



**ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „КОМПЮТЪРНА ИНФОРМАТИКА“**

Милен Пламенов Близнаков

**СИСТЕМА ЗА МОДЕЛИРАНЕ И ВИЗУАЛИЗАЦИЯ
НА ДОСТЪПНИ ДИНАМИЧНИ ИЗГЛЕДИ
НА ЦИФРОВИ ОБЕКТИ**

А В Т О Р Е Ф Е Р А Т

на дисертационен труд
за присъждане на образователната 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 Георги Петров Димитров
доц. д.н. Красимир Янков Йорджев

Пловдив
2022

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 17.03.2022 г.

Дисертационният труд „Система за моделиране и визуализация на достъпни динамични изгледи на цифрови обекти“ съдържа 158 страници. Списъкът на използваната литература съдържа 191 (сто деветдесет и едно) заглавия, от които: 118 (сто и осемнадесет) публикации и книги и 73 (седемдесет и три) интернет-източника; 18 (осемнадесет) на кирилица, 173 (сто седемдесет и три) на латиница. Списъкът на авторските публикации се състои от 6 заглавия.

Защитата на дисертационния труд ще се състои на 27.05.2022 г. от 11:30 ч. в Заседателната зала на ФМИ (Нова сграда, бул. „България“ №236).

Материалите по защитата са на разположение на интересувашите се в библиотеката 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: **Милен Пламенов Близнаков**

Заглавие: **Система за моделиране и визуализация на достъпни динамични изгледи на цифрови обекти**

Modeling and visualization system of accessible dynamic views of digital objects

Milen Plamenov Bliznakov

Visualization of information in an appropriate way is the process associated with humanity since ancient times. In the beginning, people depicted and preserved objects and phenomena in drawings and pictograms. At a later stage, in order to present human abstract thought, the representation grows into logograms, hieroglyphs and ideograms, which over the centuries have evolved into complex formal languages and alphabets. Nowadays, the presentation of ideas and phenomena is closely related to the information age in which we live. Visualization is through images, text, diagrams and complex interactive connections between objects. This creates a number of opportunities to get started, the information resources can be visualized on different devices and users.

The possibilities of dynamic views have not yet been fully explored and used. The dissertation examines various concepts and meanings that are invested in terms of digital object, visualization, dynamic view and examines the process of presenting information sources and structure in an accessible way. You can also look at different stages and aspects of visualization and conceptual models for analyzing and modeling abstract data. Data visualization and post-processing need interpretation tools, which means that it is necessary to specifically find the problem of presenting data and resources appropriate from different information sources in a way. The problem is related to the automatic collection, analysis and interpretation of a huge amount of data for various aspects of data visualization in appropriate views.

The **main goal** of the dissertation research is to propose, research and test software system and tools (methods and tools) for modeling and visualization of data and information resources in accessible dynamic views.

The fulfillment of the set goal implies solving the following questions:

Question 1. Study of the general theory, existing organizations, models, standards and systems for visualization of information resources and CA;

Question 2. Creating a conceptual model and methodologies for data analysis, modeling and visualization of accessible dynamic views;

Question 3. Introduction and creation of software prototype (s) for visualization of accessible dynamic views;

Question 4. Testing of software prototype(s) for accessible views and tools for dynamic visualization of information resources and objects, following specific methodologies.

СЪДЪРЖАНИЕ

СПИСЪК НА ИЗПОЛЗВАНИТЕ СЪКРАЩЕНИЯ	5
УВОД.....	6
ГЛАВА 1. МОДЕЛИРАНЕ И ВИЗУАЛИЗАЦИЯ НА ЦИФРОВИ ОБЕКТИ.....	8
1.1. Основни понятия и определения	8
1.2. Стандарти и методи за моделиране и визуализация	11
1.3. Класификация на системи за моделиране и визуализация.....	13
1.4. Изводи	14
ГЛАВА 2. МОДЕЛ И МЕТОДИКИ НА МОДЕЛИРАНЕ И ВИЗУАЛИЗАЦИЯ НА ДОСТЪПНИ ДИНАМИЧНИ ИЗГЛЕДИ	14
2.1. Общ модел на процес за визуализация на изглед	14
2.2. Модел на процес за моделиране и визуализация на достъпни динамични изгледи.....	15
2.3. Методика за създаване на концептуална рамка	17
2.4. Архитектура на рамка на системата	19
2.5. Схема на взаимодействие на елементите в системата	20
2.6. Изводи	22
ГЛАВА 3. СОФТУЕРНИ ПРОТОТИПИ ЗА МОДЕЛИРАНЕ И ВИЗУАЛИЗАЦИЯ НА ДОСТЪПНИ ДИНАМИЧНИ ИЗГЛЕДИ	22
3.1. Динамична презентационна рамка	23
3.2. Прототип без участие на потребител при обработката и моделирането на заявен резултат (C1)	25
3.3. Прототип с участие на потребител при обработката и моделирането на заявен резултат (C2)	28
3.4. Заключение за реализираните софтуерни прототипи	30
ГЛАВА 4. АПРОБАЦИЯ И ТЕСТОВЕ НА ПОЛУЧЕНИТЕ РЕЗУЛТАТИ	31
4.1. Приложение за оперативни и управленски решения	31
4.2. Анализ на уеб страници в прототипа C1 с Vue.....	32
4.3. Експерименти за C1 с Vue	32
4.4. Визуализация на изгледи на различни по размер устройства	36
4.5. Експеримент с генератор на заявки.....	36
4.6. Визуално представяне на данни и диаграми с прототип C2.....	37
4.7. Изводи	37
ЗАКЛЮЧЕНИЕ	39
ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИОННИЯ ТРУД	41
БИБЛИОГРАФИЯ.....	42

Списък на използваните съкращения

Съкр.	Значение
АТРМ	– Анализ-Трансформация-Представяне-Интерпретация (модел)
БД	– База от данни
ВД	– Визуализация на данни
ВИ	– Визуализация на информация
ВО	– Висше образование
ДО	– Дигитален обект
ДДИ	– Достъпни динамични изгледи
ИБ	– Информационна визуализация
ПУ	– Пловдивски университет „Паисий Хилендарски“
СМВДДИ	– Система за моделиране и визуализация на ДДИ
ЦО	– Цифров обект
API	– Application Programming Interface
ARIA	– Accessible Rich Internet Applications
CSS	– Cascading Style Sheets
CSV	– Comma-separated values (file format)
DOCX	– Document Word open XML format
DOM	– Document Object Model
DPF	– Dynamic Presentation Framework
DTD	– Document Type Definition
GIF	– Graphics Interchange Format
HTML	– Hypertext Markup Language
HTTP	– Hypertext Transfer Protocol
JPEG	– Joint Photographic Experts Group (digital image format)
JRXML	– Jasper Report XML
JSON	– JavaScript Object Notation
JSP	– Jackson Structured Programming
MVC	– Model-View-Controller
OMT	– Object Modeling Technique
OOSE	– Object-oriented software engineering
PDF	– Portable Document Format
REST	– Representational State Transfer
SOAP	– Simple Object Access Protocol
SVG	– Scalable Vector Graphics
SQL	– Structured Query Language
UML	– Unified Modeling Language
URL	– Uniform Resource Locator
WAI	– Web Accessibility Initiative
WCAG	– Web Content Accessibility Guidelines
W3C	– World Wide Web Consortium
XLS	– Excel Spreadsheet format
XML	– Extensible Markup Language

Увод

Визуализацията на информация (ВИ) по подходящ начин е процес, свързан с човечеството от дълбока древност. В началото хората са изобразявали и запазвали обекти и явления чрез рисунки и пиктограми. На по-късен етап, за да се представи човешката абстрактна мисъл, представянето прераства в логограми, йероглифи и идеограми, които през вековете еволюират до сложни формални езици и азбуки. В днешно време представянето на идеи и явления е тясно свързано с информационната ера в която живеем. Визуализацията е едновременно чрез изображения, текст, диаграми и сложни интерактивни връзки между обектите. Това поражда редица предизвикателства за начина, по който информационните ресурси могат да бъдат визуализирани на различни устройства и потребители.

Възможностите на динамичните изгледи все още не са напълно проучени и използвани. В дисертацията са разгледани различни понятия и смисъл, който се влага в термините **цифров обект, визуализация, динамичен изглед** и се изследва процеса на представяне на информационни източници и структури по **достъпен** начин. Също така са разгледани различни етапи и аспекти на визуализацията и концептуални модели за анализ и моделиране на абстрактни данни. Визуализацията на данни и последващата им обработка има нужда от инструменти за тяхната интерпретация, което означава, че е необходимо специално да се изследва проблема за представянето на данните и ресурсите от различни информационни източници по подходящ начин.

Цел и задачи

Основна цел на дисертационното изследване е да се предложат, изследват и апробират софтуерна система и средства (методи и инструменти) за моделиране и визуализация на данни и информационни ресурси в достъпни динамични изгледи.

Изпълнението на поставената цел предполага решаване на **следните задачи**:

Задача 1. Проучване на общата теория, съществуващи организации, модели, стандарти и системи за визуализация на информационни ресурси и ЦО;

Задача 2. Създаване на концептуален модел и методики за анализ на данни, моделиране и визуализация на достъпни динамични изгледи;

Задача 3. Въвеждане и създаване на софтуерен прототип(и) за визуализация на достъпни динамични изгледи;

Задача 4. Тестване на софтуерен прототип(и) за достъпни изгледи и инструменти за динамична визуализация на информационни ресурси и обекти, следвайки конкретни методики.

Структура на дисертацията

Дисертационният труд се състои от Списък на използваните съкращения, Списък на фигурите, Списък на таблиците, Увод, 4 (четири) глави (представящи решения и резултати на поставените задачи), Заключение, Приложение, Списък на авторските публикации по темата, Списък на забелязани цитирания, Списък на използваната литература и Декларация за оригиналност. Списъкът на използваната литература съдържа 191 (сто деветдесет и едно) заглавия, от които: 118 (сто и

осемнадесет) публикации и книги и 73 (седемдесет и три) интернет-източника; 18 (осемнадесет) на кирилица, 173 (сто седемдесет и три) на латиница.

В **Глава 1. Моделиране и визуализация на цифрови обекти** са разгледани и представени основни понятия и определения, свързани с проблема за дефиницията на цифров обект (ЦО), за моделиране и визуализация на достъпни динамични изгледи (МВДДИ). Представени са стандарти за достъпност, развивани и поддържани от The Web Accessibility Initiative (WAI) и популяризиращи като международен стандарт. Анализирани са редица модели и методи за моделиране на ЦО и ВД, както и софтуерни средства и инструменти за тяхното представяне. В заключителния раздел на Глава 1. е представена класификация на софтуерните инструменти за моделиране и ВД и на базата на анализ на състоянието на изследванията в областта, са формулирани основна цел и задачи на дисертационното изследване.

В **Глава 2. Модел и методики на моделиране и визуализация на достъпни динамични изгледи** е предложен общ модел на процес за визуализация на изглед и модел на процес за моделиране и визуализация на достъпни динамични изгледи (МВДДИ). Разгледани са основните принципи за създаване на достъпни интернет страници според WCAG и съответни софтуерни инструменти за тестване. Въведени и изследвани са съответни методики за създаване на изгледи на различни устройства в уеб среда и на концептуална рамка на система за МВДДИ. На базата на въведените модели е предложена архитектура на рамка на система за МВДДИ и са формулирани основни функционални и нефункционални изисквания към системата. Разгледани са четири подхода за реализация на архитектурния модел в зависимост от източниците на данни и информационните ресурси. Последният раздел (изводи) аргументира адекватността на общите модели и методики, предложени в качеството на решения на Задача 2.

В **Глава 3. Софтуерни прототипи за моделиране и визуализация на достъпни динамични изгледи** са представени архитектура на рамка и три прототипа за МВДДИ, базирани на участието на потребител при обработката и моделирането на резултат от различни информационни източници. Описани са приложните услуги и модулите на архитектурната рамка на системата и съответните компоненти, участващи при реализацията на софтуерните прототипи на системата. Представени са сценарии и съставни компоненти, библиотеки и интегрирани външни системи, съдействащи за реализацията на прототипите. За всеки от тях са направени основни изводи, анализирани от гледна точка на архитектурата на системата, методиките и модела за моделиране и визуализация на достъпни динамични изгледи.

В **Глава 4. Аprobация и тестове на получените резултати** са проведени експерименти с прототипите на софтуерната система, съгласно принципите и методиките описани в Глава 2. на дисертацията. Направен е анализ на уеб страници за избор на вариант за извличане и попълване на данни в изглед. Представен е инструмент за оперативни управленски решения, базиран на един от прототипите. Приведен е тест на SQL генератор на заявки с 2 релационни БД. Част от експериментите са направени с онлайн валидатори за проверка на уеб страници според критериите за достъпност на WCAG и са изпълнени тестови симулации на изгледите за различни браузъри и различни по размер устройства. Описани са

основните резултати, постигнати с инструментите за тестване на достъпни страници и са представени част от изгледите, визуализирани като справки и диаграми.

В **Заклучението** са направени изводи от резултатите на дисертационното изследване, направено е обобщение на архитектурата и реализираните прототипи на системата. Отбелязани са получените резултати, основните научни, научно-приложни и приложни приноси на дисертацията и са формулирани възможностите за бъдещо развитие на тематиката.

Благодарности

Изказвам своите най-искрени благодарности на моя научен ръководител – проф. д.м.н. Георги Тотков за предоставената ми възможност и доверие за провеждане на дисертационно изследване в областта. 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 МУ19-ФТФ-001 „Интелигентен анализ на данни за подобряване на резултатите от обучението“ (2019 – 2020), МУ21-ФТФ-018 „Приложение на методи за анализ на големи данни във висшите училища“ (2021 – 2022).

Глава 1. Моделиране и визуализация на цифрови обекти

1.1. Основни понятия и определения

Разбирането за природата на **понятията и обектите** има две фази [Quine'60]: първо трябва да се разбере какви са обектите и второ трябва да се разбере какво казва теорията за тях. **Понятието** е мисъл за обобщение на обекти и процеси от действителността [Тотков'01]. **Обект** е общо понятие, което често се използва за определяне чрез термина субект. Според [Peirce'09] обект е всичко, за което можем да мислим и всичко за което можем да говорим.

По отношение на термина **цифров обект или дигитален обект (ЦО или ДО)** редица изследователи предлагат обобщени определения. Според [Kahn'06] дигитален обект е екземпляр на абстрактен тип данни, който има два компонента: данни и ключови метаданни. Подобно е и определението на [Hui'12], че цифровите обекти са просто обекти в мрежата, като видеоклипове в YouTube, Facebook профили, изображения на Flickr и т.н., които са съставени от данни и формализирани чрез схеми или онтологии, обобщени като метаданни. Редица автори дават определение на цифров обект чрез неговите характеристики като: [Kallinikos'10], че ЦО са белязани от ограничен набор от променливи с обобщени атрибути като редактируемост, интерактивност, откритост и възможност да бъдат лесно копирани и разпространявани; или [Saidis'07], че ЦО могат да бъдат представени като сложни артефакти, които обгръщат дигиталния материал от гледна точка на четири елемента: неговото съдържание, метаданните, връзките му с други обекти и неговото поведение.

[Ware'13] твърди, че има различни подходи за **визуализацията**. Тя може да бъде изучавана в традицията на арт-училището като графичен дизайн. Също може да бъде изучавана в компютърната графика като област, за която става дума за алгоритмите, необходими за показване на данни. Може да се изучава като част от семиотиката. Но научният подход, основан на възприятието, еднозначно задава правила за проектиране, които надхвърлят капризите на дизайна, като се основават на относително стабилната структура на зрителната система на човека.

Визуализация на данни (ВД) [Friendly'09] се разглежда от много изследователи като съвременен еквивалент на визуалната комуникация. Тя включва създаването и изучаването на визуалното представяне на данните. Визуализация на информация (ВИ) е по-широк термин от ВД, който обикновено се прилага към визуалното представяне на широкомащабни колекции от не-цифрова информация, като например файлове и линии на код в софтуерните системи. В по-общ смисъл ВИ е препратка към рисунките на форми по скалите, развитието на пиктографията и до най-ранното използване на диаграми в историята на науката и математиката. ВД е потясна област за визуалното представяне на „данните“, дефинирани като информация, която е била в някаква схематична форма, включително атрибути или променливи.

Визуализация на информация (ВИ) или информационна визуализация (ИВ) е предаването на абстрактни данни чрез използване на интерактивни визуални интерфейси [Keim'06]. ВИ е набор от технологии, които използват визуални изчисления за разширяване на човешкото познание с абстрактна информация. ВИ помага да се ускори разбирането на механизмите в един свят на нарастващ обем информация. Целта на информационната визуализация е не само да се създадат интересни снимки, а да се разширят познавателните резултати, като се опитват да направят за съзнанието това, което правят автомобилите за краката на хората [Card'08].

Според [McCandless'10]: „Като визуализираме информацията, ние я превръщаме в пейзаж, който може да се изследва с очите, нещо като информационна карта.“ Също така застъпва твърдението, че е много по-лесно човек да се ориентира в подобно представяне на информацията.

Според [Abdelzad'15] **модел** е абстракция на реалността, която помага на потребителите да се фокусират върху специфични аспекти на даден проблем или система, за да се подкрепи комуникацията, разбирането, анализа и вземането на решения. Езиците за моделиране често имат графично и/или текстово представяне, наречено конкретен синтаксис. Понятията на език за моделиране често са описани с абстрактен синтаксис, например под формата на граматика или мета-модел. Конкретните синтаксиси носят разбираемост, използваемост и често създават динамична визуализация на понятията, дефинирани на абстрактно ниво.

Моделиране е метод за получаване на информация за конкретен обект, наричан оригинал чрез: формулиране на цел; анализ на обекта; построяване на модел-аналог; изследване на свойства и отношения; сравняване на моделираните свойства; пренасяне на откритите свойства и отношения върху обект-оригинал [Тотков'01].

Компютърно моделиране е процес, който реализира представяне на обект, система, концепция във форма, различна от реалната, но близка до алгоритмичното

описание. Включва набор от данни, характеризиращи свойствата на системата и динамиката на тяхната промяна във времето [Ножнов'09].

В тълковния речник за понятието **изглед** има следните значения – вид, който се открива отнякъде; гледка, снимка; вероятност, перспектива. Значение на думата **динамичен** най-общо е: който се основава на законите на динамиката; който е изпълнен с движение и активност. В дисертационното изследването понятието има значение също като синонимите: подвижен, вариращ, изменяем.

Динамичен изглед е възможността цифров обект или група от ЦО с еднаква логическа структура да бъде представена по различни начини в зависимост от потребителя, презентационно устройство, време или събитие.

В епохалния труд „Държавата“ Платон създава доктрина за идеалните форми, която твърди, че в абстрактните форми или идеи се съдържат най-основната степен на реалността [Платон'81]. Според тази теория явленията са моментни проявления, които са подобни на Абсолютната форма, а материята е само сянка на света на идеите. В контекста на Платоновата теория за формите, динамичния изглед може да се разглежда в идеалния случай като материално изображение и копие на очакваната визуализация на идеите на потребителя.

[Burger'13] дефинира понятието **гъвкав изглед** (flexible view) като подход за моделиране при разработване на софтуер и като концепция за създаване на персонализирани, специфични за потребителя изгледи. Дава определение за гъвкавите изгледи като частични изгледи на една софтуерна система, които: 1. Могат да се дефинират по време на изпълнение на рамката за разработка; 2. Са дефинирани като екземпляри за моделиране; 3. Може да въведат нови елементи към основния модел; 4. Поддържат работен процес за онлайн моделиране; 5. Съдържат правила, описващи: а) изборът на елементи, които се показват (на базата на свойствата на екземпляра), б) кои операции за редактиране са възможни, в) как операциите по редактиране се отразяват в основния модел.

Разгледано като общо понятие **достъпността** е степента на възможността на изгледа да бъде лесно достижим и разбран от повече на брой потребители. Възможно е понятието погрешно да се смесва с нефункционалното изискване „използваемост“, което е със значение колко лесно/трудно може да се ползва една система.

Според Web Accessibility Initiative [WAI'18] смисъла, който се влага в понятието (accessible) е възможността да се направи уеб-мрежата по-достъпна за потребители с различни видове увреждания. Инициативата е част от The World Wide Web Consortium (W3C) и предлага, поддържа и развива различни методи, стратегии и стандарти [WCAG'18] за достъпност (accessibility).

Достъпността и динамичността на изгледа е тясно свързан със семиотичния характер и природа на представянето. В когнитивната наука съществува широк смисъл, обхващащ човешката познавателна способност, при която е от съществена важност системното създаване и използване на системи от символи, всеки от които свързва даден знак с определено значение, установено чрез обществени конвенции [Nöth'95]. В началото на 20 век Чарлс Пиърс нарича този процес семиоза, а науката, която го изследва – семиотика [Saussure'83].

Семиотичния изглед е дискуссионен термин и редица автори изследват [Bertin'83, Nadin'90] и критикуват [Ware'13] семиотичния подход при визуалното представяне на различни информационни ресурси.

1.2. Стандарти и методи за моделиране и визуализация

Кръгове на Ойлер [Euler'68] и диаграми на Вен [Venn'80] са начини за моделиране и графично представяне на множества и техните отношения. Те се изобразяват в равнина със затворени контури, като се представят като елипси, кръгове, понякога – квадрати или по-сложни фигури. Кръговете на Ойлер са общ случай за визуално моделиране на множествата със следните взаимоотношения помежду им: пълно, частично, никакво. Диаграмите на Вен са подобни на ойлеровите, но показват всички възможни връзки между различните множества: сечение, обединение, разлика, симетрична разлика.

В края на 60-те и началото на 70-те години на 20 век [Ross'77] разработва и тества метод, наречен „Структуриран анализ и техника за дизайн“ (SADT – Structured analysis and design technique). SADT е визуално представяне на информацията чрез диаграми, предназначени да помогнат на разработчици и експерти да опишат и разберат системите [Marca'87].

По-късно се създава метод за структурирано програмиране (Jackson structured programming – JSP), основаващо се на принципа, че структурата на програмата трябва да бъде диктувана от структурата на входящите и изходящите потоци от данни [Jackson'75].

[Stevens'74, Myers'76] също изследват и развиват идеите за структурирано-модулен дизайн и задават принципите за свързаност и сплотеност (coupling and cohesion) като подход към модулния софтуер. Развиват и допълват графичното представяне на „потока“ на данни чрез информационна система – диаграма на потока от данни (data flow diagram – DFD).

[Warnier'75] създава техника, наречена по-късно диаграми на Warnier/Orr, които помагат при проектирането на програмни структури, чрез идентифициране на изходните резултати. След обработката им методът се връща назад, за да определи стъпките и комбинациите от входящи данни, необходими за създаването им. Простият графичен метод, използван в диаграмите, прави видими нивата в системата и движението на данните между тях.

[Chen'76] дава своя принос в изследванията със създаването и развитието на обектно-релационен модел (Entity Relationship Model, ER-модел), който описва взаимосвързани неща от специфична област на знанието. Моделът се състои от типове обекти и определя взаимоотношения, които могат да съществуват между единиците, екземпляри от тези типове обекти. Явява се абстрактен модел на данни, който дефинира структура от данни или информация, която може да бъде внедрена в база данни, обикновено релационна база данни.

[Zloof'77] създава език за управление на база данни от високо ниво, който осигурява унифициран стил за заявки, дефиниция и контрол на релационна база данни. Инструментът предоставя техники, за които са необходими минимални знания от потребителя, за да работи и да сведе до минимум броя на концепциите от релационния модел. Анализаторът на системата има възможност да преобразува

действията на потребителя в изрази, изразени в SQL, като визуалния синтаксис обхваща голямо разнообразие от сложни заявки.

[Yourdon'79] разработва структурирани техники и участва като съразработчик в методи за обектно-ориентиран анализ и дизайн. [DeMarko'79] също създава структуриран метод, който се състои от 4 обекти: Контекстна диаграма (Context diagram), Диаграма на потока от данни (Data flow diagram), Спецификации на процеса (Process specifications), Речник на данните (Data dictionary).

OMT, OOSE и Бууч методите се явяват основа за създаването на **унифицираният език за моделиране** (Unified Modeling Language – UML). UML е език с общо предназначение за моделиране и развойна дейност в областта на софтуерното инженерство, който има за цел да осигури стандартен начин за визуализиране на дизайна на системата [Booch'05]. Неговата изразителност позволява на потребителите да моделират: корпоративни информационни системи, разпределени уеб-базирани приложения, вградени системи в реално време, концептуални неща като бизнес процеси и системни функции, изречения за програмни езици, схеми на база данни и софтуерни компоненти за многократна употреба.

Статичния модел на **иконите** се явява разпространен начин за представяне на информация [Пиктограмм'19]. [Horton'94] представя иконите като заместител на думите, но не по-добри от думите; разглежда механизма на възприемане на иконите от потребителя; дава обосновка за използването на икони като поставя дискусия за еволюцията на иконата от пещерното рисуване, чрез идеограма и народното изкуство, до пътните знаци; класифицира начините на визуално представяне на идеите; показва начини за формулиране на граматически правила за визуалните езици и споделя опит в организирането им и процеса на развитие.

[Ward'02] изследва иконите с по-обобщеното понятие **глифи** и ги определя като графични обекти, които предават една или повече стойности на данните чрез атрибути като форма, размер, цвят и позиция. Според него глифите са широко използвани при визуализацията на данни и информация и са особено подходящи за показване на сложни, многовариантни набори от данни. Поставянето или разположението на глифите на дисплея може да предаде значителна информация относно самите стойности на данните, както и връзките между точките за данни.

Стандартите за достъпност, в общия случай, се постигат чрез изпълнение на съответните официални директиви на World Wide Web Consortium (W3C), респективно The Web Accessibility Initiative [WAI'18]. WAI развива и поддържа стратегии, ръководни принципи и ресурси, и ги популяризира като международен стандарт за осигуряване на достъпност до Интернет. Той има няколко работни групи и групи по интереси, които работят върху насоки, технически доклади, образователни материали и документи, които се отнасят до различни компоненти на уеб достъпността. Тези компоненти включват уеб съдържание, уеб браузъри и медийни плейъри, инструменти за авторство и инструменти за оценка.

Инициативата предлага и развива общо 18 принципа [Енков'13]: 14 във версия WCAG 1.0 и 4 принципа в WCAG 2.0. В стандарта са предложени 3 приоритетности (степени на достъпност) като покритието им осигурява определено ниво или съответствие на достъпност (A, AA, AAA). От 2018 г. се развива версия WCAG 2.1, като добавя нови критерии (за мобилна достъпност, за хора със слабо зрение, за хора с

когнитивни и обучителни затруднения), дефиниции, които подкрепят критериите, насоки за организиране на допълненията и две помощни бележки към раздела за съответствие [WCAG'18]

1.3. Класификация на системи за моделиране и визуализация

В таблицата по-долу се оценяват софтуерни системи по 6 критерия, за които се предполага, че са основни свойства, отразяващи достъпността и динамичността на представените резултати на системите за моделиране и визуализация.

Система	ВАД	ВМ	ИБ	ВП	ДП	ВЕД
Astah	0	3	2	2	3	1
ATL	0	3	2	0	2	1
ChartStudio	3	1	2	2	2	3
Crystal Reports	1	3	3	1	3	2
D3	2	0	2	0	3	1
DataMelt	3	1	2	1	3	1
DataHero	3	1	2	3	3	2
Dia	0	3	2	1	2	2
EMSG	1	2	3	2	2	2
FastReport Delphi	3	2	3	1	3	3
JasperSoft	1	3	3	2	2	3
KNIME	2	3	3	1	3	2
MATLAB	3	3	2	1	2	3
Modelio	0	3	2	2	3	1
Oracle Reports	2	3	3	1	2	3
PragmaDev Studio	0	3	2	0	3	1
Qlik	3	1	3	1-3	3	2
R	2	0	2	1	2	1
SQL Server Reporting Services	1	3	3	2	3	3
Tableau	3	1	3	2	3	2
VAX	0	2	2	1	2	1
ZebraBI	2	1	3	2	2	1

Таблица 1. Класификация на системи за моделиране и визуализация на данни [Близнаков'19].

Легенда:

- 0 – най-ниско ниво или не притежава функционалността; 1 – ниско ниво; 2 – средно ниво; 3 – най-високо ниво.
- ВАД – възможност за анализ на данни (0 – 3);
- ВМ – възможност за моделиране (с доп. познания – 0, без познания – 3);
- ИБ – интерактивни инструменти за визуализации: заложили в продукта интегрирани функционалности за манипулация на информационните ресурси (0 – 3);
- ВП – вид потребител (за разработчици – 0, без техн. умения – 3);

- ДП – динамично представяне: различни начини и техники за представяне на информацията – диаграми, графики и др. (0 – 3);
- ВЕД – възможност за експорт на данни във формати (0 – 3).

Разгледаните стандарти, модели и методики показват възможностите за моделиране и визуализация на динамични изгледи. Стандартите за достъпност на The Web Accessibility Initiative предоставят подход и директиви за реализиране на достъпни изгледи, но независимо от това описаните системи не се придържат към методиката за реализиране на достъпни изгледи по стандартите на WAI. Често срещан недостатък на разгледаните системите е, че не обръщат сериозно внимание на създаването на софтуерни техники и инструменти в уеб базирана среда.

1.4. Изводи

Визуализацията на данни и изгледи е понятие с изключително широко значение и се използва в значителен брой области на човешкото познание. Възможността за представяне на информацията като достъпни изгледи може да бъде разглеждана основно в две направления – според смисъла на понятието за достъпен и според принципите на достъпност на WCAG. На базата на проведен анализ в Глави 1 и 2 в дисертационния труд е избран да се използва уеб базиран подход за визуализация на изгледа, съчетан с няколко сценария и подхода за извличане и обработка на ресурсите от различни източници на данни.

Глава 2. Модел и методики на моделиране и визуализация на достъпни динамични изгледи

В настоящата глава е представен модел за визуализация на достъпни динамични изгледи. Описани са модели на процеси за визуализация на изгледи, методики и рамка за моделиране и визуализация на ДДИ. Представени са характеристики на главните модули и са описани основните функционални и нефункционални изисквания за софтуерна система за МВДДИ.

2.1. **Общ модел на процес за визуализация на изглед**

Изводите, които следват предполагат следните 4 етапа за общ модел на визуализация на изглед на ЦО. По-долу са представени етапите на модела и е направено резюме за всеки от тях, заедно с представени подходи за осъществяването им.

1. Анализ на данните (**Analysis: A**) – изследване на логическата структура на информационните ресурси и ЦО, които трябва да се визуализират;
2. Трансформация или Моделиране (**Transformation: T**) – възможността за промяна или обновяване на ЦО и данните, за да могат да бъдат визуализирани по подходящ начин; структурни промени като съединяване, разделяне, изтриване и др.;
3. Представяне или Визуализация (**Presentation: P**) – конкретното изобразяване (изглед) на данните според потребителя и презентационната форма;

4. Значение или Интерпретация (Meaning or Interpretation: M) – значението или интерпретираното тълкуване, което се влага във визуализирания изглед; Интерпретацията се явява функция на изгледа.

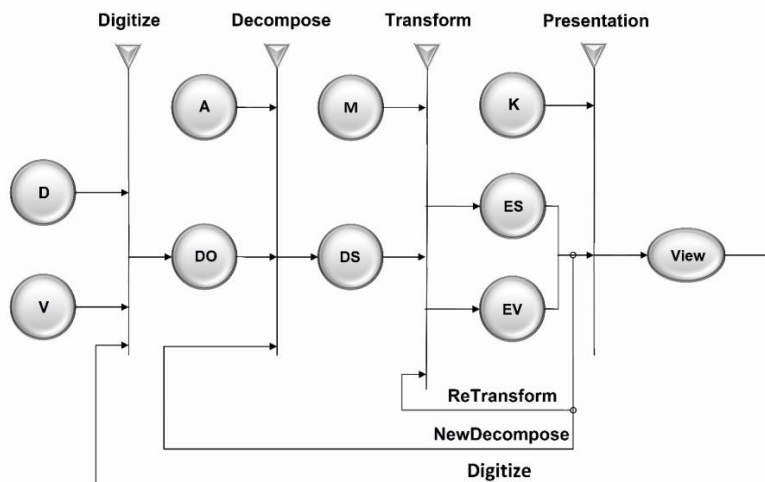
Основния път, който следва процеса в модела е: Анализ (A) --> Трансформация (T) --> Представяне (P) --> Интерпретация (M). Но в общия случай етапите от модела АТРМ могат да не са в строга линейна последователност. Например анализа (A) може да бъде едва след като има известно първоначално представяне (P) на данните – първичен изглед. Също така визуализацията (P) може да бъде преди етапа на трансформация (T) на ЦО, след което да следва ново представяне (P). Единствено интерпретацията (M) е едва след като има вече визуализиран (P) изглед. Движението по етапите в модела може отново да продължи след M в нов анализ или трансформация.

2.2. Модел на процес за моделиране и визуализация на достъпни динамични изгледи

На базата на анализа, проведен в глава 1 и описанието на общ модел за визуализация на изглед (АТРМ) може да се направят изводи за това кои са елементите и процесите, свързани с модел на процес за моделиране и визуализация на достъпни динамични изгледи. Тези елементи, заедно с техните означения са следните:

- *D* – информационни ресурси и данни;
- *V* – съхранени изгледи;
- *Digitize* – процес на дигитализация на данните в цифрови обекти;
- *DO* – цифрови обекти, върху които ще се извършва анализа;
- *A* – алгоритъм, който се прилага за анализа на цифровите обекти;
- *Decompose* – процеса на анализ на ЦО;
- *Transform* – процеса на моделиране и трансформация на данните в структурни и визуални компоненти;
- *DS* – данни, обработени при процеса на анализ *Decompose* и готови за процеса на трансформация *Transform*;
- *M* – модел или шаблон, който се прилага върху структурата *DS*;
- *ES* – данни и метаданни, съдържащи структурни компоненти, които са подмножество на изгледа;
- *EV* – визуални компоненти, които са подмножество на изгледа;
- *K* – критерий или метод за визуализация, който се прилага върху *ES* и *EV*;
- *View* – изглед; информационни ресурси и данни, резултат от обединение на *ES* и *EV* по критерий *K*;
- *Presentation* – процес на визуализация на моделираните компоненти в динамичен изглед *View* ($View = ES \cup EV$) по критерий *K* със следващо:
 - архивиране (*Digitize*) на резултати от процеса;

- *NewDecompose* – процеса на нов анализ на структурните и визуални компоненти;
- *ReTransform* – процеса на повторно изпълнение на трансформация за елементите *ES* и *EV*;



Фигура 1. Модел на процес за моделиране и визуализация на достъпни динамични изгледи.

Представената схема на взаимодействие между основните елементи на процеса е в основата на проектирането и създаването на система за ВМДДИ. Описанието следва общия модел за визуализация на изглед с допълнени етапи и методи, които се прилагат при обработката на информационните ресурси.

На диаграмата модела на процеса за ВМДДИ има 4 основни прехода: Дигитализация, Анализ, Трансформация, Визуализация. При етапа на дигитализация информационните ресурси и съхранените изгледи се съхраняват и подготвят за следващите преходи по движението в модела. След етапа на *Digitize* върху цифровите обекти *DO* се прилага анализ *Decompose* за обработка и форматиране с алгоритъм *A*. След анализа, данните *DS* са готови за етапа на трансформация *Transform*. Върху *DS* се прилага определен модел *M*, резултат от който е създаването на структурни и визуални компоненти *ES* и *EV*. Тези елементи са части на структурно-визуална форма, върху която може да се приложи определен критерий *K* при етапа на визуализация *Presentation*. Крайният резултат от процеса е динамичен изглед *View*, който се явява множество от визуални и структурни елементи *ES* и *EV* по модел-шаблон *M*, визуализирани по критерий *K*. Необходимо е да се отбележи, че в етапа от процеса преди момента на визуализация (*Presentation*) е възможно да има нов анализ (*Decompose*) или трансформация (*Transform*).

2.3. Методика за създаване на концептуална рамка

Основните фази за създаването на концептуална рамка и система за МВДДИ се формират от следните (в непоследователен ред) стъпки:

- Следване на етапите на процесите на обобщен модел за визуализация на изглед и модел за моделиране и визуализация на достъпни динамични изгледи;
- Проучване на възможността за осигуряване на достъп до системата на голям брой потребители, независещ от специфични умения и способности;
- Проучване на възможностите за осигуряване на интерактивни инструменти за моделиране на динамичните изгледи;
- Проучване на подход при създаване на софтуерна система и визуализация на изгледите за различни по вид устройства – достъпност от гледна точка на устройство и презентационна форма;
- Описание и следване на директивите и принципите за достъпност по критериите на WCAG;
- Осигуряване на подходящи техники и инструменти за тестване и оценка на системата за достъпни динамични изгледи.

Десктоп приложение vs. Уеб базирана система

Разпространените основни подходи, които се използват за създаване на софтуерни приложения са:

- Десктоп система.
- Уеб базирана система.

Главната идея, която надделява е, че една **уеб базирана система е по-достъпна** за голям брой потребители (аспект на достъпността) без предварителни условия (развити специализирани технически умения, необходимо е единствено наличие на браузър). Допълнителни предимства на едно приложение в уеб среда са: по-лесна поддръжка; интеграция с външни системи; бърз достъп за развитието на системата само до едно място – сървър; няма ограничение за работа със системата само от определен компютър или определено място; не е необходимо да се правят инсталации на клиентски машини. Недостатъци са: по-трудно реализируема – изискват се владееенето на комплекс от техники и езици за програмиране; в повечето случаи една уеб базирана система трудно се конкурира като бързодействие с десктоп система.

Предизвикателство за реализирането на **динамична интерактивна система в уеб среда** е използването на голям набор от инструменти като: сървърни езици; декларативни, презентационни и стилови езици; скриптов езици, позволяващи динамична промяна на поведението в рамките на уеб страницата; осигуряване на възможност за интеграция с приложни библиотеки и осигуряване на връзка с външни API.

За визуализацията на данни на различни по размер устройства в уеб среда се използват 2 подхода за преизмеряване на уеб страниците:

- „responsive“ дизайн;
- адаптивен дизайн.

При „responsive“ дизайна се създава гъвкаво или относително оформление на уеб страниците за визуализация на различни устройства. При подобен подход някои от елементите е възможно да изчезнат или автоматично да бъдат намалени. Всичко това е свързано в известна степен с намаляване скоростта на зареждане на данните.

При адаптивния дизайн се декларират няколко различни оформления на сайта с определени фиксирани размери за различните устройства:

- от 320 до 767 пиксела за смартфони;
- от 768 до 959 – за планшети;
- над 959 – за компютри.

Принципи за създаване на достъпни интернет страници според WCAG

Web Accessibility Initiative (WAI) разработва и предлага принципи за осигуряване на достъпност за W3C-технологии, като HTML, XML, SMIL, CSS и SVG [WAI'18]. Проектирането и създаването на достъпен дизайн според WAI, в общия случай, се постига чрез съобразяването с конкретните принципи, критерии и официални директиви на World Wide Web Consortium (W3C), които трябва да се следват при проектирането и реализацията на уеб страници [Енков'13].

От 2018 г. [WCAG'18] се развива версия WCAG 2.1, като добавя нови критерии (за мобилна достъпност, за хора със слабо зрение, за хора с когнитивни и обучителни затруднения); дефиниции, които подкрепят критериите; насоки за организиране на допълненията и помощни бележки към раздела за съответствие.

Също така, за да бъде реализиран като достъпен даден уеб сайт, се налага при проектирането му да се осигури [Тотков'13]: достъпен дизайн; валидност на XHTML и CSS; внедряване на достъпни технологии; логическо разделение на сайта; структурно разделение на сайта.

Софтуерни инструменти за тестване на достъпни интернет страници според принципите на WCAG

Име	URL-адрес и описание
WAVE	https://wave.webaim.org/ – Онлайн инструмент за оценка на достъпно уеб съдържание за хора с увреждания. Идентифицира грешки и бариери в насоките за достъпност на уеб съдържанието (WCAG). С възможности за включване и изключване на форматирането и стила на страниците.
Tingitun Checker	https://checkers.eiii.eu/en/pagecheck2.0/ – Онлайн инструмент, използван за разкриване на бариери за достъпност в уеб съдържанието за хора с увреждания спрямо специфични насоки на WCAG. Възможност за проверка на HTML и PDF документи.
AInspector WCAG	https://addons.mozilla.org/en-US/firefox/addon/ainspector-wcag/ – Разширение на Firefox за проверка на уеб страници за функции и проблеми, свързани с изискванията на WCAG 2.0 ниво А и АА.
axe DevTools	https://www.deque.com/axe/devtools/ – Разширение на Google Chrome за коригиране на проблеми с достъпността по време на

	разработката на уеб сайт.
AChecker Accessibility	https://achecker.achecks.ca/checker/index.php – Онлайн инструмент за анализ и автоматизирана оценка на достъпността на уеб страници и PDF документи. Има възможност за валидиране на CSS форматирането.
User-Agent Switcher	https://chrome.google.com/webstore/detail/user-agent-switcher-for-c/djflhoibgkdhkhcedjklpkjnoahfmg – Разширение на Google Chrome за симулиране и превключване на различни версии, както на мобилни, така и на настолни браузъри.
A11y Validator	https://color.a11y.com/Contrast/ – Онлайн инструмент за анализ на цветовия контраст на уеб страница или избрана цвят двойка, според насоките на WCAG.
Toptal Colorlab	https://www.toptal.com/designers/colorfilter – Онлайн инструмент за избор на безопасни цветове в процеса на проектиране на уеб страницата. За хора с цветен дефицит или нарушена способност за виждане на цветно зрение.

Таблица 2. Инструменти за тестване на достъпни интернет страници.

2.4. Архитектура на рамка на системата

Използваният подход в изследването предполага следната концептуална архитектура за рамка на система за моделиране и визуализация на ДДИ.



Фигура 2. Концептуална архитектурна рамка на система за МВДДИ.

Представените модули в рамката следват напълно естествено подхода при проектирането. На диаграмата са представени четири слоя на архитектурната рамка:

1. Потребителски интерфейс;
2. Система за сигурност.
3. Основни модули;

4. Информационни ресурси. Слой за потребителя, в частния случай при реализацията в уеб среда, ще бъде браузър. Системата за сигурност съдържа функции за аутентификация и оторизация и е представена на диаграмата с името „Оторизация“. Тя дава възможност за входиране и задаване на права и роли на потребителите в системата.

Слой „Основни модули“ съдържа основните елементи на архитектурната рамка на базата, на които ще се реализира приложението. На диаграмата модула за трансформация и моделиране е представен с името „Моделиране“, а динамични изгледи и диаграми е представен като „Динамични изгледи“, при останалите модули има точно съвпадение на имената. Четвъртия слой „Информационни ресурси“ съдържа наличните източници и бази от данни, предварително съхранени изгледи и външни системи от които ще постъпват необходимите данни, заявени за анализ, обработка и последваща визуализация.

Подходи за реализация на архитектурния модел в зависимост от източниците на данни

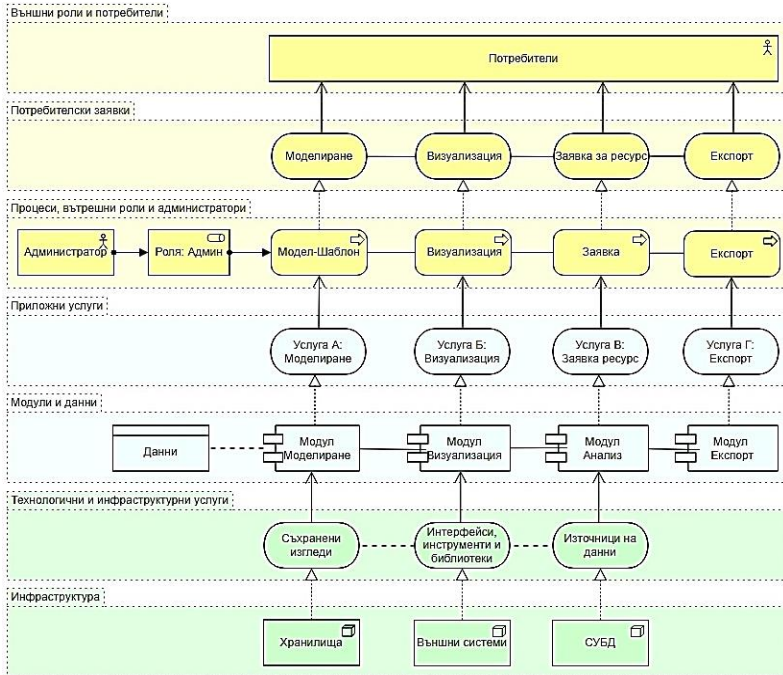
Като следствие от диаграмата на архитектурната рамка се явяват следните похвати за реализирането на прототипи на система за МВДДИ, разделени в зависимост от начина по който информационните източници захранват динамичните изгледи:

- П1.** Чрез информационни ресурси, постъпващи директно от източници на данни, които се обработват и визуализират от системата.
- П2.** Чрез визуализация на съхранени готови изгледи, налични в бази от данни или външни приложения за системата.
- П3.** Чрез външни системи, които съхраняват метаданни за модела на изгледа и са свързани с подходящи бази от данни и хранилища, а системата за МВДДИ се обръща към тях чрез интегрирани уеб услуги.
- П4.** Комплексен подход с възможности от изброените по-горе.

За реализиране на прототипите са избрани П1 и П3, осъществени съвместно с подходите за възможността за участието на потребител при трансформацията и моделирането на изгледа (виж Глава 3).

2.5. Схема на взаимодействие на елементите в системата

Посоките в схемата са от гледна точка на ролите, модулите и ресурсите, които осигуряват заявените приложни услуги и захранват изгледите с данни. Разделени са в три общи групи, симетрични на три архитектурни слоя: външни приложения и инфраструктура; модули и приложни услуги на системата; роли и потребителски заявки към системата. Прекъснатите линии или стрелки са взаимодействия между елементите от една и съща група или ниво. Непрекъснатите линии са за компонентите, които обменят съобщения между различните архитектурни слоеве или между различните роли. Частите, между които няма изобразена връзка, могат да ползват общи ресурси като данни или услуги.



Фигура 3. Модел на взаимодействие между компонентите на обкръжението на СМВДДИ.

Изобразените нива и елементи са:

- Външни потребители и роли, на които системата предоставя услуги.
- Потребителски заявки – услуги, които се обслужват от системата за използващите я потребители.
- Процеси, с които се реализират потребителските заявки и услуги. Вътрешни актьори и роли също могат да реализират част от определени услуги (например създаване и тестване на модел на изглед).
- Приложните услуги са част от слоя на системата и реализират приложните процеси. Осигуряват подходящи инструменти и потребителски интерфейси (GUI) или интерфейси (API) за комуникация с различни външни елементи за системата.
- Приложението реализира модулите и услугите и управлява активите от данни.
- Технологичните услуги поддържат съхранени изгледи и източници на данни, външни за системата от приложения (напр. платформи, капацитет, софтуерни разработки). Независимо, че инструментите и библиотеките

могат да са интегрирани в системата, в схемата са описани като част от външните технологични услуги.

- Инфраструктурата е част от слоя, който осигурява платформи, хранилища, системен софтуер и устройства за управление на информационните ресурси.

2.6. Изводи

Резултатите от изследването във 2-ра глава осигуряват широк спектър от средства за създаване на система за МВДДИ – от базови теоретични модели и архитектурна рамка, до методика за проектиране и създаване на програмни приложения за МВДДИ.

Основните етапи, които се следват за създаването на концептуална архитектурна рамка на СМВДДИ са изведени от модела за визуализация на ДДИ: анализ, трансформация, представяне, интерпретация. За реализацията на софтуерни прототипи е предпочетена уеб базирана среда, за която са създадени подходящи модули и инструменти за изграждане и генериране на изгледите.

Методиката и софтуерните прототипи, разработени на базата на общия теоретичен модел, са апробирани за разнотипни обекти (справки и репорти, уеб сайт, софтуерни приложения за взимане на експертни решения и др.) и процеси за МВДДИ във ВО с различна сложност. Така например, следвайки подходите, в 3-та глава са създадени софтуерни прототипи за моделиране и визуализация на ДДИ, и са проведени експерименти за МВДДИ на специфични обекти, елементи на информационни ресурси от университетски системи и данни. В хода на проведените експерименти, след въвеждане на необходимите промени, е потвърдена целесъобразността и ефективността на предложените подходи и на методиката за създаване на софтуерни приложения, свързани със система за МВДДИ.

Глава 3. Софтуерни прототипи за моделиране и визуализация на достъпни динамични изгледи

При реализацията на софтуерен прототип за МВДДИ са използвани описаните модел, архитектура и подходи в Глава 2. при създаването на софтуерно приложение в уеб среда. Реализациите на прототипите са разделени в две категории, условно наречени сценарии:

- C1.** сценарий без участие на потребител¹ при трансформацията и обработката на заявен резултат и съответната промяна на изгледа. При определена заявка от потребителя, системата автоматично обработва и връща резултата – динамичен изглед, предварително създаден или базиран на предварително зададени условия.
- C2.** сценарий с участието на потребител при трансформацията на анализирания резултат и моделирането на изгледа. Тук потребителя има възможност да моделира представения първичен изглед чрез интегрирани средства в уеб средата.

¹ потребител без технически познания и специализирани умения.

При сценарий 1 (C1) за реализацията на прототипите са използвани помощни инструменти на системата от приложения **JasperSoft** и възможностите на рамката **Vue.js**. За сценарий 2 (C2) е създаден **автоматичен SQL генератор** на заявки и са имплементирани допълнителни **JavaScript библиотеки** за визуализацията на потребителските инструменти и функционалности в браузъра.

Сценариите (C1 и C2) са реализирани с динамична презентационна рамка (Dynamic Presentation Framework – DPF).

3.1. Динамична презентационна рамка

На базата на теоретичните резултати е създадена архитектура на динамична презентационна рамка (DPF) за МВДДИ. Рамката предоставя възможности за създаване на прототипи и инструменти за анализ и обработка на заявки от уеб браузър и последващи интерактивни представяния, трансформация и визуализация на данни. При реализацията на рамката се следват методиките от гл. 2 за създаване на изгледи на различни устройства в уеб среда и принципите за достъпни интернет страници според WCAG.

DPF предоставя функционалности за анализиране и интерпретиране на данни в подходящи изгледи от различни информационни източници. Реализирана е чрез програмния език PHP и използва MVC (Model-View-Controller) парадигмата.

DPF съдържа следните модули и прилежащите им компоненти:

1. Модул за анализ (A), съдържащ функционалности от компонентите: SQL генератор, Диспечер на заявки, REST уеб сървис.
2. Модул за трансформация (T), съдържащ функционалности от компонентите: RAWGraphs, Jsreports, XML парсер, D3.js, Vue.js.
3. Модул за визуализация (B), съдържащ функции от компонентите: Контролер за стилове (style control module), RAWGraphs, Jsreports.
4. Модул за експорт (E), съдържащ функции от изброените по-горе компоненти: REST уеб сървис, RAWGraphs, Jsreports.

Архитектурата на динамичната презентационна рамка следва стандартния тип тристепенна архитектура с три слоя – интерфейсен, логически слой, разделен на модули и слой за източници на данни. В основата на DPF е модулно-компонентния слой (логическия), чрез който потребителят може да заяви или да генерирана изглед по определен критерий и да види резултат от заявката. Този слой се явява ядрото на презентационната рамка за МВДДИ. В различните сценарии DPF създава автоматично изглед или позволява на потребителите чрез предварително дефинирани условия да модифицират някои атрибути на изглед като местоположение, цвят, размер на шрифта и т.н., за да се визуализира изгледа по удобен за потребителя начин.

REST уеб сървис е софтуерна архитектура за реализация на уеб услуги, която е алтернатива на SOAP. Обикновено се основава на свободно дефинирани HTTP методи за достъп до ресурси чрез URL-кодирани параметри и използването на JSON или XML за предаване на данни. В архитектурната рамка се явява компонент, част от модула за анализ, който осъществява връзката между слоя с източниците на данни и частта, която се грижи за обработката им в Т (при C1 с Vue.js). Съществува като интерфейс за връзка между отделните слоеве, компонентите на които, могат да бъдат заменени

или развивани независимо един от друг при условие, че съобщенията, които разменят остават в стандартния формат. В модула за експорт (при C1 с участието на JasperSoft) функциите на REST уеб сървис се използват за връзка с подходящия формат на данните, който трябва да бъде визуализиран или свален от потребителя-клиент (HTML, PDF, XLS и др.).

Диспечер на заявки е елемент на системата, който дава възможност да разпредели заявката от потребителския интерфейс към определени функционалности или модули на системата, в зависимост от атрибутите на заявката или зададени условия. При сценарии C1 (с участието на Vue.js) се използва за анализ на заявен ресурс от потребителя и пренасочване към източник на данни или компонент за обработка на заявката. В C2 се използва съвместно с SQL генератора за анализ на правата на съответния потребител и конфигуриране на SQL заявката.

SQL генератор е компонент, създаден за изграждане на SQL заявки от потребители без специфични познания. Има възможност за формиране на заявки с лесен за използване интерфейс за плъзгане и пускане, без да се пише нито един ред код. Генераторът автоматично създава необходимата заявка на базата на маркираните атрибути от потребителя и се обръща към конкретната релационна база данни за изпълнение.

RAWGraphs [Mauri'17] е библиотека с отворен код за визуализации на данни, която предоставя функционалности за проекция на данните върху визуални променливи и моделиране на сложни визуализации. Предоставя възможности чрез интерактивно участие на потребителя да манипулира определени параметри и свойства на данните и да променя структурни и визуални елементи на изгледите. Данните е възможно да бъдат експортирани във формати, с отворен код, които могат да бъдат допълнително редактирани.

Jsreports е библиотека на Javascript за редактиране, преглед и експортиране на документи в уеб среда. Компонентите на дизайнера на библиотеката предлагат възможност за работа с данни в JSON или CSV формат и създаване на собствени типове персонализирани елементи, които могат да се плъзгат и пускат в уеб страниците с последваща визуализация и запазване на създадените шаблони в HTML, PDF и Excel [Jsreports'20].

XML парсер – DOM парсер за манипулиране и добавяне на различни атрибути към дървото на текущия уеб изглед, който се представя на потребителя. Използва се за манипулиране на определени тагове от XML дървото при обработката на изгледи в XML формат.

D3.js е представен в Глава 1, Javascript библиотека, която се използва за трансформация на данните за визуално представяне в диаграми и графики в частта за динамични изгледи в сценарии 2.

Vue.js е MVVM Javascript рамка за изграждане на потребителски интерфейси и приложения. Разполага с адаптивна архитектура, която се фокусира върху декларативното изразяване и състава на компонентите. Съдържа пакети, функции и помощни инструменти за създаване и управление на потока и състоянието на данни в приложенията [VueJS'20]. Независимо, че в реализираните прототипи се използва като компонент в слоя за трансформация, рамката предлага също и допълнителна библиотека, фокусирана само върху презентационния слой.

Style Control Module – служи за промяна и контрол на различни стилови параметри за визуализация (color, text-align, font-size и др.). Реализацията на компонента основно осигурява пакет за управление на визуализацията и стила на уеб страницата. Реализирана е с използване на CSS стилове за създаване на responsive и адаптивен дизайн.

Допълнително са използвани асинхронните технологии jQuery и Axios. **jQuery** е популярна JavaScript библиотека с отворен код, предназначена да опрости обхождането и манипулацията на DOM дървото на HTML, както и обработката на събития, CSS анимацията и Ajax. **Axios** е HTTP клиент, базиран на заявки, който работи както в браузъра, така и в среда Node.js. Той предоставя API интерфейс за работа с XMLHttpRequests и Node. Използването му позволява лесен начин за интеграция на бекенд нивото и взаимодействие с REST услуги [Axios'20].

В таблицата е описана връзката на всеки компонент с модулите, сценариите, програмен език и технология, използвани за всеки от тях.

Компонент	Част от модул	В сценарий	Технология
REST уеб сървис	A, E	C1	PHP, JSON
Диспечер на заявки	A	C1, C2	PHP
SQL генератор	A	C2	PHP, jQuery
XML парсер	T	C2	PHP, XML
Vue.js	T	C1	JavaScript, Axios, JSON
RAWGraphs	T, B, E	C2	JavaScript, CSS, JSON
Jsreports	T, B, E	C2	JavaScript, CSS, JSON
Стилов контролер	B	C1, C2	HTML, CSS
D3.js	T	C2	JavaScript

Таблица 3. Компоненти и модули на DPF в сценариите.

3.2. Прототип без участие на потребител при обработката и моделирането на заявен резултат (C1)

Компонентите, които са част от технологията на **C1 с участието на инструменти и функционалности на JasperSoft** са следните: 1. REST уеб сървис – елемент от аналитичния модул, който играе ролята на свързващото звено между логическия слой и услугите, предоставени от JasperServer, съответно и експорт на изгледите в общоизвестни формати; 2. Стилов контролер, част от визуализационния модул, с възможност за представяне на изгледа в браузъра на клиента. Като обработката и трансформацията на динамичния изглед се осъществява с помощта на софтуерни инструменти на JasperSoft за създаване, промяна и съхранение на модели на изгледи.

Всички разработени модели-шаблони на изгледа са съхранени на JasperServer. JasperReport Server играе междинна роля между двата архитектурни слоя:

- Прототипът прави заявка към услугите на JasperServer чрез REST сървис;
- Интерфейсът Web Service отговаря на HTTP заявки от клиентското приложение.
- Уеб услугата стартира избран шаблон и генерира отчет, като JasperServer се обръща към източниците на данни за конкретни резултати, които пълнят изгледа;

Моделът на изгледа се създава с инструмент дизайнер – iReport Designer (версия 5.2.0), който е част от комплекса от програмни приложения на JasperSoft.

Дизайнерът е десктоп система с вградени инструменти за манипулиране на модела-шаблон и интегрирана функционалност за реално експериментиране на представянето на изгледа, при което се зареждат структурните и визуални елементи, заложиени в модела и се запълва изгледа с данни. Шаблонът, създаден с дизайнера представлява моделиран XML файл с разширение jrxml. Файлът съдържа структурата и форматиращите правила на модела-шаблон, определени по DTD схема (Document Type Definition) на JasperSoft.

JRXML структурата е композиция, която се създава по време на формирането и изграждането на шаблона и представлява дефиниция на модела по който се генерира определения изглед в презентационния слой. Този шаблон може да бъде променен чрез предоставени функционалности и инструменти в дизайнера и има възможност да се динамизира начина на визуализация според предварително зададени параметри и критерии. Данните, които се зареждат в инстанцията на модела – динамичния изглед, могат да бъдат от различни хранилища и източници като по подразбиране представянето в уеб базирана среда е в HTML формат.

Изгледът служи за извеждане на резултата от заявката в браузъра на потребителя при HTML-формат. При избран друг формат справката се сваля на компютъра на потребителя. Също така потребителят няма възможност да моделира шаблона на изгледа, който е предварително съхранен на външен източник спрямо СМВДДИ – JasperServer. Независимо, че системата е уеб базирана, изгледа се създава от десктоп приложение, а комуникацията до презентационния слой става с помощта на предоставени уеб услуги и интегриран уеб консуматор от страна на клиента. Функционалностите от етапа на трансформацията и моделирането на изгледа не са част от динамичната презентационна рамка, а се явяват елементи на външна система с подходящи инструменти и функции за обработка и манипулиране на модела на изгледа. Това доказва, че изгледа може да не се създава в системата за МВДДИ, а да се доставя в готова форма от външно приложение и да се представи в уеб браузъра.

Компонентите на архитектурната рамка на DPF, които участват в C1 с Vue.js са: Диспечер на заявки, REST уеб сървис, Vue.js рамка, Стилов контролер.

Сценарият е реализиран чрез интеграция със съществуващия университетски уеб сайт на ПУ „Паисий Хилендарски“ – uni-plovdiv.bg. Сайтът е създаден посредством MVC модел на PHP, което допълнително помага за по-лесното свързване на компонентите на DPF рамката в конфигурацията му. На нивото на логическия слой се използва – аналитичния модул, представен от диспечер на заявки и REST уеб сървис; и трансформационния модул с представител Vue.js. В класическия случай уеб сайта е свързан с релационна база данни MySQL, където се съхранява част от съдържанието на страниците в HTML формат, които след заявка се представят в браузъра. За реализирането на C1 е направена нова връзка към MSSQL релационна база, където се съхраняват данните, които се достъпват от REST уеб сървис и се подготвят за визуализация с Vue рамката.

Според модела от Глава 2. анализа (А) се осъществява от диспечера на заявки, като REST сървиса достъпва и извлича данните, Vue компонента осъществява

обработката и трансформация на ресурсите (Т), а презентацията (Р) се реализира с помощта на стилев контролер.

Реализацията на сценария с Vue рамката е осъществена по следната схема:

- Изпраща се заявка за определена страница от брауъра.
- Диспечера анализира заявката и подава сигнал към сървиса за извличане на данни от базата.
- На следващия етап към трансформационния слой се подават данните, заедно с ID на страницата и шаблона на изгледа, който трябва да се обработи.
- След което Vue.js оформя изгледа с подадените данни и параметри.
- На последния етап се зарежда представянето в брауъра чрез стилев контролер.

Изгледа е реализиран чрез синтаксис на Vue рамката в HTML форма. Структурата на синтаксиса съдържа атрибути, свойства и контролни условия, които се вграждат чрез директиви и двойни скоби в кода на страницата. Като заедно с извлечените данни се съдържат и метаданни за изгледа, който трябва да се зареди. На базата на заявката от брауъра Диспечера изпраща ID на изгледа, който ще бъде зареден с ресурсите извлечени от REST уеб сървиса.

ID	Име на модела	Тип на изгледа	Примерна заявка
1	V1	Катедра	uni-plovdiv.bg/pages/index/2013/
2	V2	Ректорско ръководство	uni-plovdiv.bg/pages/index/14/
3	V3	Ръководство на факултет	uni-plovdiv.bg/pages/index/80/

Таблица 4. Съответствие на уеб страници с модели на изгледи в C1 с Vue.

Изгледът се явява визуализирани данни на административния или преподавателския състав на ПУ в уеб страницата, където е инжектиран като частичен елемент в характерната визия на сайта. Като за целта са вградени и допълнителните стилове на добавения изглед.

Сценарий C1 с Vue.js е реализиран чрез добавяне, инжектиране на нов слой към съществуващ такъв в сайта на Пловдивски университет. Динамичността е осъществена с помощта на диспечер, който осигурява анализа на пристигащите заявки от брауъра и уеб услуга, предоставяща данни за визуализацията на изгледа.

В зависимост от информационните ресурси, които се получават сценария попада в P1 – данни постъпващи директно от БД, част от които се обработват от трансформационен слой и се представят чрез визуализационен компонент от прототипа за ДДИ. При този подход потребителят не участва директно при трансформацията и моделирането на обектите, а резултата се получава обработен наготово при заявката на желанния ресурс. Ресурсите се намират в релационна база, като визуализацията им предварително се подготвя чрез аналитичен и модифициращ компонент на системата. В зависимост от данните, те се свързват с подходящия шаблон на изгледа и се вграждат в уеб страницата на сайта.

3.3. Прототип с участие на потребител при обработката и моделирането на заявен резултат (C2)

При този сценарий потребителят след като получи определена визуализация на изгледа, има възможност да го манипулира и моделира с помощта на предоставени инструменти и функционалности от софтуерния прототип. Сценарият е реализиран като инструмент за създаване на справки и динамични изгледи на резултата, заявка от релационна база данни.

При създаването на изглед за справка е необходимо да се съобрази, че справката може да бъде създавана от потребители без познания и възможности за създаване на SQL заявки [Близнаков'18]. За тази цел е създаден SQL генератор на заявки за БД и са интегрирани допълнителни JavaScript библиотеки за визуализацията на потребителските изгледи в браузъра. Реализацията на генератора става чрез представяне на релационната структура на базата данни в модел на неориентиран граф, където отношенията между таблиците се явяват ребра на графа. JS библиотеките са имплементирани за манипулиране и промяна на справката в уеб средата и предоставят възможност за експорт и съхранение на изгледите на справките от потребителя.

```

176 + public function bfs($listToExplore, $graph) {
177
178     $graph[$listToExplore[0]]["visited"] = 1;
179 + while (count($listToExplore) > 0 ) {
180         $nodeIndex = array_shift($listToExplore);
181
182 +     foreach($graph[$nodeIndex]["links"] as $childIndex) {
183 +         if ($graph[$childIndex]["visited"] == 0) {
184
185             $graph[$childIndex]["visited"] = 1;
186             $graph[$childIndex]["path"] = array_merge($graph[$nodeIndex]["path"], array($nodeIndex));
187             array_push($listToExplore, $childIndex);
188         }
189     }
190 }
191
192 return $graph;
193 }
```

Фигура 4. Код за намиране на най-кратък път в графа.

На базата на намерения път генераторът присъединява към заявката, участващите таблици, чиито колони са избрани. SQL заявката се създава като в SELECT формата се добавят избраните полета от интерфейса и заедно с това се присъединяват следващите таблици (JOIN), участващи в пътя на графа. След което се продължава по намерения минимален път и се присъединяват и останалите таблици и прилежащите полета. Някои колони са невидими в графичния интерфейс за обикновения потребител като ID-та и външни ключове – те са описани в структурата на графа като възможност за реализиране на операциите за присъединяване, по които да се създаде SQL заявката.

В потребителския интерфейс е добавена възможност за ограничаване на множеството от избраните записи (WHERE клауза) от потребителя, чрез функционалност за създаване на филтър в потребителския интерфейс. Всяко избрано поле има възможност да участва във филтъра на заявката чрез добавяне като отделен запис в потребителския интерфейс. Всеки запис представлява: име на поле, релация и стойност (текст, цифра, дата). При добавяне на повече от един запис, филтъра го

добавя към (WHERE клауза) като AND. В разширения интерфейс се добавя възможност и за OR.

Визията на контейнера на изгледа е опростена за по лесна и интуитивна навигация. Има възможност за преглед и съдържа бутони за опресняване, редактиране и запазване на изгледа. Структурата на изгледа е в JSON формат и се съхранява в релационна БД. JSON формата се явява шаблон на изгледа, който може да се използва като модел за евентуална бъдеща употреба. Структурата му може да бъде извлечена, визуализирана в разбираем от потребителя изглед и променена с вградените инструменти в прототипа, след което да се запише новата редакция.

Инструментът за преглед на изгледа и последващото моделиране е реализиран чрез jsreports компонент – JavaScript библиотека за манипулиране на визуални елементи в браузъра. Библиотеката предлага възможности за работа с данни в JSON и CSV формат с последваща визуализация в уеб страницата и „drag and drop“ възможности, запазване на изгледите и създадените модели на изгледи от потребителя.

Моделът на изгледа, включващ стойности за позиции, шрифт, големина, тип, имена на полетата, се запазва в JSON формат, който на по-късен етап може да бъде достъпен и променен.

За визуално представяне на данни и динамични диаграми в прототипа е интегриран компонент, чрез който обикновен потребител има възможност да създава и променя специфични по вид графики и диаграми, представящи различни проекции на данни. Тази функционалност е реализирана с помощта на RAWGraphs – JavaScript библиотека за манипулиране и създаване на диаграми с участие на потребител, който има възможност да моделира и да настройва визуални елементи на изглед, представен като диаграма. Диаграмите могат да бъдат експортирани във формати, с отворен код, които могат да бъдат допълнително редактирани. Има възможност за избор на модел на диаграма от налични такива като: кръгова и дървовидна, клъстерна дендограма, сравнителна, план и бар-диаграма, хоризонтална, радиална и други.

При реализацията са използвани D3.js компоненти и функции, библиотека с отворен код за визуализация и проектиране на правилния визуален интерфейс за данните в браузъра. Основният принцип на визуализация при подобен подход е възможността за употреба на селектори и набор от възли на DOM обектите, след което да се използват оператори и функции, за да бъдат манипулирани. Изборът на елементите и селекторите може да се основава на HTML таг, клас, идентификатор, атрибут или място в йерархията. След като елементите са избрани, може да се приложи определен тип операции към тях. Това включва промяна и настройка на атрибути, текстове на дисплея и стилове. Селекторите също могат да се добавят и премахват. Този процес на модифициране, създаване и премахване на HTML елементи може да бъде зависим от данните, което е част от концепцията на прототипа.

За този сценарий изгледът се явява резултат от аналитичните функции и предоставени възможности на потребителя да променя и манипулира изгледа. Визуализацията му може да бъде моделирана и променена според избора на потребителя в зависимост от избраната диаграма, данните и проекциите им върху

шаблона на изгледа. Демонстрирана е възможността за създаване на заявка чрез инструмент за потребител без познания по релационна структура на БД, която е моделирана в неориентиран граф като средство за анализ на данните и автоматизирано създаване на SQL-заявка. Това е с цената на опростяване на заявката и подобна реализация губи част от средствата за създаването на сложни множества между структурите на релационна база.

3.4. Заключение за реализираните софтуерни прототипи

Прототипите за моделиране и визуализация на достъпни динамични изгледи в уеб браузър са реализирани с използване на различни способы чрез помощни външни системи и интегриране на рамки, библиотеки и инструменти, необходими за целите на изследването. Прилага се стратегията от Глава 2. като прототипите основно се разглеждат от гледна точка на участието на потребител при моделирането на изгледа и подходите в зависимост от начините, по които информационните източници захранват изгледите с данни.

Етапа на анализа от концептуалния модел се постига чрез техники и инструменти, подходящи за обработка на данните и помощни приложения за анализиране на връзките и ресурсите на източниците на данни. Моделирането се прилага чрез интегрирани инструменти в уеб браузър, като се използват два подхода: потребител не взима участие при моделирането на изгледа и потребителя активно променя и моделира изгледа на данните в уеб средата. При визуализацията са използвани интерактивни компоненти, софтуерни библиотеки и са направени експерименти с инструменти, съдействащи представянето на изгледа да бъде достъпно според принципите на WCAG. За реализацията на прототипите е избрана уеб среда, заради възможността на достъп на голям брой потребители без допълнителни инсталации.

Прототип	Подход от гл. 2	Инструменти и библиотеки	Изглед/Моделиране от потребител	Източници на данни
C1 с JasperSoft	ПЗ	JasperServer – външно десктоп приложение за системата	Изглед – модела е съхранен на външна система. Потребител не може да го моделира	Поддържани от JasperServer
C1 с Vue	П1	Интегрирана рамка на Vue.js	Изглед се създава в реално време. Потребител не може да го моделира	Релационна БД
C2	П1	Интегрирани JavaScript библиотеки на Jsreports, D3, RAWGraphs	Изглед се създава в реално време. Потребител може да го моделира	Релационна БД

Таблица 5. Прототипи за моделиране и визуализация на ДДИ.

Глава 4. Аprobация и тестове на получените резултати

В настоящата глава са описани тестове и експерименти с прототипите на софтуерната система, съгласно принципите от Глава 2. Представен е инструмент за оперативни решения, базиран на един от прототипите. Направен е анализ на уеб страници за избор на вариант за извличане и попълване на данни в изглед. Приведен е тест на SQL генератор на заявки с 2 релационни БД. Част от експериментите са реализирани с онлайн валидатори за проверка на уеб страници според критериите за достъпност на WCAG и са направени тестови симулации на изгледите за различни по размер устройства. Описани са основните резултати, постигнати с инструментите за тестване на достъпни страници и са представени част от динамичните изгледи, визуализирани като диаграми.

4.1. Приложение за оперативни и управленски решения

На основата на софтуерния прототип (C1 с JasperSoft) е създаден инструмент за оперативни и управленски решения – LATch [Doneva'20]. Приложение, което е презентационна рамка за моделиране и генериране на справки и динамични изгледи в уеб-среда. Архитектурата на приложението е с три слоя – интерфейсен, приложен слой и слой за източници на данни. С помощта на системата за проектиране на шаблони за отчети JasperSoft Studio е внедрена основната функционалност на слоя за приложения на DPF и неговата бизнес логика. При създаването на инструмента е разработен допълнителен стил към дизайна. Ползите от LATch помагат на университетските ръководители да вземат оперативни, управленски или средни управленски решения.

Слоят данни на приложението LATch включва различни бази данни на институционалната информационна инфраструктура (информационна система за студенти, Moodle и др.), както и самото хранилище на JasperReports Server. JasperReport Server се обръща към тях, за да извлече необходимите данни при генериране на отчети.

Етапите по които се реализират изгледите на справки в приложението са следните:

1. Създаване от администратор на шаблон на изглед с дизайнера за моделиране на JasperSoft. С инструмента се настройват полета, таблици, стил, шрифт, цвят и различни визуализации на данните в модела.
2. Създаване на заявка към релационна БД, резултата от която се попълва в шаблона. Често заявката и модела-шаблон на изгледа се създават паралелно, като едновременно се правят е тестови визуализации на изгледа в дизайнера.
3. Импортиране на създадения шаблон на JasperSoft сървър, където компилиран jrxml файл се качва като репорт, заедно с метаданни за необходимите атрибути и параметри, както и допълнителни изображения налични в създадения шаблон.
4. Вход в инструмента за оперативни и управленски решения от потребител без технически познания.

5. Избор и заявка на справка и евентуално попълване на желани параметрични стойности при стартиране на шаблона.
6. Експорт на изгледа на справката в желан формат.

Първите три стъпки се осъществяват от администратор, който подготвя заявката и модела на изгледа. Потребител без технически умения участва при заявяване на определена справка, подаване на параметри и експорт в желан формат, част от идеята на прототипа – обикновен потребител не моделира изгледа.

Допълнително прототипът може да се използва и като интегриран софтуерен инструмент, предназначен за анализ за експерти по качеството. Инструментът позволява да се наблюдава и подобрява учебния процес и може също така значително да помогне при подготовката на доклади за вътрешна и външна оценка на качеството на висшето образование. Въз основа на резултатите от инструмента за анализ, потребителите могат да предприемат мерки за подобряване на качеството на обучението и постиженията на студентите [Gaftandzhieva'20].

4.2. Анализ на уеб страници в прототипа C1 с Vue.

Анализът е направен с цел да се проучи начина, по който е необходимо да се реализира възможността за извличане и попълване на данни в сайта от две релационни БД. Съществуват два варианта за реализация на модула „Диспечер“ в зависимост от начина по кой път (S или V) да се насочи заявения ресурс:

- В зависимост от данните, които се получават от REST уеб сървиса, за да се визуализира резултата;
- В зависимост от предварително заложиени критерии за наличните уеб страници на сайта.

Изборът за вариант е на базата на проучване, проведено за сайта на uni-plovdiv.bg. Анализът обхваща периода 01.05.2020 г. – 01.05.2021 г. За изследвания период страниците от тип „Катедри“ и „Ръководство“, които се използват в сценарий C1 показват, че са били заявени общо 52792 пъти, което е под 1 процент (0,64%) от общия брой заявки към всички страници на сайта.

На базата на проведеното изследване е избран подхода (вариант 2), при който компонентите на изгледа се зареждат в зависимост от предварително заложиени критерии за наличните уеб страници на сайта. Критерият е заложен като условие дали заявената страница е подстраница или част от меню „Катедри“ или „Ръководство“.

4.3. Експерименти за C1 с Vue

В прототип C1 (Vue) са създадени изгледи чрез интеграция със съществуващия университетски уеб сайт на Пловдивски университет – uni-plovdiv.bg. Реализацията на изгледите е осъществена за част от уеб страниците, с помощта на Vue рамка и REST сървис с достъп до резултата от заявените данни.

Експериментите в раздела са направени единствено върху изгледите, които са реализирани с Vue рамката и с помощта на описаните инструменти и методика за тестване, описани в Глава 2.

Необходимо е да се отбележи, че инструментите не могат да намерят абсолютно всички препятствия, ограничаващи достъпността. Те могат да забележат

само бариери, които могат да бъдат открити автоматично. Резултатът от потребителските тестове се основава на ограничен набор от изпитания и също така не може да се покрият всички възможни тестове. Следователно оценените страници е възможно да съдържат прегради, които не са отчетени от нито един от инструментите.

Тест с инструмента WAVE Web Accessibility Evaluation Tool

Онлайн инструментът WAVE помага за откриване на грешки и бариери в насоките за достъпност на уеб съдържанието. Предоставя допълнителна информация за откритите грешки и осигурени достъпни функционалности за потребителя. Инструментът предоставя възможност и за тестване на страниците с включени и изключени стилове за съответната страница.

След завършването на теста се изобразява меню със следните шест секции, които показват бариери, несъответствия и помощни функционални елементи за оценената страница:

- Грешки – елементи или съдържание в документа, които правят страницата трудна или невъзможна за четене и разбиране от хора с увреждания;
- Контрастни грешки – нисък контраст между цветовете на текста и фона, който е необходим за потребители със слабо зрение. Секцията определя присъствието на текст със съотношение на контраста по-малко от 4,5:1 или при текст (по-голям от 18 пункта или 14 пункта удебелен) има съотношение на контраста по-малко от 3:1;
- Сигнали – обръща се внимание на определени атрибути в HTML документа, които е възможно да възпрепятстват определени потребители да достигнат информацията;
- Свойства – представят функционалност на потребителя за навигация по елементите или да прескача основното съдържание на страницата, значително помага на потребителите на клавиатурата при навигиране в уеб страницата;
- Структурни елементи – представят семантично и визуално значение в структурата на документа и улесняват навигацията по страниците за потребителите на помощни технологии;
- ARIA елементи – (Accessible Rich Internet Applications) набор от атрибути, които дефинират начини да направят уеб съдържанието на страниците (особено тези, разработени с JavaScript) по-достъпни за хора с увреждания.

Резултатът от тестовете на страниците и съответните изгледи, реализирани с прототипа показват 0 грешки и 0 контрастни грешки. За различните тестове показва диапазон между 6 до 9 сигнала, от 15-18 свойства, до 30 структурни елементи и над 100 дефинирани ARIA атрибути.

Тест с инструмента Tingtun Checker

Tingtun Checker предоставя услуги за автоматична проверка на достъпност за уеб страници и PDF документи. Отговаря на критериите за достъпност на уеб съдържанието 2.1 (WCAG 2.1). Резултатът от тестове се основава на набор от изпитания по критериите за достъпност.

0 barriers found on the web page		0 barriers found on the web page	
Checked page:	https://uni-plovdiv.bg/pages/index/1993/	Checked page:	https://uni-plovdiv.bg/pages/index/840/
Time:	2021-12-17T13:03:33Z	Time:	2021-12-17T13:04:55Z
Applied Tests:	Total: 310 Fail: 0 Verify: 0 Pass: 310	Applied Tests:	Total: 323 Fail: 0 Verify: 0 Pass: 323
Score:	100.00 (where 100 is the best)	Score:	100.00 (where 100 is the best)
0 barriers found on the web page		0 barriers found on the web page	
Checked page:	https://uni-plovdiv.bg/pages/index/2010/	Checked page:	https://uni-plovdiv.bg/pages/index/627/
Time:	2021-12-17T13:06:59Z	Time:	2021-12-17T13:08:30Z
Applied Tests:	Total: 323 Fail: 0 Verify: 0 Pass: 323	Applied Tests:	Total: 325 Fail: 0 Verify: 0 Pass: 325
Score:	100.00 (where 100 is the best)	Score:	100.00 (where 100 is the best)

Фигура 5. Тестове за достъпност на изгледи на уеб страници.

Не са намерени грешки, възпрепятстващи достъпността. Като оценките на резултатите от теста са обединени в 4 групи от стойности:

- Total – общ брой оценени атрибути;
- Fail – брой несъответствия;
- Verify – брой атрибути за проверка;
- Pass – общ брой валидни.

Тест с инструмента AInspector WCAG

AInspector WCAG е разширение за браузър Firefox, което може да оценява отделна уеб страница въз основа на изискванията за достъпност на (WCAG) 2.0, ниво А и АА.

Инструментът залага на 4 категории за оценка на достъпността:

- V – Нарушения;
- W – Предупреждения;
- MC – Елементи за ръчна проверка;
- P – Елементи, които отговарят на критериите.

При теста са намерени 4 несъответствия, 2 от които не са свързани с интегрираните изгледи:

- цялото съдържание трябва да се съдържа в ориентири или контейнерни елементи – критерий 1.3.1, Ниво А;
- етикетът трябва да има съдържание – критерий 3.3.2, Ниво А.

Несъответствията са налични в хедъра на сайта и не са част от интегрираните динамични изгледи. Допълнително има 2 бариери, свързани с тях:

- идентифициране на маркирането на таблицата като данни или оформление – критерий 1.3.1, Ниво А;
- текстът на връзката трябва да описва целта на връзката – критерий 2.4.4, Ниво А.

Решени са по следния начин, следвайки критериите на WCAG.

1. Препоръка: Използване на `role="presentation"` на елемента на таблицата, за да се идентифицира ясно маркировка на таблицата за оформление.

Подобен подход допълва HTML, така че взаимодействията на елементите от потребителския интерфейс, които обикновено се използват в приложенията, могат да бъдат прехвърлени към помощни технологии, когато няма друг механизъм.

2. Препоръка: Използване на `aria-label`, `aria-labelledby` или атрибута `title`, за предоставяне на по-описателно достъпно име, когато текстовото съдържание на връзката не може да бъде променено.

След направените допълнения и добавените нови атрибути в таблицата, тестът не открива несъответствия, свързани с динамичните изгледи в сайта.

Тест с инструмента ахе DevTools

Безплатното разширение за браузър ахе DevTools е бърз и мощен инструмент за разработчици и дизайнери за тестване на достъпност.

Намерени са 2 несъответствия в заглавната част на сайта – елементите трябва да имат достатъчен цветови контраст – препятствия, които не са свързани с вградените динамични изгледа.

Тест с инструмента AChecker

AChecker Web Accessibility Checker прави анализ и автоматична оценка на съответствието на уеб сайтове и представя доклад за намерени проблеми на достъпността на страниците.

Не са открити грешки при тест и сканиране на достъпността на изгледите при зададените опции ниво WCAG 2.0 (Level A) от инструмента.

Допълнително свойство на инструмента е възможността за тест на валидация на CSS форматирането. Тестът показва валиден CSS за изгледа на уеб страницата.

Тест с инструмента User-Agent Switcher

Разширението на Chrome е потребителски агент за симулиране, както на мобилни браузъри, така и на настолни браузъри. Има възможност да се превключва между нивове на потребителските агенти, идентифициращи браузъра и операционната система към уеб сървъра

С User-Agent Switcher не са открити определени препятствия за достъпните изгледи на прототипа.

Тест с инструмента A11y Validator

С онлайн инструмента за определяне на цветовия контраст на адрес <https://color.a11y.com/Contrast/> е направен тест на изгледите.

Анализът показва проблемите с цветовия контраст на уеб страница или избрана цветова двойка; според насоките на WCAG 2.1. Цветовия контраст се отнася до това как ярки или тъмни цветове се появяват един срещу друг на екраните; особено по отношение на относителната яркост в сивата скала, както се възприема от човешкото око.

Текущото съотношението на цветовия контраст за тестваните изгледи е 11:1 при необходимо 4,5:1.

Тест с инструмента Toptal

Използването на онлайн инструмента Toptal Colorblind Colorlab помага на разработчиците да избират безопасни цветове по-рано в процеса на проектиране. Предназначен е за тестване на страници, които се ползват от хора с цветен дефицит и нарушено цветно зрение.

Предлага следните автоматични тестове за страницата:

- филтриране на изображения;
- филтриране на изображения без GIF;
- филтриране на таблици със стилове.

Допълнително, инструмента предлага следните 4 филтъра по избор, които могат да се приложат върху изгледа на уеб страницата:

4.4. Визуализация на изгледи на различни по размер устройства

На адрес <https://responsivedesignchecker.com/> се намира инструмент за тестване на отзивчив дизайн на уеб страници. Симулира различни по размер (настолни, таблети и смартфони) и резолюции на устройства.

Необходимо е да се отбележи, че дизайна е свързан със стила на уеб сайта и при реализирането на достъпните изгледи това е съобразено и е взето под внимание. Експериментите са осъществени само за интегрираните изгледи с Vue рамката.

Тъй като в колоните на таблиците в изгледите присъства сравнително малко съдържание е възприет подхода на сплескване на съдържанието хоризонтално. Което показва, че няма проблеми за мобилен екран и оформлението остава видимо и сходно на визуализация с по-голяма резолюция. При тестването с инструмента не са открити отклонения за изгледите на различни устройства или резолюции.

4.5. Експеримент с генератор на заявки

Генераторът на заявки е част от аналитичния модул, с помощта на който обикновен потребител има възможност да изгражда справки и да извлича данни без необходимост от познаване на структурата на релационна БД и програмни техники за създаване на заявки.

Направени са експерименти с две БД, които са част от инфраструктурата на ПУ „Паисий Хилендарски“, съдържащи данни за студенти, преподаватели и съответстващите им информационни обекти свързани с тях. С графа е симулирана структурата на релациите на бази данни MSSQL и MySQL. Описанието на графа е в JSON формат.

Симулирана релационна структура от MSSQL и MySQL

Симулираната MSSQL БД съдържа информация за студенти, преподаватели и административен състав на Пловдивски университет. Описаната релационна структура в графа е за частта от таблиците, съдържаща информация и данни за обучаващите се студенти в университета (Приложение 1).

При създаването на заявка се избират определени полета, които участват в справката. В конкретния пример са избрани 4 полета: Специалност, Курс, Дата записване, OKC и е зададен филтър за първи курс на студент.

След избора на избраните елементи генератора създава автоматично SQL-заявката и се обръща към БД за извличане на данните от базата.

Етапите, по които потребителя реализира справката са:

1. Избор на желани полета и евентуално добавяне на филтър (WHERE клауза).
2. Избраните полета и филтъра се трансформират в заявка от SQL-генератора и се подава за изпълнение към БД, откъдето се извличат данните.
3. Представянето на данните е в контейнер в брауъра и могат да се експортират или потребителя може да приложи промяна на визуализацията с вградените инструменти за манипулиране на изгледа.

Вторият експеримент е за симулирана структура на БД MySQL, съдържаща данни за кандидатстудентска информация: имена на кандидати, желани специалности, изпити, оценки от диплома за завършено средно образование и др. Избраните полета на диаграмата са: първо име на кандидат, вид оценка и стойност на оценка, без избран филтър. Генерираната заявката извлича автоматично данните от базата.

4.6. Визуално представяне на данни и диаграми с прототип C2

За представянето на данните и визуализацията им в брауъра ресурсите, с които се експериментира се зареждат от кандидатстудентска БД на ПУ.

В брауъра се визуализират четири секции:

- за зареждане на данни;
- за избор на диаграма от наличните;
- за настройка на атрибутите на диаграмата;
- за визуализация.

След зареждане на данните и съответните проекции на колоните от извадката е визуализирана стълбовидна диаграма (Bar) описваща броя на кандидат-студентите, заявили желание за полагане на съответните конкурсни изпити [ОтчетКСК'21].

4.7. Изводи

За осъществяването на прототипите се прилагат подходи и способности, при които се използват комплексни техники и разнообразни инструменти за обработка и манипулиране на изгледите и съответните им модели. Демонстрирано е, че изгледите могат да бъдат създавани като справки с помощта на външни системи и

представени в уеб приложение с подходящи интерфейси (C1 с JasperSoft). Могат да бъдат реализирани като динамични изгледи и да са премахнати значителна част от препятствията за достъпност по критериите и съответствията на WCAG (C1 с Vue). Достъпността е тествана с автоматизирани инструменти за оценка на изгледите и на различни персонализирани резолюции и видове устройства. Изгледите могат също да бъдат създавани и моделирани от потребители без технически умения с помощта на автоматизиран SQL-генератор и интегрирани програмни библиотеки, да бъдат визуализирани и експортирани като графики и диаграми (C2).

Прототип	Приложение/Експерименти	Ресурси/БД
C1 (JasperSoft)	Приложение за оперативни и управленски решения – моделите на изгледите се съхраняват в JasperServer и се създават с дизайнер, външна десктоп система.	Източници, поддържани от JasperSoft
C1 (Vue)	WAVE Accessibility Evaluation Tool – за откриване на грешки и бариери в насоките за достъпност на уеб съдържанието.	uni-plovdiv.bg, MySQL, MSSQL
	Tingun Checker – автоматична проверка на достъпност за уеб страници и PDF документи.	
	AIInspector WCAG – оценява отделна уеб страница въз основа на изискванията за достъпност.	
	ахе DevTools – инструмент за разработчици и дизайнери за тестване на достъпност.	
	Achecker – анализ и автоматична оценка на съответствието на достъпност на уеб сайт.	
	User-Agent Switcher – инструмент за симулиране, както на мобилни браузъри, така и на настолни браузъри.	
	A11y Validator – за определяне на цветовия контраст на уеб страница или избрана цвeтова двойка.	
	Toptal Colorblind – тест за хора с цветен дефицит и нарушено цветно зрение.	
	Responsive checker – визуализация на изгледи на различни по размер устройства.	
C2	Експерименти с SQL генератор на заявки в две релационни БД.	MySQL, MSSQL
	Експерименти с интерактивни инструменти за визуализация на данни и диаграми.	

Таблица 6. Реализирани приложения и експерименти с прототипите.

Заклучение

Не е трудно да се предположи, че подобни разработки, в направления близки до изследванията за представяне на динамични и достъпни форми на информацията, ще доведат:

- до създаването на един много по богат, осезаем и интерактивен визуален език за манипулиране и промяна на визуализацията на данни и информационни ресурси;
- не само до развитие на средствата за моделиране и представяне на идеи, но и до революционна промяна на техническите инструменти като посредници в комуникацията между хората.

В рамките на дисертационното изследване са решени следните основни задачи:

- Проучване на общата теория, съществуващи организации, модели, стандарти и системи за визуализация на информационни ресурси и ЦО;
- Създаване на концептуален модел и методики за анализ на данни, моделиране и визуализация на достъпни динамични изгледи;
- Въвеждане и създаване на софтуерен прототип(и) за визуализация на достъпни динамични изгледи;
- Тестване на софтуерен прототип(и) за достъпни изгледи и инструменти за динамична визуализация на информационни ресурси и обекти, следвайки конкретни методики.

С решаването на задачи 1 – 4 се постига основната цел на дисертацията – предлагане, изследване и апробиране на софтуерна система и средства (методи и инструменти) за моделиране и визуализация на данни и информационни ресурси в достъпни динамични изгледи.

Приноси на дисертационния труд

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

Научни приноси на дисертационното изследване са:

- НП1.** Създаден е общ модел на процес за визуализация на изгледи на цифрови обекти;
- НП2.** Създаден е модел на процес за моделиране и визуализация на достъпни динамични изгледи.

Научно-приложни приноси на дисертационното изследване са:

- НПП1.** Предложени са методики за създаване на концептуална рамка на система за моделиране и визуализация на ДДИ и за тестване на прототипи за МВДДИ;
- НПП2.** Предложена е архитектура на рамка за софтуерни прототипи за моделиране и визуализация на ДДИ.

Приложни приноси на дисертационното изследване са:

- ПП1.** Реализирани са модули на софтуерни прототипи за моделиране и визуализация на ДДИ;
- ПП2.** Тестване на софтуерни прототипи за достъпни изгледи и инструменти за динамична визуализация на информационни ресурси и обекти.

Перспективи за развитие

Резултати от Глава 2., 3. и 4. могат да бъдат да основата на следващи изследвания в следните 2 (две) направления:

- усъвършенстване на моделите за визуализация на изгледи на цифрови обекти и за моделиране и визуализация на достъпни динамични изгледи с развиване на съществуващите и добавяне на нови етапи в процесите;
- създаване и тестване на приложения за МВДДИ на други типове информационни ресурси от различни източници на данни.

В близка перспектива, на дневен ред са следните задачи:

- добавяне на нови интерактивни функции към съществуващите приложения за обработка на структурните елементи и манипулация на данните в уеб среда;
- добавяне на акумулиращи функции на генератора на справки за създаване на по-сложни заявки от потребител без технически умения;
- тестване на изгледите с асистирани технологии за потребители с увреждания и ограничени възможности на достъпност до уеб страниците.

В по-далечна перспектива предстои създаване на:

- методика и съответни инструменти за реализация на десктоп приложения за МВДДИ;
- разработване на възможност за създаване на инструмент за тестване и оценка на достъпни динамични изгледи в уеб среда.

Апробация

Резултати на дисертационното изследване са представени в 6 (шест) публикации – 2 (две) в специализирани списания и 4 (четири) в трудове на конференции (4 национални, една от които с международно участие). Две от публикациите са индексирани в световноизвестните бази от данни: Web of Science и Scopus. Част от резултатите на изследването – инструмент за оперативни и управленски решения, са апробирани в 1 (един) университетски проект: МУ19-ФТФ-001 „Интелигентен анализ на данни за подобряване на резултатите от обучението“ (2019-2020), финансиран от Фонд „Научни изследвания“ на ПУ „Паисий Хилендарски“.

Публикации по темата на дисертационния труд

1. Тотков Г., С. Енков, **М. Близнаков**, „Достъпен университетски портал за лица със специални образователни потребности“, Научни трудове на Съюза на учените в България–Пловдив, серия Б. Естествени и хуманитарни науки, т. XVI. Научна сесия „Техника и технологии, естествени и хуманитарни науки“, 2013, ISSN 1311-9192, стр. 95 – 98.
2. Алendarова Е., Р. Донева, **М. Близнаков**, Г. Тотков, За качеството на университетското образование и българските системи от тип „Алumni“, VII национална научна конференция за студенти, докторанти и млади учени, 27 май 2017 г., гр. Пловдив, стр. 71 – 76, ISSN:1314-9547.
3. **Близнаков М.**, Г. Тотков, Г. Пашев, Генератор на справки в университетска информационна система, научна сесия "Дни на науката 2018", Съюз на учените в България-Пловдив, 2 - 3 ноември, Пловдив, Серия В: Техника и технологии 17 (2018): стр. 92 – 95, 2018 г.
4. **Близнаков М.**, Г. Пашев, Г. Тотков, „Софтуерни инструменти за визуализация на динамични информационни ресурси“, 12-та национална конференция с международно участие „Образованието и изследванията в информационното общество“ Пловдив, май 2019 г., стр. 32 – 41, ISSN 2534-8663.
5. Doneva R., S. Gaftandzhieva, **M. Bliznakov**, Stanislava Bandeva. Learning Analytics Software Tool Supporting Decision Making in Higher Education., International Journal on Information Technologies and Security, No.2 (vol. 12), 2020, pp. 37 – 46, <http://ijits-bg.com/ijitsarchive>, Web of Science.
6. Gaftandzhieva S., R. Doneva, **M. Bliznakov**, Internal and External QA in HE: LA Tools and Self-Evaluation Report Preparation, International Journal of Emerging Technologies in Learning (iJET), Vol 15, No 16, 2020, pp. 191 – 199, SCOPUS (SJR=0.219), <https://online-journals.org/index.php/i-jet/article/view/14401/7715>.

Списък на забелязани цитирания

1. Doneva R., S. Gaftandzhieva, **M. Bliznakov**, Stanislava Bandeva. Learning Analytics Software Tool Supporting Decision Making in Higher Education., International Journal on Information Technologies and Security, No.2 (vol. 12), 2020, pp. 37 – 46, <http://ijits-bg.com/ijitsarchive>, Web of Science.
 - Kobets, V., Liubchenko, V., Popovych, I., & Koval, S. (2021, June). Institutional Aspects of Integrated Quality Assurance of Engineering Study Programs at HEI Using ICT. In Design, Simulation, Manufacturing: The Innovation Exchange (pp. 301 – 310). Springer, Cham.

Библиография

- [Близнаков'18] Близнаков, М., Г. Тотков, Г. Пашев, Генератор на справки в университетска информационна система, Съюз на учените в България-Пловдив, 2 – 3 ноември, Пловдив, Серия В: Техника и технологии 17 (2018): 92 – 95.
- [Близнаков'19] Близнаков, М., Г. Пашев, Г. Тотков, Софтуерни инструменти за визуализация на динамични информационни ресурси, 12-та национална конференция с международно участие „Образованието и изследванията в информационното общество“ Пловдив, май 2019 г., стр. 32 – 41, ISSN 2534-8663.
- [Енков'13] Енков С., Методика и средства за осигуряване на интернет достъпност за лица със специални образователни потребности, Дисертационен труд за присъждане на образователна и научна степен „доктор“, 2013, Пловдив.
- [Ножнов'09] Ножнов, В. А. "Модель учебного курса." Сборник трудов Международной научно-практической конференции ИТО-2009. 2009.
- [ОтчетКСК'21] Стоянова М., А. Илиев, Отчет за КСК'2021, <https://uniplovdiv.bg/uploads/site/pravilnici/vpravilnici/Otchet%20KSK%202021-end.pdf>, посл. посетен на 15.12.2021 год.
- [Пиктограмм'19] Созданием пиктограмм и визуальным дизайном интерфейсов - Турбомилк, <http://turbomilk.ru/>, посл. посетен на 12.04.2019 год.
- [Платон'81] Платон; Р. Радев, превод: Ал. Милев, Държавата, ISBN -COBISS.BG-ID – 1097255396, Издание – 2, Държавен полиграфически комбинат „Димитър Благоев“ – София, 1981.
- [Тотков'01] Тотков, Г., Вл. Шкуртов, Р. Донева, Основи на компютърната информатика, Университетско издателство, ПУ, Пловдив, стр. 35 – 40, 2001.
- [Тотков'13] Тотков Г., С. Енков, М. Близнаков, „Достъпен университетски портал за лица със специални образователни потребности“, Научни трудове на Съюза на учените в България–Пловдив, серия Б. Естествени и хуманитарни науки, т.XVI. Научна сесия „Техника и технологии, естествени и хуманитарни науки“, 2013, ISSN 1311-9192, pp. 95 – 98.
- [Тотков'14] Тотков и др., Пловдивски е-университет, „Ракурси“, Пловдив, 2014.
- [Abdelzad'15] Abdelzad V., D. Amyot(B), T. C. Lethbridge, Adding a Textual Syntax to an Existing Graphical Modeling Language: Experience Report with GRL, Springer International Publishing Switzerland 2015, J. Fischer et al. (Eds.): SDL 2015, LNCS 9369, pp. 159 – 174, 2015. DOI: 10.1007/978-3-319-24912-4 12.
- [Axios'20] Axios – Promise based HTTP client for the browser and node.js, <https://github.com/axios/axios>, посл. посетен на 09.03.2021 год.
- [Bertin'83] Bertin, J., Semiology of Graphics, University of Wisconsin Press, 1967/1983, ISBN 978-1-58948-261-6.
- [Booch'05] Booch G., James Rumbaugh, Ivar Jacobson, Unified Modeling Language User Guide, 2nd Edition, Addison-Wesley Professional, ISBN-13: 978-0-321-26797-9, May, 2005.
- [Burger'13] Burger, E. J. Flexible views for view-based model-driven development. In Proceedings of the 18th international doctoral symposium on Components and architecture (pp. 25-30), 17 June 2013.

- [Card'08] Card, S., Information visualization, The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Lawrence Erlbaum Assoc Inc, p.539-542, ISBN: 978-0-8058-5870-9, 2008.
- [Chen'76] Chen, P., The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems. Vol.1 (1), pp. 9–36. CiteSeerX 10.1.1.523.6679. doi:10.1145/320434.320440, March 1976.
- [D3'16] D3 – Data Driven Documents, <https://github.com/d3/d3/wiki>, посл. посетен на 15.11.2018 год.
- [DeMarko'79] DeMarko, T., Structured Analysis and System Specification. Prentice Hall, ISBN 0-13-854380-1, 1979.
- [Doneva'20] Doneva, R., Gaftandzhieva, S., Bliznakov, M., Bandeva, S. Learning Analytics Software Tool Supporting Decision Making in Higher Education., International Journal on Information Technologies and Security, No.2 (vol. 12), 2020, pp. 37-46
- [Euler'68] Euler L., Lettres à une princesse d'Allemagne sur divers sujets de physique et de philosophie [Letters to a German Princess on various physical and philosophical subjects] (Saint Petersburg, Russia: l'Academie Impériale des Sciences, 1768), volume 2, p. 95 – 126, https://books.google.bg/books?id=gxsAAAAQAAJ&pg=PA95&redir_esc=y#v=onepage&q&f=false, посл. посетен на 07.03.2019 год.
- [Friendly'09] Friendly, M., Milestones in the history of thematic cartography, statistical graphics, and data visualization, 2009.
- [Gaftandzhieva'20] Gaftandzhieva S., R. Doneva, M. Bliznakov, Internal and External QA in HE: LA Tools and Self-Evaluation Report Preparation, International Journal of Emerging Technologies in Learning (iJET), Vol 15, No 16, 2020, pp. 191-199.
- [Hui'12] Hui, Y., What is a digital object?, Metaphilosophy LLC and Blackwell Publishing Ltd Published by Blackwell Publishing Ltd, 9600 Garsington Road, pp. 380-395, Vol. 43, No. 4, July 2012.
- [Horton'94] Horton W., The Icon Book: Visual Symbols for Computer Systems and Documentation, 1st Edition, Wiley Technical Communication Library, ISBN: 978-0471599005, 1994.
- [Jackson'75] Jackson, M. A., Principles of Program Design, Academic Press, 1975.
- [Jsreports'20] Jsreports – JavaScript library for designing, editing, viewing, and exporting printable documents, <https://www.jsreports.com>, посл. посетен на 27.01.2021 г.
- [Kahn'06] Kahn R., R. Wilensky, A framework for distributed digital object services International Journal on Digital Libraries (2006) 6(2): 115 – 123, DOI 10.1007/s00799-005-0128-x, 2006.
- [Kallinikos'10] Kallinikos J.; A. Aaltonen; A. Marton, A theory of digital objects, First Monday, Peer-reviewed journal of the internet, Volume 15, Number 6 – 7 June 2010.
- [Keim'06] Keim, D.; F. Mansmann; J. Schneidewind; H. Ziegler, Challenges in Visual Data Analysis, Information Visualization (IV 2006), IEEE, pp. 9-16, 2006.
- [Marca'87] Marca D., C. McGowan, Structured Analysis and Design Technique, McGraw-Hill, 1987, ISBN 0-07-040235-3.
- [Mauri'17] Mauri, M., Elli, T., Caviglia, G., Ubaldi, G., & Azzi, M. (2017, September). RAWGraphs: a visualization platform to create open outputs. In Proceedings of the 12th biannual conference on Italian SIGCHI chapter (pp. 1-5).

- [McCandless'10] McCandless D., The beauty of data visualization, https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization, 2010, посл.посетен на 15.01.2019 год.
- [Myers'76] Myers, G.I., Software Reliability: Principles & Practices, Wiley, 1976.
- [Nadin'90] Nadin, M., Design und Semiotics, In Semiotics in the Individual Sciences, Vol II. 418-436. Koch, W. A.(ed). Bochum: Brockmeyer, 1990.
- [Nöth'95] Nöth, W.. Handbook of semiotics. Indiana University Press Bloomington and Indianapolis, p. 13 – 50, 1995
- [Peirce'09] Peirce, C., Reflections on Real and Unreal Objects, 1909, <http://www.commens.org/dictionary/entry/quote-reflections-real-and-unreal-objects-ср>. посл. посетен на 19.03.2019 год.
- [Quine'60] Quine, W. V. O., Word and Object, Massachusetts Institute of Technology, p.14, 1960.
- [Ross'77] Ross, D. T., Structured Analysis (SA): A Language for Communicating Ideas, IEEE Transactions on Software Engineering, Vol. SE-3, No. 1, pp. 16 – 34, January 1977.
- [Saidis'07] Saidis K.; A. Delis, Type-consistent Digital Objects, D-Lib Magazine, May/June 2007, Volume 13 Number 5/6, ISSN 1082-9873.
- [Saussure'83] Saussure, F., Harris, Roy, Translator. Course in General Linguistics. La Salle, Illinois, Open Court, 1983, p. 32, [1913]. ISBN 0-8126-9023-0.
- [Stevens'74] Stevens W. P., G. J. Myers, L. L. Constantine, Structured Design, IBM Systems Journal Volume 13, Number 2, 1974.
- [Venn'80] J. Venn M.A. (1880) I. On the diagrammatic and mechanical representation of propositions and reasonings , Philosophical Magazine Series 5, 10:59, 1-18, DOI: 10.1080/14786448008626877, 2009.
- [VueJS'20] Vue.js – The Progressive JavaScript Framework, <https://vuejs.org/>, посл. посетен на 14.10.2020 год.
- [WAI'18] Web Accessibility Initiative, Making the Web Accessible, <https://www.w3.org/WAI/>, посл. посетен на 12.03.2019 год.
- [Ward'02] Ward, M. O., A taxonomy of glyph placement strategies for multidimensional data visualization. Information Visualization, 1(3-4), pp. 194 – 210, 2002.
- [Ware'13] Ware, C., Information visualization: perception for design – 3rd [edition], 2013, ISBN 978-0-12-381464-7.
- [Warnier'75] Warnier, Jean-Dominique, Logical Construction of Programs, Springer, 1 edition, ISBN: 978-9020704518, 1975.
- [WCAG'18] Web Content Accessibility Guidelines (WCAG), <https://www.w3.org/TR/WCAG/>, посл. посетен на 12.03.2019 год.
- [Yourdon'79] Yourdon, E., Constantine, Larry L., Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Yourdon Press, ISBN 978-0-13-854471-3, 1979.
- [Zloof'77] Zloof, Moshe M., Query-by-example: A data base language, IBM systems Journal 16.4 (1977), pp. 324 – 343.