едитор **Šifra** уноси/приказује се шифра типа екстерног шифарника (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *tesid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив типа екстерног шифарника максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *tes_naziv*.

На слици 2.1.5 приказана је шифра типа екстерног шифарника 1 чији назив је *Kodovi za jezike*.

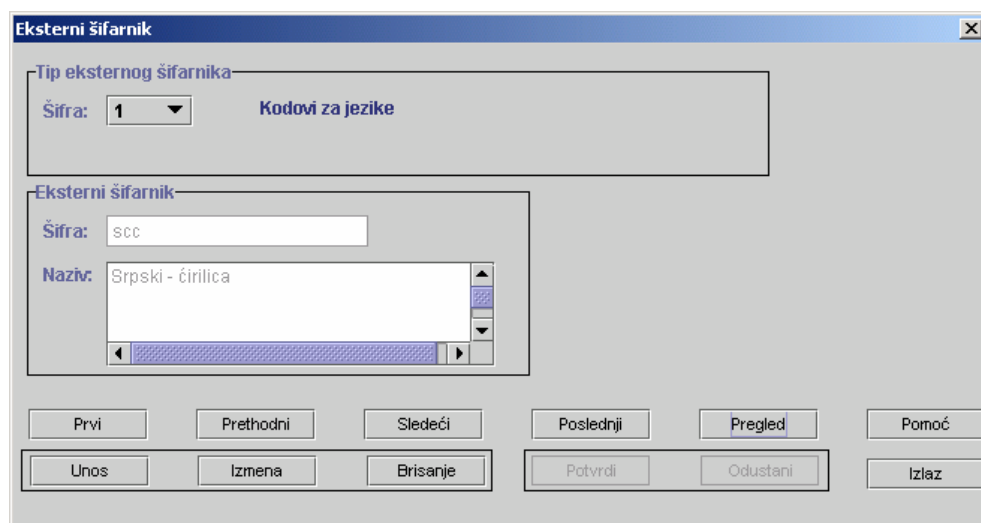
2.1.6 Прозор за ажурирање екстерног шифарника

Ажурирање шифри екстерног шифарника одређеног типа екстерног шифарника могуће је извршити помоћу прозора **Eksterni šifarnik** који је

приказан на слици 2.1.6. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Eksterni_sifarnik*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (типа екстерног шифарника) садржи све унете шифре типа екстерног шифарника. Садржај ове графичке компоненте су све вредности колоне *tesid* табеле *Tipekst_sifarnika*. Одабрана шифра типа екстерног шифарника (на слици 2.1.6 шифра типа екстерног шифарника *1*) из ове компоненте одговара садржају колоне табеле *Eksterni_sifarnik* (у којој се чувају шифре типа екстерног шифарника), која настаје на основу везе те табеле са табелом *Tipekst_sifarnika*.

У једнолинијски едитор **Šifra** (екстерног шифарника) уноси/приказује се шифра екстерног шифарника (максимално 15 знакова). Садржај овог едитора је одговарајући садржај колоне *esid*.



Слика 2.1.6 Прозор **Eksterni šifarnik**

У вишелинијски едитор **Naziv** уноси/приказује се назив екстерног шифарника максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *es_naziv*.

На слици 2.1.6 приказан је шифра типа екстерног шифарника *1* (*Kodovi za jezike*) са шифром екстерног шифарника *scc*, чији назив је *Srpski-ćirilica*.

2.1.7 Прозор за ажурирање интерног шифарника

Ажурирање шифри интерног шифарника одређеног потпоља могуће је извршити помоћу прозора **Interni šifarnik** који је приказан на слици 2.1.7. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Interni_sifarnik*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (поља) садржи све унете шифре поља. Садржај ове графичке компоненте су све вредности колоне табеле *Potpolja* у којој се чувају шифре поља. Одабрана шифра поља (на слици 2.1.7 шифра поља 001) из ове компоненте одговара садржају колоне табеле *Interni_sifarnik* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Potpolja*.

Падајућа листа **Šifra** (потпоља) садржи све унете шифре потпоља које се односе на одабрану шифру поља. Садржај ове графичке компоненте су све вредности колоне *potpoljaaid* табеле *Potpolja* придружене одабраној шифри поља. Одабрана шифра потпоља (на слици 2.1.7 шифра потпоља a) из ове компоненте одговара садржају колоне табеле *Interni_sifarnik* (у којој се чувају шифре потпоља), која настаје на основу везе те табеле са табелом *Potpolja*.

Слика 2.1.7 Прозор **Interni šifarnik**

У једнолинијски едитор **Šifra** (интерног шифарника) уноси/приказује се шифра интерног шифарника (максимално 15 знакова). Садржај овог едитора је одговарајући садржај колоне *isid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив интерног шифарника максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *is_naziv*.

На слици 2.1.7 приказана је шифра интерног шифарника *c* потпоља *a* поља *001*. Назив шифре интерног шифарника *c* је *Konačno ispravljani zapis*.

2.1.8 Прозор за ажурирање локалног шифарника

Ажурирање шифри локалног шифарника одређеног потпотпоља могуће је извршити помоћу прозора **Lokalni šifarnik** који је приказан на слици 2.1.8. Овај прозор омогућава унос, модификацију и брисање слогова из табеле *Sifarnik_potpotpolja*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Šifra** (поља) садржи све шифре поља чије бар једно потпоље садржи потпотпоља. Садржај ове графичке компоненте су све вредности колоне табеле *Potpotpolja* у којој се чувају шифре поља. Одабрана шифра поља (на слици 2.1.8 шифра поља *996*) из ове компоненте одговара садржају колоне табеле *Sifarnik_potpotpolja* (у којој се чувају шифре поља), која настаје на основу везе те табеле са табелом *Potpotpolja*.

Падајућа листа **Šifra** (потпоља) садржи све шифре потпоља које се односе на одабрану шифру поља, а имају потпотпоља. Садржај ове графичке компоненте су све вредности колоне табеле *Potpotpolja* у којој се чувају шифре потпоља придружене одабраној шифри поља. Одабрана шифра потпоља (на слици 2.1.8 шифра потпоља *d*) из ове компоненте одговара садржају колоне табеле *Sifarnik_potpotpolja* (у којој се чувају шифре потпоља), која настаје на основу везе те табеле са табелом *Potpotpolja*.

Падајућа листа **Šifra** (потпотпоља) садржи све шифре потпотпоља које се односе на одабрану шифру потпоља и шифру поља. Садржај ове графичке компоненте су све вредности колоне *potpid* табеле *Potpotpolja* придружене одабраној шифри поља и шифри потпоља. Одабрана шифра потпотпоља (на слици 2.1.8 шифра потпотпоља *l*) из ове компоненте одговара садржају колоне табеле *Sifarnik_potpotpolja* (у којој се чувају шифре потпотпоља), која настаје на основу везе те табеле са табелом *Potpotpolja*.

Слика 2.1.8 Прозор **Lokalni šifarnik**

У једнолинијски едитор **Šifra** (локалног шифарника) уноси/приказује се шифра локалног шифарника (максимално 15 знакова). Садржај овог едитора је одговарајући садржај колоне *sifpotpid*.

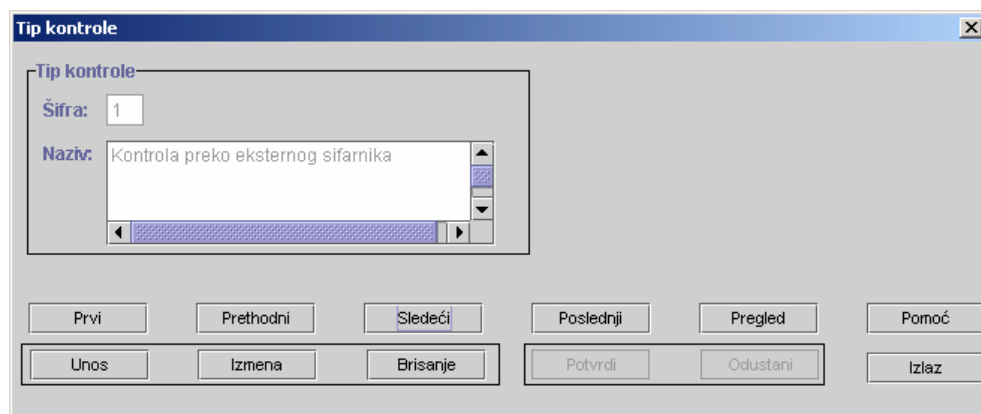
У вишелинијски едитор **Naziv** уноси/приказује се назив локалног шифарника максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *sifpotp_naziv*.

На слици 2.1.8 приказана је шифра локалног шифарника *M*, чији назив је *Matematika*. Шифра локалног шифарника *M* припада потпотпољу *I* потпоља *d* поља *996*.

2.1.9 Прозор за ажурирање типа контроле

Ажурирање шифри типа контроле могуће је извршити помоћу прозора **Tip kontrole** који је приказан на слици 2.1.9. Овај прозор омогућава унос,

модификацију и брисање слогова из табеле *Tip_kontrole*, као и навигацију кроз постојеће слокове те табеле.



Слика 2.1.9 Прозор **Tip kontrole**

У једнолинијски едитор **Šifra** уноси/приказује се шифра типа контроле (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *kontrid*.

У вишелинијски едитор **Naziv** уноси/приказује се назив типа контроле максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *ko_naziv*.

На слици 2.1.9 приказана је шифра типа контроле *1* чији назив је *Kontrola preko eksternog šifarnika*.

Моделирање и имплементација радног окружења библиотекара

Претходно поглавље односило се на концепте YUMARC формата (поље, потпоље, потпотпоље...). Да би се дефинисало радно окружење библиотекара, потребно је прво дефинисати податке у оквиру YUMARC формата. Неки од ових података служе за дефинисање појединих концепата радног окружења библиотекара (потпоља типа публикације, потпоља нивоа обраде...).

Подаци о радном окружењу библиотекара налазе се у бази података радног окружења библиотекара [Laz96, Пуп00б], која представља саставни део софтверског система БИСИС. 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 [Сур03] 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 у апликацији за обраду библиографске грађе 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 дефинишу се 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 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 потпоља дефинисана на датом нивоу обавезности.

База података радног окружења библиотекара садржи и различите формате исписа који омогућавају различите приказе извештаја и резултата претраживања базе података библиографске грађе. Поред тога, база података радног окружења библиотекара 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 нотације, приказан на слици 3.1, садржи следеће класе: *Tip_publicacije*, *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 (описан у Поглављу 2) и представља везу између ова два дијаграма.

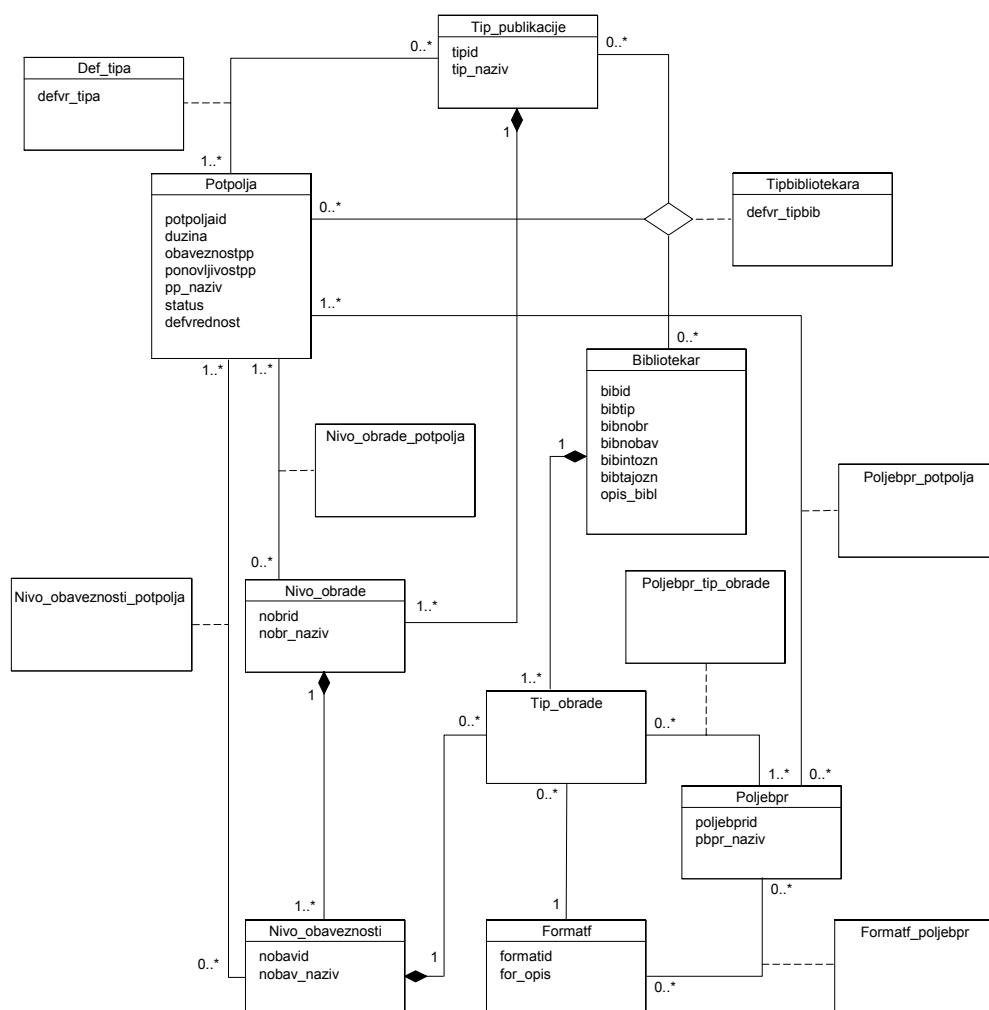
Класа *Tip_publicacije* садржи атрибут *tipid* који дефинише јединствену вредност типа публикације и атрибут *tip_naziv* који дефинише назив одређеног типа публикације.

Класа асоцијације *Def_tipa* дефинише 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 потпоља која се обрађују на одређеном нивоу обраде.

Класа *Nivo_obaveznosti* садржи атрибут *nobavid* који дефинише јединствену вредност нивоа обавезности у оквиру одређеног нивоа обраде. Такође, ова класа садржи и атрибут *nobav_naziv* којим је дефинисано значење одређеног нивоа обавезности. Класа асоцијације *Nivo_obaveznosti_potpolja* пове-

зана са везом између класа *Nivo_obaveznosti* и *Potpolja* дефинише потпоља која се обрађују на одређеном нивоу обавезности.



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*, *bibnabr* и *bibnobav* дефинишу подразумевану вредност типа публикације, нивоа обраде и нивоа обавезности који се постављају за библио-

текара након пријављивања на систем. Атрибут *opis_bibl* садржи додатне податке о библиотекару.

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

Класа *Poljebpr* дефинише префиксе претраживања базе података библиографске грађе. Атрибут *poljebprid* дефинише 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 и потпоља која се претражују у оквиру одређеног префикса претраживања.

Класа *Formatf* дефинише формате исписа предвиђене у систему. Атрибут *formatid* дефинише 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 за рад библиотекара у апликацији за обраду библиографске грађе 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 претраживања за одређени тип обраде библиотекара.

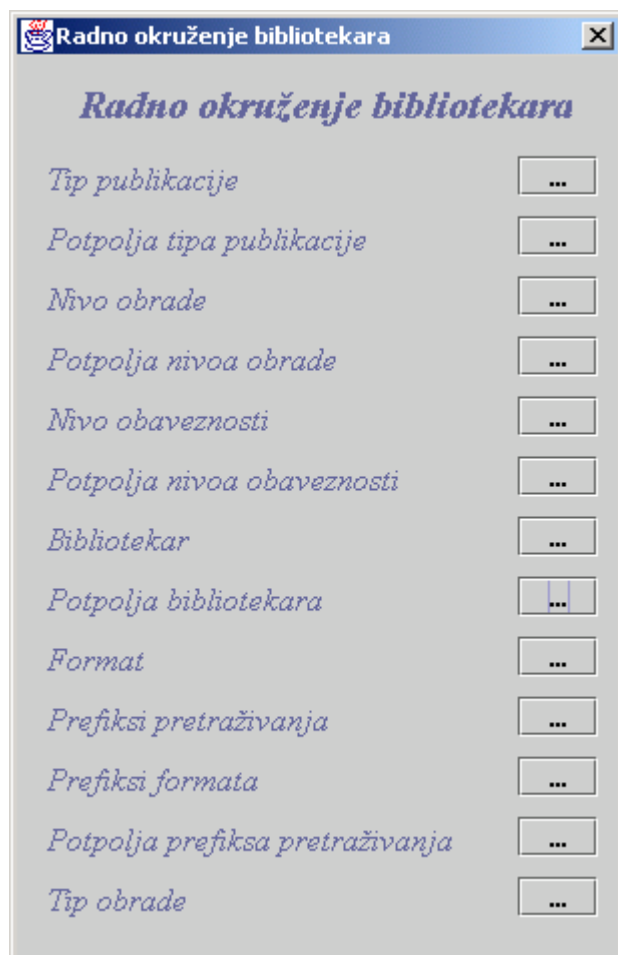
На основу дијаграма класа (слика 3.1) помоћу CASE алата *Rational Rose* може се генерисати шема базе података радног окружења библиотекара. Тада, свакој класи са дијаграма класа одговара тачно једна табела (са истим именом) чији називи колона одговарају називима атрибута класе.

3.2 Ажурирање радног окружења библиотекара

Саставни део софтверског система БИСИС вер. 3 је и апликација за ажурирање радног окружења библиотекара [Пуп01д] имплементирана у Јава окружењу.

Изглед почетног прозора за ажурирање радног окружења библиотекара приказан је на слици 3.2. Овај прозор омогућава прелазак 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 слокове. У наредном тексту приказани су прозори за ажурирање типа публикације и потпоља типа публикације. Дефинисањем ових концепата омогућена је обрада различитих типова докумената. У Прилогу 3.1 приказни су прозори за ажурирање осталих концепата радног окружења библиотекара. Приликом описивања ових прозора истакнута је веза графичких компонената прозора са колонама одговарајуће табеле базе података радног окружења библиотекара.



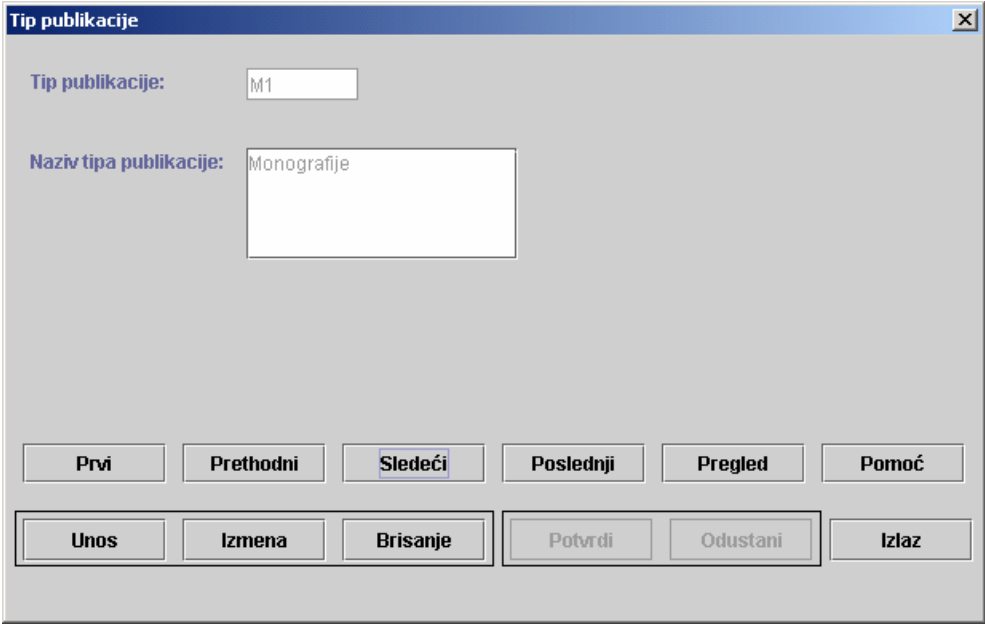
Слика 3.2 Почетни прозор за ажурирање радног окружења библиотекара

3.2.1 Прозор за ажурирање типа публикације

Први корак при дефинисању радног окружења библиотекара јесте дефинисање свих типова докумената (односно типова публикација) који се могу наћи у конкретної библиотеци. Прозор **Tip publikacije**, приказан на слици 3.3, служи за ажурирање слогова табеле *Tip publikacije* (у којој се чувају сви типови докумената који се могу обрађивати), као и за навигацију кроз постојеће слоге те табеле.

У једнолинијски едитор **Tip publikacije** уноси/приказује се вредност типа публикације (2 знака). Садржај овог едитора је одговарајући садржај колоне *tipid*.

У вишелинијски едитор **Naziv tipa publikacije** уноси/приказује се назив типа публикације максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *tip_naziv*.



Слика 3.3 Прозор **Tip publikacije**

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

3.2.2 Прозор за ажурирање потпоља типа публикације

За сваки тип публикације потребно је обрадити другачији скуп потпоља YUMARC формата. Дефинисање тог скупа врши се помоћу прозора **Potpolja tipa publikacije** који је приказан на слици 3.4. Сваком типу публикације се из скупа свих потпоља придружују она која се могу обрађивати за тај тип публикације. Уколико је потребно, постојећем скупу потпоља неког типа публикације могуће је додати нова или избацити постојећа потпоља. Осим тога, може се унети или променити подразумевана вредност сваког потпоља типа публикације. Акције над прозором **Potpolja tipa publikacije** омогућавају ажурирање слогова табеле *Def_tipa*, као и навигацију кроз постојеће слокове 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 3.4 вредност *M1*) из ове компоненте одговара садржају колоне табеле *Def_tipa* у којој се чувају вредности типа публикације.

Potpolja tipa publikacije

Tip publikacije: **M1** Monografije

Šifra polja: **001** Identifikator sloga

Šifra potpolja: **a** Stanje sloga

Podrazumevana vrednost: i

Prvi Prethodni **Sledeći** Poslednji Pregled Pomoć

Izmena Brisanje Potvrdi Odustani Izlaz

Слика 3.4 Прозор **Potpolja tipa publikacije**

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 те табеле и табеле *Polje* (описана у Поглављу 2). Одабрана шифра (на слици 3.4 шифра поља 001) из ове компоненте одговара садржају колоне табеле *Def_tipa* у којој се чувају шифре поља.

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* придружене одабраној шифри поља. Одабрана шифра (на слици 3.4 шифра потпоља *a*) из ове компоненте одговара садржају колоне табеле *Def_tipa* у којој се чувају шифре потпоља.

У једнолинијски едитор **Podrazumevana vrednost** уноси/приказује се подразумевана вредност потпоља. Садржај овог едитора је одговарајући садржај колоне *defvr_tipa*.

На слици 3.4 приказана је подразумевана вредност *i* потпоља *a* поља 001 који су придружени типу публикације *M1*.

Прилог 3.1 Прозори за ажурирање радног окружења библиотекара

У Поглављу 3 приказани су прозор за ажурирање типа публикације и прозор за ажурирање потпоља типа публикације, као и веза графичких компоненти ових прозора са колонама одговарајућих табела базе података радног окружења библиотекара. У наредном тексту приказни су прозори за ажурирање осталих концепата радног окружења библиотекара.

3.1.1 Прозор за ажурирање нивоа обраде

У зависности од природе посла коју библиотекари обављају потребно је за сваки тип публикације дефинисати један или више нивоа обраде. Дефинисање нивоа обраде за одређени тип публикације могуће је извршити помоћу прозора **Nivo obrade** који је приказан на слици 3.1.1. За сваки ниво обраде, дефинисан за одређени тип публикације, уноси се вредност нивоа обраде и назив нивоа обраде. Уколико је потребно, сваки ниво обраде дефинисан за одређени тип публикације могуће је избрисати или извршити измену његовог назива. Акције над прозором **Nivo obrade** омогућавају ажурирање слогова табеле *Nivo_obrade*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Tip publikacije** садржи све унете вредности типа публикације. Садржај ове графичке компоненте су све вредности колоне *tipid* табеле *Tip_publicacije*. Одабрана вредност (на слици 3.1.1 вредност *MI*) из ове компоненте одговара садржају колоне табеле *Nivo_obrade* (у којој се чувају вредности типа публикације), која настаје на основу везе те табеле са табелом *Tip_publicacije*.

У једнолинијски едитор **Nivo obrade** уноси/приказује се вредност нивоа обраде (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *nobrid*.

У вишелинијски едитор **Naziv nivoa obrade** уноси/приказује се назив нивоа обраде максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *nobr_naziv*.

На слици 3.1.1 приказан је тип публикације *MI* са придруженим нивоом обраде *I0*, чији назив је *Kataloška obrada*.

The image shows a software window titled "Nivo obrade". Inside, there are three main input areas:

- "Tip publikacije:" with a dropdown menu showing "M1" and a link "Monografije".
- "Nivo obrade:" with a text box containing the number "10".
- "Naziv nivoa obrade:" with a text box containing "Kataloška obrada".

 At the bottom of the window, there are two rows of buttons:

- Top row: "Prvi", "Prethodni", "Sledeći" (highlighted with a blue border), "Poslednji", "Pregled", and "Pomoć".
- Bottom row: "Unos", "Izmena", "Brisanje", "Potvrdi", "Odustani", and "Izlaz".

Слика 3.1.1 Прозор **Nivo obrade**

3.1.2 Прозор за ажурирање потпоља нивоа обраде

Да би се дефинисао скуп потпоља који је карактеристичан за дату врсту посла дефинишу се потпоља нивоа обраде. Дефинисање потпоља нивоа обраде за одређени ниво обраде у оквиру специфицираног типа публикације, могуће је извршити помоћу прозора **Potpolja nivoa obrade** који је приказан на слици 3.1.2. Овај скуп потпоља мора бити подскуп скупа потпоља дефинисаних у оквиру специфицираног типа публикације. Уколико је потребно, постојећим потпољима неког нивоа обраде могуће је додати нова или избаци-ти постојећа потпоља. Акције над прозором **Potpolja nivoa obrade** омогућавају ажурирање слогова табеле *Nivo_obrade_potpolja*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Tip publikacije** садржи све унете вредности типа публикације за које су дефинисана потпоља типа публикације и нивои обраде. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obrade* у којој се чувају вредности типа публикације за које су дефинисана потпоља типа публикације. Одабрана вредност (на слици 3.1.2 вредност *M1*) из ове компоненте одговара садржају колоне табеле *Nivo_obrade_potpolja* у којој се чувају вредности типа публикације.

Падајућа листа **Nivo obrade** садржи све унете вредности нивоа обраде које припадају одабраном типу публикације. Садржај ове графичке компоненте су све вредности колоне *nobrid* табеле *Nivo_obrade* придружене одабраном типу публикације. Одабрана вредност (на слици 3.1.2 вредност *10*) из ове компоненте одговара садржају колоне табеле *Nivo_obrade_potpolja* у којој се чувају вредности нивоа обраде.

Падајућа листа **Šifra polja** садржи све унете шифре поља које су у скупу поља дефинисаних у оквиру концепта потпоља типа публикације (за одабрани тип публикације). Садржај ове графичке компоненте су све вредности колоне табеле *Def_tipa* у којој се чувају шифре поља придружене одабраном типу публикације. Одабрана шифра (на слици 3.1.2 шифра поља *001*) из ове компоненте одговара садржају колоне табеле *Nivo_obrade_potpolja* у којој се чувају шифре поља.

Potpolja nivoa obrade

Tip publikacije: **M1** Monografije

Nivo obrade: **10** Kataloška obrada

Šifra polja: **001** Identifikator sloga

Šifra potpolja: ---bez selekcije---
7-Abeceda/pismo
a-Stanje sloga
b-Vrsta sloga
c-Bibliografski nivo

Prvi Prethodni **Sledeći** Poslednji Pregled Pomoć

Izmena Potvrdi Odustani Izlaz

Слика 3.1.2 Прозор **Potpolja nivoa obrade**

Листа **Šifra potpolja** садржи све шифре потпоља које се односе на одабрану шифру поља и које су у скупу потпоља дефинисаних у оквиру

концепта потпоља типа публикације (за одабрани тип публикације). Садржај ове графичке компоненте су све вредности колоне табеле *Def_tipa* у којој се чувају шифре потпоља одабраног поља придружене одабраном типу публикације. Поред тога, уз сваку шифру потпоља приказан је и назив потпоља. Одабране шифре (на слици 3.1.2 шифре потпоља 7, a, b, c) из ове компоненте одговарају садржајима колоне табеле *Nivo_obrade_potpolja* у којој се чувају шифре потпоља.

На слици 3.1.2 приказане су шифре потпоља поља 001 које су придружене нивоу обраде 10 типа публикације *M1*.

3.1.3 Прозор за ажурирање нивоа обавезности

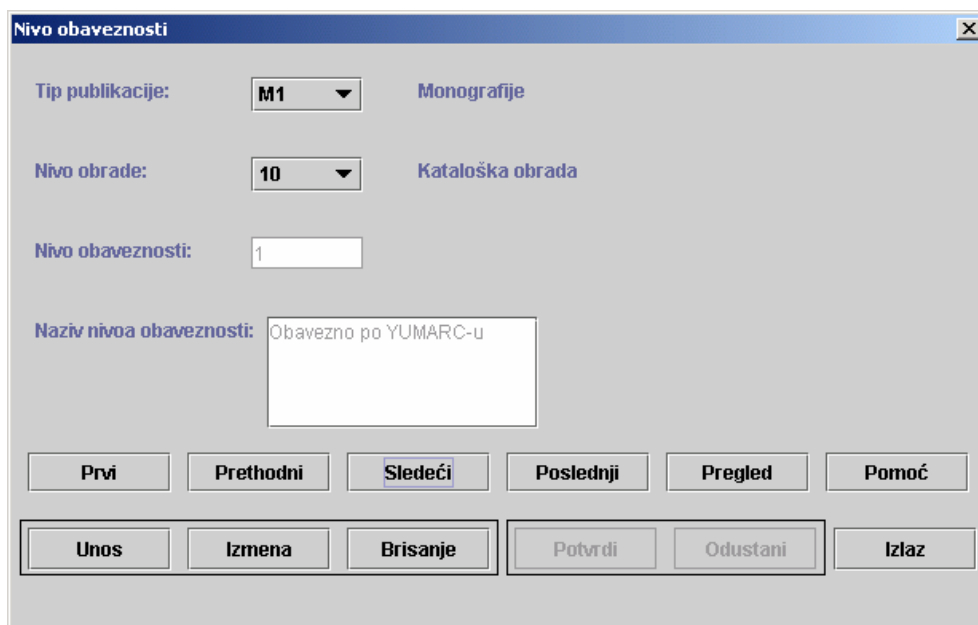
За одређени тип публикације и ниво обраде могу се дефинисати различити нивои обавезности. Дефинисање нивоа обавезности за одређени ниво обраде и тип публикације врши се помоћу прозора **Nivo obaveznosti** који је приказан на слици 3.1.3. За сваки ниво обавезности, дефинисан за одређени тип публикације и ниво обраде, уноси се вредност нивоа обавезности и назив нивоа обавезности. Уколико је потребно, сваки ниво обавезности, дефинисан за одређени тип публикације и ниво обраде, могуће је избрисати или извршити измену његовог назива. Акције над прозором **Nivo obaveznosti** омогућавају ажурирање слогова табеле *Nivo_obaveznosti*, као и навигацију кроз постојеће слоге те табеле.

Падајућа листа **Tip publikacije** садржи све унете вредности типа публикације за које су дефинисани нивои обраде. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obrade* у којој се чувају вредности типа публикације. Одабрана вредност (на слици 3.1.3 вредност *M1*) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti* (у којој се чувају вредности типа публикације), која настаје на основу везе те табеле са табелом *Nivo_obrade*.

Падајућа листа **Nivo obrade** садржи све унете вредности нивоа обраде одабраног типа публикације. Садржај ове графичке компоненте су све вредности колоне *nobrid* табеле *Nivo_obrade* придружене одабраном типу публикације. Одабрана вредност (на слици 3.1.3 вредност 10) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti* (у којој се чувају вредности нивоа обраде), која настаје на основу везе те табеле са табелом *Nivo_obrade*.

У једнолинијски едитор **Nivo obaveznosti** уноси/приказује се вредност нивоа обавезности (максимално 2 цифре). Садржај овог едитора је одговарајући садржај колоне *nobavid*.

У вишелинијски едитор **Naziv nivoa obaveznosti** уноси/приказује се назив нивоа обавезности максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *nobav_naziv*.



The screenshot shows a window titled "Nivo obaveznosti". It contains the following elements:

- Tip publikacije:** A dropdown menu showing "M1" and a label "Monografije".
- Nivo obrade:** A dropdown menu showing "10" and a label "Kataloška obrada".
- Nivo obaveznosti:** A text input field containing the number "1".
- Naziv nivoa obaveznosti:** A text input field containing the text "Obavezno po YUMARC-u".
- Buttons:** A row of six buttons: "Prvi", "Prethodni", "Sledeći", "Poslednji", "Pregled", and "Pomoć". Below this is another row of six buttons: "Unos", "Izmena", "Brisanje", "Potvrdi", "Odustani", and "Izlaz".

Слика 3.1.3 Прозор **Nivo obaveznosti**

На слици 3.1.3 приказан је ниво обавезности *1* (чији назив је *Obavezno po YUMARC-u*) придружен нивоу обраде *10* типа публикације *M1*.

3.1.4 Прозор за ажурирање потпоља нивоа обавезности

Обрада документа сматра се комплетном ако се унесу вредности за сва потпоља везана за одређени ниво обавезности. У ту сврху дефинише се скуп потпоља одређеног нивоа обавезности помоћу прозора **Potpolja nivoa obaveznosti** који је приказан на слици 3.1.4. Овај скуп потпоља, дефинисаних за одређени ниво обавезности у оквиру датог нивоа обраде и типа публикације, мора бити подскуп скупа потпоља дефинисаних у оквиру датог нивоа обраде. Уколико је потребно, постојећим потпољима неког нивоа обавезности могуће је додати нова или избацити постојећа потпоља. Акције над прозором

Potpolja nivoa obaveznosti омогућавају ажурирање слогова табеле *Nivo_obaveznosti_potpolja*, као и навигацију кроз постојеће слокове те табеле.

Падајућа листа **Tip publikacije** садржи све унете вредности типа публикације за које су дефинисана потпоља нивоа обраде и нивои обавезности. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obaveznosti* у којој се чувају вредности типа публикације за које су дефинисана потпоља нивоа обраде. Одабрана вредност (на слици 3.1.4 вредност *MI*) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti_potpolja* у којој се чувају вредности типа публикације.

Падајућа листа **Nivo obrade** садржи све унете вредности нивоа обраде (које припадају одабраном типу публикације) за које су дефинисана потпоља нивоа обраде и нивои обавезности. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obaveznosti* у којој се чувају вредности нивоа обраде (придružене одабраном типу публикације) за које су дефинисана потпоља нивоа обраде. Одабрана вредност (на слици 3.1.4 вредност *I0*) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti_potpolja* у којој се чувају вредности нивоа обраде.

Падајућа листа **Nivo obaveznosti** садржи све унете вредности нивоа обавезности које припадају одабраном типу публикације и нивоу обраде. Садржај ове графичке компоненте су све вредности колоне *nobavid* табеле *Nivo_obaveznosti* придružене одабраном типу публикације и нивоу обраде. Одабрана вредност (на слици 3.1.4 вредност *I*) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti_potpolja* у којој се чувају вредности нивоа обавезности.

Падајућа листа **Šifra polja** садржи све унете шифре поља које су у скупу поља дефинисаних у оквиру концепта потпоља нивоа обраде (за одабрани тип публикације и ниво обраде). Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obrade_potpolja* у којој се чувају шифре поља придružене одабраном типу публикације и нивоу обраде. Одабрана шифра (на слици 3.1.4 шифра поља *001*) из ове компоненте одговара садржају колоне табеле *Nivo_obaveznosti_potpolja* у којој се чувају шифре поља.

Листа **Šifra potpolja** садржи све шифре потпоља које се односе на одабрану шифру поља и које су у скупу потпоља дефинисаних у оквиру концепта потпоља нивоа обраде (за одабрани тип публикације и ниво обраде). Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obrade_potpolja* у којој се чувају шифре потпоља одабраног поља придružених одабраном типу публикације и нивоу обраде. Поред тога, уз сваку шифру потпоља приказан је и назив потпоља. Одабране шифре (на

слици 3.1.4 шифре потпоља *a*, *b*, *c*) из ове компоненте одговарају садржајима колоне табеле *Nivo_obaveznosti_potpolja* у којој се чувају шифре потпоља.

Слика 3.1.4 Прозор **Potpolja nivoa obaveznosti**

На слици 3.1.4 приказане су шифре потпоља поља *001* које су придружене нивоу обавезности *1* нивоа обраде *10* и типа публикације *M1*.

3.1.5 Прозор за ажурирање библиотекара

Сваки библиотекар одређене библиотеке мора бити регистрован у бази података радног окружења библиотекара да би могао да ради у систему. Регистрација библиотекара врши се помоћу прозора **Bibliotekar** који је приказан на слици 3.1.5. У оквиру овог прозора се, осим шифре библиотекара, интерне ознаке библиотекара, тајне ознаке и описа, уносе подразумеване вредности за тип публикације, ниво обраде и ниво обавезности. Уколико је потребно, сваког регистрованог библиотекара у бази података могуће је избрисати или извршити измену интерне ознаке библиотекара, тајне ознаке,

описа и подразумеваних вредности за тип публикације, ниво обраде и ниво обавезности. Акције над прозором **Bibliotekar** омогућавају ажурирање слогова табеле *Bibliotekar*, као и навигацију кроз постојеће слокове те табеле.

У једнолинијски едитор **Šifra bibliotekara** уноси/приказује се шифра библиотекар (максимално 50 знакова). Садржај овог едитора је одговарајући садржај колоне *bibid*.

У једнолинијски едитор **Interna oznaka** уноси/приказује се интерна ознака библиотекар максималне дужине 10 цифара. Садржај овог едитора је одговарајући садржај колоне *bibintozn*.

Bibliotekar

Šifra bibliotekara:

Tip publikacije: Monografije

Nivo obrade: Kataloška obrada

Nivo obaveznosti: Obavezno po YUMARC-u

Interna oznaka:

Tajna oznaka:

Opis bibliotekara:

Prvi Prethodni Sledeći Poslednji Pregled Pomoć

Unos Izmena Brisanje Potvrdi Odustani Izlaz

Слика 3.1.5 Прозор **Bibliotekar**

У једнолинијски едитор **Tajna oznaka** уноси/приказује се тајна ознака библиотекара максималне дужине 6 знакова. Садржај овог едитора је одговарајући садржај колоне *bibtajozn*.

У вишелинијски едитор **Opis bibliotekara** уноси/приказује се опис библиотекара максималне дужине 240 знакова. Садржај овог едитора је одговарајући садржај колоне *opis_bibl*.

Избор подразумеваних вредности типа публикације, нивоа обраде и нивоа обавезности врши се избором одговарајућих вредности на овом прозору за сваког библиотекара понаособ. Одабране вредности типа публикације, нивоа обраде и нивоа обавезности одговарају садржајима колоне *bibtip*, *bibnobr* и *bibnobav* табеле *Bibliotekar* респективно.

На слици 3.1.5 приказан је библиотекар са шифром *Goga*. Подразумевана вредност типа публикације за овог библиотекара је *M1*, нивоа обраде *10* и нивоа обавезности *1*. Интерна ознака је *111*, а тајна ознака *xxx*.

3.1.6 Прозор за ажурирање потпоља библиотекара

За сваког библиотекара у систему могу се дефинисати подразумеване вредности потпоља за свако потпоље из скупа потпоља одређеног типа публикације. Подразумеване вредности се уносе помоћу прозора **Potpolja bibliotekara** који је приказан на слици 3.1.6. Уколико је потребно, постојећим потпољима библиотекара могуће је додати нова потпоља или избацити нека. Осим тога, може се променити подразумевана вредност за свако од потпоља библиотекара. Акције над прозором **Potpolja bibliotekara** омогућавају ажурирање слогова табеле *Tipbibliotekara*, као и навигацију кроз постојеће слогове те табеле.

Падајућа листа **Šifra bibliotekara** садржи све унете шифре библиотекара. Садржај ове графичке компоненте су све вредности колоне *bibid* табеле *Bibliotekar*. Одабрана вредност (на слици 3.1.6 вредност *Goga*) из ове компоненте одговара садржају колоне табеле *Tipbibliotekara* у којој се чувају шифре библиотекара.

Падајућа листа **Tip publikacije** садржи вредности типа публикације за које су дефинисана потпоља типа публикације. Садржај ове графичке компоненте су све вредности колоне табеле *Def_tipa* у којој се чувају вредности типа публикације. Одабрана вредност (на слици 3.1.6 вредност *M1*) из ове компоненте одговара садржају колоне табеле *Tipbibliotekara* у којој се чувају вредности типа публикације.

Падајућа листа **Šifra polja** садржи све унете шифре поља које су у скупу поља дефинисаних у оквиру концепта потпоља типа публикације (за одабрани тип публикације). Садржај ове графичке компоненте су све вредности колоне табеле *Def_tipa* у којој се чувају шифре поља придружене одабраном типу публикације. Одабрана шифра (на слици 3.1.6 шифра поља 001) из ове компоненте одговара садржају колоне табеле *Tipbibliotekara* у којој се чувају шифре поља.

Падајућа листа **Šifra potpolja** садржи све шифре потпоља које се односе на одабрану шифру поља и које су у скупу потпоља дефинисаних у оквиру концепта потпоља типа публикације (за одабрани тип публикације). Садржај ове графичке компоненте су све вредности колоне табеле *Def_tipa* у којој се чувају шифре потпоља одабраног поља придружене одабраном типу публикације. Одабрана шифра (на слици 3.1.6 шифра потпоља a) из ове компоненте одговара садржају колоне табеле *Tipbibliotekara* у којој се чувају шифре потпоља.

Potpolja bibliotekara

Šifra bibliotekara: Goga

Tip publikacije: M1 Monografije

Šifra polja: 001 Identifikator sloga

Šifra potpolja: a Stanje sloga

Podrazumevana vrednost: i

Prvi Prethodni Sledeći Poslednji Pregled Pomoć

Izmena Brisanje Potvrdi Odustani Izlaz

Слика 3.1.6 Прозор **Potpolja bibliotekara**

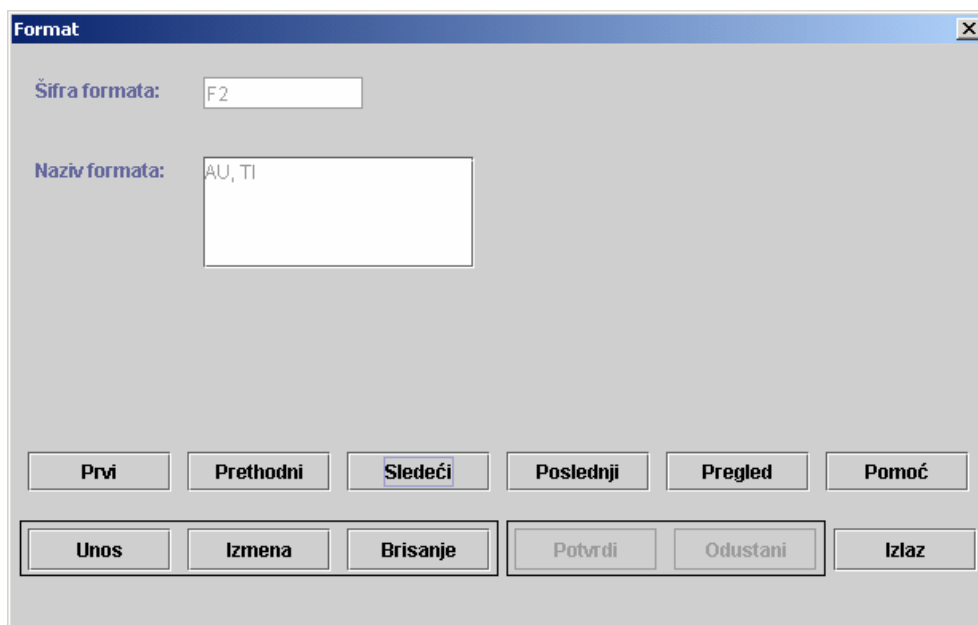
У једнолинијски едитор **Podrazumevana vrednost** уноси/приказује се подразумевана вредност потпоља. Садржај овог едитора је одговарајући садржај колоне *defvr_tipbib*.

На слици 3.1.6 приказана је подразумевана вредност *i* потпоља *a* поља *001* која је придружена библиотекару *Goga* за вредност типа публикације *M1*.

3.1.7 Прозор за ажурирање формата

Различити формати извештаја и прикази резултата претраживања базе података библиографске грађе могући су у систему уколико се дефинишу помоћу прозора **Format** који је приказан на слици 3.1.7. За сваки формат уноси се шифра формата и назив формата. Уколико је потребно, сваки формат у бази података могуће је избрисати или извршити измену његовог назива. Акције над прозором **Format** омогућавају ажурирање слогова табеле *Formatf*, као и навигацију кроз постојеће слокове те табеле.

У једнолинијски едитор **Šifra formata** уноси/приказује се шифра формата (максимално 6 знакова). Садржај овог едитора је одговарајући садржај колоне *formatid*.



Слика 3.1.7 Прозор **Format**

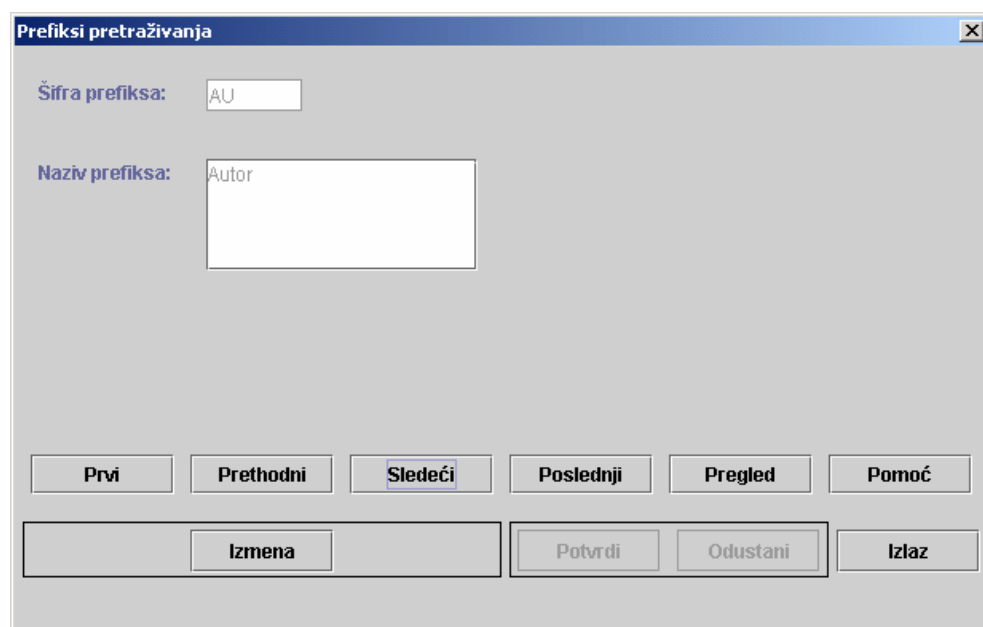
У вишелинијски едитор **Naziv formata** уноси/приказује се назив формата максималне дужине 160 знакова. Садржај овог едитора је одговарајући садржај колоне *for_opis*.

На слици 3.1.7 приказана је шифра формата *F2* чији назив је *AU, TI*.

3.1.8 Прозор за ажурирање префикса претраживања

Формирање упита над базом података библиографске грађе врши се коришћењем префикса претраживања. За сваки префикс претраживања дефинисана је шифра префикса и назив префикса. Библиотекар има могућност само да прегледа постојеће префиксе претраживања или да изврши измену назива префикса помоћу прозора **Prefiksi pretraživanja** који је приказан на слици 3.1.8. Акције над овим прозором омогућавају модификацију назива префикса, као и навигацију кроз постојеће слогове табеле *Poljebpr*.

У једнолинијском едитору **Šifra prefiksa** приказује се шифра префикса (2 знака). Овај едитор омогућава да се прикаже садржај колоне *poljebprid*.



Prefiksi pretraživanja

Šifra prefiksa: AU

Naziv prefiksa: Autor

Prvi Prethodni **Sledeći** Poslednji Pregled Pomoć

Izmena Potvrdi Odustani Izlaz

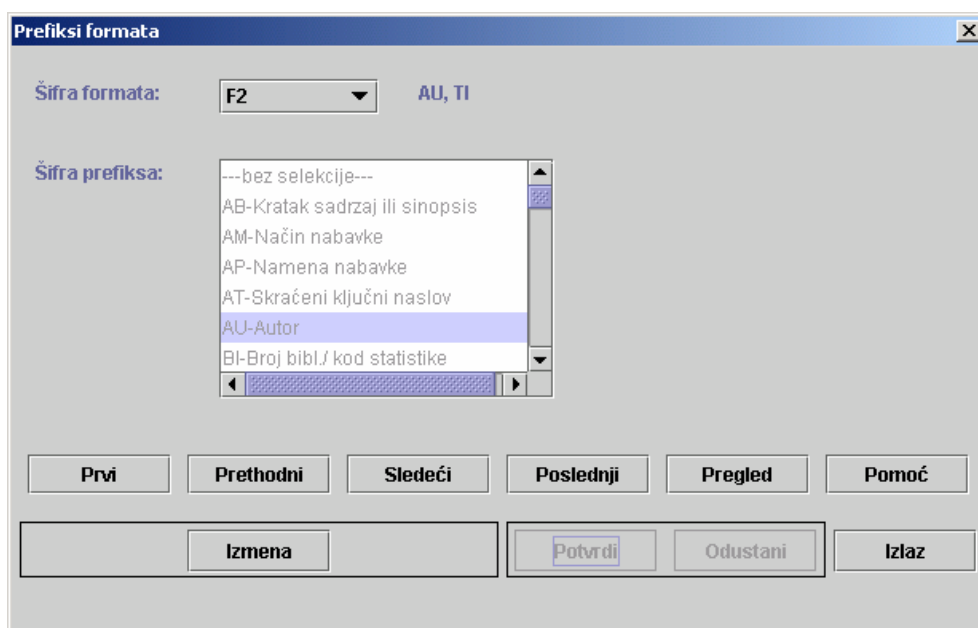
Слика 3.1.8 Прозор **Prefiksi pretraživanja**

У вишелинијском едитору **Naziv prefiksa** приказује се назив префикса максималне дужине 160 знакова. Овај едитор омогућава да се прикаже (и евентуално модификује) садржај колоне *pbpr_naziv*.

На слици 3.1.8 приказана је шифра префикса претраживања *AU*, чији назив је *Autor*.

3.1.9 Прозор за ажурирање префикса формата

Неки формати исписа се дефинишу помоћу скупа префикса претраживања. За сваки такав формат специфицирају се префикси који омогућавају приказ документа. Избор префикса за одређени формат врши се помоћу прозора **Prefiksi formata** који је приказан на слици 3.1.9. Уколико је потребно, постојећим префиксима претраживања неког формата могуће је додати нове или избацили постојеће префиксе. Акције над овим прозором омогућавају ажурирање слогова табеле *Formatf_poljebpr*, као и навигацију кроз постојеће слокове те табеле.



Слика 3.1.9 Прозор **Prefiksi formata**

Падајућа листа **Šifra formata** садржи све унете шифре формата. Садржај ове графичке компоненте су све вредности колоне *formatid* табеле *Formatf*. Одабрана шифра (на слици 3.1.9 шифра формата *F2*) из ове компо-

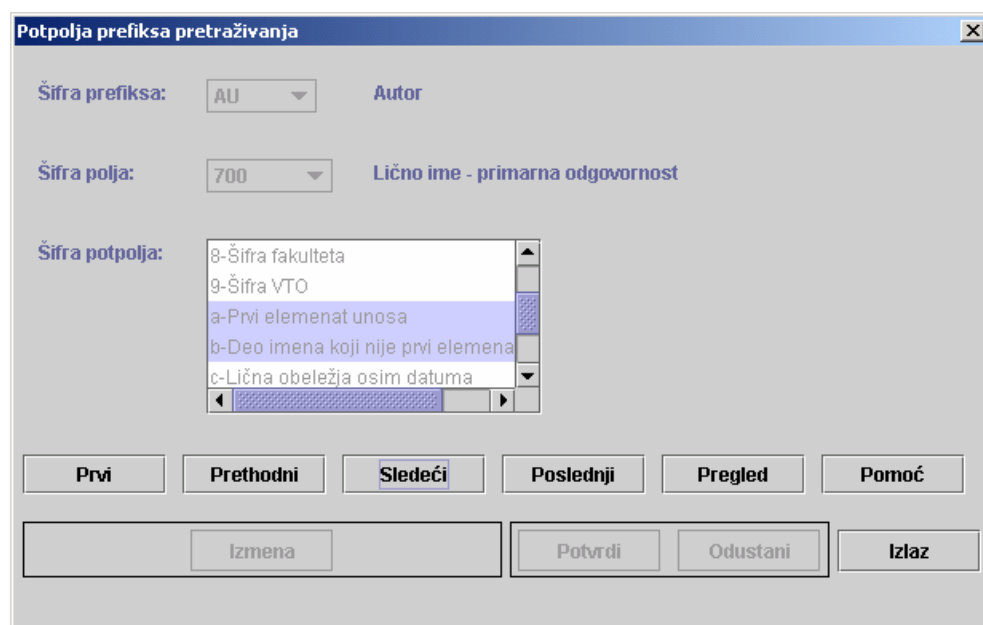
ненте одговара садржају колоне табеле *Formatf_poljebpr* у којој се чувају шифре формата.

Листа **Šifra prefiksa** садржи све унете шифре префикса претраживања као и њихове називе. Садржај ове графичке компоненте су све вредности колоне *poljebprid* табеле *Poljebpr*. Одабране шифре (на слици 3.1.9 шифра *AU* као једна од шифри префикса претраживања) из ове компоненте одговарају садржајима колоне табеле *Formatf_poljebpr* у којој се чувају шифре префикса претраживања.

На слици 3.1.9 приказана је шифра префикса претраживања *AU* као једна од шифри придружених формату *F2*.

3.1.10 Прозор за ажурирање потпоља префикса претраживања

За сваки префикс претраживања дефинисан је скуп потпоља YUMARC формата по којима се врши индексирање и претраживање базе података библиографске грађе. Библиотекар има могућност само да прегледа тај скуп потпоља за сваки префикс помоћу прозора **Potpolja prefiksa pretraživanja** који је приказан на слици 3.1.10. Акције над овим прозором омогућавају навигацију кроз постојеће слогево табеле *Poljebpr_potpolja*.



Слика 3.1.10 Прозор **Potpolja prefiksa pretraživanja**

Падајућа листа **Šifra prefiksa** садржи све унете шифре префикса претраживања. Садржај ове графичке компоненте су све вредности колоне *poljebprid* табеле *Poljebpr*. Приказана шифра префикса (на слици 3.1.10 шифра префикса *AU*) ове компоненте одговара садржају колоне табеле *Poljebpr_potpolja* у којој се чувају шифре префикса претраживања.

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. Приказана шифра (на слици 3.1.10 шифра поља *700*) ове компоненте одговара садржају колоне табеле *Poljebpr_potpolja* у којој се чувају шифре поља.

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. Поред тога, уз сваку шифру потпоља приказан је и назив потпоља. Приказане одабране шифре (на слици 3.1.10 шифре потпоља *a*, *b*) ове компоненте одговарају садржајима колоне табеле *Poljebpr_potpolja* у којој се чувају шифре потпоља.

На слици 3.1.10 приказане су шифре потпоља поља *700* које су придружене префиксу претраживања *AU*.

3.1.11 Прозор за ажурирање типа обраде

Библиотекар може да обрађује више различитих типова публикација на различитим нивоима обраде и обавезности, а све то може да се специфицира помоћу прозора **Tip obrade** који је приказан на слици 3.1.11. Осим тога, овде се уноси и пет префикса претраживања који се иницијално постављају за пријављеног библиотекара у апликацији за обраду библиографске грађе. Такође се, на овом прозору, поставља подразумевани формат исписа за пријављеног библиотекара. Акције над прозором **Tip obrade** омогућавају ажурирање слогова табела *Tip_obrade* и *Poljebpr_tip_obrade*, као и навигацију кроз постојеће слокове тих табела.

Падајућа листа **Šifra bibliotekara** садржи све унете шифре библиотекара. Садржај ове графичке компоненте су све вредности колоне *bibid* табеле *Bibliotekar*. Одабрана шифра (на слици 3.1.11 шифра библиотекара *Goga*) из ове компоненте одговара садржају колоне табеле *Tip_obrade* (у којој се чувају шифре библиотекара), која настаје на основу везе те табеле са табелом *Bibliotekar*.

Падајућа листа **Tip publikacije** садржи све вредности типа публикације за које је дефинисан бар један ниво обраде и ниво обавезности. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obaveznosti* у којој се чувају вредности типа публикације. Одабрана вредност (на слици 3.1.11 вредност *S1*) из ове компоненте одговара садржају колоне табеле *Tip_obrađe* (у којој се чувају вредности типа публикације), која настаје на основу везе те табеле са табелом *Nivo_obaveznosti*.

Слика 3.1.11 Прозор **Tip obrade**

Падајућа листа **Nivo obrade** садржи све унете вредности нивоа обраде за одабрани тип публикације, за које је дефинисан бар један ниво обавезности. Садржај ове графичке компоненте су све вредности колоне табеле *Nivo_obaveznosti* у којој се чувају вредности нивоа обраде за одабрани тип публикације. Одабрана вредност (на слици 3.1.11 вредност *10*) из ове компо-

ненте одговара садржају колоне табеле *Tip_obrade* (у којој се чувају вредности нивоа обраде), која настаје на основу везе те табеле са табелом *Nivo_obaveznosti*.

Падајућа листа **Nivo obaveznosti** садржи све унете вредности нивоа обавезности које припадају одабраном типу публикације и нивоу обраде. Садржај ове графичке компоненте су све вредности колоне *nobavid* табеле *Nivo_obaveznosti* придружене одабраном типу публикације и нивоу обраде. Одабрана вредност (на слици 3.1.11 вредност *I*) из ове компоненте одговара садржају колоне табеле *Tip_obrade* (у којој се чувају вредности нивоа обавезности), која настаје на основу везе те табеле са табелом *Nivo_obaveznosti*.

За сваког одабраног библиотекара се на екранској форми исписују вредност типа публикације, вредности нивоа обраде и нивоа обавезности које представљају подразумеване вредности за тог библиотекара.

Падајућа листа **Šifra formata** садржи све унете шифре формата. Садржај ове графичке компоненте су све вредности колоне *formatid* табеле *Formatf*. Одабрана вредност (на слици 3.1.11 вредност *FULL*) из ове компоненте одговара садржају колоне табеле *Tip_obrade* (у којој се чувају шифре формата), која настаје на основу везе те табеле са табелом *Formatf*.

Листа **Šifra prefiksa** садржи све унете шифре префикса претраживања као и њихове називе. Садржај ове графичке компоненте су све вредности колоне *poljebprid* табеле *Poljebpr*. Одабране вредности (на слици 3.1.11 шифра *SN* као једна од пет шифри префикса претраживања) из ове компоненте одговарају садржајима колоне табеле *Poljebpr_tip_obrade* у којој се чувају шифре префикса претраживања за одређени тип обраде.

На слици 3.1.11 приказан је тип обраде за библиотекара са шифром *Goga*. Вредност типа публикације је *SI*, нивоа обраде *I0* и нивоа обавезности *I*. Шифра формата је *FULL*, а шифра префикса *SN* је једна од 5 шифри префикса претраживања које се морају одабрати приликом дефинисања типа обраде за одабраног библиотекара.

Команде библиотечког софтверског система

Команде библиотечког софтверског система уведене су да би се библиотекарима обезбедио комфорнији начин рада при обради библиографске грађе. У тези [Хол97] приказан је модел команди на основу којег је дат предлог за имплементацију команди у софтверском систему БИСИС вер. 1. У раду [Рур00а] приказане су почетне идеје објектно-оријентисане спецификације команди, док је у тези [Пуп00б] дата детаљна објектно-оријентисана спецификација команди софтверског система БИСИС вер. 3. Као резултати рада на тези објављени су радови [Рур01а, Рур01б, Рур01с]. Објектно-оријентисана спецификација команди софтверског система БИСИС вер. 3 предмет је и овог поглавља.

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 библиотекар 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 у командну линију. 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 монографије [Сур04а] и у упутству за коришћење библиотечког софтверског система БИСИС вер. 3 [Сур03].

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 базе података библиографске грађе (употребом команде *edit*) који 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 наводе 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 записа који се обрађује у изабраном формату. 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 параметар 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 искључиво 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 наводе 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 претраживања, међусобно одвојених зарезима. Овако дефинисан формат омогућава испис записа по префиксима претраживања 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 библиотекар може да изабере за испис записа. Ову команду могуће је применити у моду претраживања и у моду директоријума.

Команда *list* омогућава испис извештаја (каталожних 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 каталожних листића. 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 који се обрађује. 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 за библиотекара који је преузет приликом иницијализације параметара апликације, или је постављен у току извршавања апликације. Постављање текућих вредности типа извештаја у току извршавања апликације врши се у моду претраживања тако што се уз команду *list* као параметар наведе знак ?. Тада се библиотекар приказу сви могући типови извештаја, а он је у могућности да изабере један од њих.

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.

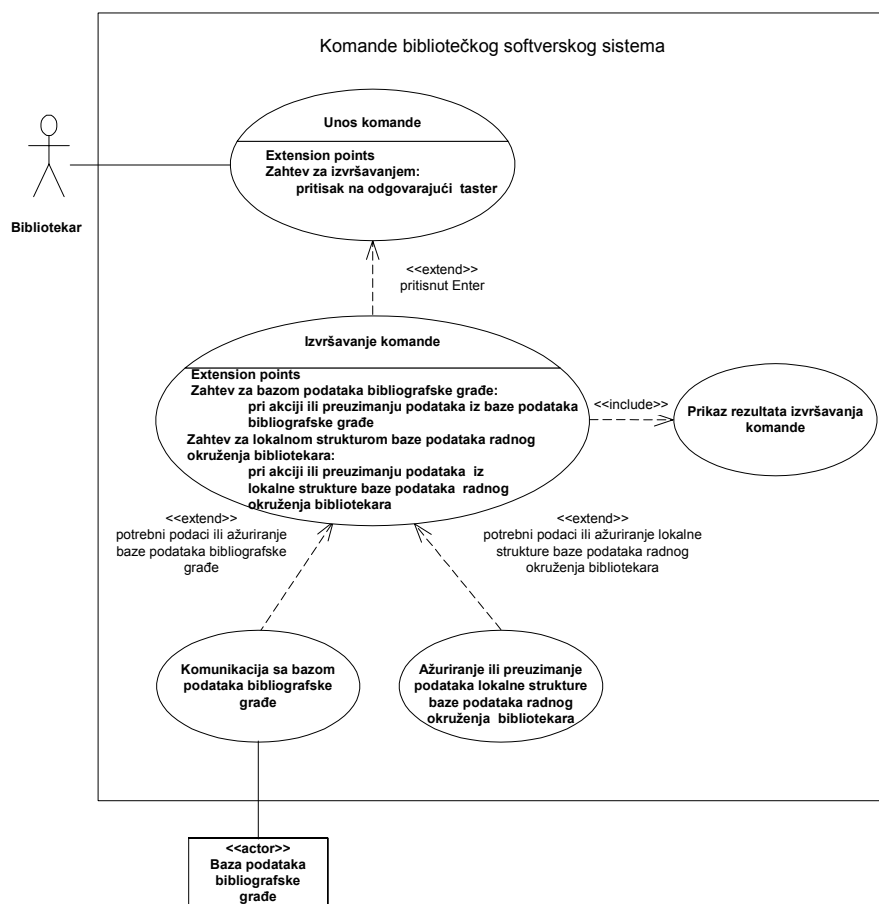
Команда *print* омогућава да се библиотекарима прикажу сложени извештаји, као што су инвентарна књига, библиографија, билтен приновљених књига. Ова команда се може применити искључиво у моду претраживања, без параметара. Детаљнији опис функционалности ове команде описан је у Поглављу 7 монографије [Сур04а] и у упутству за коришћење библиотечког софтверског система БИСИС вер. 3 [Сур03].

Команда *quit* се може применити у моду претраживања и у моду директоријума. Функционалност команде зависи од мода извршавања и од опције коју библиотекар изабере. Ако се команда зада у моду претраживања, библиотекар може да бира између две опције: опција којом се излази из апликације за обраду библиографске грађе и опција којом се остаје у овом моду. Ако се команда зада у моду директоријума библиотекар, такође, може да бира између две опције: опција којом се прелази из мода директоријума у мод претраживања, а запис који се обрађује неће бити снимљен у базу података библиографске грађе и опција којом се остаје у овом моду.

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 библиотекара који обрађује запис. Приликом задавања ове команде, пре снимања записа у базу података библиографске грађе, библиотекар се приказују основни подаци записа који се обрађује (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 наводе се параметри.

4.1 Дијаграм случајева коришћења команди

У овом одељку дат је модел случајева коришћења команди софтверског система БИСИС вер. 3. Модел случајева коришћења представља опис функционалности система с тачке гледишта корисника система, у коме се изостављају детаљи везани за начин реализације посматране функционалности. Моделом случајева коришћења дефинише се граница посматраног система у смислу одређивања учесника који комуницирају са системом, функција које ти учесници могу иницирати и у чијем извршавању учествују, као и порука и података које учесници размењују са системом. Дефинисањем границе система јасно се издвајају функције које посматрани систем обухвата.



Слика 4.1 Дијаграм случајева коришћења команди

Модел случајева коришћења команди представљен је дијаграмом случајева коришћења који је приказан на слици 4.1. 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 lokalne strukture baze podataka radnog okruženja bibliotekara*, као и учесници *Bibliotekar* и *Baza podataka bibliografske građe*. За сваки уочени случај коришћења дат је текстуални опис којим су детаљно описане функције система.

Случај коришћења: *Unos komande*

Кратак опис: Унос команде у командну линију.

Учесници: *Bibliotekar*

Услови који морају бити задовољени пре извршавања: Библиотекар се током рада у апликацији за обраду библиографске грађе позиционирао на командну линију притиском на тастер *F6* или уз помоћ миша.

Опис: Када се позиционирао на командну линију библиотекар уноси команду и по потреби параметре команде. Библиотекар може да одустане од уноса команде притиском на тастер *Esc* или да мишем пренесе фокус са командне линије на неку другу графичку компоненту. Уколико је задовољен специфициран услов *pritisnut Enter* извршава се случај коришћења *Izvršavanje komande* [Изузетак: Притиснут тастер *Esc* или мишем пренет фокус на другу графичку компоненту].

Изузеци:

[Притиснут тастер *Esc* или мишем пренет фокус на другу графичку компоненту] Уколико је библиотекар притиснуо тастер *Esc* или је мишем пренео фокус на неку другу графичку компоненту прекида се извршавање овог случаја коришћења, односно библиотекар одустаје од уноса команде.

Услови који морају бити задовољени после извршавања: Команда је унета и библиотекар је притиснуо тастер *Enter*.

Случај коришћења: *Izvršavanje komande*

Кратак опис: Извршавање команде стартовано притиском на тастер *Enter* у командној линији.

Учесници: нема

Услови који морају бити задовољени пре извршавања: Библиотекар је задао команду у командној линији и притиснуо тастер *Enter*.

Опис: Врши се парсирање садржаја командне линије. 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 lokalne strukture baze podataka radnog okruženja bibliotekara* [Изузетак: Команда неисправно задата] [Изузетак: Неуспешно извршавање команде].

Изузеци:

[Команда неисправно задата] Уколико је команда неисправно задата приказује се одговарајућа порука библиотекару и прекида се извршавање команде.

[Неуспешно извршавање команде] Уколико је извршавање команде неуспешно приказује се одговарајућа порука библиотекару.

Услови који морају бити задовољени после извршавања: Команда је извршена било успешно или не.

Случај коришћења: *Prikaz rezultata izvršavanja komande*

Кратак опис: Приказују се резултати извршавања команде коју је библиотекар унео, било оно успешно или не.

Учесници: нема

Услови који морају бити задовољени пре извршавања: Команда је извршена било успешно или неуспешно.

Опис: Приказују се резултати извршавања команде библиотекару. 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 погрешној синтакси којом је библиотекар задао команду.

Изузеци: нема

Услови који морају бити задовољени после извршавања: Резултати извршавања су приказани библиотекару.

Случај коришћења: Ažuriranje ili preuzimanje podataka lokalne strukture baze podataka radnog okruženja bibliotekara

Кратак опис: Комуникација са локалном структуром базе података радног окружења библиотекара ради преузимања или ажурирања података из ове структуре.

Учесници: нема

Услови који морају бити задовољени пре извршавања: За извршавање команде потребне су акције над подацима или сами подаци локалне структуре базе података радног окружења библиотекара. Локална структура базе података радног окружења библиотекара је успешно формирана приликом стартовања апликације за обраду библиографске грађе.

Опис: У случају извршавања неких од команди врши се преузимање података из локалне структуре базе података радног окружења библиотекара ради добијања додатних података у циљу извршавања тих команди. Неке од команди врше ажурирање података те структуре у циљу даљег рада библиотекара у апликацији.

Изузеци: нема

Услови који морају бити задовољени после извршавања: Локална структура базе података радног окружења библиотекара је успешно ажурирана или су подаци преузети из те структуре.

Случај коришћења: Komunikacija sa bazom podataka bibliografske građe

Кратак опис: Комуникација са базом података библиографске грађе ради преузимања или ажурирања података ове базе.

Учесници: Baza podataka bibliografske građe

Услови који морају бити задовољени пре извршавања: За извршавање команде потребне су акције над подацима или сами подаци базе података библиографске грађе.

Опис: У случају извршавања неких од 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 [Изузеци: Немогуће остварити комуникацију са базом података библиографске грађе].

Изузеци:

[Немогуће остварити комуникацију са базом података библиографске грађе] Уколико није могуће остварити конекцију са базом података библиографске грађе или уколико се конекција раскине током извршавања команде.

Услови који морају бити задовољени после извршавања: Запис из базе података библиографске грађе је успешно ажуриран или су записи преузети из те базе.

4.2 Дијаграм класа команди

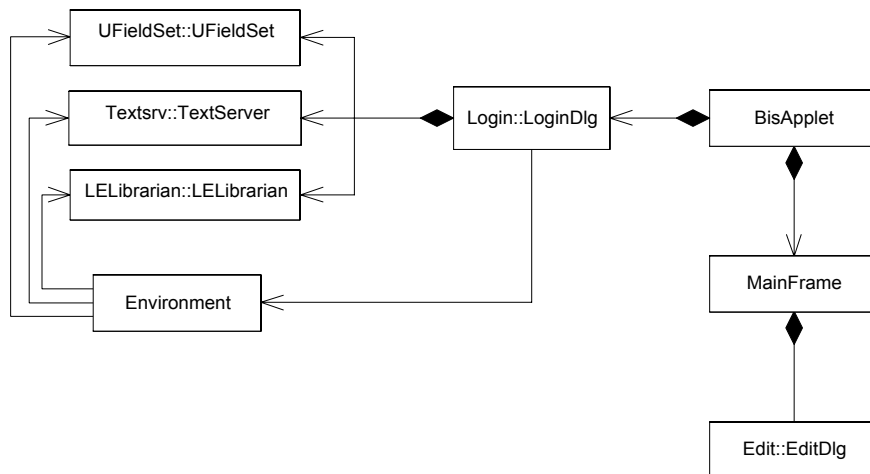
Дијаграм класа служи за приказивање статичке структуре система. Овај дијаграм приказује један поглед на систем, којим су обухваћене класе система, њихове структуре и везе између њих. Дијаграм класа се користи за приказивање улога и одговорности које ентитети, представљени класама, имају у систему. Такође, приказују се и везе између посматраних ентитета којима се описују различите законитости и ограничења која важе у систему. Дијаграм класа представља колекцију елемената модела, као што су класе, интерфејси, пакети и њихове међусобне везе.

Основна идеја приликом моделирања дијаграма класа команди била је да се искористи *Command Pattern* као један од дизајн патерна [Gam95]. Дизајн патерни описују једноставно и елегантно решење за одређене проблеме у области објектно-оријентисаног дизајна. Они омогућавају да дизајн буде флексибилнији, модуларан, поново употребљив и лак за разумевање. По узору на *Command Pattern* приликом моделирања дијаграма класа команди креирана је апстрактна класа са апстрактном операцијом *execute*. Све класе које имплементирају конкретне команде наслеђују ову апстрактну класу и реализују операцију *execute*. Приликом имплементације искоришћен је принцип полиморфизма који подржава програмски језик Јава.

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 4.2.

Стартна класа апликације за обраду библиографске грађе софтверског система БИСИС вер. 3 је класа *BisApplet*. Из ове класе приступа се објекту класе *LoginDlg* која припада пакету *Login* и који 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 у апликацији као што су, на пример, референце на објекте који омогућавају рад са локалним структурама, текст сервером и други.



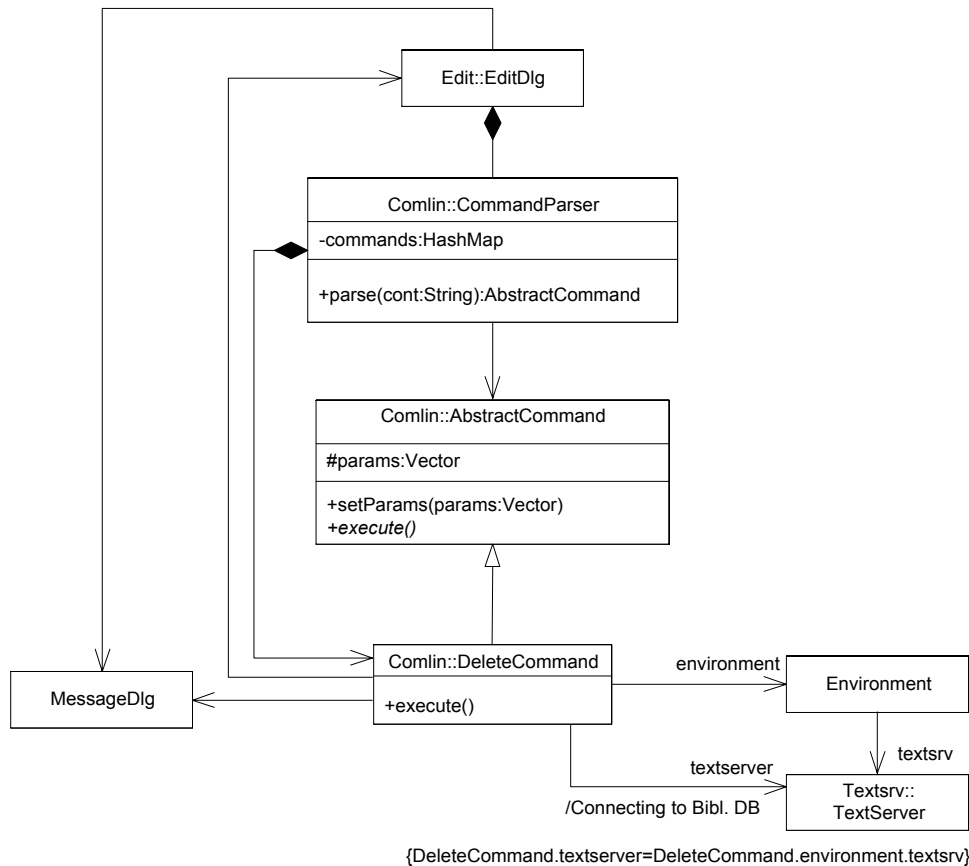
Слика 4.2 Дијаграм класа заједнички за све команде

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.

4.2.1 Дијаграм класа команде *delete*

Команда *delete*, као што је већ речено, може да се извршава у моду директоријума. Дијаграм класа команде *delete* приказан је на слици 4.3. Објекат класе *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. Овако креирани објекти чувају се у структури *HashMap*, где се налазе парови (назив_команде, објекат_команде).

Парсирање садржаја командне линије извршава се у оквиру операције *parse* класе *CommandParser*. Приликом парсирања, одваја се назив команде од параметара. Уколико је конкретна команда пронађена у структури *HashMap* на основу свог назива, операцијом *setParams* класе *AbstractCommand* постављају се параметри у променљиву *params* тако да буду доступни класи која наслеђује ову класу и имплементира конкретну команду. 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*.



Слика 4.3 Дијаграм класа команде *delete*

Како је приликом извршавања команде *delete* потребно имати везу са базом података библиографске грађе, класа *DeleteCommand* повезана је са

класом *Environment* која чува референцу на објекат класе *TextServer* који омогућава акције над том базом.

Веза класе *DeleteCommand* 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*.

4.2.2 Дијаграм класа команде *display*

Команда *display*, као што је већ речено, може да се извршава у моду претраживања као и у моду директоријума. Дијаграм класа команде *display* која се извршава у моду претраживања приказан је на слици 4.4.

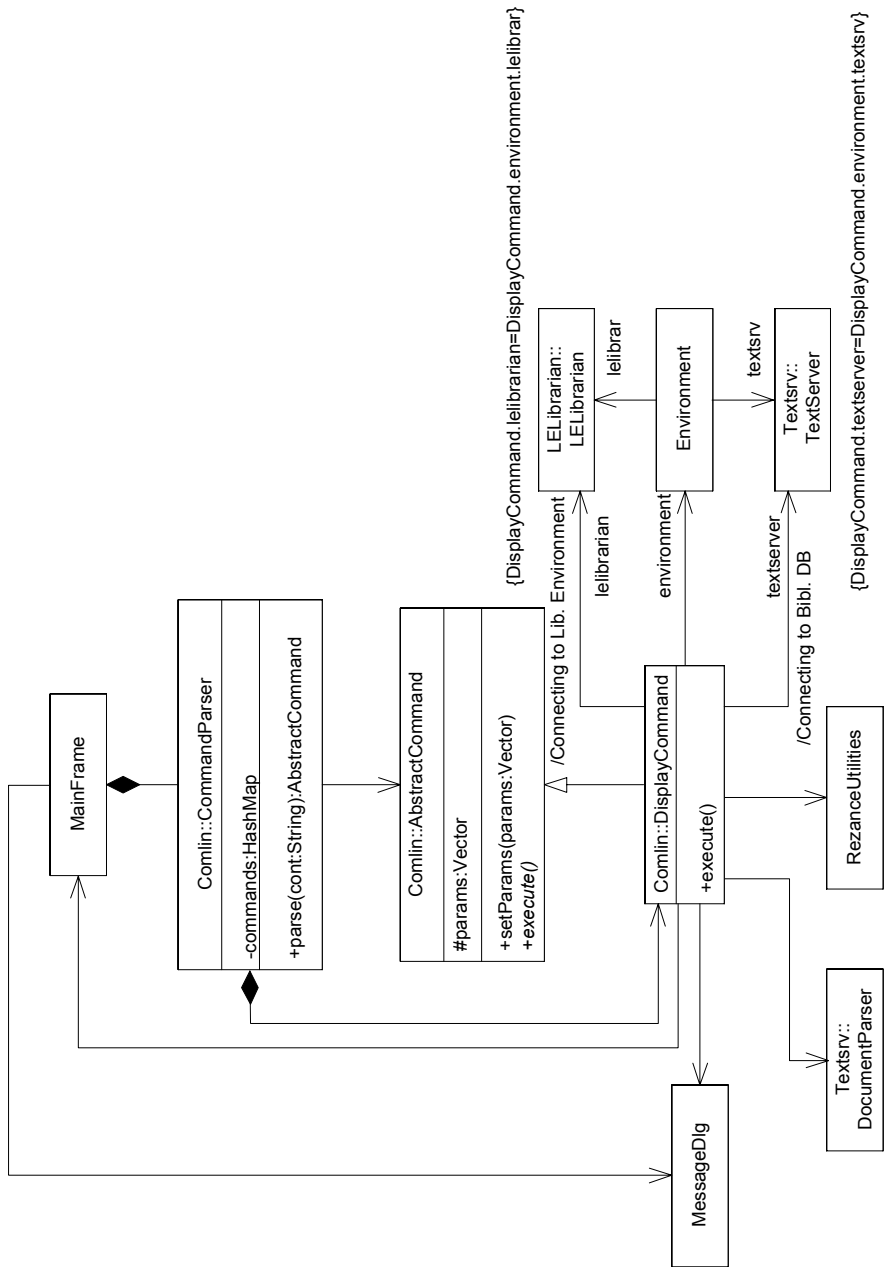
Улога класа *CommandParser* и *AbstractCommand* описана је у одељку 4.2.1. Једина разлика је што се објекат типа *CommandParser* креира у оквиру класе *MainFrame* (која имплементира мод претраживања).

Како је приликом извршавања команде *display* потребно имати везу са базом података библиографске грађе и са локалном структуром у којој се налазе подаци базе података радног окружења библиотекара, класа *DisplayCommand* повезана је са класом *Environment*. Ова класа чува референцу на објекат класе *TextServer* који омогућава акције над базом података библиографске грађе, као и референцу на објекат класе *LELibrarian* који омогућава акције над поменутом локалном структуром.

Ради формирања резултата извршавања ове команде класа *DisplayCommand* повезана је са класама *DocumentParser* (за формат исписа различит од *FULL*) и *RezanceUtilities* (за формат исписа постављен на *FULL*).

Веза класе *DisplayCommand* са класом *MainFrame* формирана је да би се одговарајућом операцијом испунила графичка компонента резултатом извршавања команде и приказала библиотекар.

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



Слика 4.4 Дијаграм класа команде *display*

4.2.3 Дијаграм класа команде *new*

Команда *new*, као што је већ речено, може да се извршава у моду претраживања. Дијаграм класа команде *new* дат је на слици 4.5.

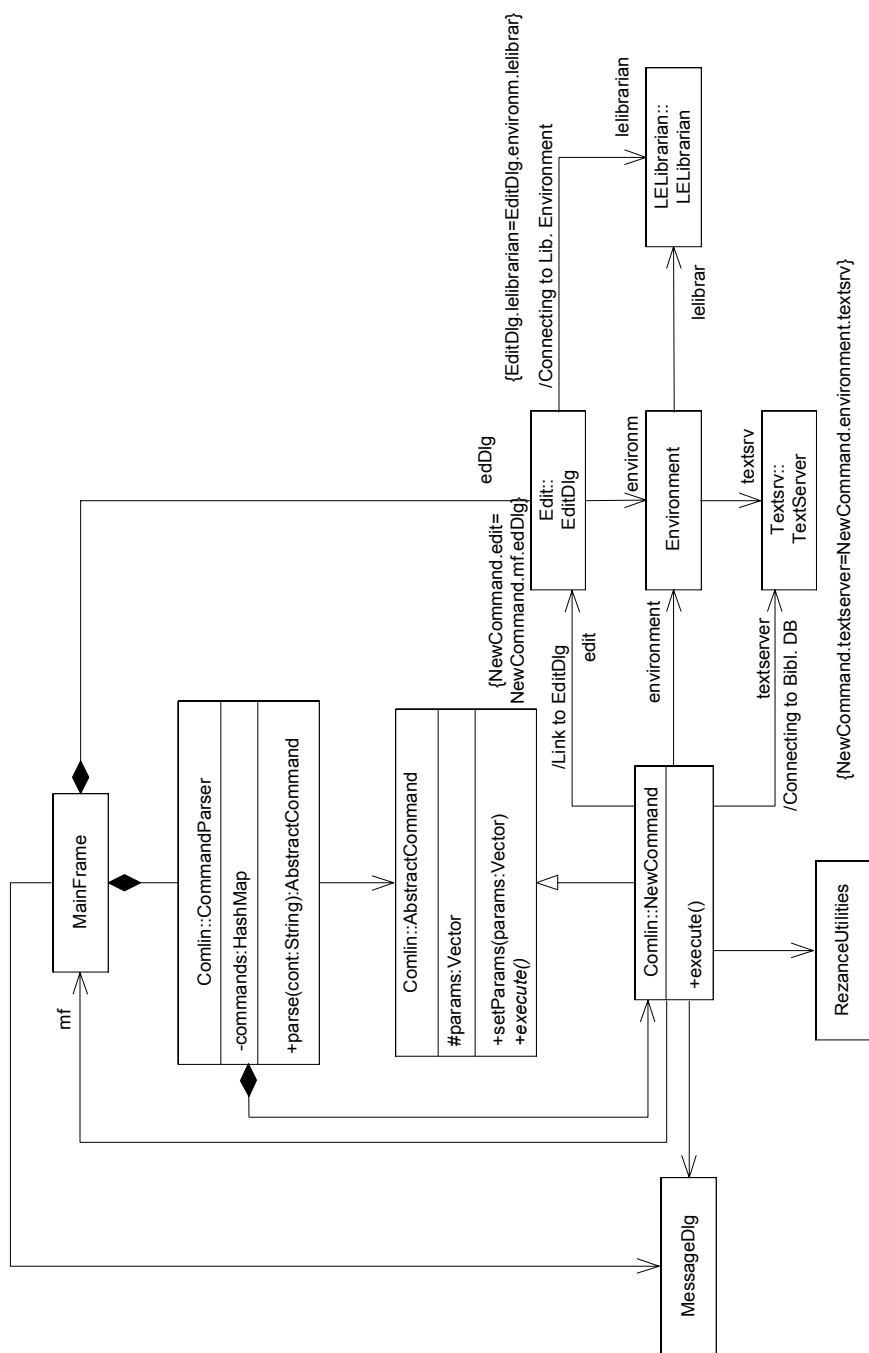
Улога класа *CommandParser* и *AbstractCommand* описана је у одељку 4.2.1. Једина разлика је што се објекат типа *CommandParser* креира у оквиру класе *MainFrame* (која имплементира мод претраживања).

Како је приликом извршавања команде *new* уз коју се задаје параметар потребно имати везу са базом података библиографске грађе, класа *NewCommand* повезана је са класом *Environment*. Класа *Environment* чува референцу на објекат класе *TextServer* који омогућава акције над том базом.

Веза класе *NewCommand* са класом *RezanceUtilities* формирана је да би се користиле операције ове класе које омогућавају распакивање записа у структуру погодну за брз приступ садржајима потпоља и приказивање те структуре библиотекару. Распакивање записа се врши када се нови запис креира на основу узорка, односно записа који постоји у бази података библиографске грађе.

Објекат класе *EditDlg* омогућава позив одговарајуће операције која на основу података локалне структуре у којој се налазе подаци базе података радног окружења библиотекара формира/прилагођава структуру новог записа. Осим тога, приликом приказивања ове структуре библиотекару позивом операције класе *EditDlg* испуњава се одговарајућа графичка компонента резултатом извршавања ове команде.

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



Слика 4.5 Дијаграм класа команде *new*

4.3 Динамички модел команди

Динамичким моделом описују се динамичке карактеристике система, односно појединачно понашање објеката у систему, као и понашање комплетног система који се моделира. За приказ динамичких карактеристика система користе се:

- дијаграм секвенци,
- дијаграм сарадње,
- дијаграм стања,
- дијаграм активности.

Дијаграм секвенци описује начин на који објекти у систему међусобно комуницирају, остварујући на тај начин очекивано понашање. На њему су приказане две димензије: вертикална, која означава време, и хоризонтална, која означава објекте. Овај дијаграм може се користити за спецификацију временских захтева и опис сложених сценарија. Могуће је описати ток порука између објеката којим се реализује одговарајућа операција у систему. Акценат је дат на опису секвенце порука, односно на опису порука и њиховом редоследу.

Дијаграм сарадње описује комуникацију између објеката, али и везе између објеката, неопходне за остваривање посматране комуникације. Дијаграм сарадње представља један дијаграм објеката на коме су, поред објеката и веза између њих, приказане и поруке које објекти међусобно прослеђују, остварујући на тај начин очекивано понашање. Дијаграмом сарадње могуће је описати скуп објеката, везе између њих, поруке које шаљу један другом, као и податке које међусобно размењују.

Дијаграм стања описује стања у којима се објекти посматране класе могу налазити током животног века, као и понашање у тим стањима и догађаје који условљавају прелазак објекта из једног стања у друго. Дијаграмом стања могуће је описати сва стања једног објекта током његовог животног века. Тај опис обухвата активности које се извршавају у појединим стањима, акције које се извршавају при преласку из једног стања у друго, као и поруке које условљавају промену стања посматраног објекта.

Дијаграм активности приказује комуникацију између објеката, али је акценат на активностима које се извршавају током посматране комуникације. Посматрају се активности које објекти извршавају, као и редослед тих активности. Дијаграм активности може се придружити пакету, класи, односно имплементацији њене операције или случају коришћења, описујући

акције и активности које се извршавају у оквиру њих, те токове контроле и података који постоје између моделираних акција и активности.

За сваку од команди које су наведене у уводном делу овог поглавља могуће је креирати динамичке дијаграме у циљу комплетне спецификације њихове функционалности. У наредним одељцима приказани су динамички дијаграми команди *delete*, *display* и *new*. Дијаграми за остале команде формирају се на сличан начин.

4.3.1 Динамички модел команде *delete*

У овом одељку дати су дијаграми сарадње, секвенци, стања и активности којима је описана динамика приликом извршавања команде *delete*. За сваки дијаграм дат је текстуални опис који га детаљно описује.

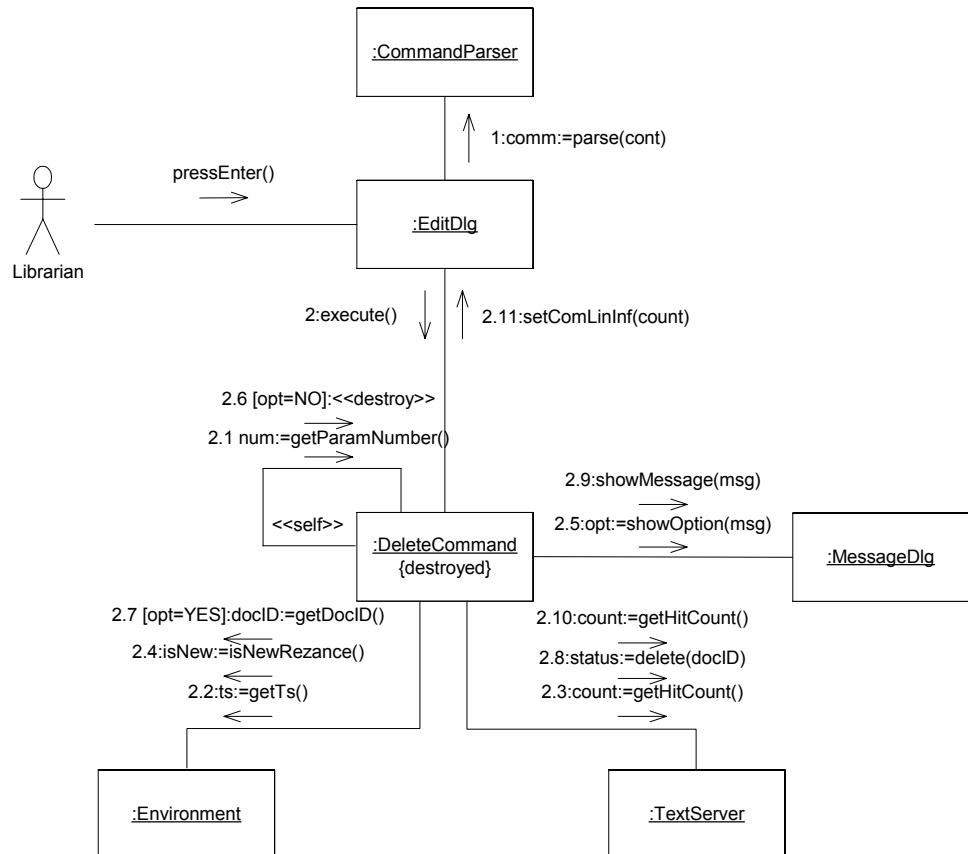
4.3.1.1 Дијаграм сарадње команде *delete*

На слици 4.6 приказан је дијаграм сарадње команде *delete*. Извршавање команде почиње када библиотекар притисне тастер *Enter* у командној линији, приказано на дијаграму прослеђивањем поруке *pressEnter* која се од библиотекара шаље објекту класе *EditDlg*. Након тога, шаље се порука *parse* објекту класе *CommandParser* у оквиру које се парсира садржај командне линије, тако што се одваја назив команде од параметара. Након парсирања, шаље се порука *execute* објекту класе *DeleteCommand* којом се иницира извршавање команде *delete*.

У оквиру поруке *execute* шаље се низ порука које представљају позиве операција, а које омогућавају извршавање ове команде. Поруком *getParamNumber* преузима се број параметара који су издвојени приликом парсирања садржаја командне линије. Након тога, шаље се порука *getTs* која омогућава да се из класе *Environment* преузме референца на објекат класе *TextServer* који обезбеђује везу са базом података библиографске грађе. Како извршавању команде *delete* претходи успешно претраживање базе података библиографске грађе, прослеђује се порука *getHitCount* објекту класе *TextServer* која омогућава да се преузме број погодака претходног претраживања. Након преузимања броја погодака, порука *isNewRezance* омогућава да се провери статус записа у циљу провере да ли је команди *delete* претходила команда *new* (запис је новоформиран). Затим се шаље порука *showOption* објекту класе *MessageDlg* која омогућава избор опције, тако да библиотекар може да одустане или потврди брисање записа. Уколико библиотекар одустане од брисања, завршава се са извршавањем команде *delete*. Уколико је би-

библиотекар потврдио брисање, прослеђивањем поруке *getDocID* добија се идентификациони број записа који се брише и прослеђивањем поруке *delete* објекту класе *TextServer* која као параметар има тај идентификациони број брише се запис из базе података библиографске грађе. Након брисања записа, прослеђује се порука *showMessage* објекту класе *MessageDlg* на основу које ће библиотекар добити информацију о томе да ли је запис успешно обрисан или је брисање било неуспешно.

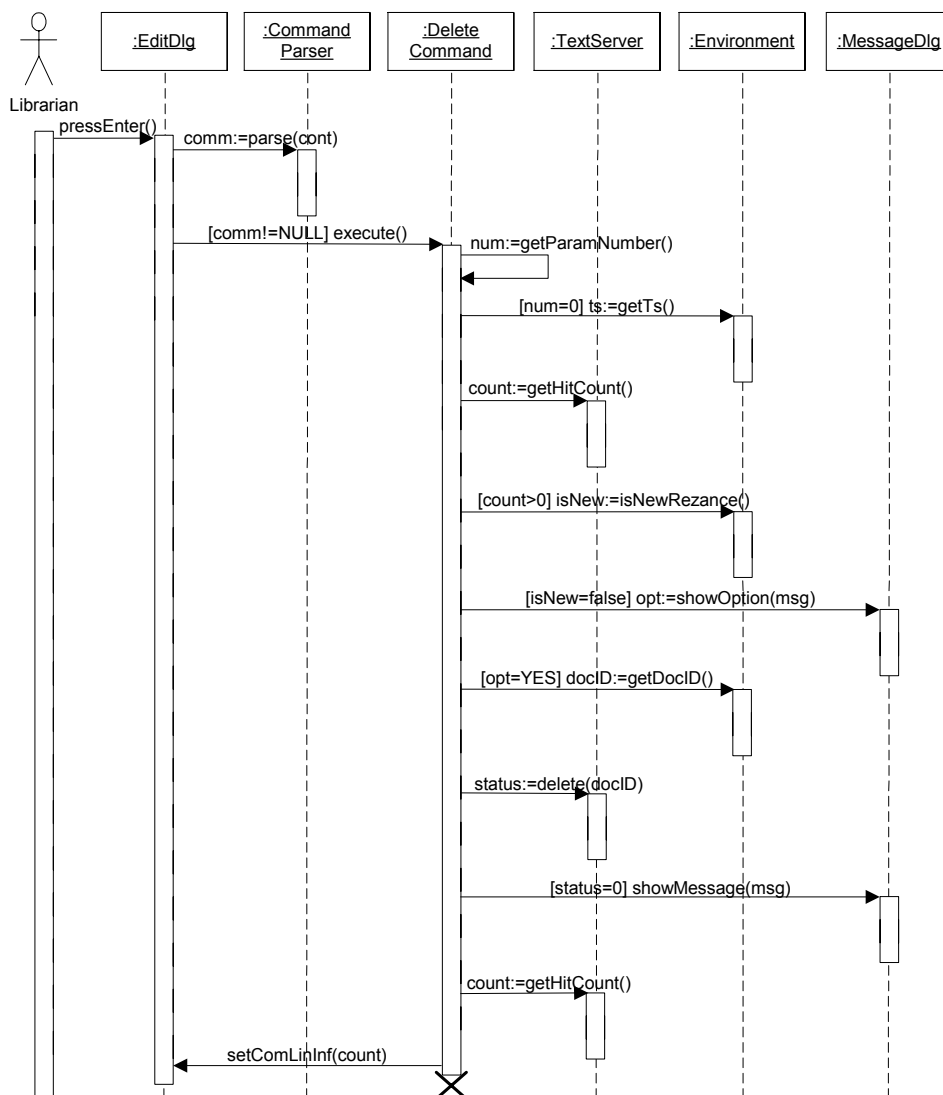
Како је након извршавања команде *delete* број погодака мањи за један, поруком *getHitCount* преузима се број погодака након извршавања команде *delete*, а поруком *setComLinInf* се омогућава да се та информација прикаже библиотекару на прозору који реализује мод претраживања.



Слика 4.6 Дијаграм сарадње команде *delete*

4.3.1.2 Дијаграм секвенци команде *delete*

На слици 4.7 приказан је дијаграм секвенци за један могући сценарио извршавања команде *delete*.



Слика 4.7 Дијаграм секвенци команде *delete*

Овим сценариом приказује се извршавање команде *delete* у случају да су задовољени сви услови потребни за успешно извршавање команде. На овом дијаграму акценат је на временској компоненти, односно хронологији слања порука између објеката. Поруке које се шаљу од објекта пошилаоца до објекта примаоца представљају позив извршавања операције објекта примаоца. Детаљи који се односе на везе између објеката приказани су на дијаграму сарадње, датом на слици 4.6.

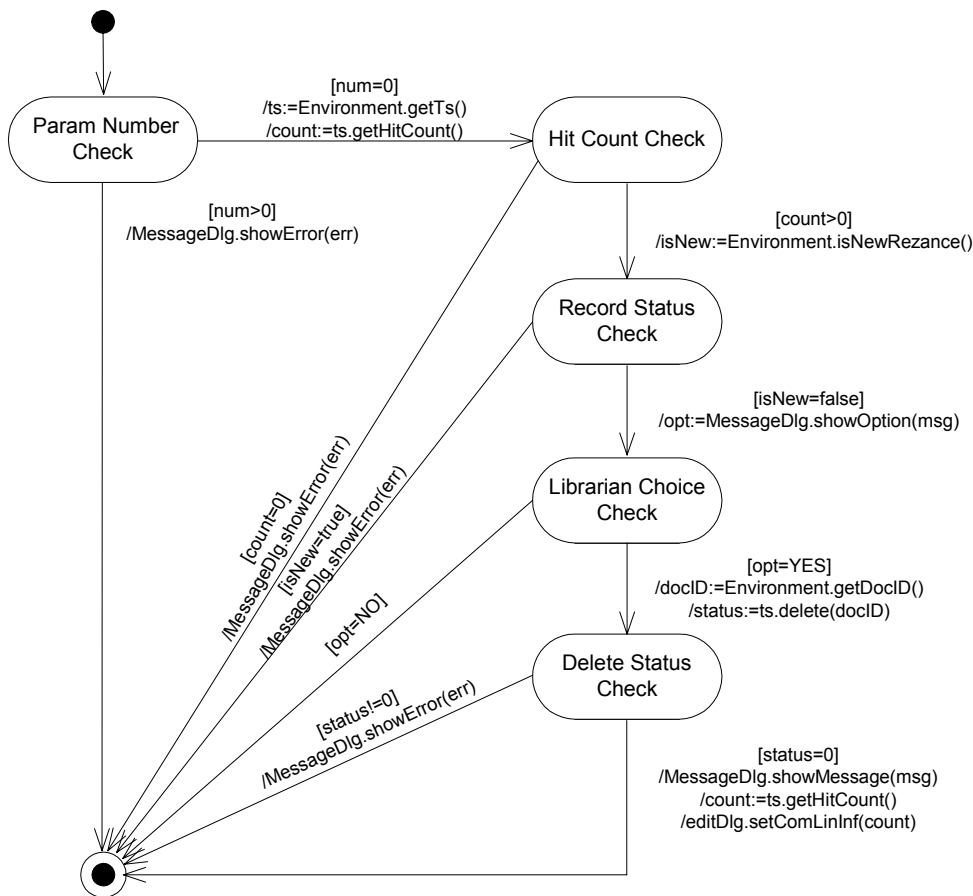
Након слања поруке *pressEnter* којом библиотекар иницира извршавање ове команде и поруке *parse* која омогућава парсирање садржаја командне линије, порука *execute* се извршава под условом да је библиотекар задао коректан назив команде и да команда припада скупу команди које се могу извршити у овом моду (мод директоријума). Порука *getTs* се прослеђује под условом да библиотекар није унео ниједан параметар, односно да порука *getParamNumber* враћа вредност нула. Порука *isNewRezance* се прослеђује под условом да је број погодака који враћа порука *getHitCount* већи од нуле. Прослеђивању поруке *showOption* претходи услов да запис није новоформиран. Порука *getDocID* прослеђује се под условом да је библиотекар потврдио брисање. Прослеђивању поруке *showMessage* претходи услов да је брисање било успешно, односно да порука *delete* враћа вредност нула. Након тога, прослеђује се порука *getHitCount*, а за њом порука *setComLinInf*.

4.3.1.3 Дијаграм стања команде *delete*

Дијаграм стања на слици 4.8 приказује сва могућа стања објекта класе *DeleteCommand*, као и акције које се дешавају приликом преласка објекта из једног стања у друго.

У стању *Param Number Check* објекта класе *DeleteCommand* проверава се број параметара које је библиотекар задао приликом уноса команде у командну линију. Ако је библиотекар унео параметар, прелази се у завршно стање и извршава се акција *showError* која библиотекару приказује одговарајућу поруку о грешци. Ако библиотекар није унео параметар, извршавају се акције *getTs* и *getHitCount*, а објекат прелази у стање *Hit Count Check*. У стању *Hit Count Check* проверава се број погодака претходног претраживања. Ако је провера погодака неуспешна, односно ако погоци не постоје, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. Ако је провера погодака успешна, извршава се акција *isNewRezance* и објекат прелази у стање *Record Status Check*. Уколико је запис новоформиран, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци, а у супротном извршава се

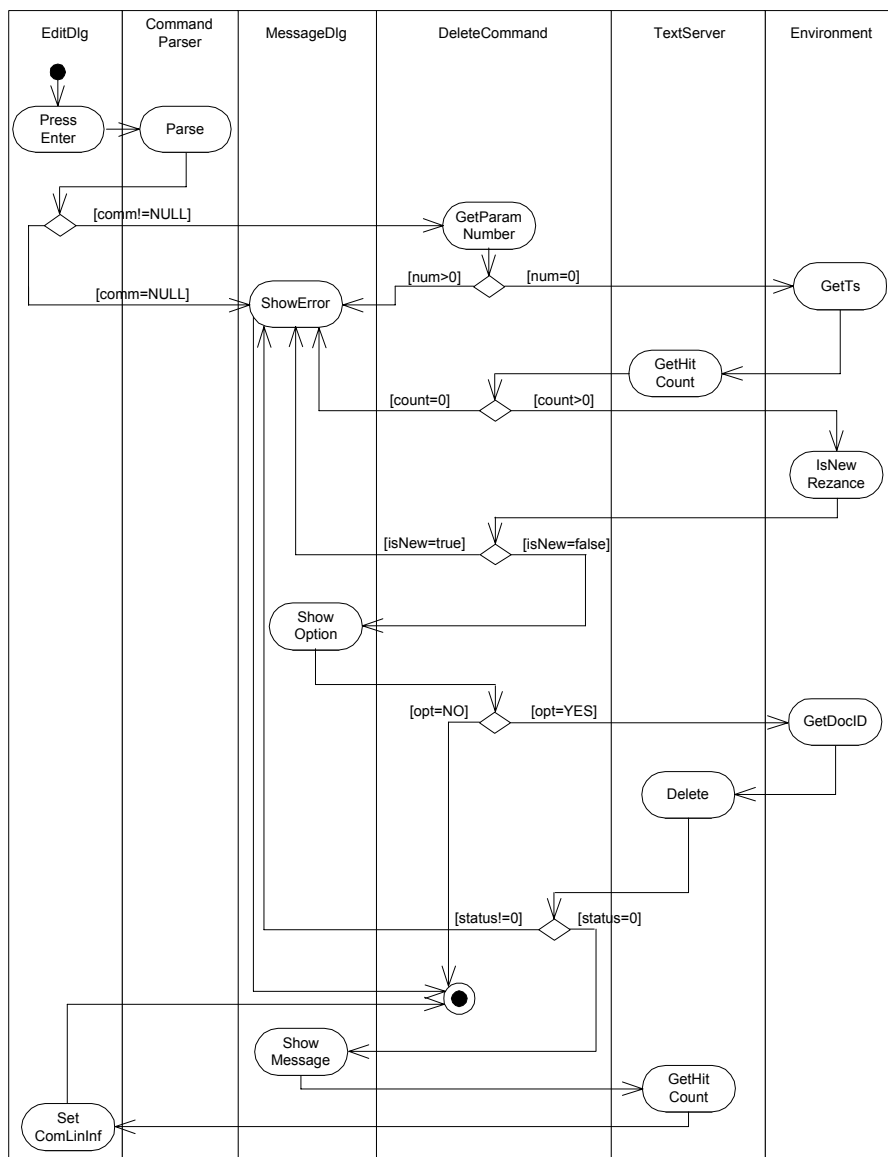
акција *showOption* и објекат прелази у стање *Librarian Choice Check*. Ако је библиотекар одустао од брисања, прелази се у завршно стање. Уколико је библиотекар потврдио брисање, извршавају се акције *getDocID* и *delete*, а објекат класе *DeleteCommand* прелази у стање *Delete Status Check* у коме се проверава да ли је брисање успешно. Ако је брисање неуспешно, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. Ако је брисање успешно, извршавају се акције *showMessage* (која приказује библиотекарку поруку о успешном брисању записа), *getHitCount* и *setComLinInf*, а објекат прелази у завршно стање, чиме се означава да је команда *delete* успешно извршена.



Слика 4.8 Дијаграм стања команде *delete*

4.3.1.4 Дијаграм активности команде *delete*

Дијаграм активности на слици 4.9 приказује стања акција, као и расподелу одговорности класа за извршавање одређених акција током извршавања команде *delete*.



Слика 4.9 Дијаграм активности команде *delete*

Класа *EditDlg* реализује акције *PressEnter* и *SetComLinInf*. За акцију *Parse* одговорна је класа *CommandParser*. *ShowError*, *ShowOption* и *ShowMessage* су акције реализоване у оквиру класе *MessageDlg*. Класа *DeleteCommand* реализује акцију *GetParamNumber*. За акције *GetHitCount* и *Delete* одговорна је класа *TextServer*, док се акције *GetTs*, *IsNewRezance* и *GetDocID* реализују у оквиру класе *Environment*. Овим дијаграмом приказује се ток акција које је потребно извршити како би се имплементирала команда *delete*.

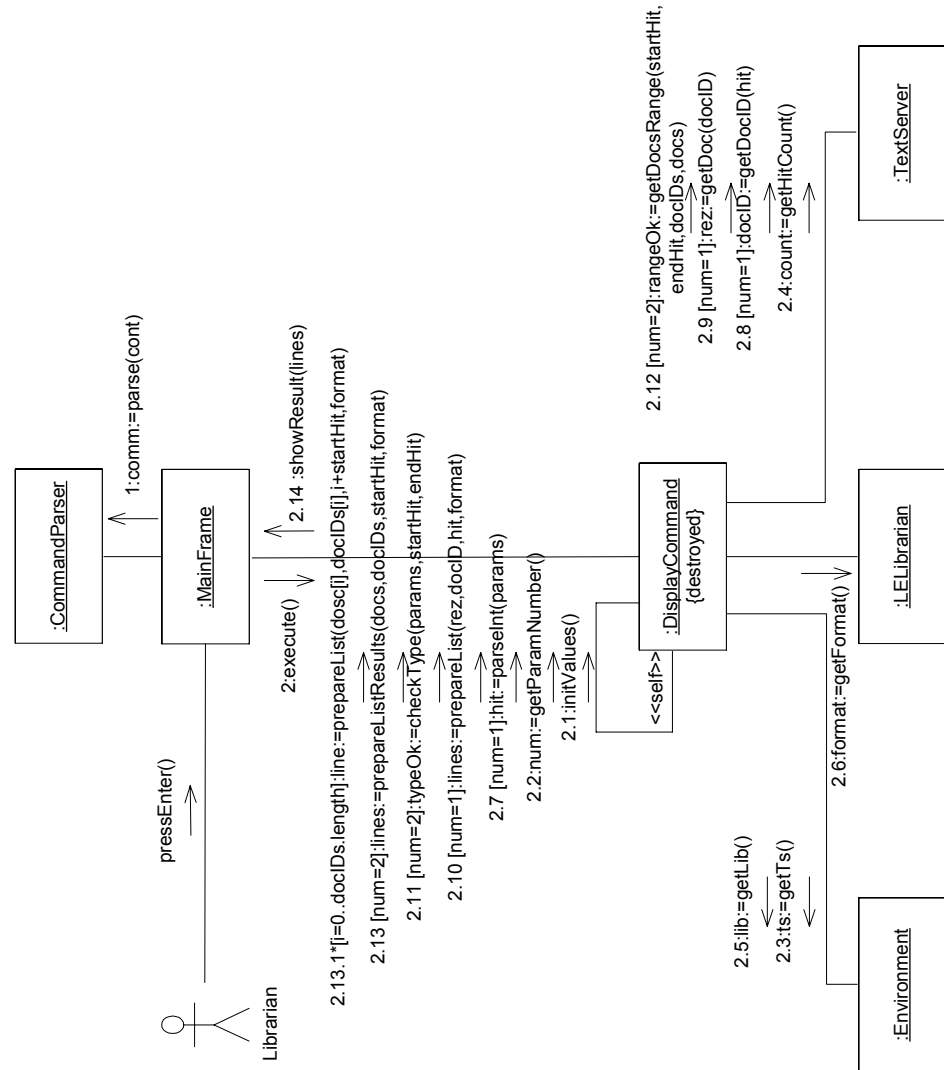
4.3.2 Динамички модел команде *display*

У овом одељку дати су дијаграми сарадње, секвенци, стања и активности којима је описана динамика приликом извршавања команде *display*. За сваки дијаграм дат је текстуални опис који га детаљно описује.

4.3.2.1 Дијаграм сарадње команде *display*

На слици 4.10 приказан је дијаграм сарадње команде *display* када се команда извршава у моду претраживања. На сличан начин може се формирати дијаграм сарадње команде *display* када се команда извршава у моду директоријума. Прослеђивање и значење порука са секвенцама 1 и 2 описане су у одељку 4.3.1.1. Једина разлика је што се команда извршава у моду претраживања и што се порука *execute* шаље објекту класе *DisplayCommand*.

У оквиру поруке *execute* шаље се низ порука које представљају позиве операција, а које омогућавају извршавање ове команде. Иницијализација параметара врши се прослеђивањем поруке *initValues*. Поруком *getParamNumber* преузима се број параметара који су издвојени приликом парсирања садржаја командне линије. Након тога, шаље се порука *getTs* која омогућава да се из класе *Environment* преузме референца на објекат класе *TextServer* који обезбеђује везу са базом података библиографске грађе. Пошто команди *display*, која се задаје у моду претраживања, претходи успешно претраживање базе података библиографске грађе, прослеђује се порука *getHitCount* објекту класе *TextServer* која омогућава да се преузме број погодака тог претраживања. Након преузимања броја погодака, поруком *getLib* добија се референца на објекат класе *LELibrarian* који омогућава рад са локалном структуром у којој се налазе подаци базе података радног окружења библиотекара. Поруком *getFormat* из локалне структуре добија се информација о текућем формату исписа постављеног за библиотекара који је тренутно пријављен на систем.

Слика 4.5 Дијаграм сарадње команде *display*

Уколико је библиотекар унео један параметар, поруком *parseInt* се проверава да ли је параметар коректно унет (односно, да ли параметар представља број). Након тога, шаље се порука *getDocID* објекту класе *TextServer* која омогућава да се из базе података библиографске грађе преузме идентификациони број записа чији се испис жели. Прослеђивањем поруке *getDoc* на основу идентификационог броја преузима се запис из базе. Након тога се прослеђивањем поруке *prepareList* формира структура за приказ на основу тако преузетог записа и текућег формата исписа.

Уколико је библиотекар унео два параметра који представљају интервал, поруком *checkType* се проверава да ли су параметри коректно унети (односно, да ли параметри представљају бројеве). Након тога, шаље се порука *getDocsRange* објекту класе *TextServer* која омогућава да се провери коректност опсега параметара и да се, уколико је опсег коректан, из базе података библиографске грађе преузму записи чији се испис жели као и њихови идентификациони бројеви. Затим се, прослеђивањем поруке *prepareListResults*, у оквиру које се за сваки преузет запис прослеђује порука *prepareList*, формира структура за приказ на основу преузетих записа и текућег формата исписа.

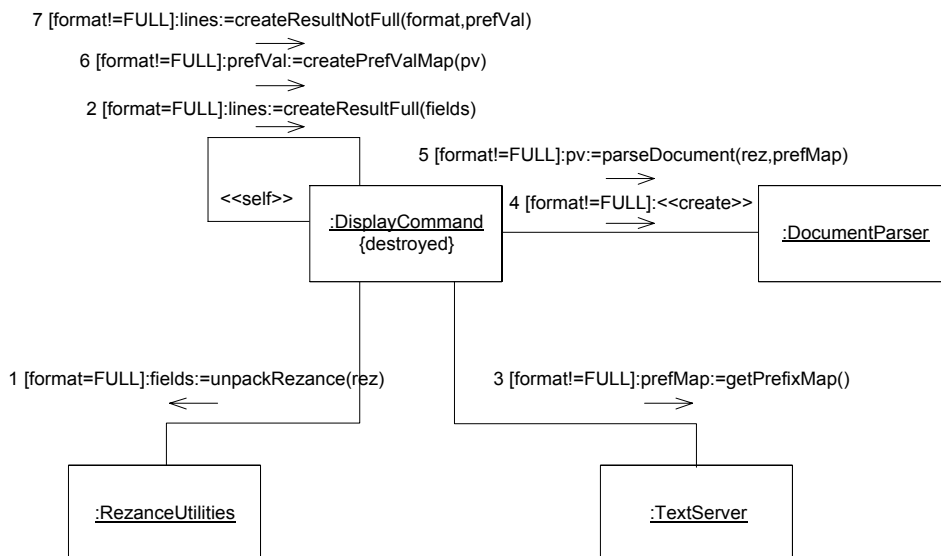
Порука *showResult* омогућава да се тако формирана структура прикаже библиотекар.

Слика 4.11 приказује сценарио извршавања операције *prepareList*. Разликују се две врсте приказа које библиотекар добија командом *display*. Једна је такозвани *FULL* формат којим се приказују садржаји свих поља и потпоља у запису, а друга су извештаји код којих се приказују само садржаји из записа који одговарају одабраним префиксима претраживања дефинисаним за тај формат.

Уколико је за библиотекара који је пријављен на систем текући формат исписа постављен на *FULL*, шаље се порука *unpackRezance* која распакује запис у структуру погодну за брз приступ садржајима потпоља. Након тога се поруком *createResultFull* креира структура извештаја у *FULL* формату која ће бити приказана библиотекар.

Уколико за библиотекара који је пријављен на систем текући формат исписа није *FULL*, шаље се порука *getPrefixMap* објекту класе *TextServer* која у структуру *HashMap* поставља парове (поље_са_потпољем, префикс_претраживања). Након тога се поруком *create* креира објекат типа *DocumentParser* и тако креираном објекту се шаље порука *parseDocument* која омогућава да се садржаји из записа придруже одговарајућим префиксима претраживања. Сва-

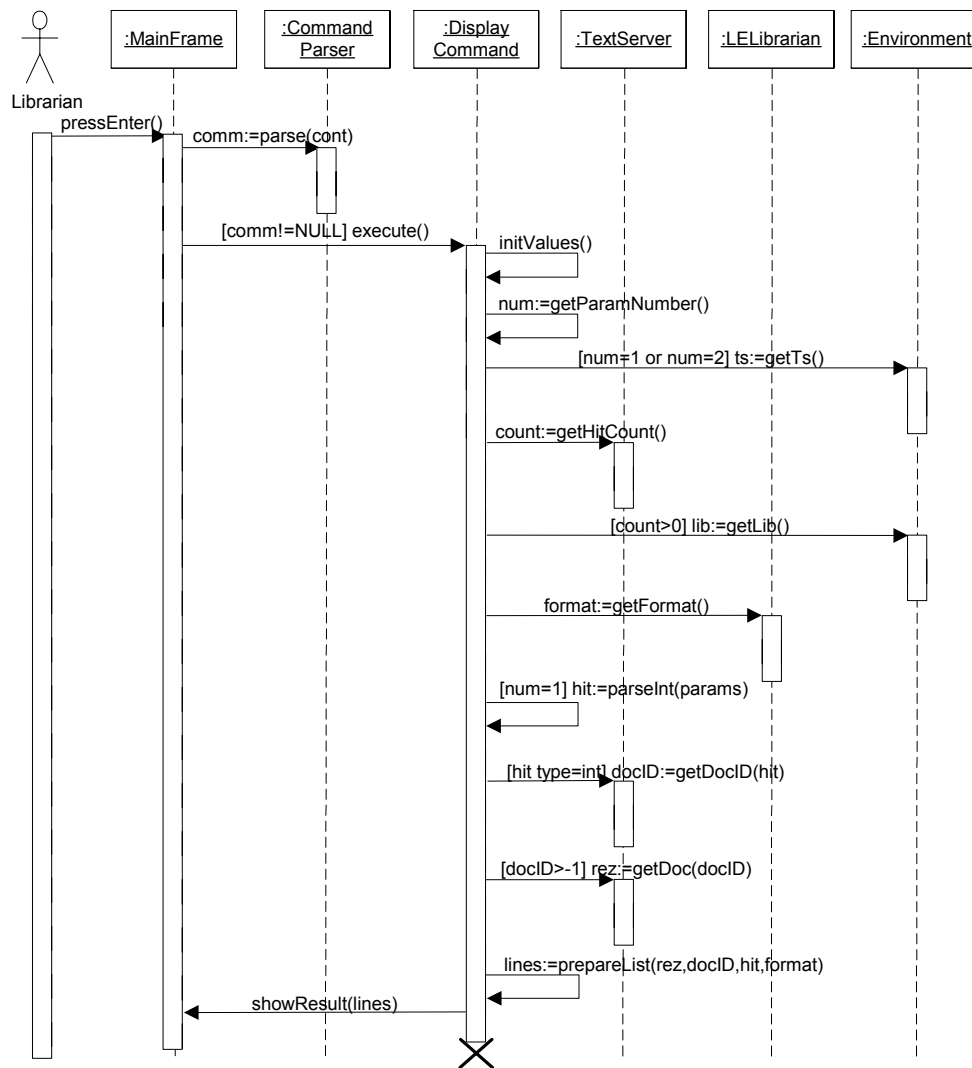
ки префикс претраживања и њему придружен садржај представљају се објектом типа *PrefixPair*. Након тога се поруком *createPrefValMap* од ових објеката креирају парови (префикс претраживања, одговарајући садржај из записа) који се чувају у структури *HashMap*. На основу те структуре и текућег формата исписа поруком *createResultNotFull* креира се извештај.

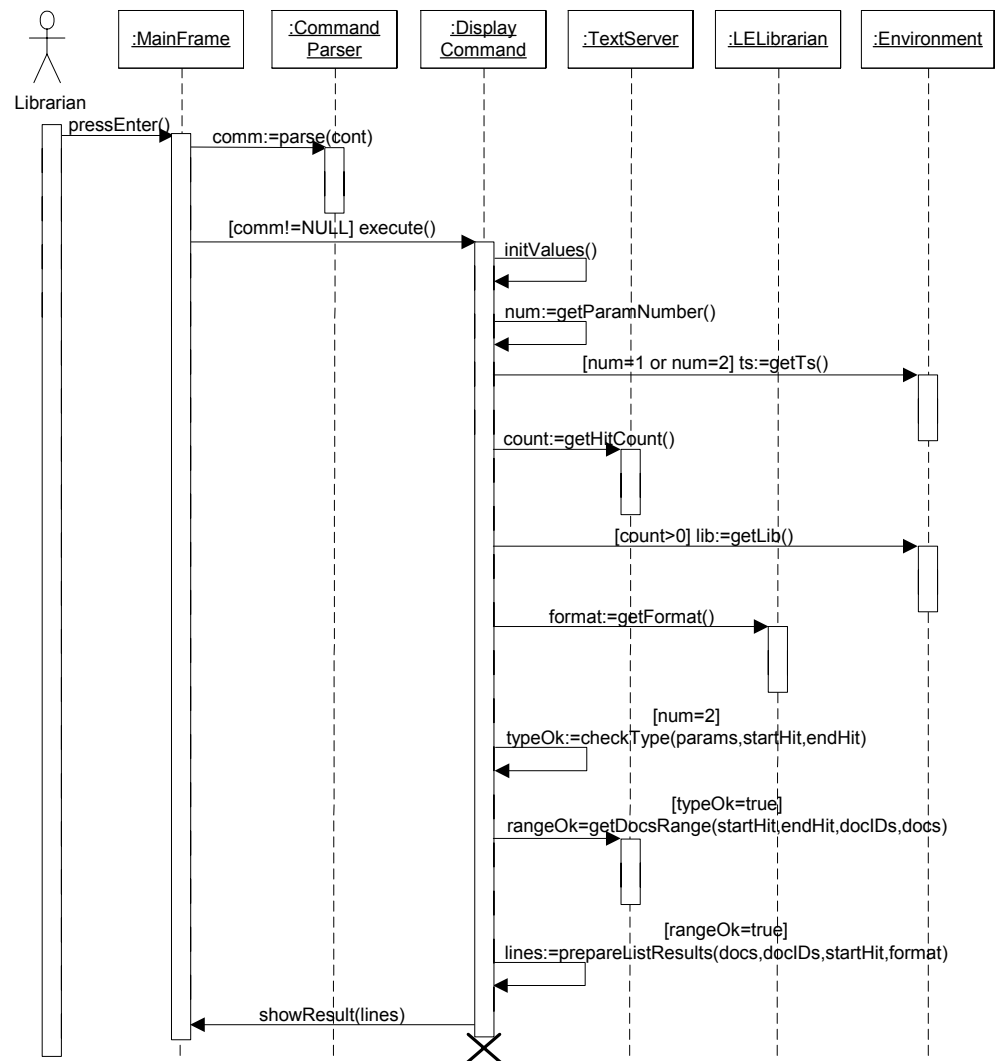


Слика 4.11 Дијаграм сарадње операције *prepareList*

4.3.2.2 Дијаграм секвенци команде *display*

На слици 4.12 приказан је дијаграм секвенци који приказује сценарио извршавања команде *display* у случају да се приликом задавања команде у моду претраживања унесе један параметар. Сценарио извршавања команде *display* у случају да се приликом задавања команде у моду претраживања унесу два параметра који представљају интервал дат је на слици 4.13. Оба сценарија приказују извршавање команде *display* у случају да су задовољени сви услови потребни за успешно извршавање команде у моду претраживања. Детаљи који се односе на везе између објеката приказани су на дијаграму сарадње, датом на слици 4.10.

Слика 4.12 Дијаграм секвенци команде *display* са једним параметром

Слика 4.13 Дијаграм секвенци команде *display* са интервалом

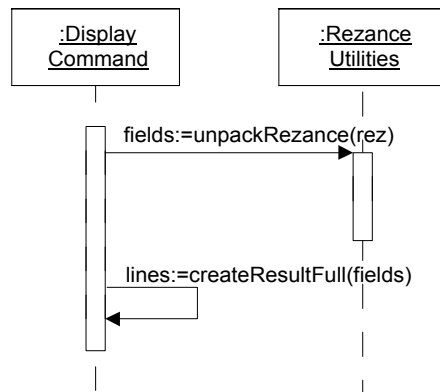
Након слања поруке *pressEnter* којом библиотекар иницира извршавање ове команде и поруке *parse* која омогућава парсирање садржаја командне линије, порука *execute* се извршава под условом да је библиотекар задао коректан назив команде и да команда припада скупу команди које се могу извршити у овом моду (мод претраживања). Након тога, прослеђује се порука *initValues*. Порука *getTs* се прослеђује под условом да је библиотекар унео

један или два параметара, односно да порука *getParamNumber* враћа вредност један или два. Порука *getLib* се прослеђује под условом да је број погодака који враћа порука *getHitCount* већи од нуле. Након тога, прослеђује се порука *getFormat*.

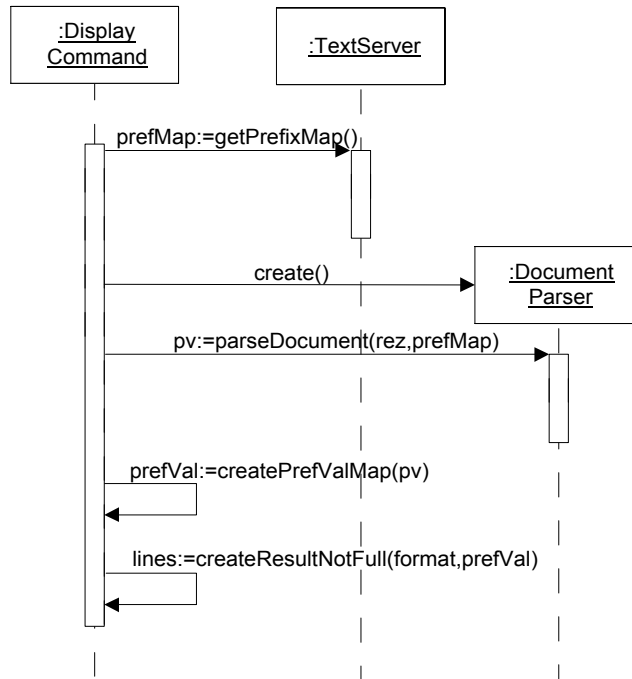
Уколико је библиотекар унео један параметар, прослеђује се порука *parseInt*. Прослеђивању поруке *getDocID* претходи услов да је унети параметар број. Порука *getDoc* прослеђује се под условом да је пронађен идентификациони број записа за задати параметар. Након тога, прослеђује се порука *prepareList*, а за њом порука *showResult*.

Уколико је библиотекар унео два параметра (који представљају интервал), прослеђује се порука *checkType*. Прослеђивању поруке *getDocsRange* претходи услов да унети параметри представљају бројеве. Порука *prepareListResults* прослеђује се под условом да је интервал који је задао библиотекар коректан. Након тога, прослеђује се порука *showResult*.

Сценарио за извршавање операције *prepareList* у случају да је текући формат исписа постављен на *FULL* приказан је на слици 4.14. Уколико је формат исписа различит од *FULL* формата извршавање операције *prepareList* може се описати сценаријом који је приказан на слици 4.15.



Слика 4.14 Дијаграм секвенци за *FULL* формат

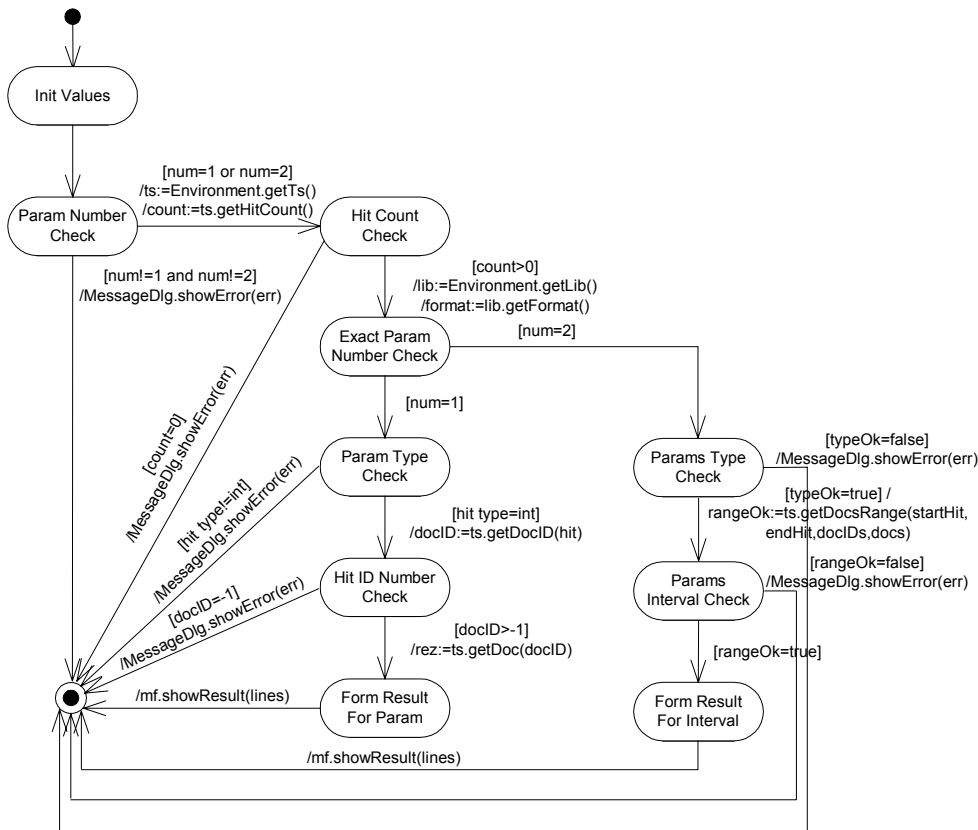
Слика 4.15 Дијаграм секвенци за формат различит од *FULL*

4.3.2.3 Дијаграм стања команде *display*

Дијаграм стања на слици 4.16 приказује сва могућа стања објекта класе *DisplayCommand*, као и акције које се дешавају приликом преласка објекта из једног стања у друго.

У стању *Init Values* објекта класе *DisplayCommand* врши се иницијализација вредности параметара који се користе приликом извршавања команде *display*. У стању *Param Number Check* проверава се да ли је библиотекар унео параметре приликом уноса команде у командну линију. Ако библиотекар није унео један или два параметра (који представљају интервал), прелази се у завршно стање и извршава се акција *showError* која библиотекару приказује одговарајућу поруку о грешци. Уколико је библиотекар унео један или два параметра, извршавају се акције *getTs* и *getHitCount*, а објекат прелази у стање *Hit Count Check*. У стању *Hit Count Check* проверава се број погодака претходног претраживања. Ако је провера погодака неуспешна, односно ако поготци не постоје, прелази се у завршно стање и извршава се акција

showError која приказује одговарајућу поруку о грешци. Ако је провера погодака успешна, извршавају се акције *getLib* и *getFormat*, а објекат прелази у стање *Exact Param Number Check* у коме се проверава да ли је библиотекар унео један параметар или два параметра (који представљају интервал).



Слика 4.16 Дијаграм стања команде *display*

Ако је библиотекар унео један параметар, објекат класе *DisplayCommand* прелази у стање *Param Type Check* у коме се врши провера да ли је параметар коректно унет (односно, да ли параметар представља број). Уколико параметар није број, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци, а у супротном извршава се акција *getDocID* и објекат прелази у стање *Hit ID Number Check*. Ако је читање идентификационог броја задатог параметра неуспешно, одно-

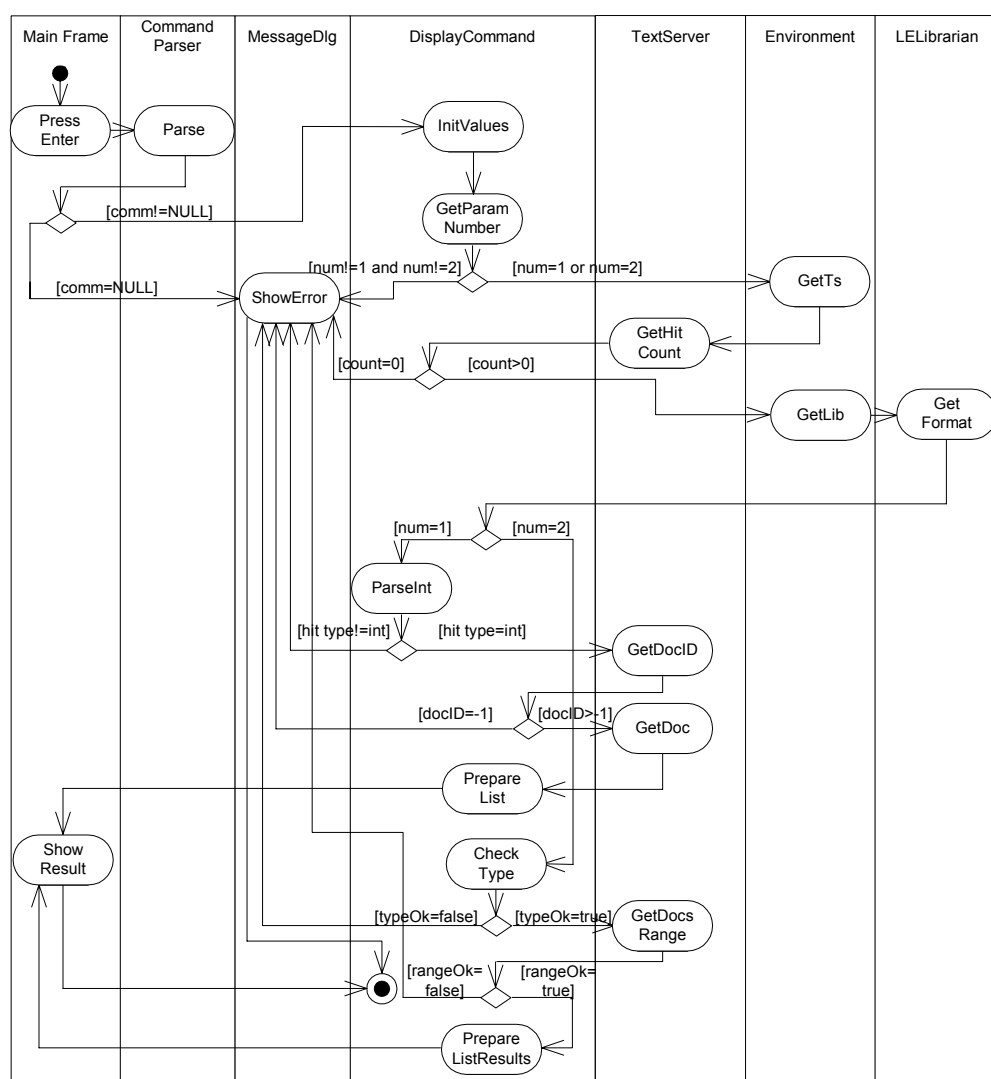
сно ако се не добије идентификациони број записа, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. У супротном, извршава се акција *getDoc*, а објекат класе *DisplayCommand* прелази у стање *Form Result For Param*. Из овог стања објекат прелази у завршно стање, при чему се извршава акција *showResult*. Тиме се означава да је команда *display* која приказује извештај о једном запису успешно извршена.

Ако је библиотекар унео два параметра, објекат класе *DisplayCommand* прелази у стање *Params Type Check* у коме се врши провера да ли су параметри коректно унети (односно, да ли параметри представљају бројеве). Уколико параметри нису бројеви, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци, а у супротном извршава се акција *getDocsRange* и објекат прелази у стање *Params Interval Check*. Ако је интервал погрешно задат, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. У супротном, објекат класе *DisplayCommand* прелази у стање *Form Result For Interval*. Из овог стања објекат прелази у завршно стање, при чему се извршава акција *showResult*. Тиме се означава да је команда *display* која приказује извештај о записима из задатог интервала успешно извршена.

4.3.2.4 Дијаграм активности команде *display*

Дијаграм активности на слици 4.17 приказује стања акција, као и расподелу одговорности класа за извршавање одређених акција током извршавања команде *display*.

Класа *MainFrame* реализује акције *PressEnter* и *ShowResult*. За акцију *Parse* одговорна је класа *CommandParser*. *ShowError* је акција реализована у оквиру класе *MessageDlg*. Класа *DisplayCommand* реализује акције *InitValues*, *GetParamNumber*, *ParseInt*, *PrepareList*, *CheckType* и *PrepareListResults*. За акције *GetHitCount*, *GetDocID*, *GetDoc* и *GetDocsRange* одговорна је класа *TextServer*. Акције *GetTs* и *GetLib* реализоване су у оквиру класе *Environment*. Акцију *GetFormat* реализује класа *LELibrarian*. Овим дијаграмом приказује се ток акција које је потребно извршити како би се имплементирала команда *display* у моду претраживања.

Слика 4.17 Дијаграм активности команде *display*

4.3.3 Динамички модел команде *new*

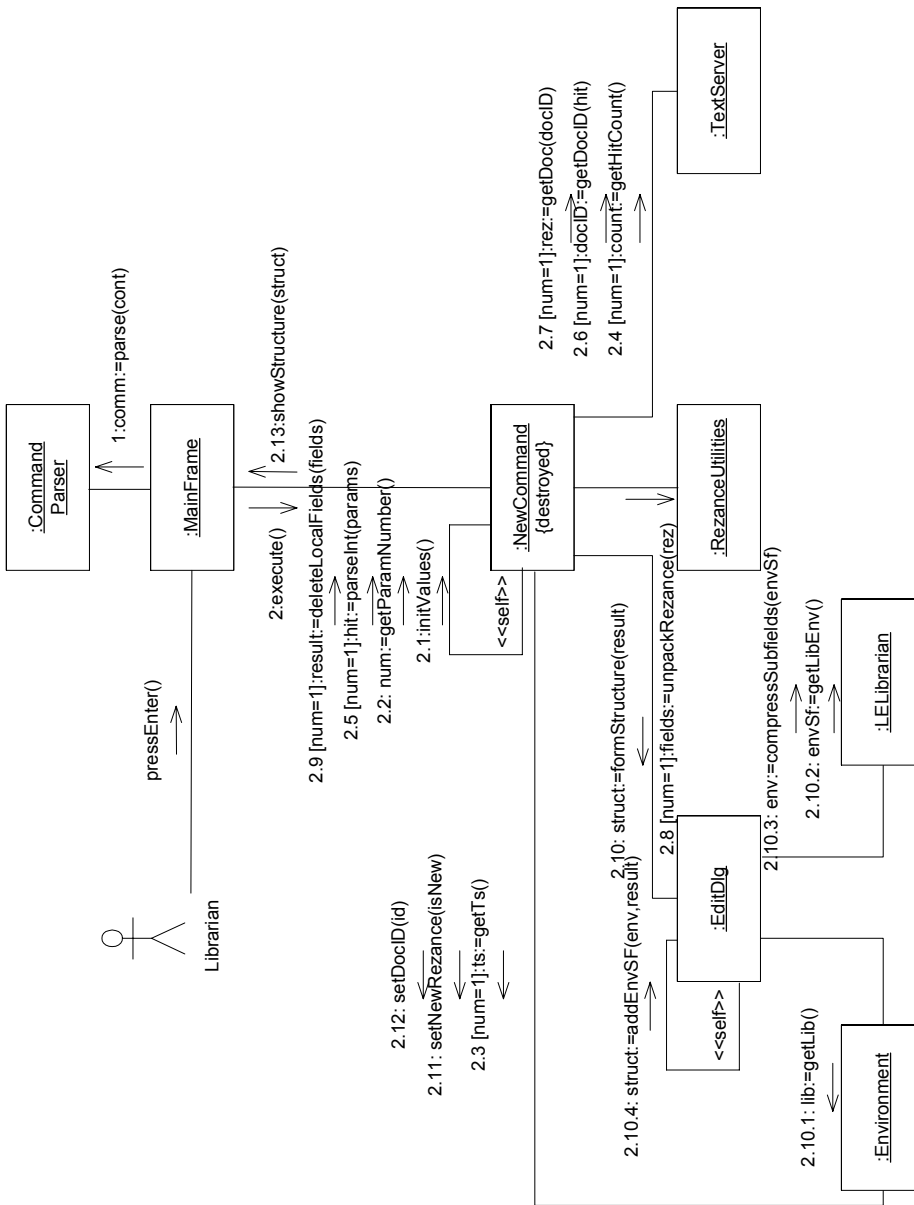
У овом одељку дати су дијаграми сарадње, секвенци, стања и активности којима је описана динамика приликом извршавања команде *new*. За сваки дијаграм дат је текстуални опис који га детаљно описује.

4.3.3.1 Дијаграм сарадње команде *new*

На слици 4.18 приказан је дијаграм сарадње команде *new*. Прослеђивање и значење порука са секвенцама 1 и 2 описане су у одељку 4.3.1.1. Једина разлика је што се команда извршава у моду претраживања и што се порука *execute* шаље објекту класе *NewCommand*.

У оквиру поруке *execute* шаље се низ порука које представљају позиве операција, а које омогућавају извршавање ове команде. Иницијализација параметара врши се прослеђивањем поруке *initValues*. Поруком *getParamNumber* преузима се број параметара који су издвојени приликом парсирања садржаја командне линије.

Уколико је библиотекар унео један параметар врши се креирање новог записа на основу узорка, при чему параметар представља редни број поготка из скупа погодака добијених након претраживања базе података библиографске грађе. При томе, шаље се порука *getTs* која омогућава да се из класе *Environment* преузме референца на објекат класе *TextServer* који обезбеђује везу са базом података библиографске грађе. Да би се креирао нови запис на основу узорка, извршавању команде *new* која има један параметар претходи успешно претраживање базе података библиографске грађе. Прослеђивањем поруке *getHitCount* објекту класе *TextServer* омогућава се преузимање броја погодака претходног претраживања. Након преузимања броја погодака, порука *parseInt* омогућава да се провери да ли је параметар коректно унет. Прослеђивањем поруке *getDocID* објекту класе *TextServer* из базе података библиографске грађе преузима се идентификациони број записа који је узорак. Поруком *getDoc* се на основу идентификационог броја преузима запис из базе. Након тога се прослеђивањем поруке *unpackRezance* распакује запис у структуру погодну за брз приступ садржајима потпоља. Након распакивања записа из структуре се бришу поља која носе локалне информације (сигнатура, инвентарни број и др.). Брисање ових поља постиже се прослеђивањем поруке *deleteLocalFields* која као параметар има запис који представља узорак.



Слика 4.5 Дијаграм сарадње команде new

Без обзира да ли се команда *new* задаје са или без параметра, прослеђује се порука *formStructure* објекту класе *EditDlg*. У оквиру ове поруке шаље се низ порука које омогућавају да се на основу података базе података радног окружења библиотекар формира структура записа на одговарајући начин. Да би се приступило подацима локалне структуре у којој се налазе подаци базе података радног окружења библиотекар, шаље се порука *getLib* која омогућава да се преузме референца на објекат за рад са локалном структуром. Поруком *getLibEnv* се, на основу параметара обраде који су постављени за библиотекар који креира запис, из ове структуре преузима списак поља и потпоља које новоформирани запис треба да садржи. Поруком *compressSubfields* овај списак поља и потпоља прилагођава се структури записа. Након тога поруком *addEnvSF* се тај списак поља и потпоља додаје списку поља и потпоља који представљају запис.

Порукама *setNewRezance* и *setDocID* се у оквиру класе *Environment* постављају вредности одговарајућих променљивих које ће указивати на то да је запис новоформирани. Након тога, поруком *showStructure* омогућава се приказ структуре новоформираног записа, тако што се у оквиру ове поруке шаље низ порука које омогућавају прелазак из мода претраживања у мод директоријума и приказ прозора са том структуром библиотекар.

4.3.3.2 Дијаграм секвенци команде *new*

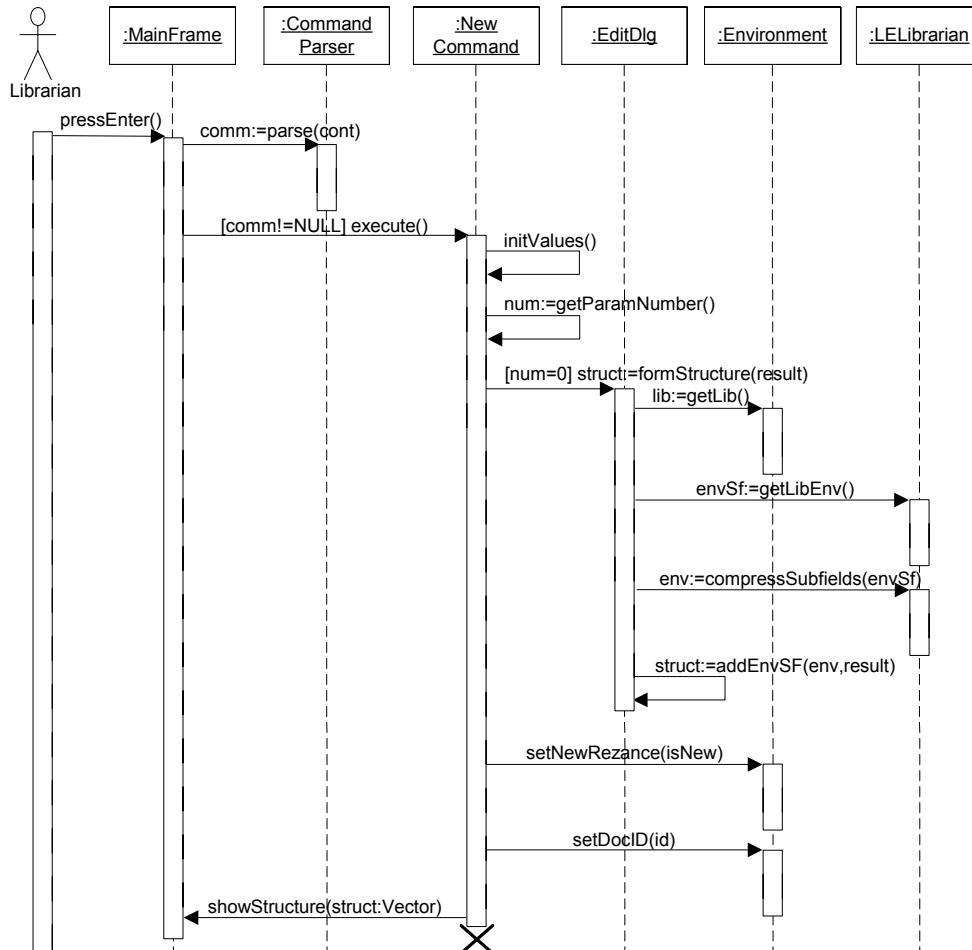
На слици 4.20 приказан је дијаграм секвенци који приказује сценарио извршавања команде *new* у случају да се нови запис креира на основу узорка, док је на слици 4.19 приказан сценарио креирања новог записа без узорка. Оба сценарија приказују извршавање команде *new* у случају да су задовољени сви услови потребни за успешно извршавање команде. Детаљи који се односе на везе између објеката приказани су на дијаграму сарадње датом на слици 4.18.

Након слања поруке *pressEnter* којом библиотекар иницира извршавање ове команде и поруке *parse* која омогућава парсирање садржаја командне линије, порука *execute* се извршава под условом да је библиотекар задао коректан назив команде и да команда припада скупу команди које се могу извршити у овом моду (мод претраживања). Након тога, прослеђује се порука *initValues*, а након ње порука *getParamNumber*.

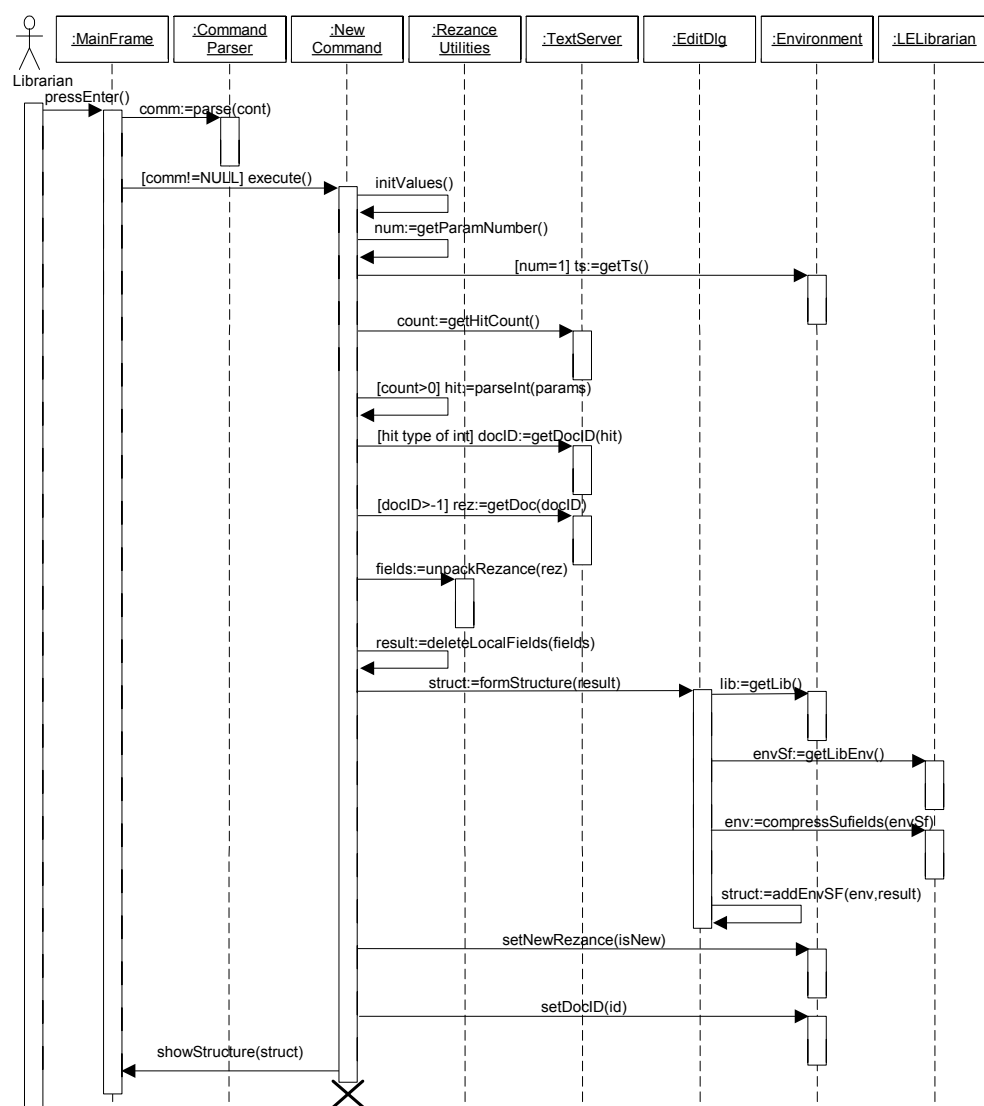
Уколико се запис креира на основу узорка прослеђује се порука *getTs*. Порука *parseInt* се прослеђује под условом да је број погодака који враћа порука *getHitCount* већи од нуле. Порука *getDocID* прослеђује се под условом да је параметар који је библиотекар унео број. Прослеђивању поруке *getDoc*

претходи услов да је пронађен идентификациони број записа за задати параметар. Након тога, прослеђује се порука *unpackRezance*, а за њом порука *deleteLocalFields*.

Било да се запис креира на основу узорка или не, прослеђује се порука *formStructure* у оквиру које се прослеђују поруке *getLib*, *getLibEnv*, *compressSubfields* и *addEnvSF*. Након тога, прослеђују се поруке *setNewRezance*, *setDocID* и *showStructure*.



Слика 4.19 Дијаграм секвенци команде *new* без параметара



Слика 4.20 Дијаграм секвенци команде *new* приликом креирања записа на основу узорка

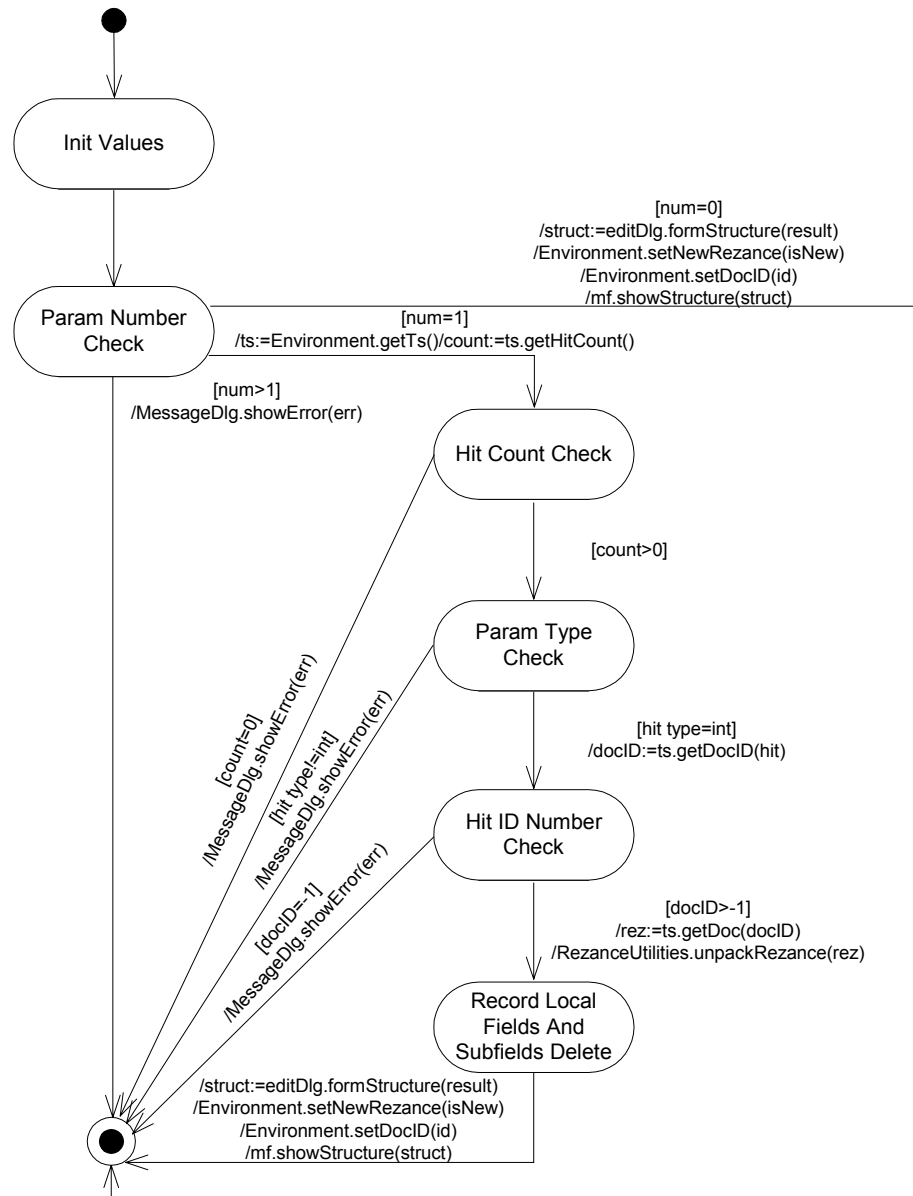
4.3.3.3 Дијаграм стања команде *new*

Дијаграм стања на слици 4.21 приказује сва могућа стања објекта класе *NewCommand*, као и акције које се дешавају приликом преласка објекта из једног стања у друго.

У стању *Init Values* објекта класе *NewCommand* врши се иницијализација вредности параметара који се користе приликом извршавања команде *new*. У стању *Param Number Check* проверава се да ли је библиотекар унео параметре приликом уноса команде у командну линију. Ако је библиотекар унео више од једног параметра, прелази се у завршно стање и извршава се акција *showError* која библиотекару приказује одговарајућу поруку о грешци.

Уколико библиотекар није унео параметре, извршавају се акције *formStructure*, *setNewRezance*, *setDocID*, *showStructure* и прелази се у завршно стање, чиме се означава да је команда *new* без параметара успешно извршена.

Ако је библиотекар унео један параметар (креирање записа на основу узорка), извршавају се акције *getTs* и *getHitCount*, а објекат прелази у стање *Hit Count Check*. У стању *Hit Count Check* проверава се број погодака претходног претраживања. Ако је провера погодака неуспешна, односно ако поготци не постоје, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. Ако је провера погодака успешна, објекат прелази у стање *Param Type Check*. Уколико параметар није коректно унет (односно, параметар није број), прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци, а у супротном извршава се акција *getDocID* и објекат прелази у стање *Hit ID Number Check*. Ако је читање идентификационог броја записа за задати параметар неуспешно, односно ако се не добије идентификациони број, прелази се у завршно стање и извршава се акција *showError* која приказује одговарајућу поруку о грешци. У супротном, извршавају се акције *getDoc* и *unpackRezance*, а објекат класе *NewCommand* прелази у стање *Record Local Fields And Subfields Delete*. Из овог стања објекат прелази у завршно стање, при чему се извршавају акције *formStructure*, *setNewRezance*, *setDocID* и *showStructure*. Тиме се означава да је команда *new* приликом креирања новог записа на основу узорка успешно извршена.

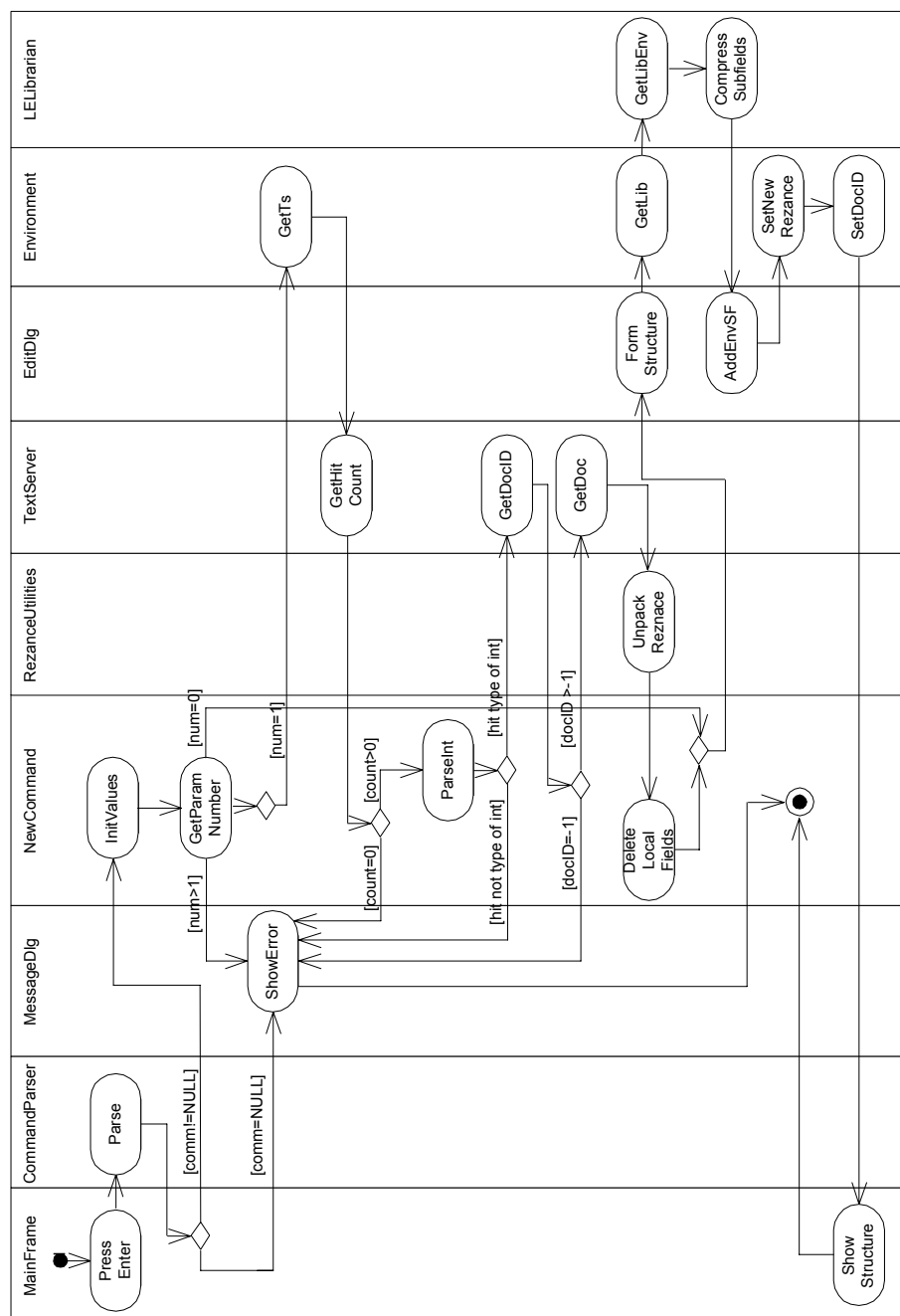


Слика 4.21 Дијаграм стања команде new

4.3.3.4 Дијаграм активности команде *new*

Дијаграм активности на слици 4.22 приказује стања акција, као и расподелу одговорности класа за извршавање одређених акција током извршавања команде *new*.

Класа *MainFrame* реализује акције *PressEnter* и *ShowStructure*. За акцију *Parse* одговорна је класа *CommandParser*. *ShowError* је акција реализована у оквиру класе *MessageDlg*. Класа *NewCommand* реализује акције *InitValues*, *GetParamNumber*, *ParseInt*, *DeleteLocalFields*. *UnpackRezance* је акција реализована у оквиру класе *RezanceUtilities*. За акције *GetHitCount*, *GetDocID* и *GetDoc* одговорна је класа *TextServer*. Акције *FormStructure* и *AddEnvSF* реализује класа *EditDlg*, док су акције *GetTs*, *GetLib*, *SetNewRezance* и *SetDocID* реализоване у оквиру класе *Environment*. Акције *GetLibEnv* и *CompressSubfields* реализује класа *LELibrarian*. Овим дијаграмом приказује се ток акција које је потребно извршити како би се имплементирала команда *new*.



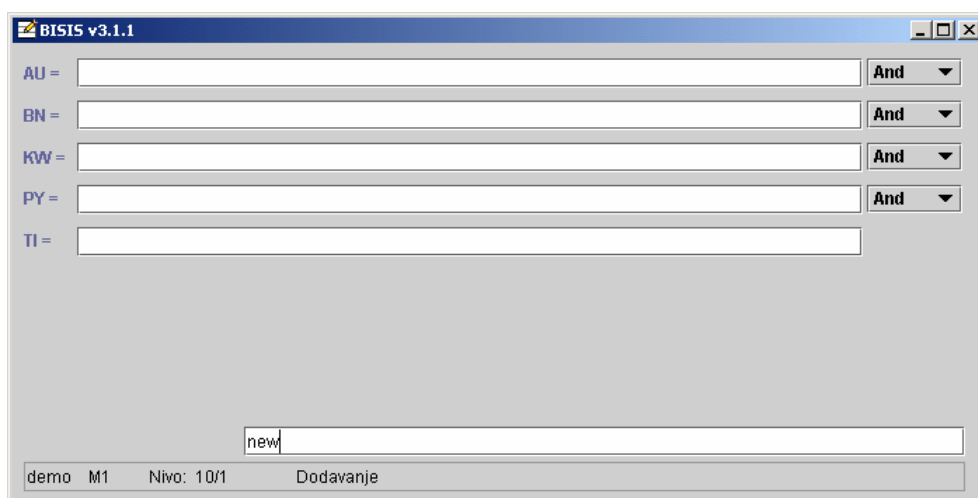
Слика 4.22 Дијаграм активности команде new

Прилог 4.1 Коришћење команди у апликацији за обраду библиографске грађе

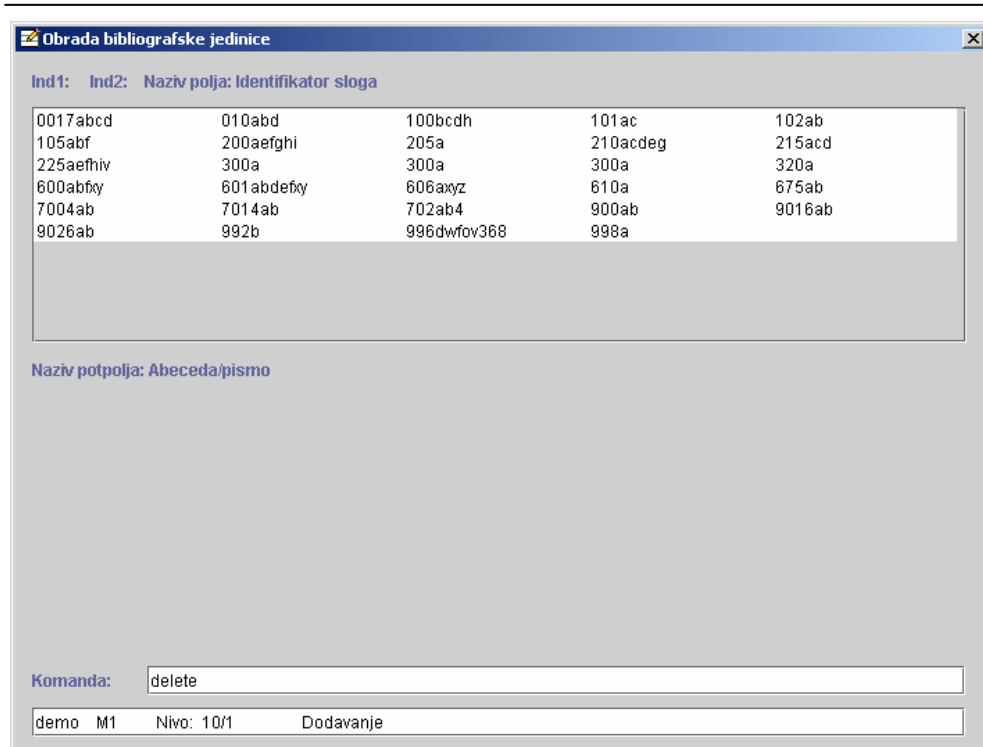
Објектно-оријентисана спецификација команди библиотечког софтверског система БИСИС вер. 3 извршена је коришћењем UML нотације, док је имплементација извршена у Java окружењу.

У овом прилогу приказани су илустративни примери коришћења команди *format*, *display*, *new*, *save*, *edit* и *delete* у апликацији за обраду библиографске грађе софтверског система БИСИС вер. 3.

Као што је речено у Поглављу 4, библиотекар може да задаје команде у моду претраживања и моду директоријума. Сваком моду одговара одређени изглед прозора. Изглед прозора који одговара моду претраживања дат је на слици 4.1.1, док је изглед прозора који одговара моду директоријума дат на слици 4.1.2.



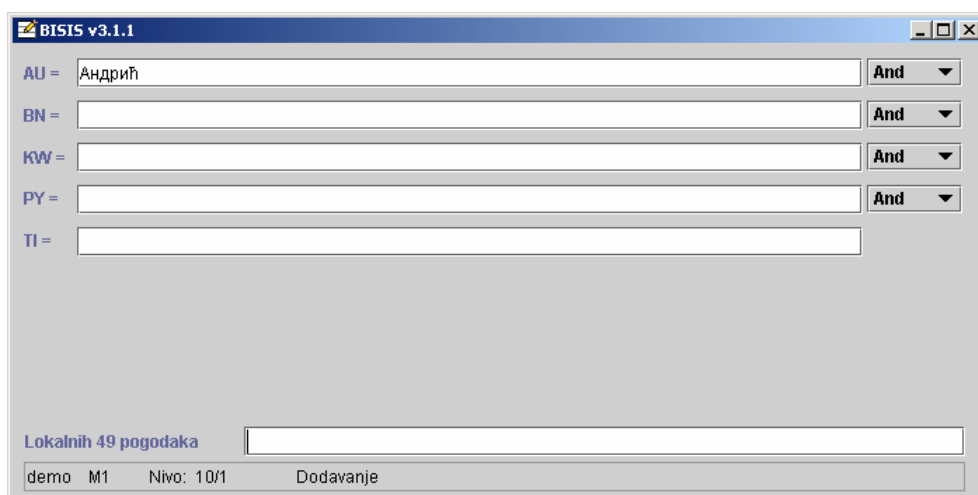
Слика 4.1.1 Изглед прозора у моду претраживања



Слика 4.1.2 Изглед прозора у моду директоријума

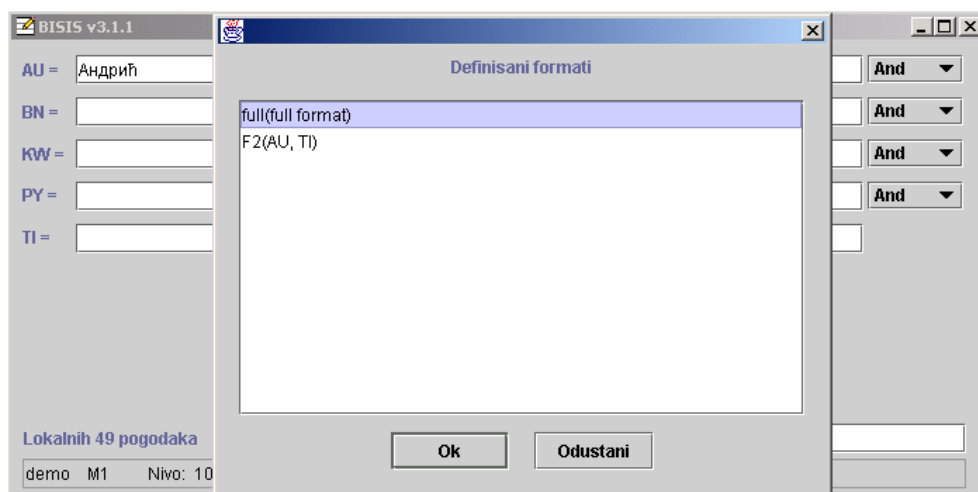
4.1.1 Пример извршавања команди у апликацији за обраду библиографске грађе

Први корак у раду библиотекара приликом обраде једне библиотечке јединице је провера да ли је та јединица већ обрађена. Ова провера врши се претраживањем базе података библиографске грађе. На слици 4.1.3 приказан је пример претраживања на основу упита AU=Андрић, при чему је број добијених погодака, тј. број записа који задовољавају постављени упит 49.



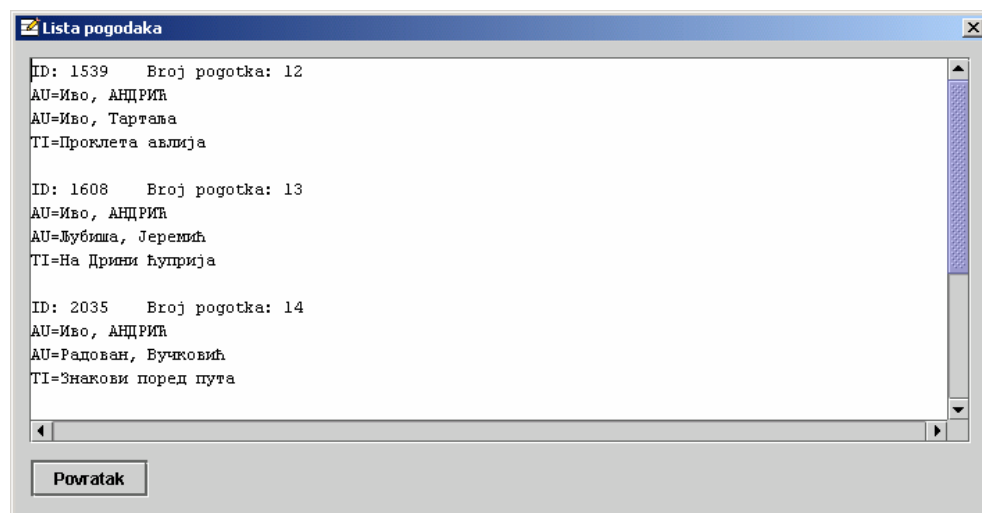
Слика 4.1.3 Изглед прозора након претраживања

Након претраживања базе података библиографске грађе и исписа броја погодака, библиотекар може да прегледа пронађене записе у текућем формату исписа. Дефинисани формати исписа могу се прегледати задавањем команде *format ?*, као што се види на слици 4.1.4. Текући формат исписа се може променити током рада у апликацији, или се може дефинисати нови формат. Промена формата исписа, или дефинисање новог формата, врши се командом *format* са одговарајућим параметрима.



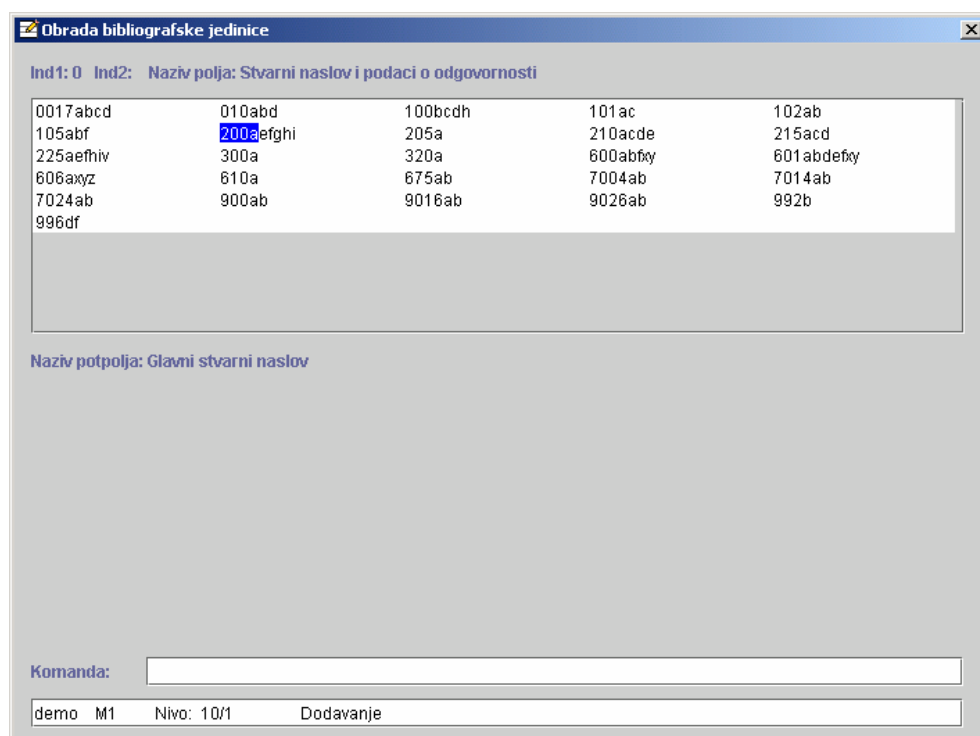
Слика 4.1.4 Изглед прозора приликом прегледања дефинисаних формата

Исписивање записа (који задовољавају постављени упит) на екрану врши се задавањем команде *display* уз коју се наводе параметри који представљају редне бројеве погодака. На тај начин библиотекар може да изврши преглед садржаја жељених записа. На слици 4.1.5 приказан је резултат извршавања команде *display 12-14* која приказује податке из записа о аутору и наслову (текући формат исписа приказује податке из записа који одговарају префиксима претраживања AU и TI).



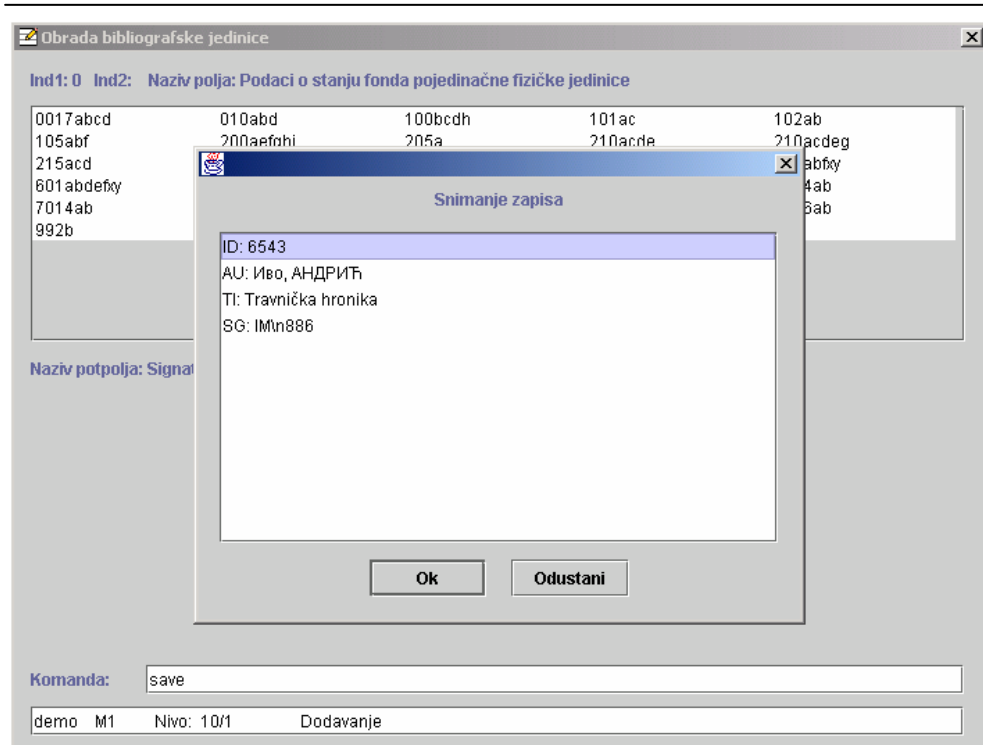
Слика 4.1.5 Приказ резултата претраживања у изабраном формату

Уколико библиотекар приликом претраживања базе података библиографске грађе није пронашао жељени запис, нови запис могуће је креирати командом *new*. Прозор за приказ структуре записа се попуњава одговарајућим пољима и потпољима која су дефинисана за текуће параметре обраде (тип публикације, ниво обраде и ниво обавезности) тог библиотекара. По унетој команди *new* прозор за приказ структуре записа добија изглед као на слици 4.1.6. Приказана поља и потпоља одговарају пољима и потпољима која су дефинисана у бази података радног окружења библиотекара за текући тип публикације M1 (монографије), ниво обраде 10 и ниво обавезности 1.



Слика 4.1.6 Изглед прозора приликом креирања новог записа

Када су унети садржаји свих потпоља дефинисаних за текући ниво обавезности, запис се може снимити у базу података библиографске грађе командом *save*. Током извршавања ове команде библиотекар се приказују основни подаци записа који се обрађује (идентификациони број записа, аутор, наслов и сигнатура), као што се види на слици 4.1.7. Библиотекар, уколико је задовољан приказаним подацима, може да потврди снимање записа у базу података библиографске грађе притиском на дугме *Ok*. У супротном, запис неће бити снимљен, а апликација ће остати у моду директоријума (тј. у моду који омогућава приказ структуре записа који се обрађује).



Слика 4.1.7 Изглед прозора приликом снимања записа

Уколико је запис успешно снимљен у базу података библиографске грађе, поновним претраживањем ове базе упитом AU=Андрић број погодака након претраживања је 50. Унети запис могуће је модификовати задавањем команде *edit* уз коју се као параметар наводи редни број поготка. Модификација записа означава, на пример, промену структуре постојећег записа додавањем нових поља и потпоља и/или промену садржаја потпоља записа. На слици 4.1.8 приказан је изглед прозора приликом уноса и/или модификације садржаја потпоља записа. Овако модификован запис могуће је снимити у базу података библиографске грађе командом *save*.

Obrada bibliografske jedinice

Ind1: Ind2: Naziv polja: Izdavanje, distribucija itd.

0017abcd	010abd	100bcdh	101ac	102ab
105abf	200aefghi	205a	210acde	210acdeg
215acd	225aefhiv	300a	320a	600abfey
601abdefy	606axyz	610a	675ab	7004ab
7014ab	7024ab	900ab	9016ab	9026ab
992b	996df			

Naziv potpolja: Izdavač, distributer

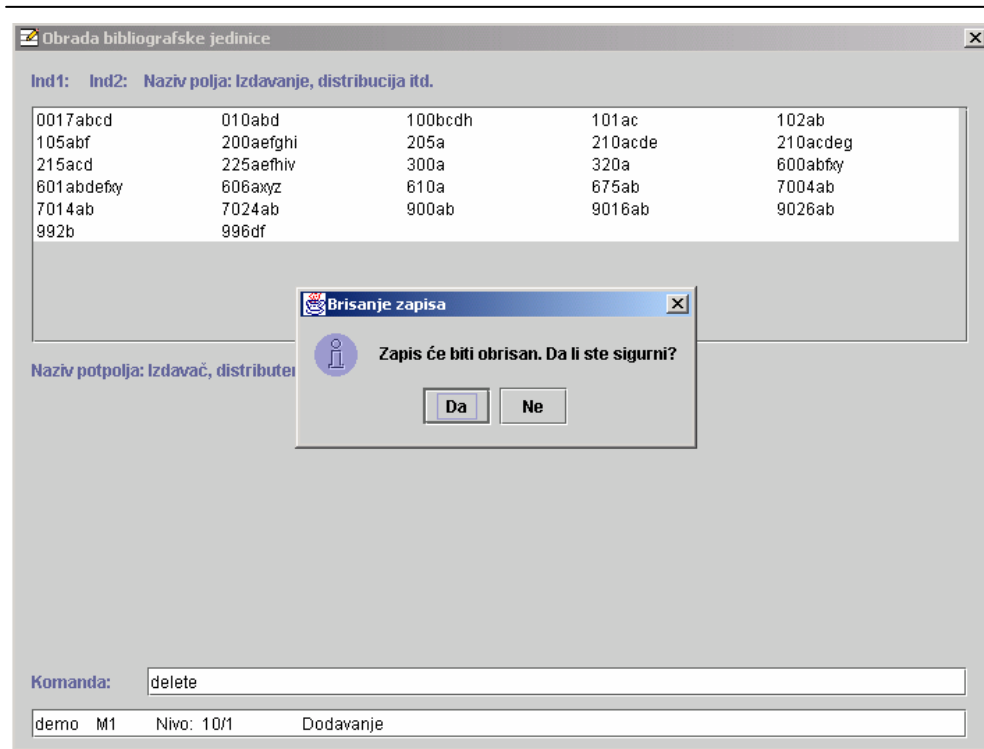
Svjetlost

Komanda:

demo M1 Nivo: 10/1 Dodavanje

Слика 4.1.8 Изглед прозора приликом уноса и/или модификације садржаја потпоља

Библиотекар може да изврши брисање записа из базе података библиографске грађе. Брисање записа врши се задавањем команде *delete* у моду директоријума. Приликом брисања записа, библиотекар се приказује упозорење да ће запис бити обрисан, а библиотекар може да потврди акцију брисања, или да одустане од ње, као што се види на слици 4.1.9. Уколико је библиотекар потврдио акцију брисања, запис се брише из базе података библиографске грађе, при чему се приказује порука да ли је ова акција успешно извршена или не.



Слика 4.1.9 Изглед прозора приликом брисања записа

Детаљан приказ коришћења осталих команди (*connect*, *exit*, *list*, *print* и *quit*) дат је у упутству за коришћење библиотечког софтверског система БИСИС вер. 3 [Сур03].

Литература

- [Boo99] Booch, G., Rumbaugh, J., Jacobson, I., *The Unified Modeling Language User Guide*, Addison-Wesley Longman, Inc., 1999.
- [Буд04а] Будимир, Г., *Контрола квалитета 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
- [Буд04б] Будимир, Г., *MARC записи и XML*, ИНФОТЕКА, фебруар 2004, година. 4, бр. 1, Београд (у штампи)
- [Вид03] Видаковић, Ј., *Моделирање и имплементација библиотечких каталошких листића у XML технологији*, Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2003.
<http://diglib.ns.ac.yu/ndltd/docs/set2/ndltd335/JovanaMagistarski.pdf>
- [Gam95] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995
- [DC] *Dublin Core format specification*,
<http://www.openarchives.org/OAI/2.0/>
- [Eck00] Eckel, B., *Thinking in Java*, 2nd edition, 2000.,
<http://www.EckelObjects.com>
- [Зар04] Зарић, М., Сурла, Д., *Имплементација протокола за прикупљање метаподатака у мрежи дигиталних библиотека теза и дисертација*. ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 99-112.
- [Зер02] Зеремски, М., *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>

- [J2SE] *Java™ 2 Platform, Standard Edition, v 1.3.1 API Specification*, <http://java.sun.com/j2se/1.3/docs/api/>
- [Laz91] *Koncepcija razvoja Sistema naučnih i tehnoloških informacija u Republici Srbiji*, Ministarstvo za nauku i tehnološki razvoj Republike Srbije, Beograd 1991. (Rukovodilac projekta B. Lazarević).
- [Laz96] Monografija *Formiranje i pretraživanje baza podataka u sistemu naučnih i tehnoloških informacija Srbije*, Redaktor Branislav Lazarević, Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd 1996.
- [Mar03] *MARC Format Overview*. Network Development and MARC Standards Office, Library of Congress, 08/13/2003., <http://www.loc.gov/marc/status.html>
- [Миј03] Мијић, В., *XML едитор за опис UNIMARC формата*, Магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2003.
<http://diglib.ns.ac.yu/ndlttd/docs/set1/ndltd133/MijicVMagistarskaTeza.pdf>
- [Мил99] Милосављевић, Б., *Текст сервер UNIMARC записа*, Магистарска теза, Факултет техничких наука, Нови Сад 1999.
<http://diglib.ns.ac.yu/ndlttd/docs/set1/ndltd6/textsrv.pdf>
- [Мил03] Милосављевић, Б., *Прошириви систем за проналажење мултимедијалних докумената*, Докторска дисертација, Факултет техничких наука, 2003.
<http://diglib.ns.ac.yu/ndlttd/docs/set1/ndltd143/xmirs.pdf>
- [NDLTD] *Networked Digital Library of Theses and Dissertations*. Viginia Tech, USA. <http://www.ndltd.org>
- [OAI] *The Open Archives Initiative*. <http://www.openarchives.org/>
- [PMH] *The Open Archives Initiative Protocol for Metadata Harvesting*. <http://www.openarchives.org/OAI/openarchivesprotocol.html>

-
- [Pup00a] Pupovac, B., Vidaković, M., *Modeliranje komandi bibliotečkog informacionog sistema u UML-u*, CD ROM YUInfo'00, Kopaonik, 2000.
- [Пуп00b] Пуповац, Б., *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>
- [Pup01a] Pupovac, B., Surla, D., *Model slučaja korišćenja komandi bibliotečkog informacionog sistema*, CD ROM YUInfo'01, Kopaonik, 2001.
- [Pup01b] Pupovac, B., *Objektno-orijentisana specifikacija komandi bibliotečkog informacionog sistema*, CD ROM YUInfo'01, Kopaonik, 2001.
- [Pup01c] Pupovac, B., *Objektno-orijentisana specifikacija komande new u bibliotečkom informacionom sistemu*, CD ROM VJInfo'01, Beograd, 2001.
- [Пуп01д] Пуповац, Б., *Радно окружење библиотекара*, ИНФОТЕКА, 2 (2001) 1-2, Београд, стр. 105-109
- [RaR02] RationalRose2002, Rational Corporation, <http://www.rational.com>
- [Rum99] Rumbaugh, J., Jacobson, I., Booch, G., *The Unified Modeling Language Reference Manual*, Addison-Wesley Longman, Inc., 1999.
- [Ста99] Станојевић, И., Сурла, Д., *Увод у обједињени језик моделирања*, Група за информационе технологије, Нови Сад, 1999.
- [Sur92] *Plan razvoja bibliotečke mreže Srbije*, Ministarstvo za nauku i tehnološki razvoj Republike Srbije, Beograd 1992. (Rukovodilac projekta D. Surla).
- [Сур03] Сурла, Д., Коњовић, З., Пуповац, Б., Милосављевић, Б., Видаковић, М., Тошић, Т., Зубић, Т., *Упутство за коришћење библиотечког софтверског система БИСИС вер. 3*, Група за информационе технологије, Нови Сад, 2003.

- [Сур04а] Монографија *Дистрибуирани библиотечки информациони систем БИСИС*, Уредници Д. Сурла, З. Коњовић, Група за информационе технологије, Нови Сад 2004.
- [Сур04б] Сурла, Д., Коњовић, З. и др. *Приказ реализације мрежне дигиталне библиотеке докторских, магистарских и дипломских радова*, ИНФОТЕКА, Београд, бр. 1-2/2004. стр. 75-86.
- [Tri98] Tričković, I., Holo, I., Pupovac, B., *Data model of the UNIMARC format in Oracle Designer/2000*, Proceedings of the XIII Conference on Applied Mathematics PRIM'98, Igalo, 1998., pp. 179-186.
- [Uni94] *UNIMARC Manual: Bibliographic Format/International Federation of Library Associations and Institutions*, IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London 1994.
- [Хол97] Холо, И., *Командни језик библиотечног информационог система*, Магистарска теза, Нови Сад, 1997.
- [Hol98] Holo, I., Pupovac, B., Tošić, T., *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