

АЛЕКСАНДЪР НАСКОВ ТРАЙКОВ

***СТАНДАРТИЗИРАНЕ И ИНТЕГРИРАНЕ НА
РАЗНОТИПНИ ИНФОРМАЦИОННИ РЕСУРСИ***

АВТОРЕФЕРАТ

**НА ДИСЕРТАЦИОНЕН ТРУД ЗА ПРИСЪЖДАНЕ
НА ОБРАЗОВАТЕЛНА И НАУЧНА СТЕПЕН „ДОКТОР“**

по област на висше образование

**4. Природни науки, математика и информатика
професионално направление**

**4.6. Информатика и компютърни науки
докторска програма „Информатика“**

Научен ръководител:

проф. д.м.н. Георги Атанасов Тотков

Рецензенти:

проф. д-р Кънчо Йорданов Иванов

доц. д-р Красимира Минкова Иванова

**Пловдив
2017**

Дисертационният труд е обсъден и насочен за защита пред научно жури, на заседание на катедра „Компютърна информатика“ при Факултета по математика и информатика на Пловдивския университет „Паисий Хилендарски“, на 08.02.2017 г.

Дисертационният труд „Стандартизиране и интегриране на разнотипни информационни ресурси“ съдържа 169 страници. Списъкът на използваната литература включва 161 източника, от които 19 на кирилица и 142 на латиница. Списъкът с авторските публикации по темата се състои от 7 заглавия.

Защитата на дисертационния труд ще се състои на 11.05.2017 г. от 11:15 часа в Заседателната зала на ПУ „Паисий Хилендарски“ (Ректорат, ул. „Цар Асен“ №24).

Материалите по защитата са на разположение на интересуващите се в Деканата на Факултета по математика и информатика, Нова сграда на ПУ „Паисий Хилендарски“ (бул. „България“ №236), всеки работен ден от 8:30 до 17 часа.

Автор: Александър Насков Трайков

**Заглавие: СТАНДАРТИЗИРАНЕ И ИНТЕГРИРАНЕ НА
РАЗНОТИПНИ ИНФОРМАЦИОННИ РЕСУРСИ**

ALEXANDER NASKOV TRAIKOV

STANDARDIZING AND INTEGRATING HETEROGENEOUS INFORMATION RESOURCES

Business process in an institution constitutes of collecting, distributing, updating and storing information from various sources. As the usage of information technologies raises in all activity fields, integrating heterogeneous resources becomes important topic.

Resources can be product of various information systems, files with different formats, paper documents and others. This leads to the growth of organizations that develop standards and specifications for integrating heterogeneous resources, tools for archiving, etc.

Purpose and tasks

The primary goal is to research, propose and test standards, methods and tools, suitable for automating the generation and archival of heterogeneous information resources – product of information systems, residing within one organization.

The following research tasks fulfill the primary goal:

Task 1. To research **theories, standards, methods and organizations** in the field of **integrating heterogeneous information resources**.

Task 2. To create data models and architecture of a **system, which collects and stores information resources**, also to **automatically generate metadata**, based on business process work in institutional information structure.

Task 3. To create **software prototype** of a system that collects and archives contents and metadata for information resources (SIHIR).

Task 4. To **experiment** with SIHIR in concrete institutional infrastructure.

СЪДЪРЖАНИЕ

ABSTRACT.....	3
СПИСЪК НА ИЗПОЛЗВАНИТЕ СЪКРАЩЕНИЯ	5
УВОД.....	6
ГЛАВА 1. СТАНДАРТИ, МЕТОДИ И СРЕДСТВА ЗА ИНТЕГРАЦИЯ НА ИНФОРМАЦИОННИ РЕСУРСИ.....	8
1.1. <i>Организации и стандарти</i>	8
1.2. <i>Библиотечни организации и стандарти</i>	9
1.3. <i>Методи и стандарти за описание на ресурси и метаданни</i>	9
1.4. <i>Интеграция на разнотипни ресурси</i>	11
1.5. <i>Цифрови хранилища</i>	12
1.6. <i>Изводи</i>	13
ГЛАВА 2. МОДЕЛИ, АРХИТЕКТУРА, МЕТОДИ И ШАБЛОНИ НА СИРИП	14
2.1. <i>Изисквания</i>	14
2.2. <i>Статични и динамични данни</i>	14
2.3. <i>Структура на метаданни</i>	15
2.4. <i>Архитектура</i>	16
2.5. <i>Методи на метаданни</i>	18
2.6. <i>Софтуерни шаблони за дизайн</i>	19
ГЛАВА 3. РЕАЛИЗАЦИЯ НА СОФТУЕРЕН ПРОТОТИП НА СИРИП	19
3.1. <i>Специализирани шаблони</i>	20
3.2. <i>Шаблон за описание на метаданни</i>	20
3.3. <i>Шаблон за генериране на еднотипови ИР</i>	21
3.4. <i>Шаблон за описание на процес за генериране на метаданни</i>	21
ГЛАВА 4. ИНТЕГРИРАНЕ НА РАЗНОТИПНИ ИНФОРМАЦИОННИ РЕСУРСИ В ПЕУ	22
4.1. <i>Средства за реализация</i>	22
4.2. <i>Реализация на подходи</i>	23
4.3. <i>Експерименти в ПеУ</i>	25
ЗАКЛЮЧЕНИЕ	26
СПИСЪК НА ПУБЛИКАЦИИ	27
ЦИТИРАНИЯ	28
БИБЛИОГРАФИЯ.....	29

Списък на използваните съкращения

БДС	–	Български държавен стандарт
БИС	–	Български институт за стандартизация
ДС	–	Динамични справки
ИР	–	Информационни ресурси
ИС	–	Информационни системи
КДА	–	Краен детерминистичен автомат
ПеУ	–	Пловдивски електронен университет
ПУ	–	Пловдивски университет
РИС	–	Разнотипни информационни системи
СИРИР	–	Система за интеграция на разнотипни информационни ресурси
СУБД	–	Система за управление на база данни
ФС	–	Файлов склад
ЦХ	–	Цифрово хранилище
BIBFRAME	–	Bibliographic Framework
CSS	–	Cascading Style Sheets
DC	–	Dublin Core
DTD	–	Document Type Definition
HTML	–	Hypertext Transfer Protocol
ISO	–	International Organization for Standardization
LOC	–	Library of Congress
MARC	–	Machine-Readable Cataloging

Увод

Бизнес процесите в организации и институции са свързани със събиране, разпространение, актуализация и съхранение на информация от различни източници. С навлизането на информационните технологии във всички сфери на дейност, проблемът за интеграция на разнотипни ресурси, става особено актуален.

Ресурсите могат да бъдат резултат от работата на различни информационни системи, файлове с различни формати, документи на хартиен носител и др., което води до развитието на организации, които се занимават с разработка на стандарти и спецификации за интеграция на разнотипни ресурси, средства за архивиране и др.

Цел и задачи

Основна цел на дисертационното изследване е да се предложат, изследват и апробират стандарти, методи и средства, подходящи за автоматизирано генериране и архивиране на разнотипни информационни ресурси – резултат от работата на информационни системи, работещи в рамките на една организация.

Изпълнението на поставената цел предполага решаване на следните задачи:

Задача 1. Проучване на **теории, стандарти, методи** и организации с предмет – **интегриране на разнотипни информационни ресурси**.

Задача 2. Създаване на модели на данни и архитектура на **система за депозиране и архивиране на информационни ресурси**, както и за **автоматично генериране на метаданни** в рамките на процеси, базирани на институционална информационна структура.

Задача 3. Създаване на **софтуерен прототип** на система за депозиране и архивиране на съдържание и метаданни за информационни ресурси (СИРИР).

Задача 4. **Експериментирание** на СИРИР върху конкретна институционална инфраструктура.

Структура на дисертацията

Дисертационният труд се състои от Списък на използваните съкращения, Списък на фигурите, Списък на таблиците, Увод, 4 (четири) глави (представлящи решения и резултати на поставените задачи), Заключение, Приложение, Списък на използваната литература, Списък на авторските публикации по темата, Списък на забелязани цитирания и Декларация за оригиналност.

В **Глава 1. Стандарти, методи и средства за интеграция на информационни ресурси** са разгледани основните понятия и определения, свързани с информационни ресурси, системи и метаданни. Специално внимание е отделено на изследвания, стандарти и методи за интеграция на разнотипни информационни ресурси, генерирани от различни процеси в бизнес организации. Анализирани са редица съществуващи цифрови хранилища за съхранение и автоматизирано генериране на информационни ресурси като от тях са проучени най-популярните в академичната област. Преведена е класификация на съществуващи платформи. В края на гл. 1, на базата на проучените източници, са направени основни изводи, които водят до формулирането на СИРИР и концепцията за нормализиране на данни.

В **Глава 2. Модели, архитектура, методи и шаблони на СИРИР** са определени изисквания за реализацията на СИРИР, базирайки се на изводите и възможностите на съществуващи ЦХ от гл. 1. Изхождайки от концепцията за нормализация на данни и необходимостта да се удовлетворят зададените изисквания, е представена архитектура на системата като се отделя специално внимание на задълженията и възможностите на всеки от шестте слоя на архитектурата. Дефинирани са метод за генериране на динамични данни, базиран на работа с йерархични структури от метаданни, и основни програмни шаблони за дизайн, които се използват за по-нататъшното реализиране на СИРИР, решавайки Задача 2.

В **Глава 3. Реализация на софтуерен прототип на СИРИР** са представени специализирани програмни шаблони като производни от дефинираните в гл. 2, целящи да поставят основи за реализацията на метода за генериране на данни. След поредица от следствия се формулират общите модели на метаструктури и множество от данни, модели за автоматизиране на процеси и модел на справки, които се използват в архитектурата на СИРИР за нормализацията и генериране на разнотипни информационни ресурс, резултат от работата на различни процеси. Представен е и модел за презентационната част на ЦХ, който се използва от крайните потребители на системата.

В **Глава 4. Интегриране на разнотипни информационни ресурси в ПеУ** е представен проектът ПеУ като подходяща среда за интеграция на СИРИР. Избрани са основните средства за реализация както и системните характеристики, използвани за провеждане на тестове. Базирайки се на представените средства, се реализират зададените в гл. 2 архитектурни слоеве и в гл. 3 модели за метаданните, автоматизирани процеси и справки като нагледно се представя тяхното приложение в работата на СИРИР чрез изпълнението на три експеримента в рамките на ПеУ. Генерираните резултати са подложени на щателен анализ, за да се удостовери валидността и успехът на експериментите.

В **Заклучението** получените резултати са обобщени по задачи 1. – 4. и са изведени основните научни, научно-приложни и приложни приноси на дисертационния труд. Формулирани са перспективи за бъдещо развитие на дисертационната тематика.

Благодарности

Издавам своите най-искрени благодарности на моят научен ръководител – проф. д.м.н. Георги Тотков, за предоставената ми възможност и доверие за провеждане на дисертационно изследване. Сърдечно му благодаря за придобитите знания и умения, проявената отзивчивост и оказаната всеотдайна и безрезервна подкрепа по време на целия период на обучение и работа върху дисертацията!

Благодаря също на колегите от катедра „Компютърна информатика“ и от Университетския информационен център за оказаната подкрепа и съдействие при провеждане на дисертационните експерименти.

Признателен съм и за финансовата подкрепа на изследването, оказана по проекти BG051-PO001-4.3.04-0064 „Пловдивски електронен университет (ПеУ): национален еталон за провеждане на качествено е-обучение в системата на висшето

образование“(2012-2014) и BG051PO001-3.1.08-0041 „Стандартизиране и интегриране на разнотипни информационни и управленски университетски системи“ (2013-2014).

Глава 1. Стандарти, методи и средства за интеграция на информационни ресурси

Информационните ресурси (ИР) представляват колекция от значими стойности (данни), генерирани чрез човешка дейност. Те се запазват в информационна структура като елементи, позволяващи предаването на съдържание, достъпно за крайни потребители [Morales, 2008]. ИР могат да се предават като входящи и изходящи данни на устройства за обработка, складиране или изпращане.

Информационните системи (ИС) са софтуерни системи, които събират, запазват и обработват данни, с цел събиране и предоставяне на информация, знания и дигитални ресурси [Zwass, 2016].

Структурирано описание на данни чрез **метаданни** е похват, позволяващ достъпност, описателност и управление на информационни ресурси, част от ИС, цифрови хранилища (ЦХ) и др. [NISO, 2004].

Схема на метаданни представлява формална структура от атрибути, които описват определен тип информационни ресурси.

Хранилище за метаданни е база данни, отговорна за физическо запазване и каталогизиране на метаданни. Моделът на метаданни в хранилището трябва да е общ и да не дефинира специфични към дадено приложение структури [Marco, 2004].

1.1. Организации и стандарти

ISO (International Organization for Standardization) [ISO] се състои от представители на интернационални организации за стандартизация (по един от нация). Основана през 23 Февруари 1947, организацията спомага за задаването на индустриални и комерсиални стандартни на световно ниво.

Друга интернационална организация е **IEC** (International Electrotechnical Commission) [IEC]. Тя подготвя и публикува стандарти за всички електрически, електронни и сходни технологии. Стандартите обхващат сектори от полупроводници до домашни уреди, слънчева енергия и други. Организацията работи заедно с ISO. Проектите, които са резултат от съвместната работа на двете организации се обозначават с „ISO/IEC“.

По отношение на стандартите, свързани с метаданни и стандартизация към ISO и IEC, един от най-значимите според Брус Баргмейер [Bargmeyer, 2002] е **ISO/IEC 11179 [ISO_11179]**.

Член на ISO и IEC е Българския институт за стандартизация (БИС) [БИС]. Тя създава нови и поддържа утвърдени стандарти като в допълнение развива и национална база данни, състояща се от европейски и чуждестранни стандарти. Всички дейности се извършват от експерти в над 70 активни технически комитети. Такъв е комитетът по „Архивна, библиотечна и информационна дейност“(ТК 16) [ТК_16]. През 2012 г. публикува стандарта **ISO 23081 [ISO_23081]**. Разглеждайки ресурсите като „документи/записи“, стандартът се фокусира върху приложимостта на метаданните и

тяхната роля в работни процеси, без да задава набор от технологични средства. При създаването на запис, първата задача на метаданните е да определят работния му контекст, като с разработването на ресурса се добавят нови нива от данни, които конкретизират неговата употреба и начин на управление, спомагайки за натрупването на информация, свързана с контекста на записа и работните процеси. Основни свойства на метаданните, описани в стандарта, са **съвкупност** и **наследяване**. Съвкупността се характеризира с възможността един документ да бъде съставен от един или повече подкласове.

Съвкупността се характеризира с възможността един документ да бъде съставен от един или повече подкласове.

С наследяването на метаданни от *по-висока към по-ниска съвкупност* се „осигурява последователност при определяне на метаданните и се гарантира, че дефинираните в по-високо ниво характеристики не трябва да се повтарят за всяко от по-ниските нива”.

1.2. Библиотечни организации и стандарти

DC (Dublin Core) [DC] – е американски библиотечен консорциум, базиран в Дъблин. Целите на организацията се определят от интернационална интердисциплинарна група от професионалисти, чиито работни позиции варират от библиотекарьство, компютърни науки до кодиране на текст.

Схемата на DC е множество от термини, които могат да се използват за описание на уеб ресурси (файлове като документи, аудио, изображения, уеб страници и др.), както и физически ресурси като книги или обекти (художествени творби).

IFLA (International Federation of Library Associations and Institutions) [IFLA] е интернационална организация, представляваща интересите на заинтересовани в библиотекарьството и информацията професионалисти.

ББИА (Българска библиотечно-информационна асоциация) [ББИА] е доброволно българско сдружение, което се занимава с промотирането и усъвършенстването на библиотечната и научно-информационна дейност. Между 20 и 21 Октомври 2016 година е проведен форум на тема „Стандартизацията в библиотеки“, на който е представен и дискутиран нововъведения стандарт по БДС за международна библиотечна статистика (ISO 2789:2015), разисквани са модерните стандарти, както и други съвременни подходи за класификация и каталогизация и съпътстващите ги системи.

1.3. Методи и стандарти за описание на ресурси и метаданни

DTD (Document Type Definition) [DTD] е множество от декларации, които приспадат към семейството от SGML (Standard Generalized Markup Language) [SGML] документи на маркиращи езици (например XML и HTML). DTD описва структурата на клас от документи чрез декларации на елементи и атрибутни списъци. Декларациите на елементи задават възможни множества от елементи в даден документ, те определят и естеството на данни, които ще бъдат показвани в елементите. Декларациите на атрибутните списъци формулират допустимите атрибути и техните типове за всички

зададени елементи. DTD може да бъде описано в XML документ или чрез външен източник.

RDF (Resource Description Framework) [RDF] се използва за моделиране на уеб ресурси. Той е с абстрактен синтаксис, който служи за база на всички езици и спецификации, способни да поддържат описанието му. Има конкретни реализации с езици като XML, JSON и Turtle и други. Основан през 1999, последна версия 1.1 е пусната през 2014 година.

Структурата се изгражда от три изрази, които следват връзката субект-предикат-обект. Субектът определя ресурс, докато предиката разграничава възможни черти и връзка с обекта на същия ресурс. Обектът на свой ред е стойност или резултат на предиката в контекста на ресурса. Множество от тройки определя RDF граф.

MARC (Machine-Readable Cataloging) [MARC] е стандарт, описващ дигитални формати за каталогизиране на библиотеки. Има три основни аспекта на записи:

- Обозначаване на поле – предоставя конкретна информация за елемента като част от запис (например автор, език, тип данни;
- Структура на запис – записите се запазват и предават в бинарен формат. Обикновено се състоят от няколко записа;
- Съдържание – тъй като MARC кодира информация само за елемента, а не за съдържанието, последното се управлява от друг стандарт.

BIBFRAME (Bibliographic Framework) [BIBFRAME] разработван под ръководството на LOC [LOC]. Той е модел на данни за библиографско описание. Разработен, за да замени MARC, той е аналог на FRBR, базиращ се на RDF и има три слоя на абстракция:

- Произведение (Work) – определя абстрактен елемент от концептуален каталог;
- Инстанция (Instance) – инстанция на произведение е абстракция, която представлява реализация на произведение (с физически или цифров произход);
- Елемент (Item) – реално копие (физическо или цифрово) на инстанция. Представя информация като локация (физическа или виртуална).

Версия две на BIBFRAME добавя нови ключови концепции, които имат връзка към основните класове:

- Агенти (Agents) – хора, организации или юрисдикции, които са асоциирани с произведение или инстанция;
- Субекти (Subjects) – произведението може да има една или повече концепции като тема, място, събитие, агенти и др.;
- Събития (Events) – тяхното описание може да служи за съдържание на произведение.

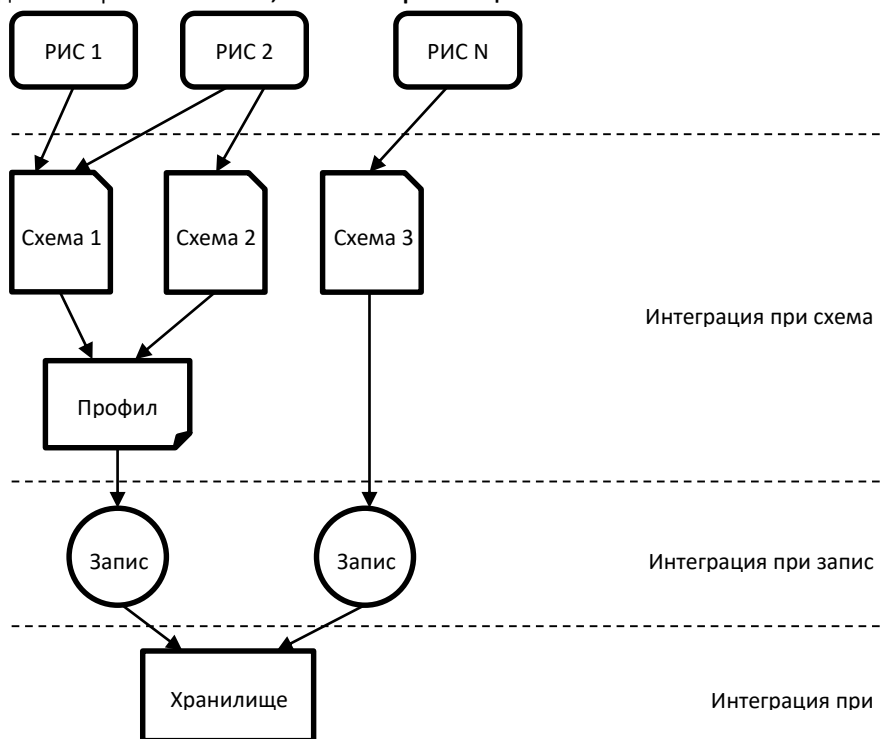
RDF речникът на модела се състои от класове и свойства. Класовете включват основните типове, и подкласове, свойствата описват характеристики на описвания ресурс, заедно с възможни връзки с други ресурси. Например, едно произведение може да е превод на оригинална.

1.4. Интеграция на разнотипни ресурси

Хранилищата, планирани и разработени според даден стандарт (като BIBFRAME), интегрират ресурси от всички отдели в организацията, които са разработени по стандарта. При наличие на разнотипни информационни системи (РИС), е необходимо да се определят възможните нива на интеграция (Фигура 6).

Множествата от РИС могат да генерират различни ресурси, чиито метаданни се базират на различни схеми. Възможно един или повече РИС да споделят общ формат, ако се подчиняват на еднакви стандарти.

ИР (записи) могат да се генерират директно на базата на една или няколко схеми, обединени в общ профил. Хранилището приема записите от различните РИС, създава връзки между тях и позволява достъп за всички заинтересовани потребители. По така зададения поток от записи, интеграционните нива могат да се идентифицират в следните направления: **схема, запис или хранилище**.



Фигура 6. Поток на запис от РИС към хранилище

В интеграция на ниво схема могат да се разграничат няколко способа. Първият е извличането на **производна** схема от основната. При сложни структури и множество

източници на данни, понякога е необходимо да се адаптират или преведат части от схемата или кодировката в един или повече формати.

В контекста на схемите, “поякога поддржката на съвместимост между близки стандарти изисква създаването на мета-диалекти или така наречените **пътеки**” (cross-walks) [Khan, 2015].

Когато се изисква свързването на няколко схеми, за да се извлече и асимилира най-важната информация от записите, се изготвят **профили** [Heery, 2000], състоящи се от елементи на една или повече схеми

Интеграция на ниво запис се изисква предимно, когато схемата на метаданни е вече разработена по конкретни изисквания и стандарти и не подлежи на промени. Първият подход на това ниво е **обръщането (превод) на запис** във формат, сходен на друга схема. Обикновено, всички записи се събират на централно място, в което се подготвят за обръщането към общ централизиран формат. Превода на записи може да бъде изключително труден, ако структурата на елементите е сложна и включва подмножества. Изисква се обхождане на елементите в дълбочина и използване на специални **речници** за правилно асоцииране на записите.

Интеграцията на ниво хранилище поставя изискване за консистентност при откриването и работата със записи в хранилища. Нужно е те да разпределят адекватно новопостъпилите ресурси по категории.

Друг подход за интеграция на ниво хранилище е приемственост на различни бази данни. Такава интеграция е възможна, ако се избере едно ключово поле, за връзка измежду всички атрибути на схемите.

1.5. Цифрови хранилища

Цифровото хранилище (ЦХ) съдържа както самите ресурси (напр. под формата на „асети“), така и метаданните, които ги описват. Системата от метаданни е основен компонент на всяко хранилище. Някои от основните ЦХ са:

- **DSpace** [DSpace] е софтуерно приложение за управление на дигитални хранилища с отворен код – резултат на съвместен проект с участието на лабораториите на Масачузетския технологичен институт и компанията Хюлет Пакар;
- **Fedora** [Fedora] е безплатна софтуерна платформа за изграждане на **цифрови** хранилища;
- **EPrints** е софтуерна платформа с отворен код за изграждане на дигитални хранилища, разработена от Университета в Саутхамптън;
- **EQUELLA** [EQUELLA] е платена софтуерна платформа за изграждане на цифрови хранилища за учебни материали и документи с изследователско, медийно и библиотечно съдържание;
- **Opus** (Online Publications of the University of Stuttgart) [OPUS] е безплатна софтуерна платформа, разработена през 1998 г. от Щутгартския университет, която позволява да се конфигурират сървъри за съхраняване на документи.

1.6. Изводи

При съпоставка на по-стари и утвърдени стандарти, като MARC, с тепърва навлизащия BIBFRAME, се разкрива ново ниво на абстракция, което налага LOC. Целта е да се обхванат повече възможности за описание на ресурси, които биха били актуални и за в бъдеще. Тази концепция се разработва върху представата за съществуващите към момента ресурси, акомпанирани с техните характеристики и технически възможности.

Стандартите разчитат на употребата на конкретни формати (например, RDF за LOM, DTD за EAD стандартите) на данни, които не винаги се поддържат от всички ИС в дадена организация. По настоящем, няма платформа, която да поддържа всички налични стандарти, формати и програмни способности.

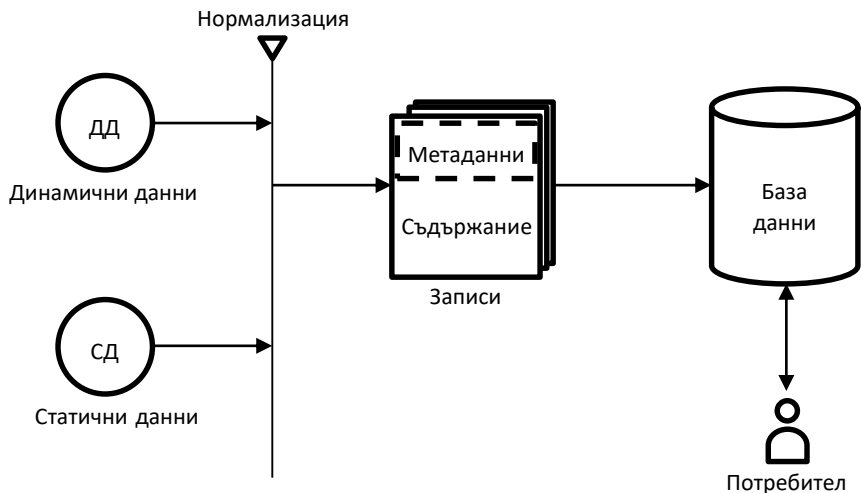
Дори и това да се промени, след известно време ще се появи нов стандарт или формат, който трябва да се интегрира. Може да се вземе за пример бъдеща схема на метаданни, която се описва и предава чрез JSON – доста по-лек от XML [JSON_XML] и широко използван при комуникация между РИС до толкова, че „е започнал да измества уеб услуги, които традиционно работят с XML“ [Sporny, 2010]. Въпрос на време е организации като DC и LOC да наложат приемственост с JSON формата.

Наблюдава се преходност в модерните организации, при които информационните технологии се изменят, а с тях и изискванията и представите за обработка и предаване на ресурси.

Категоризирането на записи трябва да се осъществява динамично и често (без ръчна намеса). Ресурсите могат да се генерират извън системата, на базата на самостоятелни процеси, или да разчитат на платформа, която да образува метаданни и да връща резултат в следствие на зададени аргументи, играейки ролята на посредник между различни РИС. В този контекст, можем да разбием ИР на две основни групи: *статични и динамични данни*. *Статичните* се разглеждат като крайни пакети от данни (документи), при чието генериране не участва платформата. Те нямат схема на метаданни и могат да се състоят от различни формати. *Динамичните данни (справки)* са директен резултат от вътрешните процеси на хранилището, позволявайки успешно извличане на метаданни.

Структурата на статичните данни не е достъпна, следователно търсенето на статичен ресурс може да се извършва чрез извличане на съдържание. Динамичните ресурси поддържат и възможност за филтрация и съпоставяне по атрибути на метаданни. Процесът на обработка, аташиране на метаданните и запазване на съдържанието на двата типа данни, се нарича **нормализация** (Фигура 7).

В дисертационното изследване се предлага платформа, която реализира работна рамка, която поддържа нормализация на двата типа данни (статични и динамични) – **Система за интегриране на разнотипни информационни ресурси (СИРИР)**.



Фигура 7. Нормализация в СИРИП

Глава 2. Модели, архитектура, методи и шаблони на СИРИП

На база на проведения в гл. 1. анализ, са определени изисквания за моделиране на данни на ИР и за дефиниране на архитектура и функционални възможности на СИРИП. Предложени са три подхода за генериране на метаданни за ИР при интегриране на ИС. Разгледани са софтуерни шаблони за дизайн, които се използват за дефиниция на софтуерни подходи за реализиране на динамична система от метаданни.

2.1. Изисквания

Моделите на данни на ИР трябва да спазват следните **критерии**:

- да позволяват изменения на схемата си (**адаптивност**);
- да разширяват и коригират класификацията си (**динамичност**);
- да бъдат генерирани чрез интегрирани работни процеси (**автоматизираност**);
- да обхващат всички характеристики и свойства на ИР (**описателност**);

СИРИП трябва да удовлетворява следните **най-общи условия**:

- да се базира на отворени (за следващо развитие) софтуерни решения (**отвореност**);
- да споделя своето съдържание с други информационни системи (**споделеност**);
- да предлага средства за лесно намиране и извличане на ИР (**достъпност**);
- да е оперативно съвместимо с други технологии и системи (**стандартизираност**), и т.н.

2.2. Статични и динамични данни

Статичните данни (документи) се предават на системата като реализирани ресурси, в чието създаване не е участвал СИРИП. Те се класифицират в две групи:

- изображения - използвайки библиотеката за разпознаване на текст, системата разпознава и извлича текстови и цифрови данни от графичното изображение);
- четими документи - основното текстово съдържание се извлича под формата на редица от знаци, без сегментация на текстови, числови или други смислови единици.

Динамичните данни (Динамични справки - ДС) се създават в СИРИР. Те са комбинация от съдържание (текст) и метаданни, извлечени по време на изпълнение на интегриран в системата процес. Благодарение на набора от метаданни, те биват добре описани, а в следствие на това и лесно намирани.

2.3. Структура на метаданни

Естеството на големите организации изисква класифицирането на документи по групи и под-групи (отдели), затова се придава структурно свойство към метаданните, като се взима предвид, че всеки клас може да има n (където $n \in \mathbb{N}$) на брой подкласове. Формално те могат да бъдат дефинирани като растящо дърво $M = (C, E)$, чиято двойка се състои от множество от метаданни C и множеството от ребра E , където всеки елемент E представлява два елемента от C . Метаданните са подредена редица $C = (t, P, R)$, където $t \in T$, T е множеството типове метаданни, а P и R са множества съответно за свойства и представяне на данните.

Дефинира термина **наследяване „отдолу-нагоре“** (обратно наследяване) на класове [Chirila, 2016], при който всички свойства на дъщерните възли се предават на родителския клас в рекурсивен ред - $M = \bigcup_{i=1}^n M'_i, i \in \mathbb{N}$. По този начин родителите обединяват множествата от свойства и откриват сеченията на сходност.

Не се изключва и възможността всеки клас от редицата на същото ниво, да презаписва сходните свойства на предходния елемент или да извършва други действия в зависимост от контекста на сечението (например, при аритметичен и лингвистичен). Така $M = \bigcup_{i=1}^n c(M'_i)$, където c е функцията за всеки контекст.

Метаданните като резултат от изпълнението на процес

Структурата на всяка ДС се състои от една или повече (под)секции, съдържащи данни. Те са резултат от изпълнението на автоматизиран процес (по ISO 17369), изпълняващ действия за извличане на релевантна информация. Без да се взима под внимание техният контекст, действията винаги имат строга последователност, като стартирането на следващо действие може да зависи от успешното изпълнение на предходно. Такова поведение може да се опише чрез краен детерминистичен автомат (КДА) с подредена редица от 5 елемента $A = (Q, \Sigma, \delta, q, F)$ [Maheshwari, 2016], където:

- Q е крайно множество, чиито елементи са действията;
- Σ е крайно множество, представляващо входните аргументи към всяко действие;
- $\delta: Q \times \Sigma \rightarrow Q$ е функцията за преминаване към следващо действие;
- q е началното действие от Q ;
- F е подмножество на Q , чиито елементи са действията, определящи крайното състояние при генериране на документ;

Резултатът от всяко изпълнение на δ , води към следващо действие Q и попълва множеството Σ , докато не достигне действие от множеството F . Така след като всички действия са изпълнени, Σ образува пълен комплект от резултати, които в следствие на предходното твърдение за структурата на документа, могат да се разглеждат като подсекции или свойства на метаданни, принадлежащи към самия обект (ДС). С помощта на обратно наследяване, този обект може да намери сечението, или сходните свойства между подсекциите си.

Класифициране на метаданни

За да се автоматизира разпределението на гори от отдели и класифицирането им в една обща структура, се предлага задаване на *акумулиращи функции*. Те представляват набор от критерии, по които се извличат подходящи ДС и се разпределят по категории. Една категория може да има повече от една функция.

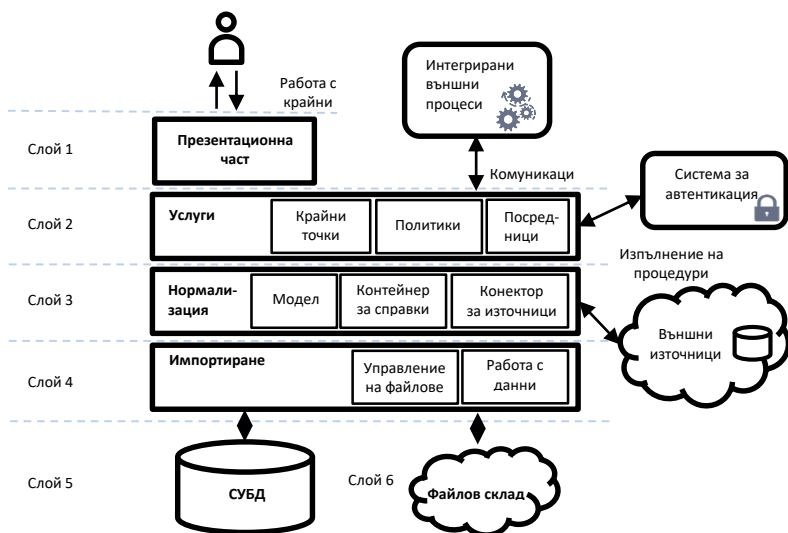
Категориите могат да се добавят по всяко време и в различна дълбочина и широчина на класификацията. След задаване на акумулиращи функции, всички сходни ИР биват автоматично откривани и добавяни към тях.

2.4. Архитектура

В СИРИР могат да се идентифицират 6 (шест) основни слоя (Фигура 10). Цел на подобно разслояване е да се капсулират специфични групи от функционалности по такъв начин, че при обновяване на някоя от тях, това да не засяга системата като цяло.

Тя може да се разглежда като алтернативен подход и ядро за приложения, които да бъдат двигател и посредник при генерирането и автоматизирането на динамични данни (справки), чрез осигуряване на входни и изходни точки за приемане и предаване на ресурси. Силата на СИРИР е в асимилирането на данните в момента, в който преминат през механизмите на системата, така че те да се интегрират с други типове документи, без да се обременяват със стандартите, които са спазвали. Дори съхранението на физически документи се разглежда като придатъчно (допълнително) към справките явление, което може да бъде постоянно за определени типове документи или преходно за ресурси, използвани от външни процеси като способ за временно утвърждаване на състояние.

Архитектурата трябва да се съобразява с изводите и поставените изисквания за нормализация на данните. В допълнение е необходимо системата да предоставя среда за работа на отделни потребители и групи и за протичане на организационни процеси, свързани със събиране, обработване, съхраняване и разпространение на разнородни информационни ресурси (документи, справки, статии, мултимедийни файлове и др.). Последното изисква от СИРИР да разполага с подходящи за информационни процеси софтуерни инструменти, спомагащи за ръчно и/или автоматично генериране и депозиране на разнотипни информационни ресурси (файлове) и метаданни към тях



Фигура 10. Архитектура на СИРИП

Слоеве на архитектурата са както следва:

- **Презентационна част** – уеб сайт, който предоставя визуални средства за работа със СИРИП като инструменти за депозиране на документи и справки по категории;
- **Услуги** – изпълнява всички дейности като депозиране, търсене по критерии и извеждане на документи се идентифицират и изпълняват от слоя за услуги. Разграничат три компонента, базирани на SOA: интерфейси, политики и посредници;
- **Нормализация** – има за цел да приведе („нормализира“) входни данни за документи и категории, доставени от съответни услуги, в унифицирана форма, достъпна и „разбираема“ за всички модули, обслужващи СИРИП. Статичните и динамичните данни преминават през този слой, за да се преобразуват в еднотипови ИР. Те се обработват в СИРИП чрез три основни модула: **конектор за източници, контейнер за справки и модели**;
- **Импортиране** – Слой за импортиране работи с нормализирани данни за справки и категории. Изграден е от два модула, които се грижат за запазването и откриването на нормализирани метаданни и за физическото управление на изходни файлове – **работа с данни и управление на файлове**;
- **Система за управление на База данни** - Този слой представлява конкретна система за управление на база данни (СУБД), която има достатъчно способности, за да изпълни функционалностите, заложили в предишните слоеве;
- **Файлов склад** - Файловият склад (ФС) представлява платформа за архивиране на различни видове ресурси (по изискване на ISO 23081). Трябва да поддържа разнотипни файлове с различни размери и кодировка. ФС не се интересува от първоизточника и начина на генериране.

2.5. Методи на метаданни

Метод за генериране на метаданни за ИР при интегриране на ИС

На база нива на интегриране на ИР в глава 1, могат да се изведат **три основни подхода** за генериране на метаданни:

Подход 1. Използване на специализирани инструменти за едновременно генериране на ИР и съответни метаданни;

Подход 2. Интегриране на шаблон за едновременно създаване на съдържание и метаданни на еднотипови ИР;

Подход 3. Надграждане на външни процедури по един от следните начини:

- a) „Дописване“ по конвенция на процедурата;
- b) Използване на инструменти от подход 1. („вграждане“ на процедурата в съответни инструмент).

Подход 1. е най-лесен за употреба, защото има достъп до всички вътрешни функции на СИРИР.

Въпреки че първият подход е най-подходящ, много организации биха прибягнали до втория, за да избегнат необходимостта от повторно описване на множество схеми. Третият подход е най-малко желан, но понякога неизбежен. Ако дадена организация разполага с достатъчно ресурси, е препоръчително, тя да пренапише функциите (подход едно). Когато външните процедури имат уникален достъп до данни или начин на генериране на ИР, който иначе не може да бъде възпроизведен от системата, остава единствено третият подход.

Метод за търсене в динамична система от метаданни

Търсене в динамична система от метаданни е необходимо се използва за откриване на конкретни документи в гора от отдели или при класифициране чрез акумулиращи функции. Неговата заявка се образува като булев израз, при който елементите се обединяват в конюнкция, а съпоставките на техните сечения в дизюнкция. Структурата на израза изглежда по следния начин:

- M е дърво с всички документи (записи в БД), където множеството S са наличните метаданни, а P и R са множества съответно за свойства и представяне на данните;
- F са използваните филтри ($F \in C$);
- di_n са елементите на метаданни, $di_n \in R, n \in \mathbb{N}$;
- s_m са сеченията, $s_m \in P, m \in \mathbb{N}$;
- f е булева функция, която изпълнява условието на даден елемент - $f(di_n, s_m)$;
- Ако BQ е функция, която връща резултата от откритите елементи, то $BQ(M, F) = \bigvee_n^R \bigwedge_m^P f(di_n, s_m)$.

Модел за генериране на еднотипови ИР чрез шаблон

ДС нямат физически документ, а се материализират с допълнителен инструмент за генериране на файлове. Той може да се реализира с помощта на готови софтуерни решения за създаване на документи във физически формат.

Необходимо е да се определи подход за подреждане на елементите (съдържание и метаданни). Този проблем може да се реши с помощта на под-компонент шаблон към контейнера за справки.

Компонентът позволява определяне на специфични шаблони за всяка справка, които задават позицията и формата на метаданни, съдържание и заглавни елементи. В случай, че дадена справка няма собствен шаблон, тя използва глобален шаблон по подразбиране. Структурата на глобалния шаблон е разделена на три части:

- Заглавна част – състои се от идентификационен номер, дата на създаване и обновяване и автор (ако е наличен), разделени на нови редове;
- Метаданни – подредени в параграфи, като всеки параграф е определен чрез името на метаданните, а под него са изброени със запетая двойките елемент-стойност. Важно е да се отбележи, че метаданните са секции към конкретна справка и тяхното наименование не зависи от конкретен стандарт;
- Текст – свободен текст, ако има такъв. Той може да е слят или логически разбит на страници.

2.6. Софтуерни шаблони за дизайн

Шаблоните за дизайн улесняват многократната употреба на успешни дизайни и архитектури“ при проектиране и интеграция на обектно-ориентиран код [Gamma, 1994]. Те се използват при дефиниция на софтуерни подходи за реализиране на динамична система от метаданни. Използват се три софтуерни шаблона:

- **Композиция** - е структурен шаблон фокусиран върху свързване на обекти в дървовидна структура. Този шаблон позволява еднакво третиране на единични обекти и на йерархични структури на обекти от един и същ тип;
- **Абстрактна фабрика** - определя интерфейс или абстрактен клас, създаващ семейства от взаимосвързани обекти, без да задава конкретни класове. Приложим е когато системата трябва да остане независима от начина на генериране и композиране на нейните обекти (продукти);
- **Миниобект** - структурен шаблон с обектен обхват, който е подходящ за ефикасна работа с големи количества обекти [Gamma, 1994]. Той е споделен и се използва в няколко контекста едновременно, без да взима под внимание какви са те. Така се намаля използването на много памет като се ограничава повторемостта при генериране на много обекти.

Глава 3. Реализация на софтуерен прототип на СИРИП

В Глава 3. са реализирани, заложените в гл. 2 модели на данни чрез шаблоните за описание на метаданни, за генериране на еднотипови ИР и за описание на процес и създаване на метаданни. Те са подходящи за подходи 1, 2 и 3б за генериране на метаданни и са следствие от дефинициите на шаблони за фиксирана дървовидна структура и контекст, които имат общ обхват на действие.

3.1. Специализирани шаблони

В резултат на обратното наследяване при метода за генериране на ДС, се появява необходимостта за две дефиниции на шаблони. Техните свойства се мултиплицират извън възможностите на метаданните и автоматизираните процеси:

- **Връзка** – структурен шаблон с обектен обхват и взаимства от дефиницията на шаблон „Композиция“ с няколко основни разлики: веднъж добавен елемент, не може да се премахва или премества на друга позиция, а всички свързани компоненти, могат да съдържат под-елементи, като се елиминира необходимостта от листа; методът за изтриване на компонент е премахнат, като на негово място е добавена специализирана функция за приемане на родител от дъщерен клас;
- **Контекст** - В метода за обратно наследяване е описана концепция за намиране на сечение на общи елементи в множество. Метаданните трябва да работят с общи обекти. Част от концепцията се покрива от шаблона „Миниобект“, при която основният проблем е управлението на външни състояния на конкретни миниобекти. За тази цел се дефинира сходен на „Миниобект“ структурен шаблон с обектен обхват, наречен „Контекст“. При новия шаблон, миниобектите (тук определяни като контексти) се свързват с клиенти чрез посреднически клас, наречен референция.

3.2. Шаблон за описание на метаданни

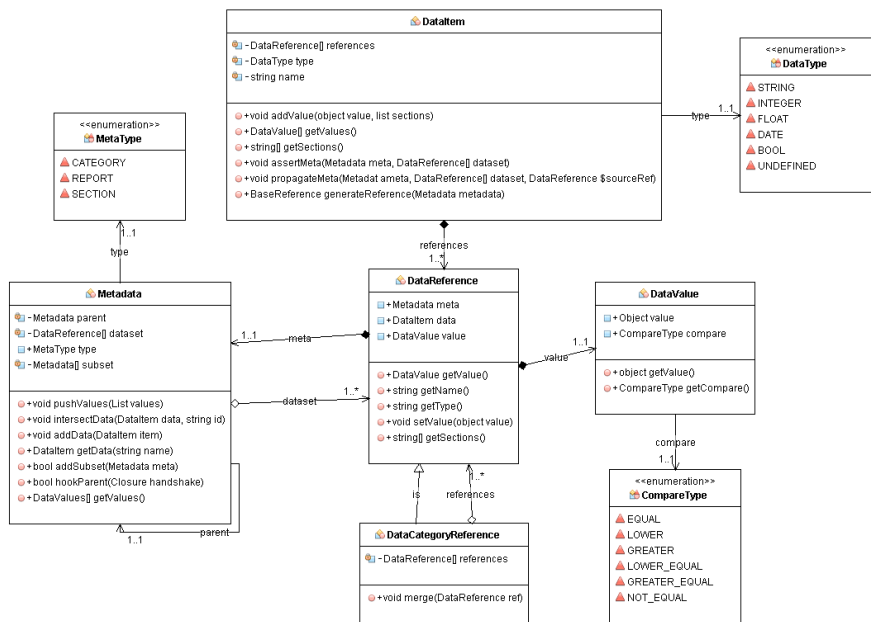
Свойството за намиране на сечение на контексти при дървовидна структура е приложимо за общ модел на метаданни и в допълнение покрива изискванията за структура на данните в метода за автоматично намиране на сходства.

На Фигура 47 е изобразен общ модел на структура на метаданни, която описва разнотипни ресурси. Структурата представлява дърво, което може да се адаптира към различни изисквания. Всяко звено в йерархията е представено от отделен възел метаданни, които имат описателни свойства, с помощта на множество от данни.

Данните са стандартизирани по типове и се използват от всички звена (без да се нарушава контекста на метаданните) за консистентност и лесна филтрация.

Участници са:

- Метаданни - базиран на шаблона „Връзка“ и поддържа йерархична структура, при която действията представляват показване на множество от данни;
- Елемент данни - имплементация на шаблона „Контекст“, който запазва основните си действия за създаване и за управление на референции;
- Референция - представлява управление на конкретна стойност на характеристика;
- Стойност - позволява на крайните потребители да предават начин за съпоставка на данни;



Фигура 47 Общ модел на метаданните

3.3. Шаблон за генериране на еднотипови ИР

Справките са в основата на резултатите, генерирани от автоматизирани процеси. За да се нормализира работата с тях, трябва да се представят няколко основни модела.

Контейнерът за справки предоставя интерфейс за ползване от клиенти чрез слабо обвързване, като позволява намиране на справка. Потребителите не се „интересуват“ от къде идва справката, а използват нейното име, за да я идентифицират. Резултатът е работеща рамка (шаблон) на справка, която може да генерира инстанции по предварително дефинирани аргументи.

3.4. Шаблон за описание на процес за генериране на метаданни

Базиран на принципа на КДА. Автоматите се разглеждат като динамични процеси, които са съвкупност от стъпки. За успешното изпълнение на процес може да са необходими входни аргументи, предоставени от потребителите, участници в процеса.

Резултатът от такива процеси може да е справка или други консолидирани изходни стойности. В общия план, резултатите зависят от аспекта, в който се разглежда процесът. Има няколко участника:

- Процес - описва серия от стъпки, които се изпълняват в предварително определена последователност;
- Стъпка - представлява едно действие от низа от възможни действия в процес;
- Списък с аргументи - представлява списък с аргументи, с които работят отделни стъпки. Той преобразува подреден списък в асоциативен масив;

- Област за аргументи – управлява аргументите на всички стъпки в процес;
- Функция за изпълнение - обвивка, която нормализира всички възможни програмни действия;
- Резултат от стъпка - резултатът се обработва от стъпката и се предава на процеса под формата на обект;

Глава 4. Интегриране на разнотипни информационни ресурси в ПеУ

Реализациите в тази глава се базират на изискванията и архитектурата в гл. 2 и шаблони, реализирани в гл. 3.

Проведен е експеримент със СИРИР като ядро на Цифрово хранилище (ЦХ) като част от електронното обучение [Донева, 2014] и динамично управление и оценяване на качеството във висшето образование [Тотков, 2016а] в Пловдивския електронен университет (ПеУ) [ПЕУ, Тотков, 2014, Тотков, 2014а], на база съществуващите инструменти за автоматизация на процесите по извличане на данни от различни информационни системи [Гафтанджиева, 2014].

Цифрово хранилище [Трайков, 2016] има за цел да складира справките, генерирани от подсистемите на ПеУ и научни разработки, медийни файлове и други ресурси, необходими за осъществяване на дейностите на университета. При добавянето на нов ресурс, ЦХ трябва да класифицира и запази метаданните му, заедно с физическо копие на ресурса. Системата трябва да открива сходни документи с помощта на методите за обратно наследяване и сечение на свойства.

Бизнес логиката на ПеУ е изградена с помощта на работни процеси [Пашев, 2015], като в контекста на системата, те описват дейности в университета, за които автоматично се генерира потребителски интерфейси. Структурата на бизнес процесите наподобява граф, в който всеки възел може да е визуален панел или резултат от взаимодействие с потребител. Подобен резултат е генерирането на справка в ЦХ или запазването на съществуващ файл в хранилището. И в двете ситуации, ЦХ приема файлове и на база входните аргументи по предварително зададен КДА, генерира клас с подсистеми от модели.

Динамичните справки в ЦХ се базират на разработената номенклатура на дела със срокове за съхранение на ПеУ [Алендарова, Тотков и др., 2016]. Нейната структура е на две нива: първото обхваща основни функции на университета (управление, образование, наука и др.), а второто – прилежащите дела и съответните им справки.

4.1. Средства за реализация

Използвани са следните средства:

- Elasticsearch [ELASTIC] - за съхраняване и търсене на цифровизирани документи;
- PHP [PHP] – програмен език;
- Laravel [LARAVEL] – работна рамка;
- Swift – файлов склад;
- HTML, CSS, JavaScript и jQuery – за описание на уеб страницата;
- Tesseract [TESSERACT] - Разпознаване и извличане на текст от изображение (OCR);

4.2. Реализация на подходи

В раздела са представени два примера за генериране на метаданни и ИР, базирани на под-ходите за интегриране на ИС, моделите и шаблоните за метаданни и процеси – реализация на вътрешен процес по подход 1. и интегриране на външна схема по подход 2. Представени са реализациите на презентационната част (слой 1) и слоя „Услуги“.

Реализация на вътрешен процес по подход 1.

Създаден е специализиран процес, който в три стъпки свързва процедури от база данни с вътрешен процес (Фигура 52). Процесът се добавя към справка, която се зарежда в контейнера за справки. Всяка стъпка изисква входни параметри, които се определят от списък с аргументи. Тъй като всички стъпки имат едни и същи аргументи, списъка се клонира за всички тях.

```
$cb = new ContextCallbackBuilder("exec", $this->context);
$args = new ArgumentList(["argument 1", "argument 2"]);
$process = new Process();
$step1 = new Step("section 1", $cb->getCallback("Procedure1"),
$args->cloneObject(), "section 2");
$step2 = new Step("section 2", $cb->getCallback("Procedure2"),
$args->cloneObject(), "section 3");
$step3 = new Step("section 3", $cb->getCallback("Procedure3"),
$args->cloneObject());
$process->addSteps( $step1, $step2, $step3 );
$this->addReport( new ProcessReport("test report", "test category",
$process) );
```

Фигура 52. Създаване процес от три стъпки и добавянето му в справка

Резултатът от всички стъпки се акумулира в процеса като метаданни и тяло на ИР. По този начин вътрешният за СИРИР процес може да адаптира външни източници от данни при генерирането на ДС. Изпълнението на тези процеси може да става циклично, при задействане от външна система или ръчно.

Интеграция на външна схема по подход 2.

Създаден е източник на справки от схеми (SchemaSource), който извлича данни от различни контексти на външни системи за справки. На Фигура 54 е представен метод, който генерира справки за всяка съществуваща схема от контекста.

```
private function parseReports(){
    $schemas = $this->context->getSchemas();
    foreach( $schemas as $schema ){
        $process = new Process();
        foreach( $schema->getSchemaSets() as $set ){
            $step = $set->generateStep();
            $process->addStep($step);
        }
    }
}
```

```

        $this->addReport( new ProcessReport($schema->getName(),
$schema->getDepartment(), $process) );
    }
}

```

Фигура 54. Обработване на справки от схеми на външни системи

Контекстът определя релативните данни, необходими за създаването на стъпки от процес. Представен е пример с интеграция на Jasper. Справката се извлича по име и отдел (Фигура 55). Тъй като се открива чрез контейнера, нейната връзка с Jasper не е от значение. Тази абстракция от първоизточника унифицира всички ДС и предлага общи средства за работа. Подобно на подход едно, справката се изпълнява, като резултатът формулира тялото на ИР и структурата от метаданни.

```

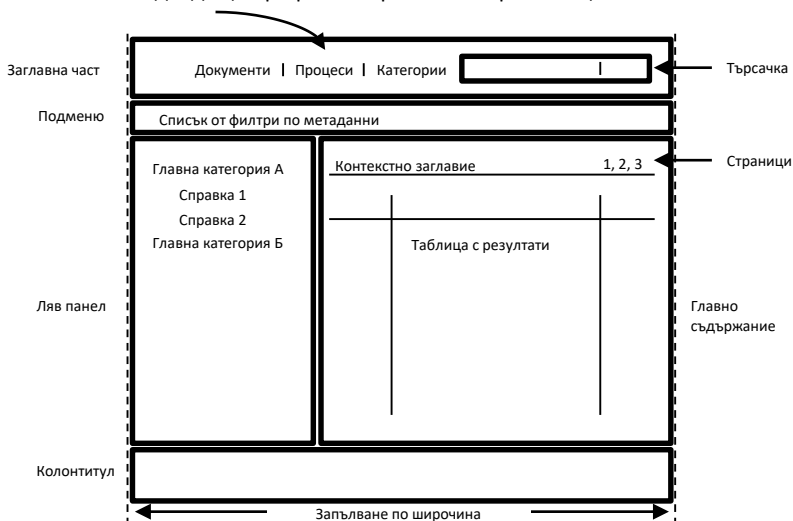
BaseReport report = container.getReport("jasper report", "jasper
category");
ReportResult body = report.generate();
Metadata $metadata = report.getMetadata();

```

Фигура 55. Генериране на ИР и метаданни от Jasper справка

Реализация на презентационна част

Моделът на този слой (Фигура 56) използва способите за уеб програмиране, зададени като най-подходящи при реализирането на презентационната част.



Фигура 56. Рамка на уеб презентационна част на СИРИП

Уеб сайтът се разделя на 5 основни части: заглавна част, подменю, ляв панел, главно съдържание и колонтитул.

4.3. Експерименти в ПеУ

Проведени са експерименти с реални студентски данни в ПеУ. Получените резултати са представени и анализирани.

Експеримент 1. Автоматично генериране на справки и сходности между ресурси. При първоначални тестове на ЦХ, се генерират справки от тип „Заявление“ и „Ведомости“ (ФС-31 от номенклатурата) при кандидатстване за стипендии към Пловдивския Университет.

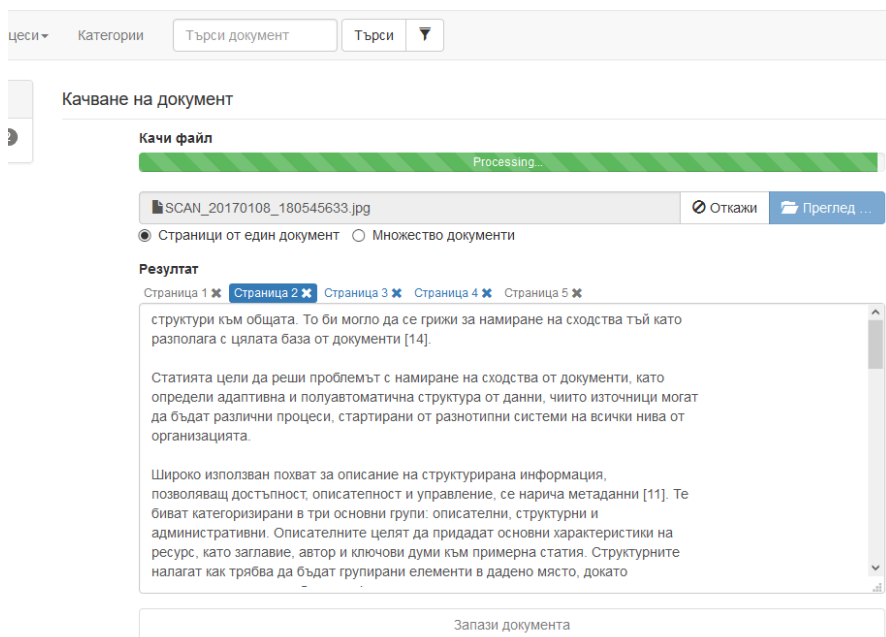
The screenshot shows a web application interface for a digital storage system. At the top, there is a navigation bar with tabs: "Хранилище", "Справки", "Процеси", and "Номенклатура". Below this, there is a search bar labeled "Търси справка" and a dropdown menu for "Търси". A sidebar on the left contains a list of categories: "Общ успех", "Доход", "Вид документ", "До дата", "Роднина", "От дата", "Дата на създаване", "Дата на редактиране", "Факултет №", "Първо име", "ИД", "Фамилия", and "Бащино име". The main content area displays a table of applications. The table has columns for "Дата", "Заглавие", and "Действия". The data rows show applications from 12-09-2016 20:12. The actions column contains buttons for "Изтегли", "Сходни", and "Детайли".

Дата	Заглавие	Действия
12-09-2016 20:12	Заявление (db05e4ca16f1bcea58871f6d315e42a46ab84613)	Изтегли Сходни Детайли
12-09-2016 20:12	Заявление (a0547b6b67c5211e4fdcec164535c7f5cc9489d4)	Изтегли Сходни Детайли
12-09-2016 20:12	Заявление (bcea7d662098d999bb1a1899382c7110f708de6a)	Изтегли Сходни Детайли
12-09-2016 20:12	Заявление (10653f53172903aa33927e21e8a5a28f64dddca)	Изтегли Сходни Детайли

Фигура 61. Цифрово хранилище

Когато студент подаде заявление за кандидатстване, бизнес процес предава информация за кандидата като набор от аргументи при задействането на ЦХ, който от своя страна зарежда предефиниран процес.

Експеримент 2. Депозирание на статични документи и разглеждане чрез услуги и интегриран изглед. Този експеримент тества разчитането на текст от изображения и извличането на съдържание от документи (pdf, doc) като запазва тяхното съдържание в базата данни за последващо търсено. Документи от този тип се генерират предимно при архивиране на научни разработки или ресурси.



Фигура 64. Ръчно прикачване на файлове с разпознаване на текст

Експеримент 3. Търсене на статични и динамични данни по съдържание и метаданни чрез услуги и интегриран изглед. След като в базата са съхранени над 530 динамични и статични документа, извлечената структура на метаданните е агрегирана с уникални стойности. Това позволява търсене не само в цялото тяло на документа, но и филтриране на определени елементи. Функционалността се гарантира от една страна с помощта на възможностите на Elasticsearch, а от друга – благодарение на модела за метаданните, осигуряващ унифициран списък от метаданни.

Заклучение

В рамките на дисертационното изследване **са решени основните задачи** и е **постигната основната цел** да се предложат, изследват и апробират стандарти методи и средства, подходящи за автоматизирано генериране и архивиране на разнотипни информационни ресурси.

Приноси на дисертационния труд

Основните приноси на дисертационния труд могат да се характеризират като научни, научно-приложни и приложни.

Научни приноси на дисертационното изследване са:

H1. Създадени модели и шаблони за динамично генериране на ресурси чрез автоматизирани процеси и структури от метаданни;

Н2. Дефинирани са подходи за интеграция на ИР и динамична система за класифициране на метаданните.

Научно-приложни приноси на дисертационното изследване са:

НП1. Създадена е архитектура на СИРИР, подходяща за обмен на данни между РИС;

НП2. Реализиран е прототип на софтуерна система за генериране, запазване и търсене на ИР.

Приложни приноси на дисертационното изследване са:

П1. Създадени и тествани са компютърни модели на общите модели от Н1;

П2. Проведени са конкретни експерименти по създаване на ЦХ в рамките на ПеУ за генериране на ИР.

Апробация

Резултати на дисертационното изследване са представени в 7 (седем) публикации - 3 (три) раздела в 1 (една) книга (Раздел II.1., Раздел II.2., Раздел II.3. в [Тотков, 2016]) и 4 (четири) в трудове на конференции.

- Резултати на дисертационното изследване са докладвани на 3 (три) конференции:
- V национална научна конференция 2013 за студенти, докторанти и млади учени;
- VI Международна конференция на младите учени – СУБ Пловдив 2015;
- Първа варненска конференция за електронно обучение и управление на знанието: Мост между средното и висшето образование.

Резултати на изследването са апробирани в 2 (два) европейски проекта:

- BG051PO001-4.3.04-0064 „Пловдивски електронен университет: национален еталон за провеждане на качествено е-обучение“, финансиран от ЕСФ;
- BG051PO001-3.1.08-0041 „Стандартизиране и интегриране на разнотипни информационни и управленски системи (СИРИУС)“.

Перспективи за развитие на дисертационната тематика

Получените резултати дават основа за следващо развитие на някои перспективни направления в областта на дисертационното изследване, като:

- Създаване на модул за дигитализация и архивиране на отчети и доказателствени документи, продуцирани от автоматизираната система за управление на качеството СИДОК [Донева, 2014, Гафтанджиева, 2014, Тотков, 2016а] в цифровото хранилище на ПУ, и др.;
- Откриване на релации между документи и ресурси, генерирани от процеси на различни информационни системи, в следствие на повторяемост на стойности и близост на свойствата;
- Използване на метода за извличане на знания от данни (data mining) на като опция за генериране на метаданни от статични данни и откриване на невидими релации между справки.

Списък на публикации

1. Алендарова Е., **А. Трайков**, Г. Тотков, За интегрирането на разнотипни информационни системи и ресурси, Сборник на V национална научна конференция за

студенти, докторанти и млади учени, „Имеон“, Пловдив, 2013, ISSN: 1314 – 9547, 203 – 207.

2. **Трайков А.**, Г. Тотков. Активна директория за управление на потребителски роли при интегриране на разнотипни информационни системи, VI Международна конференция на младите учени, СУБ Пловдив, 2015, , ISSN 1311-9192.
3. Пашев Г., **А. Трайков**, Е. Алendarова, Г. Тотков, Интегриране на функционалности от разнотипни системи в Moodle, VI Международна конференция на младите учени – СУБ Пловдив 2015, ISSN 1311-9192, 161-164.
4. **Трайков А.**, Автоматизирано генериране на метаданни за университетски документи, Първа варненска конференция за електронно обучение и управление на знанието: Мост между средното и висшето образование, МУ Варна, 01 октомври 2016 г,
5. 3 глави в книга „Университетът, документите, хората“ (ред. М. Шнитер), Студио 18, Пловдив, 2016, ISBN 978-619-7249-09-5:
 - Тотков Г., **А. Трайков**, Р. Донева, Д. Десев, Раздел II.1. „Платформи за управление на университетски дигитални архиви: състояние, проблеми и решения“, 35 – 73;
 - **Трайков А.**, Г. Тотков, Г. Пашев, Раздел II.2. „Модел и архитектура на платформа за управление на университетско дигитално хранилище“, 74 – 107;
 - Алendarова Е., Г. Тотков, Г. Пашев, **А. Трайков**, Раздел II.3. „Документни електронни панели към университетски цифрови архиви“, 108 – 138.

Цитирания

Известни са 4 (четири) цитирания на публикации по дисертационната тема.

Алendarова Е., **А. Трайков**, Г. Тотков, За интегрирането на разнотипни информационни системи и ресурси, Сборник на V национална научна конференция за студенти, докторанти и млади учени, „Имеон“, Пловдив, 2013, ISSN: 1314 – 9547, 203 – 207

1. С. Гафтанджиева „Модел и система за динамично оценяване на качеството във висшето образование“, Дисертация за придобиване на ОНС „доктор“ на Пловдивския университет „П. Хилендарски“, Пловдив, 2016

Пашев Г., **А. Трайков**, Е. Алendarова, Г. Тотков, Интегриране на функционалности от разнотипни системи в Moodle, VI Международна конференция на младите учени – СУБ Пловдив 2015, ISSN 1311-9192, 161-164

2. С. Гафтанджиева „Модел и система за динамично оценяване на качеството във висшето образование“, Дисертация за придобиване на ОНС „доктор“ на Пловдивския университет „П. Хилендарски“, Пловдив, 2016

Тотков Г., **А. Трайков**, Р. Донева, Д. Десев, Раздел II.1. „Платформи за управление на университетски дигитални архиви: състояние, проблеми и решения“, 35 – 73

3. Алendarова Е., Моделиране на потребителски е-панели за управление на информационни системи, Дисертационен труд за присъждане на образователна и научна степен "доктор"

Алendarова Е., Г. Тотков, Г. Пашев, **А. Трайков**, Раздел II.3. „Документни електронни панели към университетски цифрови архиви“, 108 – 138

4. С. Гафтанджиева „Модел и система за динамично оценяване на качеството във висшето образование“, Дисертация за придобиване на ОНС „доктор“ на Пловдивския университет „П. Хилендарски“, Пловдив, 2016

Библиография

[ББИА] Българска библиотечно-информационна асоциация, www.lib.bg

[БИС] Български институт за стандартизация, www.bds-bg.org

[Гафтанджиева, 2014] Гафтанджиева С., Р. Донева, Качество на висшето образование – осигуряване, оценяване, автоматизация, Научни трудове на Съюза на учените в България – Пловдив, Серия В. Техника и технологии, том. XII, ISSN 1311-9419, Пловдив, 2014, стр. 230 – 233.

[Донева, 2014] Гафтанджиева С., Р. Донева, Г. Тотков, Система от университетски стандарти за описание и архивиране на електронни курсове и дигитални ресурси, Пета национална конференция по Електронно обучение във висшите училища, 16 – 17 Май 2014 г., Русе, 2014, ISBN: 978-954-23-0747-1, стр. 275–280.

[Пашев, 2015] Пашев Г., Е. Алendarова, Г. Тотков, Проверяване на знанията и автоматично оценяване чрез интегриране на разнотипни системи с работни процеси, Сборник на 8-ма Нац. конференция „Образованието и изследванията в информационното общество“ (ред. Г. Тотков и Ив. Койчев), 28 май – 29 май 2015 г., Асоциация „Развитие на информационното общество“, Пловдив, ISSN 1314-0752, 119 – 128.

[ПЕУ] Пловдивски Електронен Университет, <http://pdu.uni-plovdiv.bg>

[Тотков, 2014] А. Г. Тотков и др. „Пловдивски е-университет“, „Ракурси“, Пловдив, 2014, ISBN 978-954-8852-48-7.

[Тотков, 2014а] Б. Г. Тотков и др. „Стандартизиране и интегриране на разнотипни информационни университетски системи“, „Ракурси“, Пловдив, 2014, ISBN 978-954-8852-49-4.

[Тотков, 2016а] Тотков Г., С. Гафтанджиева, Р. Донева, Модел за динамично управление и оценяване на качеството във висшето образование (с приложения в е-обучението), Първа Варненска конференция по електронно обучение и управление на знанието: Мост между средното и висшето образование, Варна, 30.09-1.10.2016 (в печат).

[Трайков, 2016] Трайков А., Г. Тотков, Г. Пашев, Модел и архитектура на платформа за управление на университетско цифрово хранилище, в „Университетът, документите, хората. Моменти от книжовната, дигиталната и неписаната история на ПУ“, Студио 18, 2016.

[Bargmeyer, 2002] Bruce Bargmeyer, Daniel Gillman, Metadata Standards and Metadata Registries: An Overview, Bureau of Labor Statistics, Washington DC, 2002

[BIBFRAME] Bibliographic Framework, <https://www.loc.gov/bibframe/>

[Bieberstein, 2008] Norbert Bieberstein, Robert Laird, Keith Jones, Tilak Mitra Executing SOA: A Practical Guide for the Service-Oriented Architect, IBM Press, ASIN B00891948Q, 2008

[Chirila, 2016] Ciprian-Bogdan Chirila, Pierre Crescenzo, Philippe Lahire, A Reverse Inheritance Relationship for Improving Reusability and Evolution, University of Nice-Sophia Antipolis, 15 Apr 2016

- [DC] Dublin Core, <http://dublincore.org/>
- [DSpace] DSpace Open Source Repository, <http://www.dspace.org>.
- [DSpaceMIT] DSpace@MIT Repository, <http://dspace.mit.edu>.
- [DTD] Document Type Definition, <http://www.w3.org/TR/html401/sgml/dtd.html>
- [ELASTIC] ElasticSearch, NoSQL аналитична база данни, <https://www.elastic.co/products/elasticsearch>
- [EQUELLA] EQUELLA Digital Repository, <http://www.equella.com>.
- [Erl, 2013] Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl), Prentice Hall, ISBN-10 0133387526, 2013
- [Fedora] Fedora Repository, <http://fedora-commons.org>.
- [Gamma, 1994] Ерик Г., Р. Хелм, Р. Джонсън, Д. Влсидес, Шаблини за дизайн. Елементи на обектно-ориентирания софтуер за многократно използване, Софтпрес ООД, 2006, ISBN 954-685-352-6
- [Gillenwater, 2010] Zoe Mickley Gillenwater, Stunning CSS3: A project-based guide to the latest in CSS, New Riders, ISBN-13 978-0321722133, 2010
- [Heery, 2000] Rachel Heery, Robina Clayphan, Metadata Application Profiles, International Conference on Dublin Core and Metadata Applications, 2000
- [IEC] International Electrotechnical Commission, <http://www.iec.ch/>
- [IFLA] International Federation of Library Associations and Institutions, <http://www.ifla.org/>
- [ISO] International organization for standardization, <http://www.iso.org>, посл. посетен на 03.10.2016 г.
- [ISO_11179] ISO/IEC 11179, Information Technology -- Metadata registries <http://metadata-standards.org/11179/>
- [JSON_XML] JSON vs XML: some hard numbers about verbosity, <http://pragmateek.com/json-vs-xml-some-hard-numbers-about-verbosity/>
- [Khan, 2015] Nadim Akhtar Khan, S M. Shafi, Sabiha Zehra Rizvi, Metadata Crosswalks as a Way Towards Interoperability, 10.4018/978-1-4666-5888-2.ch177, 2015
- [LARAVEL] Работна рамка на PHP Laravel, <https://laravel.com/>
- [LOC] Library of Congress. <https://www.loc.gov/>
- [Maheshwari, 2016] Anil Maheshwari, Michiel Smid, Introduction Theory of Computation, Carleton University, 11 Apr 2016
- [Manning, 1999] Christopher Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, ISBN 9780262133609, 1999
- [MARC] Machine-Readable Cataloging, <https://www.loc.gov/marc/umb/>
- [Morales, 2008] Josep Cobars Morales, Campus as a Framework for a Networked University, Universitat Oberta de Catalunya, DOI: 10.4018/978-1-59904-885-7.ch017, 2008
- [NISO, 2004] Understanding Metadata, National Information Standards Organization, ISBN 1-880124-62-9, 2004
- [PHP] PHP - Hypertext Processor, интерпретационен език за генериране на динамични уеб страници, <http://php.net/>
- [RDF] Resource Description Framework, <https://www.w3.org/RDF/>
- [Sporny, 2010] Web Services: JSON vs. XML, <http://new.digitalbazaar.com/2010/11/22/json-vs-xml/>

[TESSERACT] Tesseract OCR library, <https://github.com/tesseract-ocr/tesseract/wiki/Documentation>.

[Zwass, 2016] Vladimir Zwass, Information system, Encyclopedia Britannica, 2016

