

**МИННО-ГЕОЛОЖКИ УНИВЕРСИТЕТ – София**

**инж. Николай Иванов Янев**

Методология и подход за разработване на web  
базирана информационна система

## **АВТОРЕФЕРАТ**

на дисертация

За получаване на образователна и научна степен "доктор"

Специалност: 01.01.12 "Информатика"

Професионално направление: 4.6 "Информатика и компютърни науки"

Научен ръководител: доц. д-р Волин Карагъзов

София, 2011г.

Дисертационният труд е обсъден и насрочен за защита от разширен катедрен съвет на катедра “Информатика” при Минно-геоложки университет “Св. Иван Рилски” – София.

Данни за дисертационния труд:

- страници – 127;
- фигури – 34;
- таблици – 3;
- цитирани литературни източници – 121;
- публикации, свързани с темата на дисертацията – 8.

Номерацията на таблиците и фигурите в автореферата съответства на номерацията в дисертационния труд.

Защитата на дисертацията ще се състои на .....  
от..... часа в зала ..... на МГУ – София на  
заседание на Научното жури.

*Автор:* Николай Иванов Янев

*Заглавие:* *Методология и подход за разработване на web базирана  
информационна система*

*Тираж:* 20 броя

*Печат:* Издателска къща „Св. Иван Рилски” при МГУ - София

# **I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД**

## **Увод**

Създаването на софтуер за минното дело следва логиката на пазарните принципи. Спецификата и сложността на решаваните в тази област задачи изисква високо специализирани приложения.

В списъка на множеството съвременни проблеми, съпътстващи еволюцията на информационните системи (ИС), съществено място заема аспектът на избор на инструментариум и подходът при реализацията, поддържането и развитието на съответната ИС. Анализът на този проблем е една от важните задачи, на които е посветена настоящата дисертационна работа.

За автора на настоящия дисертационен труд приложенията в областта на минното дело са естествен обект за изследване, поради значителния професионален стаж в обкръжение на съответните специалисти.

За целите на минната индустрия няма разработен специален стандарт на инструментариум за изграждане на ИС. Това налага комбиниране на стандартите и инструментариума с оглед на решаваната задача, което е една сложна за осъществяване дейност, изискваща умения, творчество и нерядко сериозни научни изследвания.

## **Същност, цели и задачи на дисертационния труд**

Целта на научния труд е да предложи, опише, реализира и валидира методология за разработване на web базирана ИС в минното дело, а така също и с общо предназначение.

Дефинирани са следните задачи за постигане на целта:

1. Да се анализират възможностите (технологии и инструментариум) за реализиране на съвременна, web базирана ИС;

2. Да се анализират популярните методологии за изграждане на ИС;
3. Да се разработи методология за реализиране на web базирани ИС, ориентирана към потребностите на минната индустрия;
4. Да се документира подход при прилагане на методологията;
5. Да се валидира предложената методология, чрез примери за приложението ѝ.

## **Публикуване на резултатите от дисертационното изследване**

Основните теоретични и експериментални резултати на дисертационния труд са изложени в 8 публикации – 7 на научни сесии и конференции и 1 в списание.

## **Структура и обем на работата**

Дисертационният труд е в обем 127 страници, съдържащ текст, 34 фигури, 3 таблици, списък с използваните литературни източници и сайтове.

## **II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД**

### **Глава 1: Развитие на технологиите за съхранение и обработка на информацията**

В глава 1 е направен обзор на технологиите за съхранение и обработка на информацията. Акцентира се върху релационния модел данни, базираните на него ИС с web интерфейс и системите за подпомагане вземането на решения. Разгледани са и програмните инструменти за разработване на web интерфейс и управление на данни.

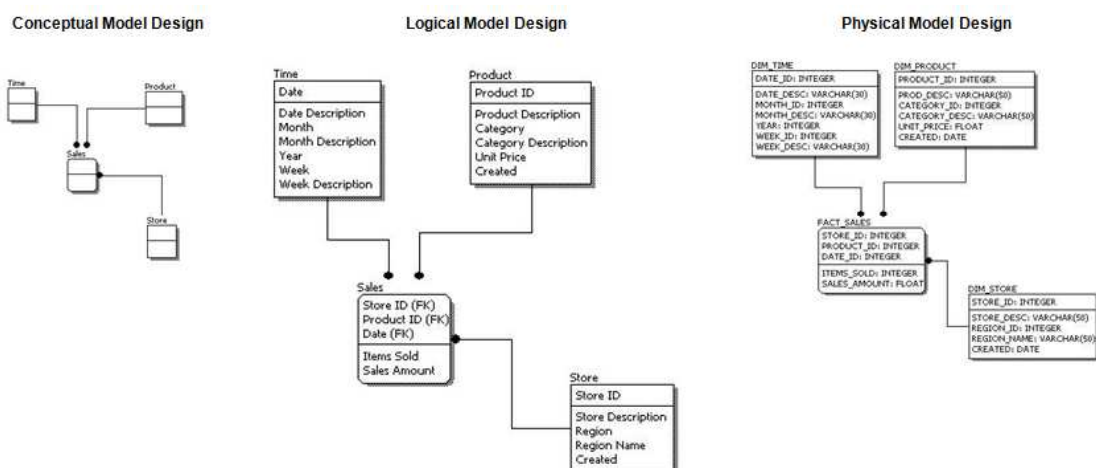
#### **1.1 Бази данни**

Базата данни (БД) може да се разглежда като съвкупност от взаимосвързани данни, организирани и представени по начин, който

позволява използването им от различни потребители за удовлетворяване на различни информационни нужди на дадена организация.

Разглеждат се три нива на абстракция (фиг. 1.2) при разработване на модел на данните в една БД:

- външен модел - отразява информационните нужди. На тази база се определя структурата на данните, бизнес правилата и операциите, които могат да бъдат изпълнявани върху тези данни;
- концептуален модел – основното му предназначение е детайлно изясняване на структурата, семантиката, връзките и ограниченията в БД, независимо от система за управление на база данни (СУБД), която ще се използва за реализирането ѝ и статично описание на съдържанието на БД. Най-популярните такива модели са Entity-Relationship (ER) и Enhanced ER [97];
- вътрешен (логически) модел - създава се като се преобразува концептуалният модел според изискванията и ограниченията на конкретна СУБД. Съвременните СУБД се базират на релационния или негови производни модели, като например обектно-релационния модел.



Фиг 1.2 - Етапи при разработване на БД

*Релационен модел* - предоставя висока степен на независимост на данните, тъй като при него се отстранява необходимостта от познаване на вътрешното представяне на данните и начина на достъп до тях. Предимство на релационния модел е и еднообразното представяне на класовете от обекти и връзките между тях.

## 1.2 Web базирани бази данни

### 1.2.1 Сървъри за бази данни

Основните принципи на технологията "клиент-сървър" от гледна точка на БД могат да бъдат разделени в 5 групи [17]:

1. функции на въвеждане и представяне на данните (Presentation Logic);
2. функции, определящи основните алгоритми за решение на задачите на приложението (Business Logic);
3. функции обработващи данните в приложението (Database Logic);
4. функции управляващи информационните ресурси (Database Manager System);
5. служебни функции, играещи роля на свръзка между функциите от изброените групи.

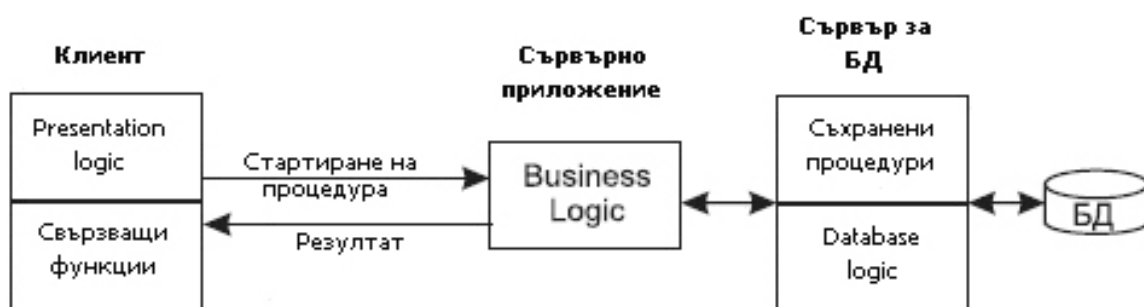
Моделът на сървъра за бази данни е представен на фиг. 1.5.



Фиг. 1.5 - Модел на активен сървър за БД

Преимущество на модела е, че трафика на обмена на информация между клиента и сървъра рязко намалява. Недостатък е голямото натоварване на сървъра.

С цел намаляване на натоварването на сървъра е предложен три слоен модел - модел на сървърно приложение. В него се въвежда допълнително междинно ниво между клиента и сървъра. То съдържа едно или няколко сървърни приложения. Архитектурата на три слойния модел е показана на фиг. 1.6.



Фиг. 1.6 - Модел сървърно приложение

Този модел притежава по-голяма гъвкавост, в сравнение с двуслойния модел. Най-значимите му преимущества са в случаите когато клиенти изпълняват сложни аналитични изчисления над бази данни, отнасящи се до областта на OLAP-приложенията.

При използване на СУБД с мултипроцесорни платформи става възможно стартирането на няколко сървъра за БД на различни процесори, като всеки от сървърите е многопотокен. При оперативна обработка на информацията в хранилища за данни този подход е най-перспективен.

## 1.2.2 Програмни инструменти за реализиране на web интерфейс към БД

### 1.2.2.1 Вградени (комерсиални) инструменти

Комерсиалните СУБД предоставят модули с добре развити графични интерфейси за разработване на ИС, Data Warehouse, OLAP и дори Data Mining приложения. Тези производители на СУБД са разработили собствени езици за програмиране базирани на SQL. В Microsoft SQL Server, е реализиран Transact/SQL, а в Oracle - PL/SQL.

### 1.2.2.2 Описателни езици

Описателните езици (markup language) служат за описване на данни, като чрез някои от тях дори могат да се определят начина и стъпките, необходими за представянето на тези данни.

В ИС езиците за маркиране се използват за:

- реализиране на web интерфейс - HTML, XML и XML базирани езици, най-често XHTML и SVG. Съвременни тенденции са свързани с две конкуриращи се спецификации, в процес на разработване на мястото на старите HTML 4.x и XHTML 1.x стандарти - HTML 5 и XHTML 2.0;
- съхранение на данни – XML;
- пренос на данни - XML и XML базирани езици, за целите на минната индустрия се използват GML, XMML, IREDES и SVG.

### 1.2.2.3 Скриптов езици

Нарастването на съхраняваните в www мрежата мултимедийни данни доведе до появата на интерфейси - CGI (Common Gateway Interface) и APIs (Application Programming Interfaces), предназначени да обслужват потребителските заявки, свързани с търсене на определени данни във



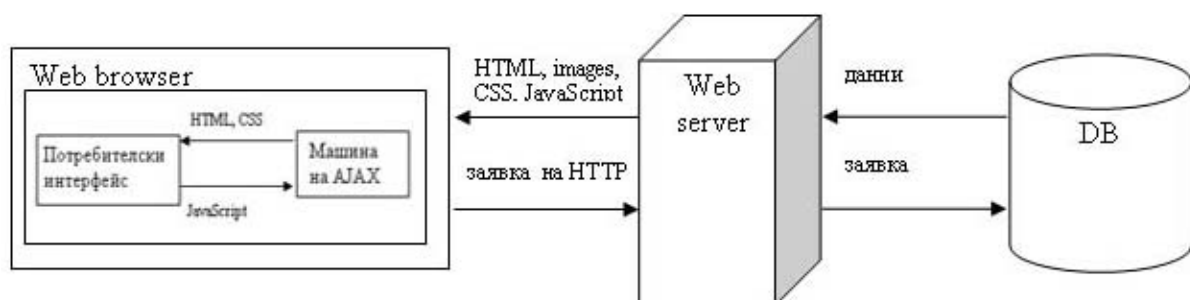
файловата структура на Web сървърите, предоставяне, чрез Интернет, достъп до БД, за обработка и систематизиране на изпращаната от Web сървъра към клиента информация.

Съществен недостатък на CGI е, че става дума за самостоятелни приложения - всеки път когато потребител се обърне към скрипта се заделя ново място в паметта на сървъра и се изпълнява ново копие на програмата. Това води до разход на много памет и процесорно време от страна на сървъра.

Проблемите на CGI се решават с използване на сървърни APIs. Тези приложения се вграждат в самия сървър, което означава че използват едно и също място в паметта и най-важното: при всяко ново потребителско обръщение към скрипт, който е свързан с даденото приложение, се създава само нова нишка в сървърния процес.

Традиционните web приложения, за да се свържат със сървъра, използват HTML форма. Така свързването става след презареждането на страницата. Поради тази причина обикновените web страници са по-бавни и с по-малко възможности в сравнение с тези използващи асинхронна комуникация със сървъра [78].

За реализиране на асинхронна комуникация се използва AJAX (Asynchronous JavaScript and XML) (фиг. 1.9).



Фиг. 1.9 - Модел на AJAX

Недостатъците на AJAX са:

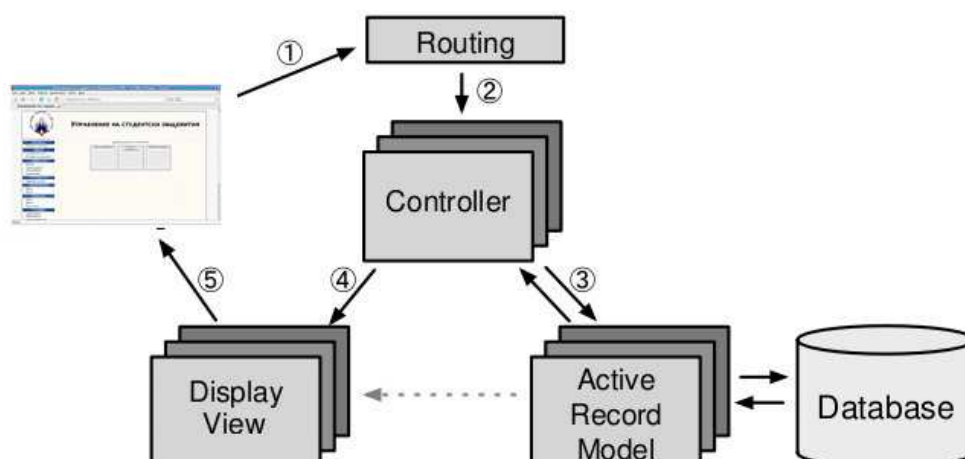
- Неочаквано поведение - проблеми със забавянето на отговора създава усещане, че приложението не работи;
- Браузърът трябва да разрешава изпълнението на JavaScript;
- Различна имплементацията на JavaScript в различните браузъри.

#### 1.2.2.4 Framework

За улесняване на разработчиците на web приложения много програмисти започнаха да публикуват свои библиотеки с готови класове и функции. Така се стигна до разработването на нови архитектури, даващи възможност за реализиране на сложни решения с използването на прости структури – framework.

Съществува голямо разнообразие от framework софтуер. Повечето такива разработки се базират на популярен език за програмиране като Ruby, PHP, ASP.net, Java, Python и др. По отношение на възможностите им най-съществените разлики са свързани с вградените им функционалности за работа с Object-relation mapping (ORM) и събитийно ориентирано програмиране (EDP).

Целта на ORM (в Ruby on Rails - ActiveRecord) е да се улесни и ускори начина на разработване на web приложенията с бази данни (фиг. 1.11).



Фиг. 1.11 – MVC и ORM (ActiveRecord)

### 1.2.3 XML бази данни

XML документите могат да бъдат съхранявани в текстови файлове, XML хранилища или БД. Има две основни причини, поради които много компании предлагат те да бъдат съхранявани в БД [18]:

- Управлението на големи количества XML данни;
- Интеграция – съхраняване на релационни данни и XML документи заедно.

Три са основните типа XML бази данни:

*XML Enabled Database* - използват релационния модел за съхранение на данните. Това изисква средства за регламентиране на съответствие между XML (йерархичният модел на данните) и релационния модел на данните;

*Native XML бази от данни* - Native XML базите от данни използват йерархичния XML модел за вътрешно съхраняване и обработка на XML данни. По този начин не се налага преобразуване към релационен модел;

*Hybrid XML Database (HXD)* - при тях данните могат да бъдат третирани и като Native XML Database и като XML Enabled Database.

## 1.3 Системи за подпомагане вземането на решения

### 1.3.1 Общи сведения

Управленските ИС и системите обработващи транзакции имат редица ограничения. Системите обработващи транзакции управляват стандартни процеси и съхраняват данни за тях, но не могат да ги анализират. Управленските ИС осигуряват отчети за управлението, но от тях не може да се извлече информация за ефективното им използване от мениджърите. С цел да се избегнат тази недостатъци,

през 70-те години на миналия век, започнаха да се разработват системи подпомагащи вземането на решения.

Основна им характеристика е, че те осигуряват средства за анализ на данните, изграждане на модели, оценяване на алтернативи. Анализите получени чрез тях осигуряват ясна обосновка за обясняването на дадено решение. Подобни решения обикновено са уникални за всяка ситуация, за тях няма ясна процедура, променят се бързо, факторите за оценката им не могат да се определят лесно предварително и се вземат сравнително рядко.

Техниките, които се използват в тях, за подпомагане вземането на решения биват:

- Ориентирани към използването на модел – симулация, оптимизация;
- Ориентирани към използването на данни – OLAP, data mining;
- От областта на изкуствения интелект – експертни системи, невронни мрежи, размита логика, интелигентни агенти.

### *1.3.2 Хранилище за данни (Data Warehouse)*

Data Warehousing (DW) възниква като ново понятие, отразяващо приложението на съвременните информационни технологии за преобразуване на огромните масиви от данни в значима информация, която да подпомага вземането на бизнес решения. Този процес е стратегически важен за обединението на различните информационни източници, с оглед по-доброто им използване за аналитична интерактивна обработка (Online Analytical Processing - OLAP).

Днес основно се прилага стандарт за изграждане на хранилище, при който базата е автономна и релационна, като данните се представят по

различни начини и могат да се разглеждат с голямо разнообразие от средства – от OLAP до Data Mining софтуер.

### *1.3.3 Качество на данните (Data Quality)*

За да бъде информацията в DW коректна, данните постъпващи в склада трябва да бъдат „качествени“.

Съществуват множество теоретични рамки за разбирането на качеството на данните [20] [21] [23] [27]. Почти 200 принципа на качество са описани от различни автори [34] [45] [48] [49], но липсва единна класификация. Тези принципи са ангажирани във всички етапи на процеса за управление на данните - улавяне, цифровизацията, съхранение, анализ, представяне и използване. Загубата на качеството на данните във всеки един етап намалява приложимостта им. Всички те имат своя принос към крайното качество или "годност за използване" на данните и се прилагат и към всички аспекти на данни.

### *1.3.4 OLAP технологии*

Релационният модел е предпочитан за съхранение, актуализация и извличане на конкретни данни от огромни по обем бази. Същевременно обаче, дори и организиран като склад от данни, той не позволява бързо и лесно структуриране, анализиране и отсяване на съществената информация. За постигането на тази цел се използва оперативната аналитична обработка в реално време - OLAP.

Обикновено тази технология включва средства за интерактивен анализ на данните, които се извличат от различни бази и се обобщават за нуждите на даден потребител. OLAP средствата предоставят възможности за представяне на данни в различни разрези, без да ги

актуализира. Те са значително по-сложни от традиционните системи за обработване на транзакции.

Използването на OLAP съвместно с хранилища от данни, предлага мощен и гъвкав метод, съчетаващ предимствата на двата подхода.

### 1.3.5 Data Mining

Подобно на OLAP системите, софтуерът от тип Data Mining (DM) се използва също за анализ на данните, но обхваща технологии, позволяващи да се откриват в „сурови“ данни по-рано неизвестни, нетривиални, практически полезни и достъпни интерпретации на знания, необходими за вземане на решения в различни сфери на човешката дейност.

Задачите, които DM решава могат да се класифицират като [104]:

*Descriptive* – описващи основните характеристики на данните в БД;

*Predictive* – извършващи изводи, на база актуалните данни, вследствие на които се правят прогнози

DM анализа намира приложение в области, където не са достатъчни само статистически и аналитични методи за изграждане на подходящите модели. В тези области преобладават нееднородни, хетерогенни, променливи и в големи количества данни. Това става при всички компании, които извършват обработка на данни при осъществяването на ежедневните си дейности и операции.

### 1.3.6 Системи с изкуствен интелект

Най-популярните типове системи с изкуствен интелект са невронните мрежи (НМ) и експертните системи (ЕС). При НМ прогнозата се формира без участие на човека, а при ЕС се включват един или няколко специалиста от висок клас като елементи. Друга разлика е, че НМ работи на

принципа на предаване на информация от едни слой неврони към друг и изменение на информацията по време на предаването ѝ, на база предварително обусловени евристически правила, а в ЕС съществува строг логически скелет - линията на разсъждение е предварително заложена, като алгоритми, правила и използвани параметри.

### *1.3.7 Enterprise Resource Planning*

Enterprise Resource Planning (ERP) системата е компютърно интегрирана, многомерна и многофункционална система, изградена на база клиент-сървър технология, основана на бизнес – модела за планиране, контрол и глобална оптимизация на цялостната логистична верига на процесите по доставката, производството и пласмента на готовите изделия и съпътстващите ги финансови операции.

## **1.4 Езици за описание на данни**

### *1.4.1 Релационни езици*

Релационните езици са езици за описание и обработка на данни в релационните БД. Те се делят на две основни категории – езици на релационната алгебра и езици на релационното смятане.

Освен двете категории езици съществуват и т.нар. *междинни релационни езици*, които имат възможности, присъщи на езиците на релационното смятане и релационната алгебра. Най-разпространения такъв език е Structured Query Language (SQL).

### *1.4.2 Structured Query Language*

За разлика от релационната алгебра, където могат да се дефинират само заявки за справки, SQL е пълен език за програмиране. Освен операторите за дефиниране на заявки, той обхваща и оператори на език

за описание на данните, както и оператори за управление и администриране на БД.

SQL не поддържа контрол на потока и други възможности за създаване на комплексни приложения. За да бъде използван SQL като език за програмиране трябва да се разширят неговите възможности. Това може да бъде направено по някои от следните начини:

- Вграждане на твърдо кодирани SQL конструкции в програма написана на друг език – вграден SQL (Embedded SQL);
- Създаване на програми, които да генерират SQL конструкции по време на изпълнението си – динамичен SQL (Dynamic SQL);
- Използване на SQL99;
- Модулен език - Използване на модули, които да бъдат извиквани от програми написани на други езици. Дава същите резултати както при Embedded SQL. Предимство е, че няколко приложения могат да получат достъп до един и същ модул;
- SQL/XML - разширение на SQL 2006 за обработване на XML документи.

#### *1.4.3 Data Mining езици*

Data Mining езиците следва да отговорят на следните очаквания [79]:

- Да осигурят възможности за избор и предварителна обработка на данни от различни източници;
- Възможност за работа с шаблони;
- Върху използваните модели трябва да могат да бъдат налагани ограничения;
- Да се проследяват процесите след прилагането на шаблона;
- Резултата да бъде четим и лесен за извличане.



Най-популярните Data Mining езици са MSOL, DMOL, MINE RULE и OLE DB for DM. Всички те се базират на SQL. Анализ и сравнение на тези езици е направен от Botta [80].

Друг език за Data Mining е PMML (Predictive Model Markup Language). Той е XML базиран език за описание на статистически, data mining модели и изкуствени невронни мрежи [81].

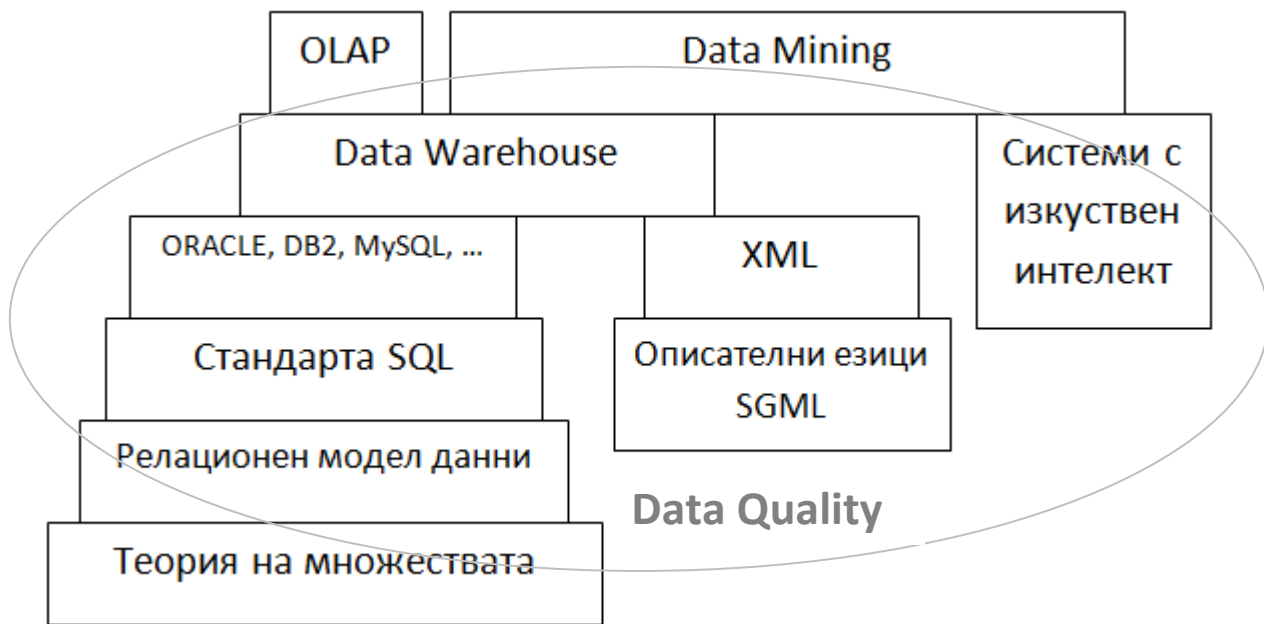
#### *1.4.4 Езици за XML данни*

За извличане на XML данни се използва езика XQuery. Той се базира на системата типове XML Schema и е съвместим със стандартите XSL и XPath, свързани с XML. XQuery позволява формирането на изрази за обхождане на йерархическата структура на XML документа, от друга страна „чистия“ SQL (без XML разширение) не допуска такава възможност. Чрез използване на SQL/XML става възможно вграждането на XQuery изрази в SQL заявки.

### **1.5 Изводи**

В областта на информационните технологии през последните години се забелязва тенденция за изместване на интереса от увеличаване на мощността на изчислителните средства към подобряване на Интернет съвместимостта им и повишаване на тяхната интелигентност, т.е. към създаване на нови подходи основани върху знания даващи възможност за по-прецизен и задълбочен анализ на данни, откриване на скрити зависимости, възможност за on-line режим на работа и т.н.

Съвременните тенденции са отразени на фиг. 1.16, която представлява разширение на фиг. 1.3.



Фиг. 1.16 Съвременни технологии за съхранение и обработка на данните

Наличието на подобни системи води до следните предимства за ползвателите им:

- Позволява на потребителите им (дори и такива, не притежаващи много висока квалификация) да могат да решават успешно текущите си задачи, по начин по който го биха сторили експерти;
- Осъществяване на достъп до данни без значение на географското им местоположение;
- Подобрява качеството на данните.

Приложено за бизнес цели това означава следното:

- Осъществяване на разпределена обработки на данни;
- Създаване на развити системи за комуникация;
- Отстраняване на пречките в системите за интеграция на "организация - външна среда", пряк достъп до световните информационни потоци;
- Създаване и развитие на системи за електронни поръчки и търговия;
- Поддръжка на социални мрежи.

## **Глава 2: Описание на методология и подход за разработване на web базирани ИС**

Тази глава описва най-съществената част от научния труд. Тук са предложени методология и подход за разработване на web базирана ИС. Разгледани са най-популярните методологии за разработка на софтуер и е представена мотивацията за създаване на нова методология.

### ***2.1 Методологични аспекти при разработката на софтуерни приложения***

Методологията е изключително важна за изграждането на ефективно софтуерно приложение. Тя се изразява в избор на подходящ подход и използването на удачна комбинация от модел на жизнения цикъл, схема за моделиране и съответните модели. Тази комбинация от своя страна определя какви техники и средства ще се използват по време на процеса на изграждане на софтуерното приложение.

Обикновено към методологиите се предявяват следните изисквания:

- Съкращаване на времето за разработка;
- Намаляване на изискванията за квалификация от страна на разработчици и потребители;
- Контрол на процесите на разработка и идентифициране на резултатите.

През годините са предлагани множество методологии за изграждане на софтуерни приложения всяка със своите силни и слаби страни. В практиката не се е наложила универсална методология, подходяща за всякакъв вид проекти. Много често една методология се формира, за специфичен процес, базирани на конкретна технология, технически, организационни, проектни и екипни съображения.

## **2.2 Базови методологии**

Основните методологии за изграждане на софтуерни, в частност web ориентирани, приложения са:

- Водопад (Waterfall model)
- Спирален модел (Spiral model)
- Гъвкави методологии (Agile software development) – Scrum и Екстремно програмиране.
- Бърза разработка на приложения (Rapid application development – RAD)

## **2.3 Избор на методология за разработване на web базирана ИС**

Една добре подбрана (разработена) методология обръща внимание, както на организацията на данните, така и на информационните процеси. Проблем при използването на методологии е ниската степен на повторяемост при внедряването им. Много често в практиката е невъзможно да се използва “на готово” някоя от известните методологии. Тогава в зависимост от спецификата на решавания проблем се разработва подходяща методология или се модифицира съществуваща.

## **2.4 Методология за разработване на web базирана информационна система**

Изграждането на web приложение е сложен процес, включващ интеграция на множество иновационни технологии. Практиката при разработването на приложения с web интерфейс показва, че липсват ясно обособени правила и методи за оценка и подбор на приложимостта на програмни инструменти за изграждането им. Освен това съществуват множество технологии и архитектури, което прави избора на инструментариум още по-труден.

За адекватната работа на едно web приложение е изключително важно в процесите на проектиране на разработка да се отчете динамичното развитие на информационните технологии – нарастването на компютърната мощ и развитието на Интернет и съпътстващите го технологии.

Работа в многопотребителски режим (характерна за web средата) води до изисквания за високо ниво на сигурност и бързодействие. Следва да се вземе под внимание и типа web – *статичен, динамичен или семантичен*.

За бързодействието на ИС с web интерфейс е важен начина за разпределение на функциите между клиента и сървъра, както и прецизното изграждане на БД на концептуално, логическо и физическо ниво. При РБД най-често стремежът е към привеждане на данните в 3-та нормална форма. В зависимост от използваната технология обаче може да се окаже, че данни в по-ниска нормална форма биха повишили бързодействието на системата. В Ruby on Rails например се налагат ограничения върху структурата на БД (специфични имена на таблици, първични и външни ключове, първичните ключове да са от 1 колона и т.н.), които водят до автоматизиране на редица дейности при съставянето на модела и контролерите.

Дизайнът на едно приложение е от голямо значение. Той е “лицето” на разработката и до голяма степен на негова база масовия потребител формира първоначалното си мнение за системата. Дизайнът на приложението следва да предоставя динамична и интерактивна среда за работа. Информационната и графичната среда трябва да бъдат съобразени и улесняващи потребителя - "приятелски" настроена към неговите познания, намерения и потребности. Наличието на иновационни технологии (осъществяване на асинхронна връзка със

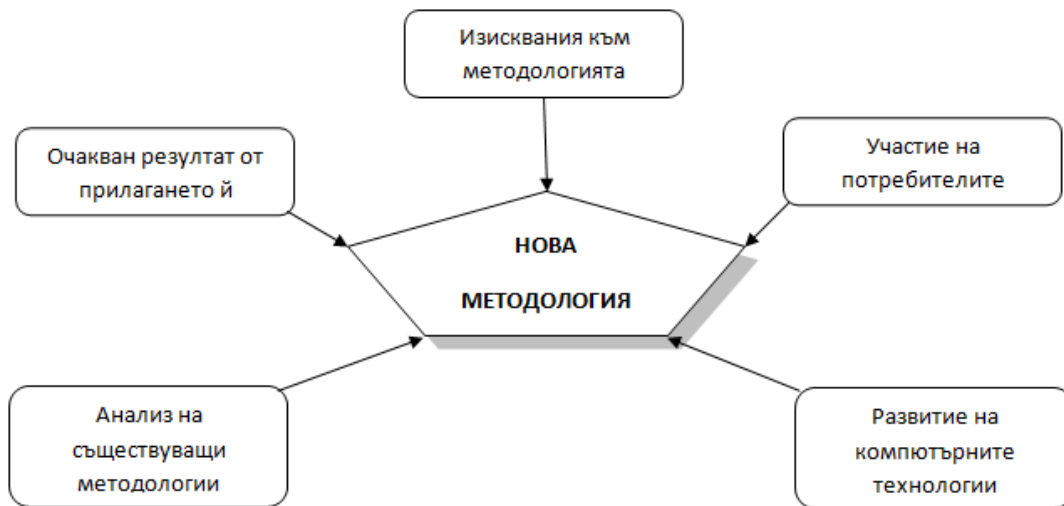
сървъра, използване на web анимация, инструменти за комуникация и обмен между потребители и т.н.), характерни за web 2.0 и 3.0, също се приема добре от потребителите. И в същото време web приложението трябва да се интерпретира по еднакъв начин поне от най-популярните версии на браузъри.

Инструментариумът за реализиране на web приложението трябва да отговори на няколко условия:

- Осигуряване на преносимост;
- Платформена независимост;
- Работа с възможно по-широк кръг СУБД;
- Възможност за работа с XML документи;
- Лесен синтаксис за постигане на възможно по-добра функционалност, включително и автоматизирано изпълнение на определени функции;
- Възможност за комбиниране с други технологии;
- Възможност за разширяване (еволюционно развитие) – с оглед динамичното развитие на web услугите и инструментите е важно системата да е адекватна на съвременните тенденции.

Стандартните методологии не отчитат спецификата на web приложенията. Това налага формулиране на методология подпомагаща изграждането на подобни системи.

Схема за изграждане на нова методология е представена на фиг. 2.8.



Фиг. 2.8 Създаване на нова методология

Предложената методологията се базира на RAD модела, като го разширява и до усъвършенства в две основни направления:

1. Комуникация с потребителите и тяхното обучение;
2. Интегриране на крайния продукт към вече функциониращи системи.

Целите пред методологията са:

- Да подпомогне изграждането на продукт удовлетворяващ информационните потребности на бъдещите му ползватели;
- Да осигури възможности за добавяне, редактиране и отстраняване на функционалности;
- Да осигури възможности за интегриране към други системи;
- Да осигури активно участие на потребителите във всички етапи на разработката;
- Да се намали цената на продукта, чрез приоритетно използване на софтуерни инструменти с отворен код;
- Да осигури възможно най-дълъг жизнен цикъл на резултантния продукт.

## Етапи на жизнения цикъл

- I. Изучаване и обследване на предметната област.
  1. Анализ на изследвания обект
  2. Установяване на текущите ограничения и проблеми
  3. Установяване на потребителските изисквания
  4. Ясно дефиниране на поставените цели и задачи от страна на възложителя
  5. Оценка на проекта
- II. Проектиране
  1. Извличане на работните данни
  2. Анализ на работните данни съвместно с експерти от съответната област
  3. Изследване и анализ на съществуващи ИС, работещи в съответна област
  4. Изграждане на концептуален модел на данните
  5. Описание на бизнес процесите в режим на непрекъснат диалог с възложителя и бъдещи ползватели на системата
  6. Създаване на модел на приложението
- III. Разработка
  1. Формулиране на изискванията за интеграция на приложението
  2. Преобразуване на концептуалния модел в релационен
  3. Определяне на структурата на приложението
  4. Описание на потребителския интерфейс, максимално близък до този на работещите до момента ИС
  5. Избор на тип сървър за БД
  6. Избор на СУБД
  7. Избор на програмни инструменти за реализиране на приложението
  8. Анализ на икономическата реализируемост



9. Създаване на прототип на приложението
10. Тестване на прототипа или модул
11. Оптимизиране на БД въз основа на резултатите от III.10
12. Изготвяне на документация

#### IV. Внедряване

1. Осигуряване на хардуер и софтуер
2. Извършване на предварителни настройки на базовия софтуер (операционна система, сървъри, компилатори и т.н.)
3. Инсталиране на системата
4. Запълване на системата с реални данни
5. Тестване
6. Изготвяне на документация за ползване на системата
7. Изготвяне на приемо-предавателен протокол

#### V. Обучение на персонала

1. Разделяне на обучаемите на групи съобразно дефинираните в системата потребителски групи
2. Оценка на компютърната грамотност на всяка група
3. Избор на тип обучение, съобразен с V.2, за всяка група
4. Провеждане на обучението
5. Оценка на нивото на придобиване на умения за работа със системата

#### VI. Поддръжка

1. Категоризиране на типовете грешки и конфликтни ситуации възникнали в процеса на работа
2. Отстраняване на грешки съобразно категорията (приоритета) им
3. Разпространение на нови версии на използваните програмни инструменти
4. Пренасяне на приложението на други платформи

## VII. Развитие

1. Изготвяне на препоръки за модифициране, съобразени с възникнали в хода на експлоатацията нужди и/или приложение на нови технологии
2. На база VII.1 пренаписване на съществуващи модули и/или добавяне на нови функционалности, следвайки гореописаните етапи

По-детайлно описание на някои от етапите:

*Дефиниране на задачата* – дефинирането на задачата е от съществено значение за последващите етапи за реализиране на приложението. Предефинирането на задачата, на следващ етап на разработката, обикновено е свързано със значителни промени по структурата и логиката ѝ. Това води до увеличаване на времето за изпълнение и вложения труд. На този етап трябва да бъдат описани данните с които ще работи ИС, тяхната същност, тип и размерност. Важен фактор на този етап са източниците на тези данни.

*Анализ на съществуващи ИС* – целта е да бъдат отстранени недостатъците и заимствани добрите практики от съществуващите ИС. При установяване на добра практика на потребителите на система с изчерпан жизнен цикъл, дизайнът на новата трябва да бъде максимално близък до него. По този начин ще се улесни прехода от старата към новата система и ще се намалят разходите за обучение.

*Избор на инструментариум* – изборът на инструментариум е свързан с типа на задачата, областта и степента на подготовка на екипа изпълнител, финансирането, а не рядко и с времето за изпълнение. За да се стигне да конкретен избор следва да се направи обстоен анализ

на практиките в областта, съществуващите програмни инструменти, архитектури и технологии.

От повечето ИС се очаква да се впишат (интегрират) към вече работещите такива или най-малко да могат да осъществяват Import/Export от и към други системи. Изключение до известна степен правят ERP приложенията предлагащи цялостно управление на предприятието.

От особена важност за коректната работа на ИС е да се анализират данните, с които ще работи системата. В следствие на този анализ се избира модела данни – мрежови, йерархичен, релационен, обектен и т.н. Трябва да се отчете и начина за достъп до ИС – сървърна или десктоп.

*Реализация* – реализацията се изразява в приложение на избрания инструментариум за постигане на дефинираните в задачата цели. На този етап се изгражда информационен модел, специфицира се диалога и се формират програмните модули.

На етапа на реализация е нужно да се разработят заложените в заданието функционалности, а и допълнителни такива свързани с:

- авторизиране на потребители с цел повишаване на нивото на сигурност на системата;
- удобен за потребителя интерфейс съобразен с нивото на подготовка на персонала, който ще използва системата;
- възможност за комуникация с външни ИС и приложения;
- осигуряване на възможност за надграждане (усъвършенстване) на системата.

*Експериментиране (апробация)* – на този етап разработената система се поставя в реални условия за определен период от време (зависещ от

отговорността на системата и практиката в съответната област) и се оценява поведението ѝ. В следствие на получените резултати е възможно работата по проекта да приключи с внедряване на системата или при незадоволителен резултат да се преразгледат етапи от реализацията, при нужда и избора на инструментариум. В етапа на апробация следва да участват експерти от съответната област с цел системата да бъде поставена във възможно по-голям брой ситуации, максимално доближаващи се до реалните условия.

Често в процеса на апробация на ИС, целящи да заменят остарели такива, се налага паралелната им работа за определен период от време. Друг важен аспект при апробацията на ИС е тестването им в режим на работа на няколко потребителя едновременно, включително и от една и съща група.

Добра практика е екипът разработил приложението да се грижи за поддръжката му за период от минимум 6 месеца, а също и провеждането на обучение на персонала, който ще експлоатира ИС.

*Участие на потребителите* в процесите на разработка и реализация на ИС. Дори и с риск от удължаване на процеса на вземане на решение, потребителите трябва активно да участват при:

- Определяне целите на проекта;
- Описание и анализ на работните данни;
- Описание и анализ на съществуващи ИС – техните функционални възможности, предимства и недостатъци;
- Анализ на алтернативни решения;
- Изготвяне на системната спецификация;
- Проектиране, планиране и провеждане на обучението.

*Качество на данните:* Някои от принципите на доброто управление на данните са:

1. Търсене на ефективност при събирането на данни и процедурите за качествен контрол
2. Обмен на данни, информация и инструменти, когато е възможно
3. Използване на съществуващи стандарти или разработване на нови, стабилни стандарти за връзка с други системи
4. Стимулиране на развитието на мрежи и партньорства
5. Съобразяване с добри бизнес практики за събиране на данни и управление
6. Намаляване на дублирането при събирането на данни и контрол на качеството на данните
7. Използване и изследване на нуждите на потребителите
8. Гарантирането на изчерпателна документация и процедури за събиране и обработка на метаданни
9. Графична визуализация на данните.

Тези принципи следва да се прилагат на всеки етап от жизнения цикъл на методологията.

## ***2.5 Критерии върху които се базира подхода***

Изследването на възможностите за усъвършенстване на ИС реализирани със свободно разпространявани инструменти, доведе до извеждане на следните характеристики при разработването им:

1. Финансови
  - a. Финансова целесъобразност на провежданото усъвършенстване;
  - b. Цена на нужния хардуер и софтуер.
2. Потребителски
  - a. Адекватност на потребителската среда;

- b. Възможности за обучение.
3. Приложни
- a. Подбор на функционално адекватни инструменти;
  - b. Ниво на владееене на инструментариума;
  - c. Тип на работещите системи.
4. Качество на данните
- a. Подобряване на достъпа до данните;
  - b. Подобряване на начина за представяне на данните.

## **2.6 Приложимост на методологията**

При традиционния подход на организация, когато специализираните функции се включват в работа една след друга, висока ефективност е недостижима. Бързото реагиране на външни изменения изисква сътрудничество между различни специализирани отдели и служби. Постоянно общуващи и обменящи информация, те могат да действат бързо, съгласувано и едновременно в различни направления. Информационните технологии са изключително полезни в случай на такива координиращи процеси. По отношение на внедряването им особена важност е типът на приложението.

Една по-обща класификация на типовете приложения е предложена на фиг. 2.9



Фиг. 2.9 - Класификация на типовете приложения

Различните видове системи трябва да са интегрирани помежду си, за да се осигури движение на информацията между йерархичните нива на

управление. Това обаче е скъп процес. Обикновено в практиката интегрирането между системите се реализира еволюционно. Повечето системи се изграждат отделно една от друга, тъй като паралелното им изграждане изисква огромни инвестиции. Този еволюционен подход сега показва някои негативни последици, които произтичат главно от слабата интеграция между отделните системи. Допуска се дублиране или противоречия в данните. Поради това се намалява ефективността, появяват се тесни места при функционирането на организацията.

ERP продуктите са сложни за използване, защото са разработени, за да бъдат пригодни за всякакъв тип организации, а това не винаги е удобно. Практиката показва, че внедряването на една ERP система е продължителен и стресиращ за потребителите процес. Освен това цената им е твърде висока и непосилна за повечето минни предприятия в България. Справка в страницата на производителя на най-популярната ERP система – SAP (<http://www.sap.com/bulgaria/about/customers/index.epx>), показва че е закупена единствено от “Мини Марица Изток ЕАД”, но и там процесът на интеграция е в начален стадий.

Описаната методология е насочена към частично (еволюционно развиващо се) решение, но може да бъде приложена и към ново решение (без аналог до момента).

## ***2.7 Специфика на разработката на еволюционно развиващи се информационни системи***

Първият етап при изграждане на такива приложения е да се определи типът им. След това на база оценка на текущото решение на проблема се разработва нова система или се усъвършенства вече съществуваща с цел удължаване на жизнения и цикъл.

Предимствата на подхода на „еволюционното развитие на данните” са:

- икономически изгоден - не се налага подмяна на използваната ИС. Когато е налице сложна ИС с развити функционални възможности подмяната ѝ с качествено нова обикновено е неоправдано. В подобни условия се предпочита надграждане на системата с допълнения подобряващи качеството на данните, реализирани със съвременни инструмент;
- подмяната и/или добавянето на модули може да стане в условията на непреустановено функциониране на системата;
- дава възможност приложението да се внедри на части (модули), като по този начин се улесняват потребителите при усвояването ѝ и се опростява внасянето на промени в системата;
- наличие на постоянна обратна връзка с потребителите;
- възприема се по-добре от поддържащия персонал – до голяма степен се запазва съществуващия интерфейс;
- съкращава времето и разходите за обучение на персонала.

### **Глава 3: Практически приложения на методология и подход за разработване на ИС с web базиран интерфейс**

В глава 3 е описана апробацията на предложената в глава 2 методология при създаването на реални ИС.

#### ***3.1 Приложение на методологията при изграждане на допълнителни модули към съществуващи ИС***

##### ***3.1.1 Система “VEZNA”***

Поставена бе задача за усъвършенстване, чрез web базиран интерфейс, на съществуваща в „Мини Марица - изток” ЕАД ИС. Обект на подобрението в разглеждания случай е Програмен продукт за пренос и

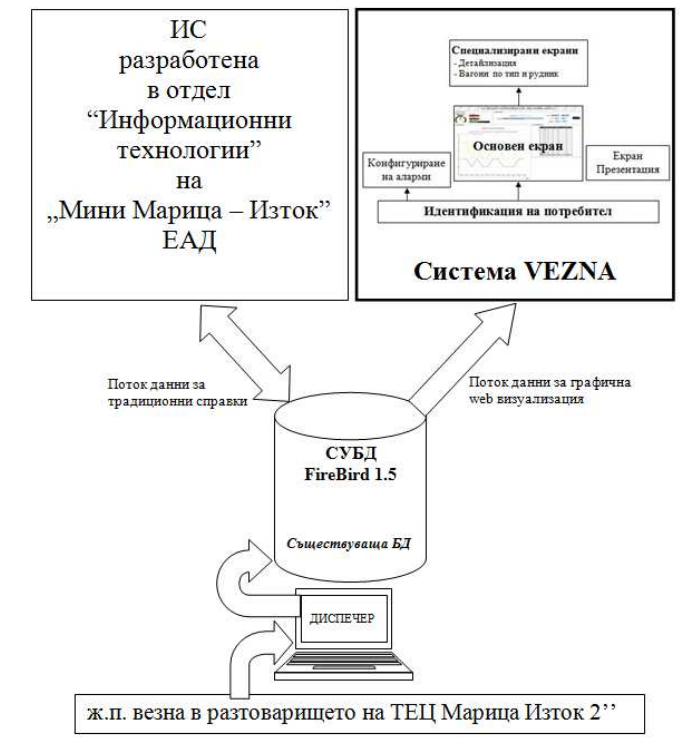


обработка на информация от ж.п. везна в разтоварището на ТЕЦ „Марица Изток 2”, разработка на отдел “Информационни технологии” на „Мини Марица – изток” ЕАД.

За решаване на поставената задача бе предпочетен подходът описан в глава 2, защото даде възможност да се използва съществуващата база данни, а интерфейсът на разработената система „VEZNA” да бъде подобен на вече използвания при обработка на тези данни.

Работата по реализиране на система „VEZNA” премина през няколко етапа:

1. Проучване на съществуващи ИС в „Мини Марица Изток” ЕАД и избор на работен обект /ж.п. везна в разтоварището на ТЕЦ „Марица Изток 2”/ - ИС покрива потребностите от натрупване, обработване и предоставяне на необходимата информация. Анализът на функционалните характеристики откроява отсъствието на специализирана компонента за визуализиране на данните в съответствие с модерните технологични решения в областта на информационните технологии.
2. Проучване на работния обект и събиране на информация.
3. Обработка на събраната информация - Система „VEZNA” използва входни данни от БД на работещата ИС отчитаща информацията от ж.п. везна в разтоварището на ТЕЦ „Марица Изток 2. От друга страна тя контролира достъпа на потребителите и им предоставя данни в съответна графична форма чрез web базиран достъп. На фиг. 3.1 е представена схема илюстрираща структурата на система „VEZNA” в рамките на съществуващата ИС и във взаимодействие с поддържаната в нея БД.



Фиг. 3.1 Схемa илюстрираща структурата на система „VEZNA” в рамките на съществуващата ИС

4. Реализиране на системата - С оглед повишаване ефективността на работата бе решено да се добавят нови компоненти, осигуряващи web базиран достъп до данните и тяхната графична визуализация. Като инструментариум са използвани свободно разпространявани програмни продукти - FireBird сървър за БД, PHP за изграждане на интерфейс, FlashDevelop и Open Source Flex SDK за реализиране на контроли и графика.

Повишаване на качеството на данните се изразява в добавяне на web интерфейс и графична визуализация на данни. Но главното в случая се състои в опростяване на достъпа на потребителя до широк набор от съпътстващи данни, обогатяване на формите на представяне на данните, като освен табличната се използва графичната и сигнализиращата концепция. Всичко това прави работата на диспечерите и потребителите много по-лесна, като предоставя необходимата информация в лесно усвояема форма и удобна за бързо осмисляне и адекватно реагиране.

Прилагането на web интерфейс цели осигуряване на по-високо, съвременно ниво на достъп до данните чрез използване на клиент – сървър архитектура.

Апробацията на системата VEZNA с реални данни доказва категорично функционалността, съвместимостта и приложимостта на разработката.

Система VEZNA е изградена на модулен принцип. По този начин бе улеснено модифицирането ѝ. Една година след като бе въведена в експлоатация системата разшири своя обхват – клиентите на ж.п. везната се увеличиха от един на четири, като при всеки клиент имаше различни изисквания за отразяване на преминалите пълни и празни влакове и различно конфигуриране на контролите и алармите. Благодарение на модулното изграждане на системата тези съществени промени бяха извършени в кратък срок (около 3 седмици).

### ***3.2 Приложение на методологията при изграждане на ИС заменящи съществуващи ИС***

#### ***3.2.1 Система “OCENKA”***

Реализирана бе ИС с web интерфейс за многокритериална оценка и избор на вариант на техническо решение на базата на интегрален количествен показател. Системата е предназначена, както за ползвателите ѝ в минните предприятия, така и за студентите в МГУ, давайки им възможност за on-line решаване на този тип задачи.

Като теоретична основа за изграждане на информационната система послужи описаната от проф. Милчо Велев методология [87]. Авторът извършва оценка и избор на технико-технологични решения на база набор от множество количествени и качествени показатели. Решението на тази задача включва избор на подходящи показатели, метод за

оценка, определяне на интегрален количествен показател (ИКолП) и интегрален качествен показател (ИКачП). За всеки от проектните варианти и намиране на най-малкия по стойност ИКолП и най-високия по-стойност ИКачП.

При множество варианти, подготвителната и изчислителната работа, по прилагане на процедурите за изчисление на ИКолП са рутинни, продължителни и обемни. За намаляване на рутинната работа и създаване на възможност за многократно прилагане на процедурата първоначално са използвани програми, написани на Pascal [87]. На следващ етап е създадена електронна таблица "ОЦЕНКАИП", реализирана с Microsoft Excel. Разработката, описана от доц. Дерменджиев [88], решава задачата за избор на предпочитан вариант, на база ИКолП. Въпреки че предостави значително повече възможности, от програмите на Pascal, системата притежава редица ограничения:

- работи само с един набор експертни оценки -  $\varphi$ ;
- използва фиксиран брой показатели-10 и варианти-6;
- не се извършва оценка по качествени показатели и комплексна оценка;
- ограничения породени от естеството на използвания инструмент (Microsoft Excel) , като управление на многопотребителски достъп, лиценз и др.

Поради горепосочените недостатъци, бе потърсена нова, по-мощна технология за обработка, представяне и съхраняване на информация. През последните 15 години, като такива се наложиха информационните системи с достъп през Интернет, базирани на релационни БД.

Целта бе да се реализира ИС за многокритериална оценка и избор на вариант на техническо решение на базата на интегрален показател (ИП), в web среда.

Третираната реализация е осъществена на базата на свободно разпространяван софтуер: PHP като език за връзка с базата данни и MySQL сървър за база данни, като по този начин се дава възможност за запазване на резултатите от предходни изчисления и използването им в бъдещи такива. Други нейни предимства са: – мащабируемост, преносимост, използване на потребителски заявки и др., присъщи на web базирания интерфейс. Дава се възможност и за export и import на данни от и в XML формат. Добавени са и подсистеми за качествено и комплексно оценяване. Изградени са модули за идентификация на експерт и потребител.

Възможностите на системата са разширени и в частта за избор на варианти – най-малко 1, най-много – всички налични в БД, а изчисленията се осъществяват по един или няколко набора от тегла за  $\varphi$ . В случая на избор на няколко варианта за тегла на  $\varphi$  се дават и обобщени данни. Възможно е добавяне на нови и изключване на вече съществуващ количествен и/или качествен показател.

### 3.2.2. Web сайт на МГУ

През 2006г., бе направена оценка на представянето на МГУ “Св. Иван Рилски” в Интернет. В работещия към момента сайт бяха констатирани редица пропуски. Заключение на групата извършваща анализа бе, че сайтът на МГУ не само не изпълнява предназначението си – информираност и реклама, а и съдържа зле поднесена, подвеждаща информация и е антиреклама за МГУ.

Поставена бе задача да се създаде съвършено нов, като облик и принцип на управление, web сайт като:

- се използва макета на вече съществуващия сайт;
- запази се и се разшири функционалността му;

- информацията се съхранява централизирано;
- интегрира се към други ИС, работещи в МГУ.

След анализ на готови решения (комерсиални и безплатно разпространявани) работния екип стигна до заключение, че те не са подходящи за постигане на набелязаните цели. Основният аргумент бе че те предлагат по-общи решения, неотразяващи спецификата на университета.

В края на 2007г. бе реализиран нов сайт на МГУ (фиг. 3.5). Информацията в него се управлява от централизирана БД. По този начин бяха решени проблемите свързани с представянето и съдържанието на информацията в сайта. Като инструменти бяха избрани свободно разпространявани с отворен код - сървър за БД MySQL 5.0 и език за реализиране на интерфейса – PHP. За реализиране на динамичен и интерактивен интерфейс и форматиране на данните са използвани инструментите CSS, JavaScript и AJAX. Избора на софтуерни продукти за реализиране на сайта бе направен по методологията описана в глава 2, като определящо бе натрупания от разработчиците опит с тези продукти, бюджета на проекта и възможността за безпроблемно интегриране на сайта към вече съществуващи системи изградени с подобен инструментариум, като например платформата за електронно обучение.



фиг. 3.5 – Входна страница на сайта на МГУ

Сайтът е изграден на модулен принцип, което улесни въвеждането и отстраняването на нови функционалности. Успешно са добавени модули за прием, магистърски тези, конкурси по ЗРАСРБ и годишник на МГУ.

Разработения продукт бе добре приет от потребителите. За това говорят следните факти:

- Според данни от независимия брояч [tycho.bg](http://tycho.bg) за последните 3 години посетителите на сайта се увеличават с около 20% на година.
- В анкета, проведена през 2009г., сред студенти от МГУ, 70% от анкетираните посочиха web сайта като основен източник на информация.

### *3.2.3 Система „Управление на студентски общежития”*

Информационна система „Управление на студентски общежития” (ИС УСО) е разработена в отговор на необходимостта от централизирано управление на специфичните дейности в тази област – настаняване и преместване на студентите в общежитията, разпределяне и заплащане на наеми и абонаментни услуги, извличане на справки за текущото разпределение в общежитията, финансова отчетност и поддържане на пълен архив.

Следвайки описаната в глава 2 методология, работата по ИС УСО започна с анализ на работещата към момента система и данните съхранявани в нея. Установено бе, че системата изпълнява ограничен набор от функции - управление на лични данни за наемател, настаняване на наемател и премахване на наемател. Освен това системата работи само с един потребител без възможност за достъп през [www](http://www), а голяма част от досието на наемателя се съхранява на хартиен носител.

Системата УСО бе проектирана и изградена изцяло на основата на свободен софтуер с отворен код: фреймуърк - Ruby on Rails; система за управление на версиите – SVN (Subversion); система за управление на бази данни – MySQL 5.0.

Изборът на софтуерни инструменти бе продиктуван и от значителния обем задачи, които трябваше да реши системата и повишените изисквания по отношение на сигурността на данните в нея. Бе проведен експеримент, който установи, че реализирането на подобна функционалност с PHP би увеличило програмния код с около 40%.

Прилагането на SVN ускори процеса на разработка, спестявайки усилията за създаване и прилагане на patch файлове, като напълно автоматизира този процес.

Достъпът до системата чрез VPN, осигурява първото ниво на защита на информацията. Следващото ниво на сигурност се осъществява с реализиране на потребители с различни права на достъп в системата.

Основните предимства, които предлага разработената и вече внедрена ИС „УСО“ са:

- Централизирано съхранение и управление на данни;
- Цялостно спазване на условията за сигурност, интегритет и конфиденциалност на съхраняваната информация;
- Актуална информация както за наемателите, така и за състоянието на контингента на Университета;
- Осигурена е възможност за развитие (усъвършенстване) на системата, чрез разработването ѝ на модулен принцип;
- Достъп до ИС чрез web интерфейс.

ИС „УСО“ е внедрена и функционира в Минно-геоложки университет „Св. Иван Рилски“ от 2007г. До момента, с цел обогатяване на функционал-



ността на системата, са направени две доработки свързани с въвеждане на картон на наемател и увеличаване на възможностите на модула "Каса".

### **3.3 Приложение на методологията при изграждане на нови ИС**

#### *3.3.1 Проект "Training of Project Managers in Clean Technologies Promotion in the Balkans"*

Проектът TRAC е структуриран в няколко подзадачи сред които:

- WP2 – Development of specific databases
- WP6 – Development of interactive module

Целта на подзадачата WP2 бе изграждане на БД съдържаща информация за енерго-спестяващи компании от Европейските страни, компании замърсители и големи консуматори на енергия за Балкански регион, както и с националните (България, Румъния, Словения) и европейски законодателни рамки в областта на екологията. Изискванията поставени към българския TRAC тим, във връзка с изграждане на базата данни бяха следните:

1. На база получената от партньорите информация да се създаде обобщена база от данни;
2. Да се изгради модул (Info Module) за интегриране към сайта на проекта TRAC, позволяващ достъп до базата данни чрез Интернет;
3. Да се изгради модул (Interactive Module) който се интегрира във web сайта на проекта TRAC и който притежава следните основни функции - свързване на мениджъри, реализиране на табло за обяви, администриране на потребителите на Info Module.

Следвайки подхода описан в глава 2 работата по проекта премина през следните етапи:

*1 Събиране на данни* - За попълване на базата от данни всеки един от TRAC партньорите бе натоварен със задача за събиране и изпращане на информация във формат MS Access или MS Excel. Събрана бе информация за повече 1500 компании от 19 Европейски страни, групирани в 19 отрасли и над 300 нормативни документа.*2 Анализ на данните* - Събраните данни се характеризират със следните особености:

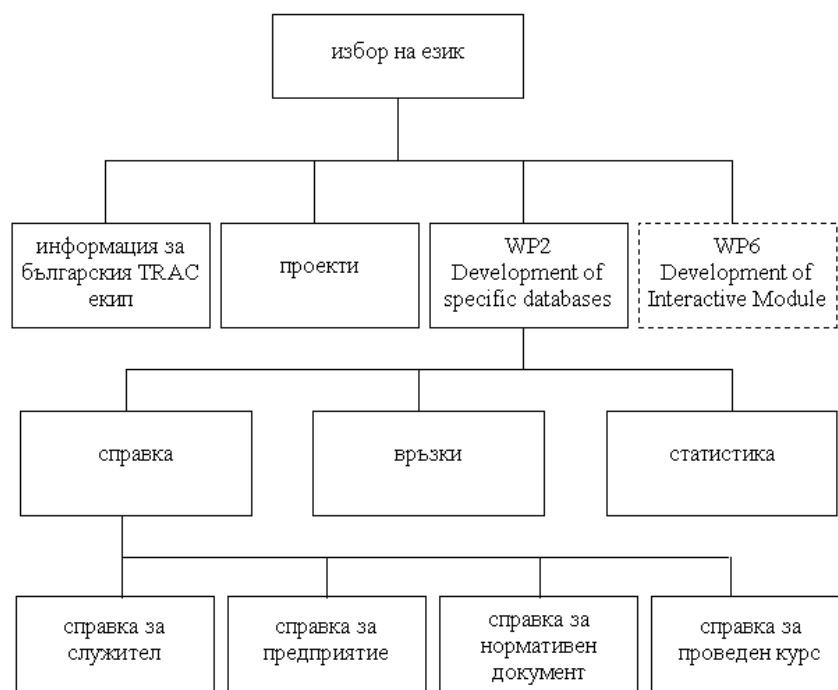
1. Многоезичност – TRAC партньорите от Испания, Германия, Румъния и Словения изпратиха данни на националните си езици, а данните за компаниите от Гърция и Европейските страни, не участващи в проекта, бяха на английски;
2. Различен формат – използвани бяха формати MS Access и MS Excel;
3. Разлики в структурата.

Екипът разработващ ИС трябваше да може да се справи с тези проблеми, а освен това бе нужно да отговори на изискванията предявявани към ИС с web интерфейс – висока сигурност и бързодействие, възможност за работа в многопотребителски режим и добре разработена система за привилегии.

*3 Избор на програмни инструменти* - MySQL, като сървър за база данни и PHP, като APIs.

#### *4 Реализация*

**Info Module** - За целите на WP2 бе разработена ИС за търсене и визуализация на информацията от БД – Info Module. Връзката между web-страниците и БД се осъществява посредством PHP скриптове.



фиг. 3.8 Схема на „Инфо модул”

**Interactive module** - Състои се от две независими части – административна част, която е ориентирана към управлението на информацията в таблиците на базата данни разработена в подзадачата WP2 и потребителска част, предназначена за извличане на създаване на заявки от потребителите, управление на таблото за обяви и синхронизация на данните между таблиците на Интерактивния модул и таблиците разработени в подзадачата WP2.

Екипът ни бе натоварен и с изборът на софтуер за електронно обучение.

Основните изисквания към платформата за обучение бяха:

- Осигуряване на среда за комуникация между учител и ученик;
- Възможност за web базиран достъп;
- Преносимост на учебните курсове.

За изпълнението на тази задача бяха апробирани различни платформи за електронно обучение. Предпочетена бе системата с отворен код MOODLE.

### *3.3.2 Електронно обучение в МГУ*

Платформата за електронно обучение MOODLE бе апробирана през 2003г. във връзка с изпълнение на проект TRAC. След като системата бе тествана в продължение на 3 месеца. Резултатите от работата ѝ бяха представени на партньорите по TRAC проекта. Реализацията бе оценена високо и системата бе възприета като среда за електронно обучение в рамките на проекта. Качени бяха 24 курса, а обучените курсисти са над 200.

За целите на МГУ „Св. Иван Рилски” също бе възприета системата MOODLE (<http://moodle.mgu.bg>). Определящо за този избор бе опита на внедрилия я екип с тази платформа.

Работата по тази тематика бе представена на поредица от работни срещи по проекта TRAC, конференции за електронно обучение и Научно-образователно Експо'2005 “Българските университети и технологични инкубатори – център на иновационни ИКТ разработки”, проведено в рамките на Есенния технически панаир в Пловдив. На Експо'2005 проекта "Електронно обучение в МГУ" бе отличен с първа поощрителна награда.

### **3.4 Изводи**

Главната цел при всички гореописани разработки бе повишаване на качеството на данните, като по този начин се:

1. Подпомагат ползвателите на тези системи при вземането на управленски решения;
2. Осигурява се web интерфейс – така системите стават достъпни за по-широк кръг потребители, в реално време и без оглед на географското им местоположение;

3. Подобрява преносимостта на данните, чрез експортирането им в XML формат.

За постигане на тази цел предложената в глава 2 методология бе приложена при изграждането на три типа системи, съобразно конкретните изисквания и особености на разработките:

1. Създаване на качествено нов продукт. Този метод бе предпочетен в случаите при които нямаше работеща ИС в съответната област. Приложен бе при проектите TRAC и електронно обучение в МГУ;
2. Замяна на съществуваща система, поради изчерпване на жизнения ѝ цикъл, с нова. По този начин бяха реализирани проектите УСО, web сайт на МГУ и OCENCA, както и доработки към тях. Този метод бе предпочетен в случаите при които използваните, за реализиране на ИС, инструменти са остарели и не позволяват интегриране и надграждане.
3. Интегриране на системата като модул в съществуваща ИС. Подходът бе предпочетен при проекта VEZNA. Той е по-евтин, тъй като не се налага подмяна на използваната информационна система. От тук произлиза и друго предимство на подхода – възприема се по-добре от поддържащия персонал, поради това че не се налага реорганизиране на цялата дейност.

Подходът на еволюционно развитие не отрича радикалното преминаване към нови платформи и цялостни преустройства. В някои случаи разширяването на ИС не е възможно, най-често поради спецификата на използваните програмни инструменти. Подобни преходи се налагат и поради развитието на инструментариума за разработване на ИС.

### III. ЗАКЛЮЧЕНИЕ

Съвременните инструменти се нуждаят от механизми за обработка на данните с които работят. През последните 10 години информационните технологии (ИТ) са една от най-динамично развиващите се области на Информатиката.

Стратегическата роля на ИТ в съвременния свят се изразява във възможността на мениджърите адекватно да реагират на динамиката на пазара, освен това създава, поддържа и увеличава конкурентното преимущество с цел извличане на максимална полза.

Съвременното състояние на ИТ може да се характеризира със следното:

- наличие на голямо количество програмно-апаратни комплекси и платформи за ефективно управление и съпровождане на производството, промишлено функциониращи БД и хранилища на знания в голям обем, съдържащи информация за всички обществени дейности;
- наличие на технологии, обезпечаващи интерактивен достъп на всеки потребител до информации и ресурси;
- разширение на функционалните възможности на ИТ, обезпечаващи разпределението на работата на базите и хранилищата на данни с данни с разнообразна структура и съдържание, мултиобектни документи, хипертекст; създаване на локални и интегрирани проблемно-ориентирани ИС с различно предназначение на основата на мощни сървъри и локално - изчислителни мрежи;
- включване в ИС на специализирани интерфейси позволяващи взаимодействие с експертни системи, системи за взимане на решения, системи поддържащи изпълнение, системи за машинен превод и други технологии и средства.

Изменението на минно-геоложките, минно-технологичните и социално-икономическите условия налага да се усъвършенстват и да се търсят технически решения, които осигуряват конкурентна продукция.

В настоящия дисертационен труд е направен обзор на технологиите и инструментите за реализиране на ИС с web интерфейс. Очертани са и перспективите в тази област. На база анализ на съществуващи методологии е предложена методология и подход за разработване на web базирана информационна система в минното дело и с общо предназначение. Приложението ѝ е предпоставка за изграждане в кратки срокове на продукт удовлетворяващ максимално информационните потребности на бъдещите му ползватели на ниска цена. Това се постига, като се осигурява активно участие на потребителите във всички етапи на разработката, а за програмната реализация се използват свободно разпространяване инструменти съобразени с квалификацията на разработчиците.

Методологията е апробирана при изграждането на 6 ИС и последвали доработки към тях за разширяване на функционалността им. Всяка една от тези системи е внедрена, а отзивите от потребителите са положителни.

## **Основни постижения**

Основните постижения са:

- Направен е сравнителен анализ на подходи за разработване на ИС с web интерфейс, като се отчита спецификата на информационното осигуряване на управленческите процеси в минното дело по отношение тяхната интеграция и зависимост от конкретни производствени процеси;
- Направен е сравнителен анализ на съществуващи методологии за разработване на софтуерни продукти;

- Предложена са методология и подход за разработване на ИС отчитащи спецификата на предметната област на приложение – минното дело;
- Предложената методология и подход за разработване на ИС е апробирана при създаването на 6 ИС работещи в минното дело, и с общо предназначение.

### **Перспективи и направления за бъдещи изследвания**

1. Разширяване на обхвата на приложение на предложената методология с цел изграждане на хранилища за данни и системи използващи знания.
2. Съществено направление за бъдещи проучвания е и специфицирането на приложен софтуер, базиран на “изчисления в облак” (Cloud computing), ориентиран към нуждите на минните предприятия.



## ПУБЛИКУВАНИ РЕЗУЛТАТИ ОТ ДИСЕРТАЦИЯТА

### *Научни сесии и конференции*

1. Янев, Н., К. Иванов, В. Карагъзов, И. Щърбанова, Й. Анастасова. 2003. *“Възможности за приложение на Web интерфейс при работа с база данни за минното производство“*. Юбилейна международна научна сесия на МГУ “Св. Иван Рилски” 2003 г.

2. Янев, Н., И. Щърбанова, К. Иванов. 2004. *“Многоетапна схема за създаване на база данни при мултинационална структура на доставчиците на данни”* Шести международен симпозиум ТЕХНОМАТ & ИНФОТЕЛ 2004

3. Ганчев, Г., К. Иванов, Н. Янев, Д. Белчевски, 2005. *“Електронно обучение по прилагане на чисти технологии на Балканите – проект TRAC по програма Леонардо да Винчи”*, ЕНЕРГИЕН ФОРУМ 2005 *“Предизвикателства към енергетиката в условията на либерализиране на енергийния пазар”*.

4. Янев, Н., К. Иванов, 2006. *“Компютъризирана система за комплексна оценка на техническо решение”*, Международна научна сесия на МГУ “Св. Иван Рилски” 2006.

5. Yanev N., I. Shtarbanova, D. Beltchevski, Y. Anastasova, K. Ivanov, 2008. *“Management Of Students' Dormitories In University Of Mining And Geology “St. Ivan Rilski” Through Framework And Open Source Technologies”*, "COMPUTER SCIENCE `2008", 18 - 19 September 2008, Kavala, Greece

6. Янев, Н., К. Иванов, И. Казанджиев, Н. Халачев, Г. Георгиев, П. Костова. 2010. *“Повишаване качеството на данните чрез прилагане на системата VEZNA”*, Международна научна сесия на МГУ “Св. Иван Рилски” 2010.

7. Янев, Н. 2011. *“Възможности за приложение на съвременни XML базирани технологии в минната промишленост”*, Международна научна сесия на МГУ “Св. Иван Рилски” 2011.

### **Списания**

1. Янев, Н., К. Иванов. 2006. *“Система за избор на техническо решение базирана на съвременни информационни технологии”*, списание "Минно дело и геология".

## **ЛИТЕРАТУРА** (по реда на цитирането й в автореферата)

- [97] Elmasri, R., B. Shamkant. 1994. Navathe, Fundamentals of Database Systems, Addison Wesley
- [17] Карпова, Т.С. 2001. "Базы данных: модели, разработка, реализация", издательстве "Питер"
- [78] Hunter, D., K. Cagle, C. Dix, R. Kovack, J. Pinnock, J. Rafter. 2001. Beginning XML, Wrox
- [18] Чонг, Р., И. Хейкс, Р. Хужа. 2008. Въведение в DB2 Express-C, IBM Corporation
- [27] Chapman, A. D. 2005. Principles of Data Quality, version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen
- [20] [http://en.wikipedia.org/wiki/Data\\_quality](http://en.wikipedia.org/wiki/Data_quality)
- [21] Ivanov, K. 1972. "Quality-control of information: On the concept of accuracy of information in data banks and in management information systems". The University of Stockholm and The Royal Institute of Technology. Doctoral dissertation
- [23] Price, R., G. Shanks. 2004. A Semiotic Information Quality Framework, Proc. IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain and Complex World, Prato
- [27] Chapman, A. D. 2005. Principles of Data Quality, version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen
- [34] Redman, T.C. 2001. Data Quality: The Field Guide. Boston, MA: Digital Press
- [45] Helfert, M, C. Herrmann. 2002. "Proactive data quality management for data warehouse systems". Proceedings of the 4th Intl. Workshop on Design and Management of Data Warehouses
- [48] Vassiliadis, P., M. Bouzeghoub, C.Quix. 2000. "Towards Quality-oriented Data Warehouse Usage and Evolution". Information Systems, 25(2)

- [49] Daniel, F., F. Casati, T. Palpanas, O. Chayka, C. Cappiello. 2008. "Enabling Better Decisions through Quality-aware Reports", International Conference on Information Quality (ICIQ), MIT
- [104] Han, J., M. Kamber. 2001. *Data Mining: Concepts and Techniques*, Morgan Kaufmann
- [79] Boulicaut, J.-F., C. Masson. 2005. Data mining query languages. In: *The Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach (Eds.), 2005
- [80] Botta, M., Boulicaut J.-F., C. Masson, R. Meo. 2004. Query languages supporting descriptive rule mining: a comparative study. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of LNCS. Springer-Verlag
- [81] <http://www.dmg.org/>
- [88] Дерменджиев, К., Е. Ножарова. 1996. Електронна таблица за многокритериална оценка и избор на вариант на техническо решение на базата на интегрален количествен критерии OCEKKAIP инструкция за ползване. МГУ
- [87] Велев, М. 1997. Методичен практикум за миннотехнологични решения. МГУ