



Софийски университет "Св. Климент Охридски"
Факултет по математика и информатика

Калинка Михайлова Калоянова

Анализ и проектиране на информационни системи

Хабилитационен труд

във връзка с конкурс за професор по информатика и компютърни науки
(бази от данни, анализ и проектиране на информационни системи)

2013
София

СЪДЪРЖАНИЕ

Увод.....	3
Глава 1. Анализ и проектиране на информационни системи: основни концепции и принципи на разработка	5
Глава 2. Анализ и проектиране при транзакционни системи	12
Глава 3. Анализ и проектиране при многомерния модел на данните.....	34
Заклучение.....	58
Литература.....	60
Приложение.....	64

УВОД

Бурното развитието на комуникационните и информационните технологии през последните години, световната глобализация, уеб технологиите и новите архитектурни парадигми доведоха до ускорено натрупване на големи количества от данни във всички области на човешката дейност. От първостепенна важност за всяка организация е да използва тези данни, съсредоточени най-вече в различни видове информационни системи, за да се развива успешно.

С всяка изминала година информационните системи стават по-комплексни, а техните потребители – по-взискателни. Независимо от големия брой методи и технологии за разработка на софтуер, пред създателите на този тип приложения продължават да стоят редица предизвикателства.

Настоящият хабилитационен труд има за цел да изследва ролята на анализа и проектирането при разработката на съвременните информационни системи.

Глава 1 има обзорен характер. Представени са основни концепции, свързани с разработката на бази от данни и информационни системи. Разграничени са транзакционните и аналитичните системи.

Следващите две глави разглеждат последователно двата типа системи, като започват с анализ на предметната област, поставят определени проблеми и представят резултати, постигнати при решаването на някои от тях.

Глава 2 разисква ролята на изискванията на клиентите при разработката на софтуерни продукти. Описани са няколко метода за събиране и анализиране на изискванията към разработката на софтуер, които имат пряка връзка с предложения подход за анализиране и обобщаване на изискванията на потребителите и използването им за оптимизирани проектни решения.

Глава 3 изяснява специфичните характеристики на многомерния модел на данните и формулира въпроси, които подлежат на разрешаване. Изследвана е ролята на данните, които постъпват в складовете от данни. Предложен е подход за разрешаване на разредеността на данните в многомерния модел.

В заключението са обобщени резултатите, описани в предходните глави и са посочени някои насоки на бъдещо развитие.

Основните резултати, описани в хабилитационния труд, са публикувани в 12 научни публикации. Част от резултатите са свързани с работата по няколко научно-изследователски проекта към Фонд „Научни изследвания” на СУ”Св. Кл. Охридски”:

- Оптимизация на интегрирането и анализа на информация, (№54/2007 от ФНИ/СУ);
- Възможности за оптимизация на многомерния модел на данни, (№165/2008 от ФНИ/СУ);
- Информационни системи – методология, класификация, интеграция, (№7/2009 от ФНИ/СУ);

- Имплементиране на БД в комплексни бизнес и ИТ приложения, (№134/2012 от ФНИ/СУ).

Хабилитационният труд се базира на изследвания и резултати, описани в 12 публикации. За 9 от тях до началото на април 2013 г. са забелязани 24 цитирания, които са описани в **Приложение: Забелязани цитирания на публикациите, въз основа на които е разработен хабилитационният труд.**

ГЛАВА 1: ИНФОРМАЦИОННИ СИСТЕМИ: ОСНОВНИ КОНЦЕПЦИИ И ПРИНЦИПИ НА РАЗРАБОТКА

Обзор на областта

Информационните системи (ИС) са най-често използваните софтуерни продукти в нашето съвремие. Всяка организация използва в ежедневната си дейност различни информационни системи. Те са основно средство за нейното успешно развитие поради няколко причини:

- обемът на данните непрекъснато нараства;
- данните се превръщат във важен корпоративен капитал;
- използването на ИС позволява обединение на информационните ресурси;
- с навлизане на новите технологии крайните потребители все по-често извършват самостоятелно голям брой бизнес операции чрез използване на информационни системи.

В основата на една информационна система е структурата от данни, в която се съхранява информацията – **базата от данни (БД)**. Самата *база от данни* се дефинира като *организирана съвкупност от данни, съхранявани за дълъг период от време и предназначени за използване от различни приложения чрез ефективни средства за достъп и поддръжка* [Martin, 1983].

Едно от основните преимущества разработването на базите от данни е възможността да се използват едни и същи данни за различни приложения на основата на т.нар. 3-слойна архитектура, която разделя потребителските приложения от концептуалното и физическото представяне на данните.

Тази архитектура описва три нива, които се фокусират върху различни аспекти на представянето на данните:

- вътрешна схема, която описва физическо съхранение на данните;
- концептуална схема, която описва структурата на цялата база от данни – отделните елементи и връзките между тях, ограничения и др.;
- външни схеми, предназначени за различните приложения, които позволяват едни и същи данни да се използват от различни потребители.

Независимостта на отделните нива е ключов фактор за успешна реализация поради промените, които неизбежно настъпват при реалното използване на системите.

Ефективната разработка на всяка информационна система изисква внимателно изучаване на структурата на данните, изграждане на адекватен модел на базата от данни и създаването на приложение, което работи с тези данни.

Тези концепции са в основата на подхода, ориентиран към данните - т.нар. data driven approach [Simsion & Witt, 2004].

Разширяването и усложняването на обхвата на дейността на информационните системи и функционалността (услугите), които те предлагат, както и сложните процеси на внедряването и поддържането им, водят до по-отчетливо разграничение на различните фази в тяхната разработка и експлоатация.

Обособяват се два жизнени цикъла на информационната система и на самата база от данни. В [Elmasri & Navathe, 2006] те са дефинирани като *микро и макро жизнен цикъл* на информационната система. Под *макро жизнен цикъл* се разбира жизненият цикъл на ИС, докато *микро жизненият цикъл* се свързва със самата база от данни.

Дейностите, свързани с **микро жизнения цикъл**, са насочени по-конкретно към разработката на базата от данни и включват:

- проектиране на базата от данни;
- програмна реализация;
- зареждане/прехвърляне на данните;
- съвместимост със стари приложения;
- тестване и валидация;
- пускане на системата в експлоатация;
- наблюдение и поддържане.

Макро жизненият цикъл включва фази, които са типични за разработката на софтуерни продукти:

- анализ на осъществимостта;
- събиране и анализиране на изискванията;
- проектиране;
- реализация;
- валидиране и тестване;
- внедряване, оперативна работа и поддържане.

Редица източници [Elmasri & Navathe, 2006], [Simsion & Witt 2004], [Garcia-Molina & Ullman & Widom, 2009] дискутират параметрите на добрия проект за една информационна система. Наред с очакванията за бързина, устойчивост, удобен интерфейс и лесно поддържане, от първостепенна важност е системата да отговаря и на други показатели: удовлетворяване на изискванията за информационно съдържание и удовлетворяване на изискванията за ефективно управление на процесите и производителността.

Постигането на добри резултати в тази насока не е лесно поради много причини. Основно предизвикателство тук е неформалният процес на дефиниране на изисквания и различният начин на мислене на крайните потребители, които не успяват да опишат своята представа за системата. Повечето модели на представяне са силно затруднени

при отразяване на промени в схемата на базата от данни след първоначалното ѝ създаване, така че добре дефинираните изисквания са в основата на успешните решения.

Тези обстоятелства поставят въпросите, свързани с анализа и проектирането на една ИС, като най-важните в цялостната разработка на този тип софтуер.

Целта на анализа е да идентифицира изискванията към разработваната система. В рамките на тази дейност се комуникира с потребителите, за да се изяснят техните виждания, да се разрешат противоречията и евентуалната несъвместимост на различните гледни точки и да се документират всички изисквания.

При проектирането се създава модел, който трябва да съответства на документираните изисквания и да уточни редица параметри като потребителски интерфейс, вход и изход, процеси, архитектура, хардуерна и софтуерна платформа, върху която ще работи системата. Получената спецификация на системния проект се одобрява от двете страни – потребители и разработчици.

И двете дейности - анализ и проектиране - работят по посока на изграждане на модел на системата, но само анализът се занимава с въпроси, които клиентите и/или потребителите трябва да обсъждат и разберат. Проектирането фокусира върху създаването на по-детайлни модели, насочени към ефективност и оптимизация и включва много елементи, които са извън обсега на конкретния интерес на клиента [Shelly & Rosenblat, 2010].

Транзакционни и аналитични системи

Изборът на едно или друго решение при проектирането на информационните системи зависи до голяма степен от функционалността, която те трябва да предлагат.

Една от най-разпространените класификации на системи за обработка на информацията се базира на начина на тяхното функциониране и употреба и ги дели на два типа:

- *Транзакционни системи (OnLine Transactional Processing systems - OLTP);*
- *Аналитични системи (OnLine Analytical Processing systems – OLAP).*

Основната цел на преобладаващата част от оперативните системи за управление на бази данни е ежедневната обработка на информация (въвеждане, изтриване, обновяване на данни) и изпълнение на заявки – запитвания към данните. Сред тях са системи, използвани от различни организации – банки, мобилни оператори, здравни заведения, застрахователни фирми, университети, държавни организации.

Някои системи имат по-специфична насоченост – системи за планиране на ресурсите (Enterprise Resources Planning systems – ERP systems), системи за управление на взаимоотношенията с клиентите (Customer Relationship Management systems– CRM systems).

Предназначението на системите от този род е да автоматизират и улесняват ежедневната работа в организацията. Те трябва да поддържат ефективност при изпълнението на структурирани и повторяеми заявки, които се провеждат в отделни изолирани транзакции, като се стремят да осигурят оптимална скорост на транзакцията, да поддържат съгласуваността на данните и възможности за тяхното възстановяване [Garcia-Molina & Ullman & Widom, 2009]. Това определя и названието на тези системи - *OLTP systems*.

Транзакционните системи използват бази от данни, които боравят с ежедневно променящата се информация (*покупки, продажби, текущи цени, актуално състояние на погасявани кредити и др.*).

С течение на времето организациите натрупват огромни количества данни от техните транзакционни системи. Обемът на тази информация затруднява процеса на съхранение и достъпа до нея и ограничава възможностите за ефективното ѝ използване. Това налага данните да бъдат организирани в друг вид структури – хранилища на данни или складове от данни (Data Warehouses - DW). Те предоставят инструменти за систематично организиране, визуализиране и анализиране на данни, събирани от различни системи в продължителен период от време. По тази причина системите, които се изграждат върху складовете от данни, се наричат **аналитични системи** – *OLAP systems*.

OLAP системите позволяват бърз, интерактивен и съгласуван достъп до обобщено представяне на информацията с възможности за динамичен, многоразмерен анализ по различни показатели – *време, място, тип дейности* и др. Тези данни могат да дадат важна за бизнеса информация. Например, какви са били параметрите на продажбите през последните няколко години; как са се променяли цените в определен период от време или географски регион.

Всеки от двата типа системи - транзакционни и аналитични, поддържа важни функции при обработката на информация.

Основната цел на една транзакционна система е да осигури бързодействие при получаване на справките, както и ефективност и гъвкавост при съхранение и обновяване на ежедневно променящата се информация.

При складовете от данни ударението е върху съхранението на значителни количества данни във вид, удобен за различни анализи - статистики, сравнения на данни от предишни години, обобщения по различни параметри. Тази информация има за цел да отговори на различни въпроси, свързани с текущото и миналото състояние на някакъв тип бизнес, както и да даде потенциални насоки за неговото развитие.

Традиционните системи поддържат онлайн обработка на транзакции (OLTP), която включва добавяне, изтриване и обновяване на данни, както и отговор на различни запитвания, относно съдържанието на данните - заявки. Организацията на базите от данни в тях трябва да бъде оптимизирана по отношение на този тип операции. В някои случаи заявките засягат само една малка част от базата от данни или се извършват операции, с които се актуализират няколко записа във връзка с един транзакционен

процес. Моделите, които съответстват на този тип операции, са се развивали в продължение на няколко десетилетия и включват последователно йерархични, мрежови, релационни, обектно-релационни или изцяло обектно-ориентирани модели [Molina & Ullman & Widom, 2009].

Организиран по начин, целящ оптимално представяне на отделните елементи и връзките между тях, тези модели на бази от данни не са подходящи за аналитични заявки. Необходими са решения, които позволяват обобщаване на огромни количества детайлни данни и използване на различни модели на представяне на информацията в аналитичните системи.

При складовете от данни се изисква тематично-ориентиран модел, удобен за интерактивен анализ на данните [Inmon, 1997]. Те трябва да бъдат проектирани така, че да осигуряват ефективно извличане, обработка и визуализация на данните за аналитични цели и вземане на решения. Затова тук се използва специфичен многомерен модел, който не само съхранява данните, но и позволява тяхното категоризиране на базата на предварително дефинирани критерии (измерения) – например, време, географско положение и др. За целта данните се разглеждат като един куб в n -мерното пространство. Кубът се дефинира чрез измеренията и факти за измеренията. Така се постига възможността данните да бъдат анализирани по различни измерения.

В сравнение с транзакционните системи, складовете на данни обикновено съдържат данни, чиито количества са в пъти по-големи. В допълнение, те са събирани от множество източници, които могат да включват бази от данни с различни модели и структури на данните, а понякога и файлове от външни системи и платформи. Това също слага отпечатък върху тяхната структура и поставя редица въпроси, свързани с обединението на данните от различните източници.

От различната функционална насоченост на двата вида системи - транзакционни и аналитични, следва и прилагането на различни модели на представяне на данните в тези системи и различни проектни решения.

Отворени въпроси и мотивация на изследванията

От направеното до момента изложение става ясно, че функционалността и предназначението на една информационна система указват съществено влияние върху модела на данните и нейната архитектура. С увеличаването на обхвата и комплексността на системите (и на данните), се появяват нови предизвикателства пред техните проектанتي и разработчици.

В последните години хранилищата на данни са активна зона за научни изследвания. Очаква се и повишена изследователска дейност и в близко бъдеще.

Търсенията тук се съсредоточават върху автоматизиране на някои аспекти на складовете от данни, които в момента изискват значителна ръчна намеса, като събиране на данни, управление на качеството на данните, подбор и изграждането на подходящи за достъп структури [Elmasri & Navathe, 2006].

Редица познати проблеми като изчистването на данни, индексиране, разделяне на дялове, изгледи, оптимизация на производителност и други, могат да получат подновено внимание в резултат на прилагането на нови технологии. Добър пример за това са IBM Netezza и Oracle Exadata [Francisco & Kearney, 2011], които предлагат модерни хардуерно-софтуерни решения със значително подобрени характеристики.

Прецизният анализ на предметната област, характеристиките на конкретните данни и включването на бизнес правила в модела на склада от данни могат да направят процеса на създаване и поддръжка на склада по-точен и по-интелигентен.

В същото време не спират търсенията и при транзакционните системи. Макар този тип системи да имат по-дълга история и много от концепциите при тях да са разработвани с години, все още съществуват неизяснени въпроси и насоки за подобрене, което се потвърждава и от появата на NoSQL решенията.

Като цяло, всеки отделен елемент на процеса на разработка на бази от данни и информационни системи е многократно дискутиран. Но все още има резерви в по-малко изследваните аспекти, в прилагане на комбинации от успешни методи и решения на границата на различните области или в търсене на нови аспекти на приложение на вече използвани практики.

От друга страна, в стремежа за по-добро проектиране все повече внимание се обръща на ролята на конкретните изискванията към информационната система. Това налага особено внимание към механизма за събиране и анализиране на данните и използването на тези резултати при моделирането.

От многообразието предизвикателства пред разработката на съвременните ИС могат да се отделят няколко, които предизвикват особен интерес, затова макар и частично са предмет на изследванията в този труд:

- Използването на правилно подбрани методи в цялостния процес на разработка е ключово за успешната реализация на проектите.
- Ролята на процеса на разработка е съществена – крайният резултат е софтуерно приложение, а не само източник на данни. Следователно, една възможна насока на търсене на подобрения е внимателният анализ и разпределяне на усилията в различните фази на разработка.
- Фокусирайки се върху анализа и проектирането в първите фази на жизнения цикъл, проектантите влагат повече усилия в началото, за да смекчат появата на негативни явления в следващите фази и да намалят риска от провал на проекта или непостигане на необходимото качество на софтуерния продукт. Анализът и проектирането оказват огромно влияние върху цялостния процес на разработка на една информационна система.

Следващите две глави от хабилизационния труд разглеждат въпроси, свързани съответно с транзакционните и аналитичните системи, като в началото на всяка глава са описани специфични проблеми, чиито решения са търсени и представени с някои конкретни резултати.

И в двете направления фокусът на изследванията е върху анализа на данните и използването на резултатите от този анализ за по-ефективно проектиране на системите.

Наред с това са представени конкретни решения на специфични въпроси под формата на методи, модели и специфични софтуерни решения.

И в двете направления са правени изследвания за възможностите за оптимизация на софтуерните модели и процеси, които се прилагат при разработката на съответните системи и са търсени пътища за оптимизация и прилагане на решения, подобряващи ефективността.

Не на последно място е търсено приложение на основните принципи не само при работата на професионалистите по реални софтуерни решения, но и в процеса на обучение на студенти от специалността *Информационни системи*.

ГЛАВА 2: АНАЛИЗ И ПРОЕКТИРАНЕ ПРИ ТРАНЗАКЦИОННИ ИНФОРМАЦИОННИ СИСТЕМИ

В тази глава се дискутират предизвикателствата, свързани с определянето на изисквания при разработката на ИС. Разгледани са утвърдени подходи за структурирано събиране, анализиране и обобщаване на потребителските изисквания като подхода на концептуално проектиране, унифицираният процес и моделът на потребителските случаи и концепциите на обектно-ориентираното проектиране. Дискутирани са техни предимства, недостатъци и отворени въпроси. Предложен е иновативен подход, който използва основни принципи на разглежданите методи и ги съчетава в нова, интегрирана рамка.

Определяне на изискванията

Изследванията в областта на класическите транзакционни системи се провеждат от десетилетия. Първоначално усилията са били съсредоточени в разработката на базата от данни. Исторически базите от данни преминават от йерархичен и мрежов модел към релационен, а по-късно към обектно-релационен и обектно-ориентиран. В последните години се появиха т.нар. *NoSQL* решения, които са насочени към динамично променящи се като структура и обем съвкупности от данни [Strauch, 2011].

Най-много средства, които улесняват моделирането и проектирането, са създадени за релационния модел – водещият модел по отношение на разработката на бази от данни в момента. Сред най-използваните средства за моделиране са *моделът същност-връзки* (*Entity Relationship model - ER model*) и неговото разширение (*Enhanced Entity Relationship - EER model*). Създадени са и редица инструменти за преобразуване на концептуалния модел в логически и физически, а нормализацията се използва като основен механизъм за оптимизация на релационните схеми.



Фигура 2.1. База от данни – етапи на разработка

Въпреки че фиг.2.1 покрива всички важни елементи от разработката на базата от данни, това е по-скоро каскаден модел на разработка, който не отчита в достатъчна степен влиянието на промените, неизбежно настъпващи в процеса на работа. В същото време, повечето модели на представяне не позволяват промяна на схемата на базата от данни след първоначалната реализация, така че правилно определените изисквания са в основата на успешните решения.

Това засилва значението на елементите на микро и макро жизнения цикъл на ИС и тяхното обвързване и извежда на преден план ролята на анализа и проектирането като основни дейности в разработката на информационните системи.

Независимо от разнообразието на методи и техники за анализ и проектиране, които използват модерни визуализационни и комуникационни техники, ролята на конкретната информация, получена от клиентите и потребителите, техните реални изисквания и механизмът на превръщането им в действителни проектни решения за конкретния софтуерен продукт не са напълно изяснени.

Най-общо под изисквания за разработка на софтуерни продукти разбираме

- функционалността, услугите, които предоставя този продукт;
- ограниченията, при които той работи и/или се разработва.

Много определения и класификации, свързани с изискванията, са обобщени в [Sommerville, 2010], [Pressman, 2009]. Без да навлизаме в детайли относно различните таксономии, за целите на настоящия труд ще използваме общоприетата класификация на изискванията, която ги разделя на **функционални и нефункционални**.

Функционалните изисквания описват функционалността, която се предоставя от софтуерния продукт.

Нефункционалните изисквания дефинират системни характеристики (надеждност, време за отговор, необходима памет) и ограничения (възможности на устройствата за вход/изход, начин за представяне на данните и др.). Ограниченията могат да засягат както дейността на бъдещия продукт, така и самия процес на разработка.

Нефункционалните изисквания обикновено са свързани с качеството на разработвания продукт и понякога могат да бъдат по-важни от функционалните.

Известни са редица методи и техники за събиране, обобщаване и анализиране на изискванията [Sommerville, 2010], [Maciaszek, 2005]. Но в повечето случаи разнообразието на използваните методологии не помага на проектантския екип да избере най-подходящия метод, а понякога това дори е причина за затруднения и объркване.

Традиционните методи като презентации, доклади и други текстови описания не винаги осигуряват пълно и непротиворечиво разбиране на всички концепции. Наред с тях, редица съвременни техники за визуализация и комуникация се опитват да вземат предвид реалните очаквания на потребителите [Maciaszek, 2005]. Но ролята на данните, получени от клиентите, не винаги е добре адресирана и не е изяснен механизмът на превръщането на тези данни в конкретни проектни решения.

В това изложение ще разгледаме няколко утвърдени подхода, които предлагат добре структурирани схеми за описание на изискванията на потребителите.

Контекстуално проектиране

Контекстуалното проектиране (Contextual Design – CD) изследва задълбочено ролята на клиентите при проектирането на една софтуерна система [Reddy et al., 2003].

Подходът осигурява набор от техники, модели и процедури за един постъпков процес, който подпомага проектирането на софтуер по добре организиран и систематичен начин. Като фокусира върху това как проектантите могат да разберат нуждите на клиентите, методът е много удобен за анализ и проектиране на ИС, тъй като те винаги са насочени към потребителите.

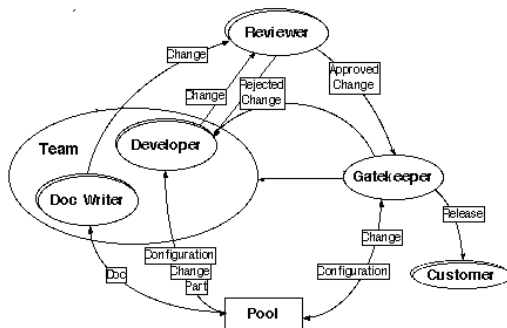
Контекстуалното проектиране използва няколко модела, всеки от които разглежда от различни перспективи работата на софтуерния продукт не като самостоятелна единица, а като елемент на организационната среда. Тези работни модели показват различни гледни точки към системата и дават възможност цялата информация, събрана от отделните интервюта с потребителите, да се опише със съответни диаграми за всеки модел [Beyer & Holtzblatt, 1997].

Петте модела, които предлага контекстуалното проектиране са:

- Модел на дейностите;
- Модел на последователността ;
- Модел на артефакта;
- Културен модел;
- Физически модел.

Моделите отчитат не само функционалността на продукта, но и неговото използване от потребителите, взаимодействията между самите потребители, средата в която си взаимодействат [Beyer & Holtzblatt & Baker, 2004]:

Най-информативен е *моделът на дейностите*. Той показва основните действащи лица, дейностите, които извършват, както и комуникацията и координацията между тях. Моделът може да посочи и някои работни продукти (артефакти). Важните за работата физически места също се отбелязват върху този модел.



Фигура 2.2. Примерен модел на дейностите [Beyer&Holtzblatt, 2009]

Моделът на последователността представя последователност от работни стъпки за постигане на конкретни потребителски намерения и цели. На него се означават и други елементи като тригери (събития, които предизвикват последователността от стъпките), възможни прекъсвания, алтернативни стъпки.

Физическите продукти, които се създават в работния процес, се описват от отделни модели на артефактите. Някои от тях могат да присъстват и в модела на дейностите.

Останалите два типа модели предоставят спомагателна информация. Културният модел има за цел да отбележи някои особености на взаимоотношенията между отделните участници в процесите, политиките и културните ценности в организацията, за която се създава системата, а физическият отразява ролята на реалната работна среда.

Според принципите на контекстуалното проектиране работата започва с изследването на контекста на реалната работна среда на потребителите на системата. Това е първата стъпка от разбирането на реалните клиенти и техните нужди. Проектантският екип провежда поредица от интервюта с клиенти на работните им места.

Обикновено много различни групи правят интервютата. За да се гарантира, че всички заинтересовани страни получават единно описание на дейностите на системата, се организира т.нар. „консолидационна“ сесия, в която отделните интервюиращи обединяват своите модели от интервютата в общ модел, представен с петте вида обобщени диаграми.

Индивидуалните бележки на интервюиращите се използват и при построяването на т.н. *групираща* диаграма (affinity diagram), която представя в йерархичен вид всички въпроси, които засягат проблемите на потребителите. Тогава екипът е готов да представи визията си за проекта, която при този подход има по-широко представяне, включващо и графични елементи, за по-голяма яснота най-вече за крайните потребители.

Контекстуалното проектиране продължава със създаването на сторибордове (storyboards), които скицират работните задачи в последователности от изображения, снимки, скици и текст. Те могат да включват не само поведението на софтуерната система, но също така и задачи, изискващи човешка намеса, първоначални идеи за потребителския интерфейс, някои бизнес правила.

Получената дотук информация се използва за изграждане на следващия елемент: Проект на средата на потребителя (User Environment Design - UED) - първото представяне на системните функции и тяхното организиране по систематичен начин, който съответства на потребителските цели [Holtzblatt, 2001].

Като последна стъпка контекстуалното проектиране използва прототипирането не толкова като форма на обсъждане на потребителски интерфейс, а като метод за доуточняване на функционалността на бъдещата система.

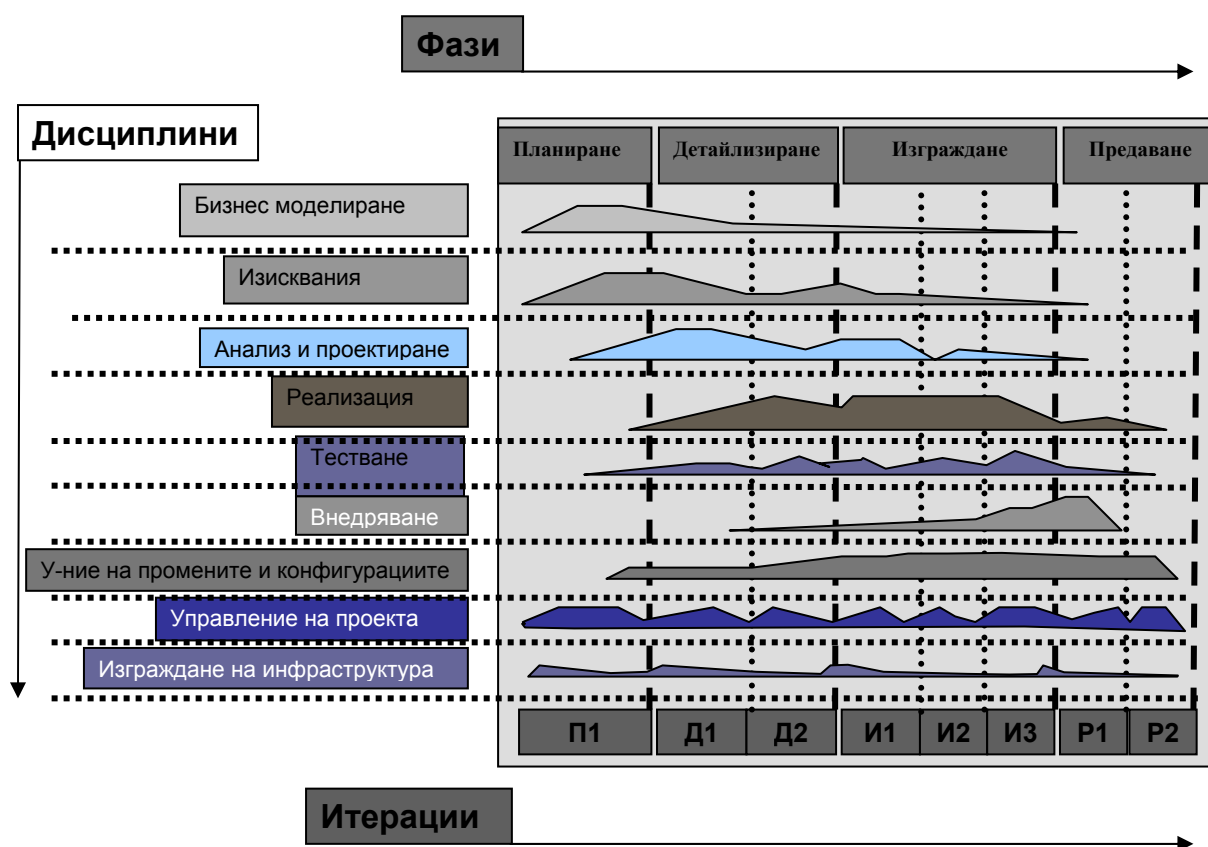
Модел на потребителските случаи

В последните години **моделът на потребителските случаи** се наложи като една от най-често срещаните форми за определяне на функционалността на софтуерните продукти и като особено подходящ в случаите на разработване на информационни системи. Макар да се използва самостоятелно, моделът се определя като част от дисциплината **Изисквания** на Унифицирания процес.

Унифицираният процес (Unified Process - UP) е последователен процес за разработка на софтуер. Поддържа четири обособени фази – Планиране (Inception), Детайлизиране (Elaboration), Изграждане (Construction) и Предаване (Transition) [Larman, 2004].

Най-разпространената практическа реализация на унифицирания процес е Rational Unified Process - RUP. В момента тази реализация се поддържа чрез серия софтуерни продукти, предоставяни от IBM.

Процесът се представя хоризонтално - чрез разпределението на отделните фази във времето, и вертикално - чрез разпределението на деветте дисциплини, които са разработени за него. Фазите от своя страна се разделят на итерации с еднаква продължителност, като броят на итерациите в отделните фази може да бъде различен. Така се поддържа гъвкав и последователен процес на работа.



Фигура 2.3. RUP - фази и итерации (примерно разпределение)

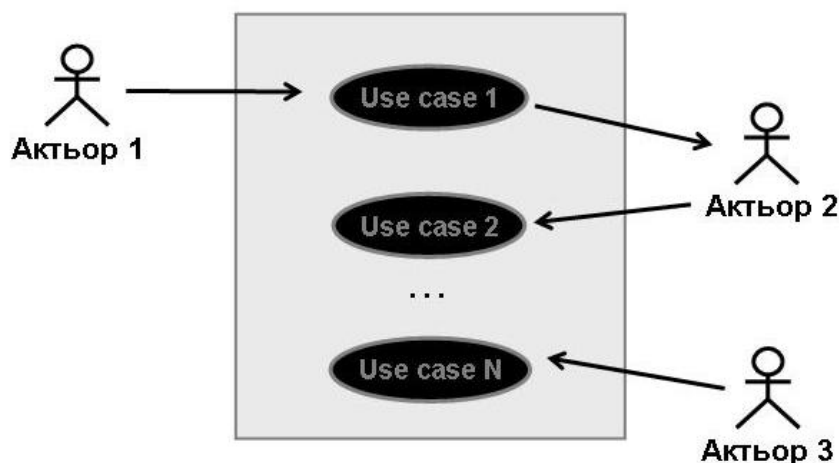
Основна парадигма на Унифицирания процес е разделянето на един проблем на малки части с цел постигането на по-добри резултати и това се осъществява чрез инкременталното използване на **потребителските случаи**, ключов артефакт, който насочва целия процес на разработка.

Потребителските случаи описват функционалността на системата като последователност от стъпки в един процес, който е ясен и разбираем за потребителите [Bittner & Spence, 2002].

Същите стъпки могат да бъдат използвани по-късно в обектно-ориентирания модел на бъдещата система на следващите фази на UP - *Детайлизиране и Изграждане* и така да се осъществи и връзката с разработчиците.

Всеки потребителски случай определя един специфичен начин за използване на системата, която е в процес на проектиране. Цялостното поведение на системата се описва от набор от такива потребителски случаи.

Потребителите, които си взаимодействат със системата, също са част от модела като актьори. Потребителските случаи и актьорите формират **модела на потребителските случаи на системата** [Cockburn, 2000].



Фигура 2.4. Модел на потребителските случаи

Актьорите също са важен елемент в модела на потребителските случаи. Един актьор може да бъде вид потребител или роля, която потребителите могат да изпълняват. [Cockburn, 2000]] разграничава два вида актьори – *главни и второстепенни*. Главният актьор има цели, които се изпълняват от системни услуги. Второстепенният актьор обикновено предоставя услуги в системата – например, администраторът на системата. [Larman, 2004] добавя т.нар. *поддържащи* актьори, за да се обхванат всички, които имат интереси в рамките на модела.

Така се описва не само поведението на системата, но и нейните граници и начинът, по който тя комуникира с актьори/потребители и други системи.

Както бе отбелязано, функционалните изисквания дават спецификацията на функциите, която трябва да поддържа системата, а нефункционалните изисквания – ограничения, които могат и да не са директно свързани с конкретни функции на системата, а да имат по-общ характер.

Унифицираният процес използва модела на потребителските случаи, за да представи функционалните изисквания. Що се отнася до нефункционалните, те са категоризирани в няколко категории. Акронимът **FURPS+** се използва, за да представи функционалните и нефункционалните изисквания при UP, като последните акцентират върху:

- използваемост (usability),
- надеждност (reliability),
- производителност (performance),
- възможности за поддръжка на системата (supportability).

Със знака „+” в името се означават помощни фактори като ограничения за ресурси, интерфейси, законови ограничения и др. [Larman, 2004].

Обектно-ориентиран анализ и проектиране

В последните години обектно-ориентираната парадигма се наложи не само при програмирането, но и при анализа и проектирането на софтуерни продукти, като успешно съчетава моделирането на тези продукти с приложната област. Реалната приложна област първо се представя като съвкупност от обекти и връзки и този модел поставя основата на бъдещата софтуерна система.

Концепциите на обектно-ориентирания (ОО) подход са многократно дискутирани в литературата [Bernd 2010], [Larman, 2004], [Schach 2010]. Тук ще отбележим само няколко елемента, които пряко засягат разработката на една софтуерни продукти на базата на този подход.

Разработката на една система може да се представи чрез три различни модела, които описват различни нейни страни [Bernd, 2010]:

- *функционален* модел, който описва функционалността на системата от гледна точка на потребителите и се представя чрез модела на потребителските случаи;
- *обектен* модел, който описва структурата на системата в термините на обекти и техните характеристики - атрибути, връзки и операции;
- *динамичен* модел, който описва вътрешното поведение на системата.

Различните модели се представят чрез съответни UML диаграми [Фаулър, 2007]. Диаграмите на класовете представят статичната структура на системата – обекти,

операции и връзки. Диаграмите на взаимодействието, диаграмите на състоянието и диаграмите на дейността описват взаимодействията между обектите и промяната на тяхното поведение.

Унифицираният процес, от своя страна, също е тясно свързан с ОО подход. На практика унифицираният процес дава рамката за обектно-ориентиран подход към разработката на софтуер.

По време на първата фаза - *Планиране*, се описва основната функционалност на разработвания софтуер под формата на потребителски случаи, които представят отделните стъпки на взаимодействието на потребителите със системата [Rosenberg & Stephens 2007]. Тези потребителски случаи изясняват функционалността на системата от гледна точка на потребителя и определят нейните граници. Така се формира функционалният модел на системата, описан чрез модела на потребителските случаи.

Въз основа на този модел, през следващите фази и итерации с помощта на различните UML диаграми се изграждат и детайлизират останалите два модела.

Мотивация на изследванията

Редица от основните концепции в областта на анализа и проектирането на ИС са експлоатирани от дълго време и все по-рядко тук се срещат изцяло нови идеи.

Една от областите, която в последните години предизвиква засилено внимание, е областта на управление на изискванията и на ролята им за построяването на модела на бъдещата система.

От множеството методи и техники за събиране, анализ и обработка на изискванията, контекстуалното проектиране предоставя един от най-систематичните и дисциплинирани подходи, които подпомагат анализаторите да извлекат необходимата информация от предметната област. Контекстуалното изследване, всички работни модели и специфичните сесии са реални стъпки, които ръководят проектантския екип в извличането на изискванията за разработката на един софтуерен продукт. Но тези стъпки не показват как да се включат елементите от работните модели в бъдещия проект на софтуерното приложение.

Унифицираният процес и по-конкретно RUP, от своя страна, предлагат добър механизъм на определяне на функционалността на системата чрез потребителски случаи и определят пътя на превръщането на информацията за използването на системата в проектантски решения.

Но когато се опитваме да въвеждаме тези методи в реални проекти, възникват няколко въпроса, свързани с първоначалното определяне на множеството от потребителските случаи, използвайки обектно-ориентиран подход.

Според принципите на UP началната фаза *Планиране*, може да бъде много кратка [Larman, 2004]. Обикновено като механизъм за определяне на актьорите, техните цели и първоначалните списъци с имената на потребителските случаи се

предлага среща на заинтересованите лица от вида „мозъчна атака”. В UP не се описва как точно да се изгради модел на потребителските случаи на базата на реалните описания на клиентите, които от една страна са обемни, а от друга – противоречиви и непълни. Но те са реалната основа, от която трябва да се оформи пълен и непротиворечив аналитичен модел, въз основа на който да се изгради бъдещият проект на системата.

За да се осъществи успешно този преход, изискванията трябва да бъдат цялостно обхванати, правилно разбрани и подходящо описани, да са разбираеми за клиентите и потребителите и да носят достатъчно информация за проектантите и разработчиците.

Следващата секция дава един възможен отговор на тези отворени въпроси като предлага интегриран подход, който използва основни принципи на контекстуалното проектиране и ги съчетава с методи и техники от UP/RUP в нова, интегрирана рамка.

В началото на секцията са описани два по-специфични резултата, свързани с приложения на RUP, които помагат за изясняването на важни детайли относно процеса на реалното използване на препоръчаните методологии.

Резултати

Практически приложения на RUP

Унифицираният процес е много подходящ за разработката на ИС, но в действителност може да се използва за широк кръг софтуерни приложения.

Този процес е използван при реализацията на **Интерактивна графична система за визуализация и трансформация на 3D обекти – StereometryLab** [Kaloyanova & Antonov, 2011]

Системата има за цел да визуализира различни геометрични обекти и да позволи редица операции върху тях – трансляция, ротация, симетрия и др. Въпреки че тук излиза от рамките на класическа информационна система, приложението на метода е много подходящо поради постепенността на изясняване на функционалността на приложението.

При разработката на системата са приложени принципите за определяне на функционалността чрез потребителски случаи и тяхната постепенна детайлизация. Едновременно с това, на практика е приложен механизмът на разделяне на процеса на разработка на фази с конкретен набор от итерации, които доведоха до успешната реализация на системата.

Таблица 2.1 описва броя итерации за отделните фази, както и промяната в статуса на различните артефакти, използвани в процеса на разработка (отразени са артефактите, засягащи анализа и проектирането). Приложена е схема, която включва по две итерации за фазите на *Детайлизиране* и *Изграждане* и по една за останалите фази.

Дисциплина	Артефакт	Плани- ране	Детайлизиране		Изграждане		Преда- ване
			И1	И2	И1	И2	
Бизнес модел	Домейн модел		с	п			
Изисквания	Модел на потребителските случаи	с	п	п	п	п	
	Визия	с	п	п			
	Допълнителна спецификация	с	п	п	п		
	Речник	с	п	п	п	п	
Проектиране	Проектиране на решението		с	п	п	п	п
Управление на проекта	План на проекта	с	п	п	п	п	п

Таблица 2.1. StereometryLab - фази, етапи и елементи

Използването на означенията

с – стартиране

п – подобрение

има за цел да покаже степента на разработване на съответните артефакти.

Работата по модела на потребителските случаи започва във фазата *Планиране*, като моделът се уточнява и усъвършенства и в следващите фази.

Обособени са няколко подгрупи потребителски случаи, които са свързани с следните дейности:

- Промяна на свойствата на сцената;
- Смяна на позицията на 3D обекти;
- Създаване на динамични 3D обекти;
- Създаване на 3D конфигурации;
- Извършване на допълнителни операции върху 3D обектите;
- Показване на допълнителна информация.

В рамките на цялата система са описани няколко десетки потребителски случаи. За тяхното описание е дефиниран шаблон (Таблица 2.2), който адекватно отразява необходимата информация за конкретния проект. Въвеждането на шаблона в първата фаза на проекта позволи пълно и съгласувано описание на функционалността в целия процес на разработка.

Описание на потребителски случай	
Име	
Главни актьори	
Заинтересовани лица	
Предусловия	
Следусловия	
Поток на събитията	
Главен сценарий	
Алтернативни сценарии	
Специални изисквания	
Отворени въпроси	

Таблица 2.2. Шаблон за описание на потребителски случай

Важна роля в анализа и проектирането на систематата играе моделът на приложната област (domain model), който описва концептуалните обекти и връзките между тях. Моделът е изграден на основа на съдържанието в описаните в първите 2 итерации потребителски случаи и е основа на диаграмите на класовете, описани в следващата фаза - *Изграждане*.

Като артефакт моделът на приложната област е ключов за по-нататъшното обектно-ориентирано проектиране на системата. Неговото създаване позволява съсредоточаване върху ключови обекти в системата и техните важни характеристики, които в последствие се детайлизират в диаграмите на класовете. Използването на редица диаграми на дейността, диаграми на последователността и диаграми на състоянията позволяват плавно преход към следващата фаза – *Изграждане*.

Фазата *Изграждане* обхваща две итерации. Тя се характеризира със стабилно множество от готови решения, всяко от които представя частична реализация на функционалността и води до постепенно доближаване до крайната версия.

Повечето артефакти са завършени изцяло, преди да се премине в последната фаза – *Предаване*, в която е направено тестването и валидирането на цялостната система спрямо предварителните изисквания.

Предложената конкретна схема на RUP е реализирана успешно и с помощта на един дипломант е създаден софтуерният продукт **Stereometry Lab**.

Интеграция на RUP с други подходи

Унифицираният процес и в частност RUP предлагат едно успешно решение за жизнения цикъл на разработка на информационни системи. Но когато става дума за интегрирани решения за големи организации, това не е достатъчно.

Една информационна система трябва не само да е проектирана добре, нейната реализация и връзките ѝ с останалите системи също са важни. Всяка ИС отразява спецификата на някаква част от дейността на една организация и обвързва работата на много хора и различни процеси. Как да се осигури синхрон на всички системи в рамките на една организация?

RUP, който обхваща пълния жизнен цикъл на разработка на един софтуерен продукт, може да се използва като основа, върху която да се надграждат елементи от други процеси и методологии.

В резултат на направен анализ на най-използваните концепции, методи, методологии за разработка на комплексни корпоративни системи бяха избрани няколко от тях: архитектура, ориентирана към услуги (*Service Oriented Architecture (SOA)*) [Papazoglou, 2007], архитектура на предприятие (*Enterprise Architecture (EA)*) [Scott, 2005], [Doucet, 2008], управление на бизнес процеси (*Business Process Management (BPM)*) [Draheim, 2010] и бяха разгледани от гледна точка на техните възможности за съвместяване с РУП.

Анализите, сравненията и обобщенията позволиха изграждането на единна концепция за интегриране на тези няколко широко използвани технологии в една обща рамка [Napolì & Kaloyanova, 2011].

В основата на предложения подход е методологията на RUP като основен, базов процес, с неговите четири фази и завършен итеративен и инкрементален подход. Към тези фази са добавени елементи от другите методологии - задачи, участници (роли), работни продукти и други елементи.

Като метод RUP предлага много добро решение за жизнения цикъл на разработка на софтуерен продукт. Той е особено подходящ за разработката на ИС, но всяка една организация използва информационните системи с определена цел, тясно свързана с нейния бизнес и специфичен начин на работа. Затова в предложения подход са включени някои от основните дейности, типове роли и работни продукти от архитектурния метод TOGAF [Temnenco, 2010]. TOGAF също използва концепцията за повтарящ се жизнен цикъл и може да се съпостави с фазите на RUP, като се отчитат архитектурните концепции и бизнес визията. Това позволява дейности, свързани с архитектурните аспекти, да се включат в началната фаза на разработка.

Включването на TOGAF с неговия фокус върху бизнес аспектите на софтуерните процеси е в пряка връзка с методите и практиките на управлението на бизнес процесите и влияе положително на работата по някои елементи като бизнес мисия, визия, бизнес правила, бизнес потребителски случаи.

Допълнително, чрез рамката на BPM, се осъществява връзката с архитектурата, ориентирана към услуги (SOA), което позволява добавянето на нейните дейности и работни продукти, релевантни към развитието на услугите: изисквания към услугите - модели на услугите, портфолиото от услуги и др., както и включването на допълнителни роли, като SOA архитект и проектант (дизайнер).

PHASES	Tasks	Workers	Workproducts
- TOGAF - ENTERPRISE ARCHITECTURE CYCLES	<ul style="list-style-type: none"> - Phase A: Architecture vision - Phase B: Business Architecture - Phase C: Information Systems Arch. - Phase D: Technology Architecture - Phase E: Opportunities and Solutions 	Enterprise Architect	Enterprise Architecture: <ul style="list-style-type: none"> - Business Architecture - Application Architecture - Information Architecture - Technology Architecture
RUP: INCEPTION PHASE (RELEVANT ACTIVITIES)			
(U) <i>Conceive New Project</i>	(U) <i>Develop Business Case</i>	Project Manager Business Architect	Business Case
(U) <i>Identify and Assess Risks</i>	(U) <i>Identify Potential Risks</i> (U) <i>Analyze and Prioritize Risks</i>	Project Manager	Risk List
(U) <i>Plan the Project</i>	(U) <i>Plan Phases and Iterations</i>	Project Manager	Software Development Plan
(U) <i>Manage the Scope of the System</i>	(U) <i>Prioritize Use Cases</i> (U) <i>Manage Dependencies</i>	SOA SW Architect	<ul style="list-style-type: none"> - Software Architecture - Software Requirement - Service Dependencies
(U) <i>Perform Architectural Synthesis</i>	(N) <i>Enterprise Architecture Review</i>	SOA SW Architect Enterprise Architect	- Enterprise Architecture
	(U) <i>Architectural Analysis</i>		- Service Portfolio
	(U) <i>Construct Architectural POC</i>		
	(U) <i>Assess Viability of Arch POC</i>		
(U) <i>Define Evaluation Mission</i>	(U) <i>Identify Targets of Test</i>	SOA Designer	<ul style="list-style-type: none"> - Software Architecture - Test Strategy
(N) <i>Identify Business Processes and Explore Process Automation</i>	(N) <i>Business Architectural Analysis</i> (N) <i>Identify Business Goals</i> (N) <i>Identify Services</i>	Business Analyst Business Architect	<ul style="list-style-type: none"> - Business Architecture - Business Goal - Business Use Cases - Business Rules
RUP: ELABORATION PHASE (RELEVANT ACTIVITIES)			
(U) <i>Refine the System Definition</i>	(U) <i>Detail the Software Requirements</i>	Requirements Specifier	<ul style="list-style-type: none"> - Service Dependencies - Software Requirements
(U) <i>Define Candidate Architecture</i>	(U) <i>Architectural Analysis</i>	SOA SW Architect System Analyst	- Enterprise Architecture
	(U) <i>Architectural Analysis</i>	Enterprise Architect	<ul style="list-style-type: none"> - Service Model - Software Architecture.
(U) <i>Refine the Architecture</i>	(U) <i>Identify Design Mechanisms</i>	SOA SW Architect SOA Designer	<ul style="list-style-type: none"> - Service Model - Software Architecture - Service Partitions - Service Collaborations - Service Specifications
	(U) <i>Structure the Implementation Model</i>	Enterprise Architect	<ul style="list-style-type: none"> - Enterprise Architecture - Implementation Model
(U) <i>Develop Components</i>			-

- (N) Service Identification			-
· (N) Domain Decomposition	(N) Functional Area Analysis	Business Architect	- Business Domain - Business Architecture - Business Analysis
	(N) Refine Business Use Case	Business Analyst	- Business Use Case
	(N) Business Process Analysis	SOA SW Architect	- Service Model
	(N) Identify Business Goals and KPIs	Business Analyst	- Business Goal
· (N) Goal Service Modelling	(N) Identify and Associate Services to Goals	SOA Designer	- Business Goal - Service Model
	(N) Existing Asset Analysis	SOA SW Architect	- Service Model - Software Architecture - Deployment Model - Design Model
· (N) Existing Asset Analysis	(N) Data Model Analysis	SOA Database Designer SOA Designer	
	(N) Business Rule Analysis	Business Analyst SOA Designer	
	- (N) Service Specification		
· (N) Perform Service Specification	(N) Apply Services Litmus Tests	SOA SW Architect	- Service Model
	(N) Service Specification	SOA Designer SOA SW Architect	- Design Model - Service Model - Software Architecture
		(N) Message Design	SOA Designer
	· (U) Perform Subsystem Analysis	(U) Subsystem Design	SOA Designer
· (N) Perform Component Specification	(N) Component Specification	SOA Designer	- Design Subsystem - Service Component
- (N) Service Realization			
· (N) Realization Decisions	(N) Perform Realization Decisions	SOA Designer SOA SW Architect	- Design Model - Service Component - Reference Architecture
Integrate and Test	<no sensitive modifications>	Test Designer Test Analyst Process Engineer SOA SW Architect	- Service Model
- Verify Test Approach			- Service Interfaces
- Integrate and Validate Build			- Service Relationships
· Integrate each Subsystem (...)			- Service Dependences
- Test and Evaluate			- Service Boundaries
· Test and Evaluate (...)			- Service Specification - Service Implementation - Service Composition
RUP: CONSTRUCTION PHASE (*)			
RUP: TRANSITION PHASE (*)			
(N) Solution Architecture Integration with Enterprise	(N) Identify Enterprise Architecture Integration Changes	SOA SW Architect Enterprise Architect	Enterprise Architecture Integration Changes

Architecture			
- TOGAF - ENTERPRISE ARCHITECTURE CYCLES	Phase H: Architecture Change Management	Enterprise Architect	Enterprise Architecture: - Business Architecture - Application Architecture - Information Architecture - Technology Architecture
Domain markers explained below: <i>(*)Activities from the construction and transition phases are not included on this table.</i> <i>(U) Represents original activities in the RUP framework that have been updated</i> <i>(N) Represents new activities that were not existing in the RUP framework</i>			
TOGAF Enterprise Architecture (TOGAF)			
Service Oriented Analysis, Design & Architecture (SOA)			
Business Process Management (BPM)			

Таблица 2.3. Интегриране на RUP, TOGAF, SOA, BPM¹

Таблица 2.3 представя част от схемата на предложената единна методология, изобразяваща етапи, задачи, роли и работни продукти. Представени са основните елементи за първите 2 фази, които са пряко свързани с дейностите анализ и проектиране.

Интегрирането на представените методологии позволява отчитането на различните страни на разработката на една система и съвместява ИС като софтуерен продукт с архитектурните и бизнес концепции на организацията. Предложената рамка е подходяща за използване при изграждането на сложни системи в големи организации.

Интегриран подход за анализ и проектиране на информационни системи

Натрупаният опит в приложението на RUP и изследването на връзките му с други методологии неизменно демонстрира основните предимства на Унифицирания процес. Едно от тях безспорно е механизмът на определяне на изискванията. Итеративният процес на определяне на функционалността на системата чрез потребителски случаи е полезен и ценен подход.

Контекстуалното проектиране, от своя страна, предоставя добро обяснение на ролята на клиента при проектирането на една софтуерната система. Неговите работни модели показват различни гледни точки към системата и дават възможност цялата многостранна информация да се опише със съответни диаграми за всеки модел.

¹ *Забележка:* Поради липсата на добре установена терминология на български език, текстът в таблицата е на английски език.

Но контекстуалното проектиране не изяснява добре как тази информация се трансформира в проектните решения за системата, а остава по-скоро в обсега на анализирането и обобщаването на изискванията.

В интегрирания подход, предложен от автора в [Kaloyanova, 2012-A] и описан по-долу, контекстуалното проектиране не се използва по строго линеен начин, както би била обичайната практика, а негови методи и техники са комбинирани с елементи от създаването на модела на потребителските случаи и е приложен последователният и надграждащ подход на UP.

Стъпка 1

Предложеният подход стартира със серия интервюта на реални потребители. Интервютата се провеждат в работната среда на потребителите. На основата на събраната от тях информация, интервюиращите подготвят:

- модели на дейностите;
- модели на последователността;
- модели на работните продукти.

Самите модели се подготвят и обсъждат в интерпретационни сесии, в които всички аналитици и други специалисти, участвали в интервютата, обсъждат получената информация и съпоставят своите индивидуални модели.

След обединение на индивидуалните модели, в друга сесия (консолидационна) се оформят обобщен модел на дейностите; обобщени модели на последователността и обобщени модели на работните продукти.

От тези модели се получават няколко основни резултата:

- Отбелязани са **групи от роли/личности**, които са свързани с работата на системата, уточнени са основните взаимодействия между тях;
- Някои **последователности от действия** са описани, като са отбелязани цели, събития, които активират последователностите, потенциални прекъсвания на последователностите, възможни разклонения и алтернативни действия;
- Описани са **работни продукти**, които са свързани с работата на системата.

Според препоръките на контекстуалното проектиране работата продължава с детайлизиране на визията, подготовка на сторибордове и оформянето на средата на потребителя, но според нашите наблюдения в този момент проектантският екип не е с достатъчно познания, за да се създаде цялостна визия на системата. Това е една от причините за малкия брой на проекти, които прилагат контекстуалното проектиране в неговата чиста форма.

Стъпка 2

Като втора стъпка на нашия подход, предлагаме да се продължи с по-дисциплинирания подход на UP и да се използва информацията от работните модели за определяне на потребителските случаи. На този етап от проекта работните модели са

консолидирани, дефинирани са основните потребители на реалните работни процеси и са показани общи модели на работа.

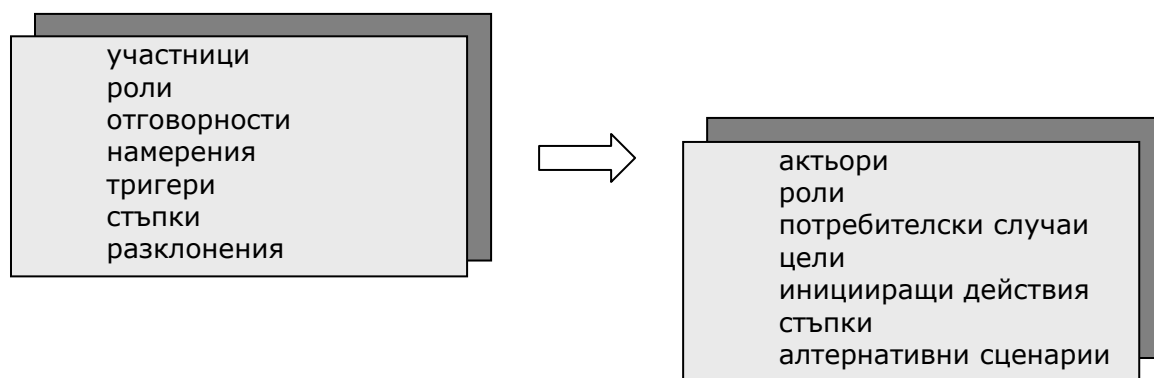
Тъй като моделите на дейностите описват роли и дейности, те са добра начална точка за определяне на различните участници/роли, които работят със системата. Тези модели показват всички значими потребители, техните дейности, важни цели и някои артефакти. Така може да се определи началното множество от участници в работата на системата.

Дейностите от моделите на последователността, както и намеренията (целите), тригерите, стъпките и алтернативните сценарии показват начините, по които потребителите работят със системата на различни нива. Макар и да не са описани в много детайли, те дават достатъчна представа за функциите, които системата трябва да предоставя.

Следователно, от информацията за модела на дейностите и моделите на последователността, голяма част от актьорите, техните основни цели и първоначален списък на потребителските случаи могат да бъдат определени на сесия от тип „мозъчна атака“, в която участват всички заинтересовани лица - потребители и разработчици. Въз основа на тези два модела, границите на системата също могат да бъдат определени.

В края на тази стъпка съществува списък на голяма част от потребителските случаи, които са описани с техните названия. Някои от тях могат да бъдат описани в кратък формат.

На фиг. 2.5 са представени основни елементи от моделите на контекстуалното проектиране и съответните елементи в модела на потребителските случаи.



Фигура 2.5. Трансформация на елементи от моделите на контекстуалното проектиране в модела на потребителските случаи

По такъв начин поне 10 % от потребителските случаи в този начален списък да бъдат напълно описани, въз основа на моделите на последователността. Това могат да бъдат най-ясните, най-рисковите или най-трудните потребителски случаи.

Важен момент тук е определянето на *шаблон за пълно описание* на потребителските случаи, с правилно подбрани секции, които да позволят достатъчно подробно описание на функционалността.

Така процесът на пълното описание на потребителските случаи започва в началото на проекта и продължава през следващите фази и итерации, позволявайки постъпково реализиране на функционалността на системата.

Таблица 2.4 представя резултатите от няколко проекта, които потвърждават ефективността на предложения подход на базата на съответствието между броя на направените модели и броя на общо генерираните потребителските случаи и на тези, описани в пълен формат.

Проект	Брой интер-вюта	Модели на дейностите	Модели на последователностите	Брой актьори	Брой потр. случаи	Брой потр. случаи в пълен формат
Проект 1	2	2	3	6	30	3
Проект 2	3	3	7	12	23	3
Проект 3	4	4 + 1	4	6	29	3
Проект 4	4	4	3	7	21	2
Проект 5	2	2	4	3	26	3
Проект 6	3	3	2	7	17	2

Таблица 2.4. Проекти, модели и елементи

Стъпка 3

След определянето на границите на системата и списъка на потребителските случаи, представата за обхвата на работата е далеч по-ясна и визията за системата може да бъде представена по-детайлно.

Концепцията за визия има представяне и в двете методологии – контекстуалното проектиране и UP, но с различни интерпретации. Тук е мястото, където двете визии – визуалната от контекстуалното проектиране и текстовата от UP, могат да се интегрират на ново равнище, като трета стъпка от нашия подход. Комбинацията на изображения и текстово описание представят пълната картина на това, което трябва да се направи в системата и е много по-разбираема за потребителите.

Сторибордовете са друг елемент на контекстуалното проектиране, който използва графично представяне. Тъй като те биха могли да представляват не само на дейностите на потребителите на системата, но също така и някои неавтоматизирани, човешки действия, първоначалните концепции за потребителския интерфейс, бизнес правила и т.н. За дизайнерския екип не е лесно да избере кои работни задачи да бъдат представени в тези серии от снимки и текст на по-ранните етапи от анализа. Сега,

когато имаме набор от потребителски случаи на използването на системата с определени предпоставки (условия) и постигнати резултати, комбинирането им в обобщена серия от дейности за постигане на определена бизнес цел и като основа за описание на бизнес процеси, е много по-лесно.

Стъпка 4

Фокусът на тази стъпка е върху информационния модел на системата. Домейн моделът е елемент от UP, който открива реални артефакти и концепции като част от специализираната предметна област и представя съществените връзки между тях.

Унифицираният процес предлага добри принципи за определяне на модела на приложната област. Списъците на предефинирани концептуални класове, които той предлага, насочват дизайнерите да изберат правилната форма на елементите на информационната структура. Описанието на потребителските случаи е другият полезен източник за елементи за модела.

Като цяло, контекстуалното проектиране не се фокусира върху информационния модел на системата. Но много важни обекти и понятия могат да бъдат намерени в неговите модели - обобщения модел на дейността и моделите на артефактите. Визията също може да се използва за дефиниране на някои концептуални класове.

Така, след изпълнението на предложените четири стъпки, е създадена добра основа за изграждане на проекта на системата и работата може да продължи със следващите три фази по класическата схема на UP/RUP.

Основните преимущества на предложената рамка от няколко стъпки са в по-детайлното описание на дейностите при събиране, обобщаване и анализиране на изискванията на потребителите и в съчетаването на различни методи и техники в пълноценен инструментариум, който съпровожда тези стъпки и създава реална връзка между потребителските изисквания и разработвания проект.

В допълнение, този подход осигурява на екипа, разработващ системата:

- работа в контекста на предметната област;
- работа в партньорство с всички заинтересовани лица;
- визуализация на концепциите;
- определяне на приоритети;
- гъвкавост.

Приложение на подхода в обучението на студенти

Предложеният подход може да се използва както при разработката на информационни системи в реална работна среда в софтуерни фирми, така и в процеса на обучение на студенти.

Работата по проекти е въведена в програмите по *Информационни системи* (бакалавърски и магистърски) на няколко нива и засяга области, свързани с анализ,

проектиране и реализация на бази от данни и информационни системи и управление на проекти за разработването на информационни системи [Kalojanova, 2012-B]. Целта на проектите е да подкрепят академичните курсове с изграждане на практически умения и опит за работа в екип, като неразделна част от обучението.

Използвани са три основни курса от учебната програма [Torі, 2010]:

- Бази от данни;
- Анализ и проектиране на ИС;
- Управление на проекти.

Студентите са организирани в екипи от 5 до 10 човека. На екипите се възлагат различни задачи (групови и индивидуални), като задачите се определят в началото на семестъра и следват строг график на предаване.

Работата по проектите в 3-те курса позволява на студентите да разглежат разработката на ИС от различни гледни точки. В курса по *Бази от данни* се набляга на проектирането на модела на данните. Студентите представят ER model на базата от данни и реализират концепцията със съответните SQL скриптове. В следващия курс - *Анализ и проектиране на ИС*, се прилага методологията на анализ на потребителските изисквания, описана по-горе. Курсът по *Управление на проекти* показва на студентите процеса на разработка на една информационна система като софтуерен продукт – планирането, организацията, методите за оценяване на сложността и обема на проекта, управлението на риска, въпросите, свързани с качеството на проекта [Kanabar, 2008].

Работата по проектите позволява на студентите да добият реална представа за приложението и полезността на голяма част от изучаваните теоретични концепции.

В допълнение, редица полезни практики съпътстват организацията на работата по проектите. Те са обобщени в [Kalojanova, 2013]. Сред най-важните от тях са :

- Работа в екип;
- Работа в реална предметна област;
- Подготовка на различни модели;
- Визуализация на идеи и концепции;
- Изпълнение на различни роли;
- Поддържане на шаблони за различни документи;
- Подготовка и представяне на презентации;
- Критични доклади върху работата на други екипи.

До момента повече от триста и петдесет студента са участвали във всички курсове. В таблица 2.5 е систематизирано приложението на практиките в отделните курсове.

Курсове	Цели на проекта
Бази от данни	<div> <div>Модели</div> <div>Екипни презентации</div> <div>Работа в екип</div> </div>
Анализ и проектиране на ИС	<div> <div>Модели</div> <div>Екипни презентации</div> <div>Индивидуални презентации</div> <div>Критични доклади</div> <div>Работа в екип</div> <div>Реална област</div> <div>Визуализация</div> <div>Роли</div> <div>Постъпкова работа</div> <div>Рецензиране на други проекти</div> </div>
Управление на проекти	<div> <div>Шаблони</div> <div>Роли</div> <div>Работа в екип</div> <div>Индивидуални презентации</div> <div>Екипни презентации</div> <div>Модели</div> </div>

Таблица 2.5. Курсове и практики

Получените резултати показват по-добро разбиране и усвояване на теоретичните концепции, дължащо се на тяхното реално използване като част от работата по проектите. Чрез прилагането на представените добри практики студентите се доближават до действителната работна среда в ИТ фирмите.

Поставянето им в различни роли по време на работата по проектите дава възможност на студентите да видят различни страни на сложните процеси, които съпътстват разработването на информационните системи. Например, въвеждането на критичните доклади на един екип относно проекта на някой от останалите екипи е много полезна практика, която им позволява не само да се поставят в ролята на оценяващи, но и да се запознаят подробно с други гледни точки и да преосмислят своите идеи.

Много от предложените практики насърчават студентите да търсят нестандартни, творчески решения и да свързват теоретичните концепции с реални елементи от различни предметни области.

Изводи

В тази глава представихме подход за структурирано събиране, анализиране и обобщаване на потребителските изисквания към разработката на информационни системи и използване на получените резултати за проектирането на системите.

Макар и да използва познати концепции, методи и техники, предложеният тук подход ги разглежда и свързва по нов начин.

Основният резултат от предложения подход е ясната последователност от стъпки, позволяваща големият обем потребителски описания да се превърне в конструктивни проектни решения. Друго предимство на подхода е свързано с това, че се предлагат конкретни методи, техники, практики, използване на набор от средства за моделиране модели, което го прави удобен за практическо приложение и позволява да се използва не само от професионалисти в сферата на разработката на информационни системи, но и за обучение на студенти в тази област.

Представеният подход е в процес на развитие. Предстои да бъде разширен, уточнен и допълнен с елементи, увеличаващи продуктивността на работата, както и с конкретни практически методи, които улесняват приложението му.

ГЛАВА 3: АНАЛИЗ И ПРОЕКТИРАНЕ ПРИ МНОГОМЕРНИЯ МОДЕЛ НА ДАННИТЕ

В тази глава са представени резултати от изследвания в областта на аналитичните системи. Дискутират се въпроси, свързани с модела на данните в тези системи, управление на бизнес зависимостите в моделираните предметни области, както и подходи за оптимизация на процесите на изчистване и консолидиране на данните.

Основни концепции при изграждане на многомерни модели на данните

Функционалността, която се предоставя от една ИС, оказва съществено влияние върху модела на данните и архитектурата на системата. В този секция са представени основни концепции при изграждането на аналитичните система. Анализирани са специфични особености на многомерния модел на данните, като основен модел за този тип системи.

Концепцията за „хранилище на данни” или „склад от данни” не е нова. Тя датира от седемдесетте години на миналия век и се свързва с някои разработки на известни компании, включително и на IBM. Но областта започва бурно развитие едва след появата на дефиницията на Инмон за „склад от данни”. В своята книга „Building the Data Warehouse” [Inmon, 1992] той определя **склада от данни** като „предметно-ориентирана, интегрирана, относително неизменна съвкупност от данни, с времеви характеристики, която е предназначена да поддържа процеса на вземане на управленски решения.”

Всеки склад от данни съдържа огромни количества обобщени данни за една организация, които обикновено включват данни от различните оперативни системи на организацията. Най-често тези данни се използват за анализи, които помагат на ръководството да взема правилни бизнес решения [Inmon, 2005].

Няколко основни типа методи се използват при работа със складовете от данни:

-- **статистически методи** - хипотетични тестове, тестове за съвпадение на разпределенията, анализи на вариантите и др.;

-- **интерактивна аналитична обработка** (OnLine Analytical Processing - OLAP) - анализ на данните и вземане на стратегически управленски решения;

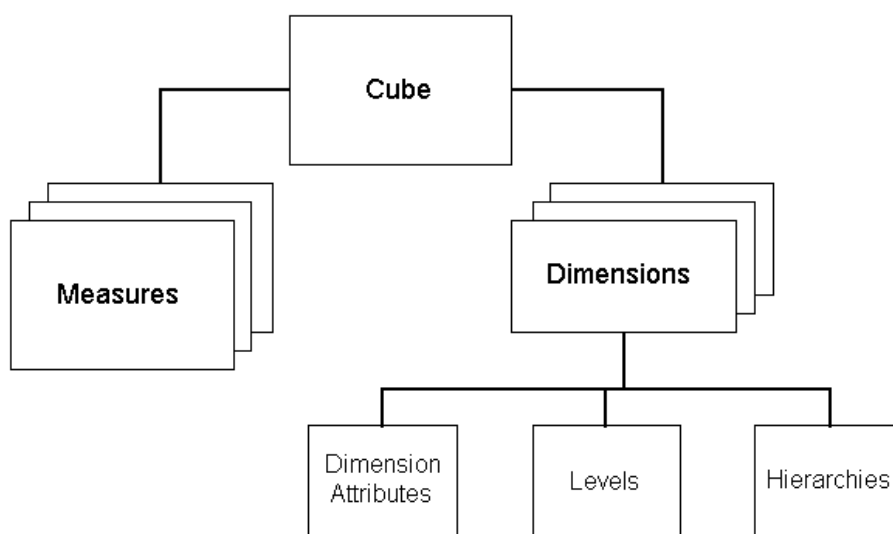
-- **методи за извличане на закономерности от данни** (Data Mining methods - DM) – позволяват откриване на нови, неизвестни зависимости в складовете от данни чрез използване на техники и алгоритми от различни области като изкуствен интелект, машинно самообучение и др.

Модели на данните

Транзакционните системи използват предимно модели на данни от тип обекти/същности и връзки между тях. Тези модели на данните са подходящи за интерактивна обработка на транзакции, но те не предоставят средства за обработка и управление на обобщени данни по различни показатели (период от време, място), които са в основата на аналитичните (OLAP) приложения. Последните изискват друг тип структуриране на данните, следователни други методи за анализ и проектиране.

Моделът на данните в една аналитична система изисква предметно-ориентирана схема, която би улеснила анализа на огромното количество данни в нея. От друга страна, тук не е необходимо да се поддържат операции за непрекъснато въвеждане и обновяване на данните, защото складът от данни се обновява, като се допълва с новите данни със съответните времеви характеристики.

Най-разпространеният модел, използван в момента в аналитичните системи, е *многомерният модел на данните*, който представя данните като един куб в n-мерното пространство. Кубът се дефинира чрез факти и измерения, свързани с фактите. Този вид представяне позволява данните да бъдат разглеждани по различните измерения или комбинации от измерения за различни видове анализи- фигура 3.1.

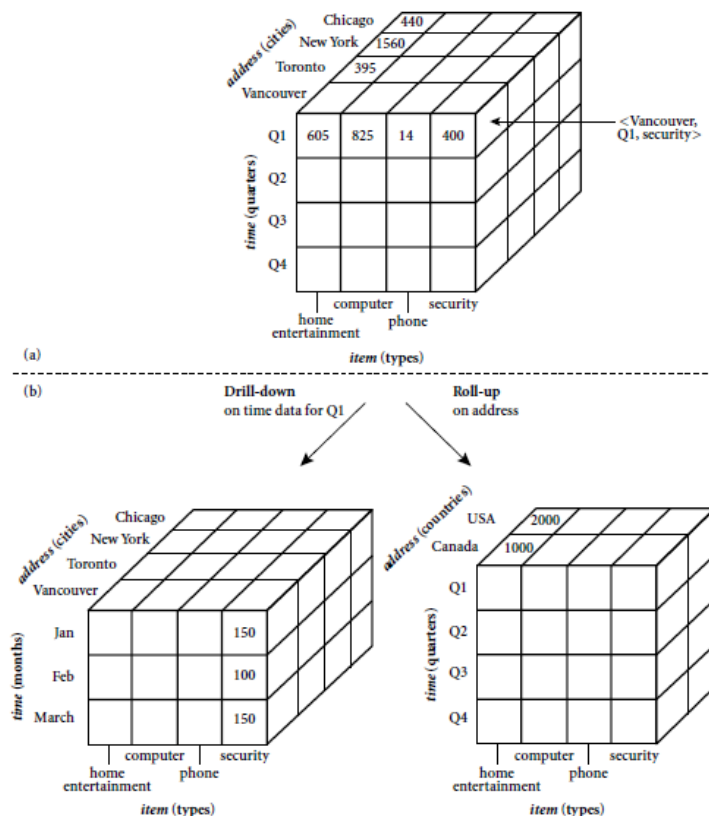


Фигура 3.1 Обобщено представяне на многомерния модел [OLAP DML, 2002]

Многомерният модел на данни е тематично-ориентиран модел. Обикновено се организира около някаква централна тема (например “продажби”), която се представя чрез така наречената *факт* таблица. Тя съдържа имената на факти или мерки, както и връзки към всяка от съответните размерности [Inmon, 2005].

Измеренията (dimentions) са множеството от критерии, от гледна точка на които една организация иска да съхранява своята информация – време, географско положение

и т.н. Всяко измерение включва йерархия, която позволява да се правят анализи на различни нива на обобщение. Например, йерархията за времето може да включва ден, седмица, месец, тримесечие и т.н.



Фигура 3.2 Операции върху многомерен куб [Han & Kamber, 2006]

Многомерният модел позволява на потребителя да разглежда и анализира данните от различни гледни точки като поддържа набор от полезни операции: [Han & Kamber, 2006]

- **Увеличаване на нивото на обобщеност(roll-up)** - преминаване от по-ниски нива на обобщеност към по-високи;
- **Намаляване на нивото на обобщеност (drill-down)** - увеличаване на нивото на детайлност;
- **Изрязване на под-куб (slice & dice)** чрез избор на две или повече размерности;
- **Завъртане(pivot/ rotate)** - операция по визуализация, която променя осите на показаните данни, за да даде алтернативен поглед върху данните.

Кубът от данни дава удобна представа за многомерния модел. Но физическото съхранение на данните използва различни архитектурни подходи.

В зависимост от връзките между размерностите и фактите, които определят куба, съществуват няколко възможни схеми на релационно представяне: *"звезда"*, *"снежинка"* и *"съзвездие от факти"*.

При схема *"звезда"* кубът се състои от централна факт таблица и измерения (по едно за всяка размерност), които са свързани към факт таблицата. Схемата *"снежинка"* е вариант на схемата *"звезда"*. При нея част от измеренията са *нормализирани*, разделяйки данните в допълнителни таблици. По този начин нормализираното измерение заема по-малко пространство, но от друга страна понижава бързодействието.

При по-сложните приложения се изисква няколко факт таблици да имат общи измерения - схема *"съзвездие от факти"*.

Видове OLAP системи

От гледна точка на архитектурното решение, за многомерния модел съществуват два основни типа – многомерен OLAP (MOLAP) и релационен OLAP (ROLAP). Третият тип - хибридният OLAP (HOLAP) е комбинация на двете технологии [Kaser & Lemire, 2003].

- ROLAP използва релационни таблици, за да симулира многомерен куб. Тази технология позволява работа с големи обеми от данни и интеграция със съществуващите релационни системи чрез SQL заявки, но е по-бавна.
- При MOLAP данните се съхраняват в многомерен куб, който е оптимизиран за сложни изчисления и поддържа висока производителност. Методът работи с големи обеми памет, необходима за съхранението на всички възможни комбинации по измеренията и техните обобщения и изисква значително време за първоначалното зареждане и изчисления.
- HOLAP използва MOLAP за справки и агрегации на високо ниво, а при преглед на детайлите се правят заявки към релационна ROLAP схема.

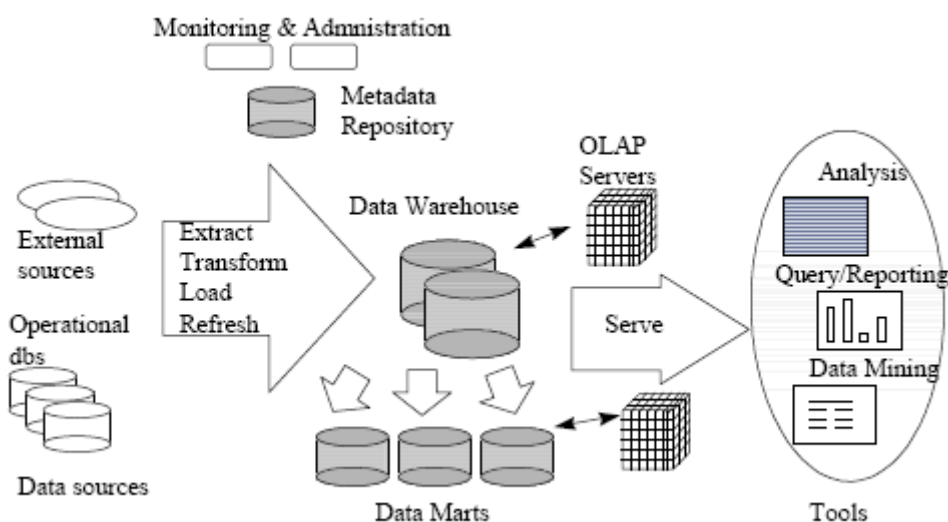
Естество на данните

Складовете от данни съдържат обединени данни, които постъпват от много различни източници: различни транзакционни системи, файлове, интернет източници (фигура 3.3). При реализацията на транзакционните бази от данни обикновено са били решавани конкретни задачи, без да се отчита евентуална бъдеща интеграция на системите. В резултат, липсва съвместимост по различни направления - методите на кодиране, конвенциите на именуване, физическите атрибути, метриките за измерване на атрибутите. Това предизвиква много проблеми при интегрирането на данните от различните системи.

За да бъдат използвани по последователен начин в аналитичната система, те се извличат от различни оперативни системи и други източници, изчистват се от противоречия, преобразуват се в съгласуван вид и се зареждат в новата система. Този

процес на извличане, трансформиране и зареждане, познат под името **ETL (Extraction, Transformation, Loading)**, има за основна цел обобщаването на информацията и трансформирането ѝ в нов, обобщен модел [Kimbal, 2002]. Това е най-тежкият, продължителен и ресурсоемък процес при създаването на склада от данни, който обикновено отнема около 70 % от времето и усилията за разработка и изисква богат набор от дейности и средства, свързани с цялостната обработка на данните [Thilini & Watson, 2006].

Фигура 3.3 представя в обобщен вид процесите при изграждането на склада от данни.



Фигура 3.3.Склад от данни- архитектура и интеграция на данни [Thilini & Watson, 2006]

Като първа стъпка се прилагат различни методи за извличане на данни от множество различни оперативни бази от данни и други външни източници, както и автоматизирани средства за изчистване на данните от грешки, противоречия и несъвместимост [Chaudhuri& Dayal, 1997].

Следват операциите по обединението на данните, зареждането им в склада от данни и периодичното им обновяване за поддържане на тяхната актуалност.

Така съхранените данни са обект на приложение от страна на специализираните средства за анализ на данни, изготвянето на различни справки, методи за извличане на закономерности от данни.

DW 2.0

Непрекъснатото нарастване на обемите от данни и бързо променящите се технологии за съхранение на информацията в последните години [Kozielski & Robert, 2009] доведоха до появата на ново поколение складове от данни, наречено **DW 2.0**

[Inmon & Strauss & Neushloss, 2008]. Новата концепция решава проблеми като интеграция на неструктурирани и структурирани данни, поддръжка на мета-данните като неразделна част от архитектурата и гъвкавост спрямо бързо променящите се бизнес изисквания.

Анализът на двете генерации складове от данни показва няколко основни разлики. Те засягат основни аспекти на този тип системи - методологията за построяване, същинска разработка и реализация на крайния продукт.

Разработката на първото поколение складове от данни следва каскадния модел, ETL процеса и използва специализирана област за предварителна обработка на данните - *staging area*.

При второто поколение водещ е спираловидният модел на разработка, който се прилага чрез т.н. *седем потоков подход*, координиращ изчистването, съпоставянето и корпоративното моделиране на данните, както и някои други дейности по управление и интеграция на данните. Като цяло DW 2.0 прилага и по-детайлен механизъм за *жизнен цикъл на данните*, в който данните преминават през четири сектора – интерактивен, интегриран, архивен и междинен сектор между последните два [Inmon et al., 2008].

За разлика от първото поколение складове от данни, което се състои изцяло от структурирани данни, DW 2.0 интегрира структурирани и неструктурирани данни. Концепцията за мета-данните също еволюира в DW 2.0. Например, всеки сектор поддържа собствени метаданни.

Важен аспект на DW 2.0 е, че архитектурата на склада от данни се разполага в среда, в която технологията може да се променя толкова бързо, колкото и променящите се бизнес изисквания.

Основни предизвикателства и ограничения на многомерния модел

Многомерният модел е основен логически модел в системите за аналитична обработка. Въпреки редица предимства в представянето на информацията по структуриран и логичен начин, някои специфични обстоятелства затрудняват неговата реализация:

- **Обобщение на данните.** Използването на този род системи за вземане на управленски решения изисква разглеждането на данни, които се извличат от множество нееднородни източници.
- **Съгласуване на данните.** Различните източници могат да съдържат данни с различна чистота или да използват несъгласувано представяне, кодиране и формат, които трябва да бъдат съгласувани и обединени.
- **Обем на данните.** В допълнение към обединените данни от различни източници, складовете от данни съхраняват и исторически данни, и обобщени данни.

- **Производителност.** Големите обеми данни изискват продължителна обработка, затова ускоряването на времето за изпълнение на запитванията и анализите е от първостепенно значение.
- **Визуализация на резултатите.** Предлагането на адекватни форми за наблюдение на ключови бизнес индикатори и представянето на резултатите в подходящи вид има важна роля в ефективното използване на хранилищата от данни.

Посочените проблеми изискват сериозно внимание на всички етапи в разработката на складовете от данни, но основната тежест е върху ранните етапи - анализът и проектирането, които при аналитичните системи са свързани не само с изискванията на потребителите относно функционалността на бъдещата система. От направеното по-горе изложение се вижда, че важна роля играят данните, които ще бъдат вложени в системата и които идват от други, вече съществуващи системи. Това изисква допълнителни усилия в процеса на разработка за изясняване на природата на тези данни с цел ефективното им трансформиране в новия модел.

Редица методи, методологии и подходи могат да се използват при анализа на данните, които ще бъдат включени в обобщения модел на бъдещия склад от данни.

В следващата секция ще бъдат представени някои резултати, които могат да бъдат обобщени в две направления:

- анализ на данните, постъпващи от различни източници, методи и средства за изчистване и съгласуване на данните;
- използване на знанията за естеството на данните за оптимизиране на многомерния модел.

Резултати

Оптимизация на процесите на изчистване и обобщаване на данните

Основната задача на складовете от данни е съхранението на големи количества информация във вид, удобен за *различни анализи*, които се използват най-вече от големите организации за бизнес цели. Освен статистики, обобщения и сравнения на данни от предишни години, складовете на данни се използват и за разкриване на скрити, неподозирани зависимости между данните чрез *методите за разкриване на знания*. Тези методи обикновено се прилагат върху обобщените данни, така че организацията на склада на данни и най-вече моделът на детайлните данни имат съществено значение за ефективността на приложението.

В [Naydenova & Kaloyanova, 2005] са изведени други допирни точки между складовете от данни, аналитичната обработка и методите за разкриване на закономерности от данни, които са систематизирани в няколко направления:

Интегриране на аналитичната обработка с методи за разкриване на знания

Интегрирането на техниките за многомерен анализ с методите за извличане на закономерности от данни (т. нар. On-Line Analytical Mining) значително увеличава силата и гъвкавостта на проучвателния процес. Например, разделянето на подкубове и прилагането на методите за разкриване закономерности от данни върху тях би улеснило аналитичния процес чрез използването на по-малки обеми от данни и обединяване на резултатите от тях.

Прилагане на методи за разкриване на знания директно върху релационни бази от данни

Методи за извличане на закономерности от данни могат да се прилагат директно върху релационните системи за управление на бази от данни. Тази интеграция е обещаваща насока за по-лесно и по-ефективно извличане на знания и има подкрепата на водещите производители на СУБД

Използване на методи за разкриване на знания в самия процес на разработка на складове от данни

Използването на методите за извличане на знания в процеса на изграждане на един корпоративен склад от данни също е важна допирна точка между дискутираните технологии.

Макар видимата част от една **Data Warehouse** система да е свързана основно с използване на обобщената информация, заложена в системата, реализацията на един такъв продукт започва с продължителен процес на извличането на данни от различните източници, тяхното трансформиране с цел синхронизация и отстраняване на противоречия и зареждането на обработените и обобщени данни в новата структура - склада от данни.

Тъй като ETL процесът е най-трудоемката и критична част от разработката на един склад, използването на **методи на извличане на закономерности от данни** върху входната за склада информация може да се използва като инструмент за изчистване и обобщаване на данните.

Тези методи могат да бъдат използвани за подпомагане на важните стъпки при разработката на един склад от данни и най-вече за автоматизиране на част от задачите, свързани с извличането, обобщаването и зареждането на данните

В [Kaloyanova, 2005] е представена една начална систематизация на методи, подпомагащи процеса на интегриране на данни от различните информационни източници в структурата на склада от данни. Изследването засяга основни въпроси, свързани с **интеграцията на данните** и предлага решения в няколко направления.

Интеграция на схеми

Най-важният проблем в процеса на интеграция на данни е съвместяването на схемите - как еквивалентни същности от многобройните източници на данни си съответстват и могат да бъдат съвместени. Статистически методи, асоциативни правила

(association rules), класификационни дървета (classification tree), индукция на правила (rule induction), и др. могат да се използват за решаването на този проблем.

Излишество на данните (зависими данни)

Излишеството на данните се дискутира на много места в обработката на данните. Смисълът на това понятие тук е свързан с възможността един атрибут да се изчисли/получи от други атрибути. Някои излишества могат да се определят на базата на корелационен анализ (*correlation analysis*).

Изчистване на данните

Изчистването на данните е нетривиална задача в областта на складовете от данни. Основните трудности тук са свързани с идентификацията на липсващи данни, наличието на некоректни или противоречиви данни от различни източници и коригирането на тези проблеми.

Грешни или замърсени (шумни) данни могат да бъдат третираны чрез методите дървета на решенията (decision trees), индукция на правила (rule induction) или методи за клъстериране (clustering methods), които отсяват появата на големи различия в данните.

Несъвместимостта на данните се проявява в противоречиви дължини на полета, противоречиви описания, противоречиви присвоявания на стойности, нарушаване на ограниченията за цялостност. Методите за клъстериране са подходящи и тук.

Липсващите данни изискват особено внимателно разглеждане и този проблем е дискутиран подробно в следващата секция.

Обикновено липсващите стойности са обозначени с празни полета или специални стойности на атрибутите. Липсващите стойности могат да бъдат определени с регресионен анализ (regression analysis), дървета на решенията и други методи.

Едно конкретно приложение на посочените по-горе подходи разглежда проблеми при интеграцията на данни в аналитичните системи, като обръща специално внимание на проблематиката, свързана с идентифицирането на лица (клиенти), за които към склада от данни постъпва информация от различни оперативни системи. Идентифицирането е на базата на българските изисквания и официални документи за идентификация [Naydenova&Kaloyanova&Ivanov, 2009].

Предложеното решение е на базата на клъстерен метод за унифициране на физически или юридически лица.

Процесът на идентифициране на клиента, който се използва, има три основни фази: подготовка и предварителна обработка (CIA_PREPARE), подбор на кандидатите за двойки (CIA_SEEK) и оценка кандидатите (CIA_EVALUATION).

CIA_PREPARE

Първата фаза естествено поддържа дейности, свързани с предварителната обработка на потребителските характеристики с цел елиминиране на правописни и

стилистични грешки, отстраняване на недостоверни данни и др. Тук влизат процедури като разделянето на неформатирани съставни данни, предварителна обработка на полета, процедури за трансформиране и валидиране (например, проверка за валидност на ЕГН според общоприетия алгоритъм, извличане на екстракти от текстовите полета и др.). В резултат се получават нови стойности на атрибутите, които ще бъдат използвани в следващите две фази.

CIA_SEEK

По време на втората фаза наличните данни се претърсват за много вероятно идентични обекти. Тъй като броят на възможните двойки е изключително голям, процесът е бавен и ресурсоемък. Необходимо е да се редуцират двойки до потенциално еднакви кандидати, които да се дооценят прецизно на следващото ниво.

CIA_EVALUATION

Целта на последната фаза е да оцени сходството за всяка двойка от получените кандидат списъци, изчислено като средно претеглена сума на степен на сходство между атрибути.

	Атрибут	Оценка на сходството
Еднакви	Персонален идентификатор	+200
	Булстат	+300
	Референтен данъчен номер	+300
	Идентификационен номер на карта	+300
	Дата на раждане (според въведени данни или извлечена от персонален идентификатор)	+50
	Име	+90
	Телефонен номер	+30
	Местоживееене (от пощенски адрес)	+20
	Пощенски код (от пощенски адрес)	+20
Различни	Пол (според въведени данни или извлечена от персонален идентификатор)	-150
	Тип на клиента	-200
	Извлечения на име	-50
	Местоживееене (от пощенски адрес)	-50
	Пощенски код (от пощенски адрес)	-20

Таблица 3.1 Степени на сходство между атрибутите

Точките, които се дават, се определят на база на евристична оценка на значението на атрибута, свързан с обектите. Методът позволява присъждане и на негативни оценки, когато някои основни атрибути са различни.

Създадена е система от точки (Таблица 3.1), които се дават за еднакви и нееднакви атрибути. Например, две лица се считат за идентични, когато сумата от техните резултати е повече от 250. Системата също така позволява лицата да бъдат изрично обявени за идентични или за неидентични.

Алгоритъмът е реализиран програмно и се използва в реално работещ склад с данни от 2007 година досега.

Някои елементи от изследването са използвани за разрешаване на друг проблем, описан в изследване на автора - откриване на несъвместимост при въвеждане на данните в транзакционни системи., Тъй като фокусът тук е насочен към многомерния модел, предложеното решение в посочената публикация остава извън обсега на този труд.

Като цяло, представените насоки за анализ на данните, които постъпват в аналитичните системи, засягат твърде широка област и не е възможно да се получат значими резултати във всички възможни направления. Но те са в основата на работата ни върху многомерния модел на данните в предишните няколко години и доведоха до резултатите по отношение на разредеността на данните, описани в следващата секция.

Разреденост на данните в многомерния модел

Изследванията на приложението на методите за разриване на закономерности в данните при ETL процеса и значимостта на наличните закономерности между данните, пораждат редица интересни въпроси.

От една страна, анализирайки данните и прилагайки горепосочените методи, можем да открием неподозирани и интересни от бизнес гледна точка зависимости. От друга страна, понякога има зависимости на семантично ниво, които са свързани с областта, бизнес модела и др. и са добре известни на аналитиците. Многомерният модел се използва активно в различни бизнес области. В тях могат да бъдат наложени определени бизнес правила, които носят допълнителна информация за ограниченията в областта. Знанието за тези зависимости би помогнало при създаването на проектното решение и би довело до по-ефективно проектиране и реализация.

Един от важните проблеми при всяка обработка на информация са липсващите данни. Този въпрос се дискутира и при транзакционните системи. Релационният модел използва нулева стойност (**NULL**), за да отбележи, че една стойност е неизвестна. В някои случаи става дума не за неизвестност, а за неприложимост на стойност за даден атрибут, докато в други – стойността е приложима, но липсва или не се знае дали съществува.[Elmasri et al., 2006].

При многомерния модел липсата на данни в клетки на куба има много по-комплексен характер и оказва съществено значение върху ефективността на работата

му. Когато големи области от куба не са запълнени с данни, това може да се използва за по-ефективната работа с него.

Резултат от изследванията в тази насока е анализът на ролята на разредеността на куба, която отчита не само наличието на празните клетки в куба (клетките, в които няма данни), но и причината, поради която данните липсват.

Изясняването на проблематиката, свързана с разредеността на данните в многомерните кубове, е свързана с уточняване на дефинициите на някои основни понятия [Naydenova&Kaloyanova et al., 2009]:

- *Измерение* D – всяко непразно крайно множество стойности.
- *Многомерно пространство* S върху измеренията D_1, D_2, \dots, D_n ($n \geq 1$) - декартово произведение $S = D_1 \times D_2 \times \dots \times D_n$.
- *Правоъгълна област* в многомерното пространство S - всяко подмножество на S от вида:

$$M \subseteq S, M = A_1 \times A_2 \times \dots \times A_n, \text{ където } A_1 \subseteq D_1, A_2 \subseteq D_2, \dots, A_n \subseteq D_n;$$

- \emptyset - „празна стойност“
- *Факт* F - множество от стойности, където $\emptyset \in F$
- *Куб* C : всяка функция от вида $C: S \rightarrow F$, където S е многомерно пространство, а F е факт
- *Клетка* в куба $C: S \rightarrow F$: всяка двойка от вида $c = (t, f)$, където t е точка от S ($t \in S$), а f е факт в куба ($C(t) = f$).

Клетката е *празна*, ако $f = \emptyset$ и е *непразна* (запълнена) – в противен случай

- *Множество от празни клетки* в куба C :

$$E(C) \subseteq S, \text{ където } E(C) = \{t \in S \mid C(t) = \emptyset\}$$

Разреденост на куб

Разредеността на един многомерен куб отразява неговата запълненост с данни [Niemi et al., 2000]. Колкото повече клетки от куба не съдържат данни, толкова повече той е по-рехав или по-разреден. Разредеността на куба измерваме с неговия коефициент на плътност.

$$\text{Коефициент на плътност на куб } C: \omega_C = \frac{|S| - |E(C)|}{|S|}$$

Някои изследванията [Potgieter'03] показват широката разпространеност на кубове с високи нива на разреденост, особено при многомерните модели,

предназначени за анализ на различни видове индустрии. Обособени са и някои основни зависимости:

- с увеличаване на броя на размерностите се увеличава и разредеността на куба.
- процентът на гъстотата на данните в повечето случаи е много под 1%.

Големият обем на празните клетки в многомерните кубове поражда съществени проблеми - увеличение на обема на базите от данни и удължено време за зареждане на данните и изчисления върху тях.

Изследванията по отношение на типа на разреденост на данните в многомерните кубове показват, че се отчитат няколко основни типа разреденост.

- *Контролирана и произволна.* Класификацията се базира на типа на разпределение на празните клетки. При *Контролирана* рехавост в цял интервал от стойности по дадено измерение клетките са празни, докато при *произволната* рехавост празните клетки са разпръснати в целия куб. [OLAP DML, 2002]).
- *Случайна, лентова, клъстерна и правоъгълна.* Класификацията се базира на физическото разположение на празните клетки, в зависимост от формата в пространството, която определят [Kang et al., 2002].

Посочените класификации на разредеността не дават информация за причините за липсата на данни в тези области от куба. Но поради спецификите на многомерния модел на данните - наличието на измерения и йерархии, проблемът с празните клетки във входните данни води до други прояви като **експлозия на данни**, която се наблюдава, когато броят на пълните клетки в базовия кубоид (клетките от най-ниското йерархично ниво) е значително по-малък от броя на останалите пълни клетки в куба. [Potgieter, 2003].

Известните до момента методи за намаляване на обема на рехавите кубове и намаляване на експлозията на данни като композитно индексирание, разделяне на куба на части, преагрегиране и др., обаче, не отчитат характера на разредеността и са концентрирани главно върху физическата оптимизация и проблемите около компресирането на съхраняваната информация [Pottle, 2008].

Прилагането на подхода на анализ на данните за създаване на по-добър обобщен модел може да даде информация за наличие на предефинирана разреденост, която може да се дължи на специфични ограничения на бизнеса. Въвеждането на нова класификация на видовете разреденост – *случайна* и *регулярна* [Naydenova & Kaloyanova, 2006] акцентира върху ролята на семантиката на стойностите на измеренията за появата на регулярна разреденост в n-мерен куб.

Следващите дефиниции формално определят *регулярната* и *случайната* разреденост.

Регулярна разреденост - всяко подмножество от точки в един n -мерен куб, които не могат да имат непразни стойности поради спецификата на моделираната бизнес област.

Случайна разреденост - всяко подмножество от точки в един n -мерен куб, които могат да имат непразни стойности поради спецификата на моделираната бизнес област.

Липсата на стойности при регулярната рехавост тук означава недопустими или неприложими стойности – напр. конкретният продукт не се продава след определена дата; клиенти, които са организации, а не частни лица, нямат пол и т.н.

Такива стойности нямат бизнес смисъл и не биха били от особен интерес за бизнес анализа. Те биха могли да се определят по време на анализа на данните и знанието за тях може да се използва при проектирането на съответния модел.

Знанието за наличие на регулярната разреденост е ценно за анализа на данните и използването му определено оказва влияние върху реализацията на многомерния модел.

Ролята на данните за появата на регулярна разреденост

Както вече бе посочено, естеството на изходните данни е важна предпоставка в процеса на проектиране.

Някои специфични характеристики на данните създават възможни ситуации, които водят до появата на регулярна разреденост.

Ирелевантни измерения

Съществуват ситуации, в които е налице частична функционална зависимост между измеренията.

Такава зависимост се наблюдава в случаите, когато за определено множество от стойности на едно измерение (или комбинация от стойности на няколко измерения) съществува точно една стойност в друго измерение, която позволява клетки с попълнени данни. За останалите стойности в другото измерение съответните клетки винаги са празни.

Например при куб, който описва седем измерения: *време, услуги, канали за продажба, региони, видове клиенти, пол и клон на бизнеса* при избор на тип на клиента "частно лице" няма стойности в измерение „клон на бизнеса“. И обратно, когато избираме "организация", „полът“ е без значение. Наблюдаваната тук разреденост е регулярна, защото предварително е ясно, че за откъслечни региони имаме празни клетки, които се дължат на несъвместимост на цяло измерение или на няколко измерения спрямо елементи на друго измерение/измерения [Naydenova&Kaloyanova, 2006].

Тази връзка определяме като ***връзка от тип ирелевантност на измерение*** .

Сегментация на измерение

В някои случаи, за дадено множество от стойности на едно измерение (ведещо) съществува определено множество от стойности в друго измерение (зависимо), за които е допустимо наличието на клетки с данни. За всички останали стойности в зависимото измерение клетките на куба са задължително празни. Например, определено множество от услуги може да се предлага в точно определено множество от офиси и да не се предлага в останалите.

Връзка от този тип определяме като ***връзка от тип сегментация на измерение***. Тя е обобщение на връзката от тип ирелевантност на измерение.

Връзката от тип сегментация на измерения в общия случай не отразява функционална зависимост между измеренията.

Изменения в измеренията в резултат на развитието на бизнеса

Промените в бизнеса по време на реалното функциониране на една аналитична система, също могат да породят регулярна разреденост. Например, такава ситуация се наблюдава при прекратяване на част от предлаганите услуги в дадена предметна област или алтернативно - при въвеждането на нови услуги. В първия случай в многомерния куб се появят празни клетки, съответстващи на новите периоди от време поради прекратяване на услугите, а във втория остават празни клетки за изминалите периоди, когато съответните услуги не са били предлагани.

Промените в тези случаи предизвикват появата на връзки от тип ***сегментация на измерения***.

Подход за намаляване регулярната рехавост за подобряване на проектните решения при многомерния модел

Разграничаването на разредеността в n -мерен куб, дължаща се на предварително известни бизнес ограничения от разредеността със случаен характер, може да доведе до оптимизация както на аналитичния процес, така и на начина на моделиране и физическото съхранение на данните.

Едно естествено решение на проблема е празните области на куба да се елиминират, като на логическо ниво се работи с един хипер-куб, а на по-ниско ниво се оперира с няколко куба с по-малка размерност, но с по-висок коефициент на плътност. Това би довело до по-добри резултати най-вече при моделиране на кубове с наличие на връзки от тип ирелевантност на измерение.

За осъществяването на този подход е развита концепцията за ***Карта на регулярна разреденост*** [Naydenova&Kaloyanova et al., 2009] като инструмент, който позволява на проектантите да опишат участъците с празни клетки. Тя съдържа клетките, които са празни поради регулярна разреденост (бизнес правила, конкретни изисквания, естествени зависимости и др.).

Карта на регулярна разреденост

Карта на регулярна разреденост R_C в куб $C: S \rightarrow F$ ще наричаме множеството от всички регулярно празни клетки $R_C \subseteq E(C) \subseteq S$.

Разликата на двете множества

$$E(C) \setminus R_C$$

определя множеството на всички клетки, които са празни поради случайна разреденост.

Представяне на картата на регулярната разреденост

Регулярната разреденост е предефинирана разреденост. Информацията за построяването на картата на регулярната разреденост се извлича от бизнес правилата, така че тя може да бъде дефинирана преди създаването на куба и да се използва при неговото проектиране. Така картата може да бъде използвана във фазите на анализа и проектирането на многомерния модел на данните и построяването на самия куб.

За целите на програмната реализация на картата на регулярна разреденост са анализирани няколко възможни модела за представянето на областите на регулярна разреденост и е избран подход на представяне чрез множества от непресичащи се правоъгълни области на регулярна разреденост, всяка от които се дефинира чрез формализирани правила. Всяко правило описва съществуващо бизнес ограничение (бизнес правило), което лесно може да бъде зададено от бизнес потребителите на аналитичната система [Naydenova&Covacheva&Kaloyanova, 2009].

При така избраното представяне всяко бизнес правило определя точно една област на регулярна разреденост в куба. Бизнес правилата се задават като набор от условия върху измеренията, като всяко условие задава набор от елементи на измерения. Въз основа на тях се обособяват определени области в n -мерното пространство. Обединението на всички правила оформя *картата на регулярната разреденост*.

По този начин картата R_C върху куба $C: S \rightarrow F$, $S = D_1 \times D_2 \times \dots \times D_n$ се представя чрез списък от правила P_1, P_2, \dots, P_k , заграждащи правоъгълни области на регулярна разреденост R_1, R_2, \dots, R_k , чието обединение е картата на регулярна разреденост R_C ,

$$R_C = \bigcup_{j=1}^k R_j, R_j = R_j^1 \times R_j^2 \times \dots \times R_j^n, \text{ където } R_j^i \subseteq D_i \text{ за } \forall i = 1..n$$

Всяко правило $P_j, j = 1..k$ може да бъде представено във вида:

rule = condition, {" AND ", condition}

където **condition** = {dimension, operator, values}

operator = {<, >, =, ≠, ≤, ≥, in, between}

При работа с правилата се спазват следните ограничения:

Всяко условие (condition) задава подмножеството R_j^i от елементи на цитираното в него измерение D_i .

Всяко измерение може да се среща най-много веднъж в правилото.

Ако дадено измерение D_i липсва в правилото, се приема, че съответната правоъгълна област R_j няма ограничения по отношение на D_i , т.е. включва всички нейни стойности.

При задаване на условията може да се зададе една стойност, списък от стойности, или двойка стойности (интервал).

Създаване на карта на регулярна разреденост

Изграждането на една карта на регулярната разреденост е постъпков процес, в който се изпълняват няколко основни операции:

- *сегментация* на многомерно пространство;
- *разширение* на сегментация на многомерно пространство от произволна правоъгълна област;
- *разцепване* на правоъгълна област върху сегментация на многомерно пространство.

Формално операциите могат да бъдат дефинирани по следния начин.

Сегментация на многомерно пространство

Нека S е многомерно пространство върху измеренията

$$D_1, D_2, \dots, D_n : S = D_1 \times D_2 \times \dots \times D_n.$$

G ще наричаме *сегментация на многомерното пространство S* , когато:

$$G(D_i) = \{D_i^1, D_i^2, \dots, D_i^{p_i}\} : D_i = \bigcup_{j=1}^{p_i} D_i^j \text{ и } D_i^j \cap D_i^k = \emptyset \text{ за } k \neq j.$$

Множествата $D_i^1, D_i^2, \dots, D_i^{p_i}$ наричаме *сегменти на измерението D_i съгласно сегментацията G* .

Разширение на сегментация на многомерно пространство от правоъгълна област)

Нека M е правоъгълна област в пространство S , а G е сегментация на S : $M = A_1 \times A_2 \times \dots \times A_n$, $A_1 \subseteq D_1$, $A_2 \subseteq D_2$, ..., $A_n \subseteq D_n$

Сегментацията G' ще наричаме разширение на G от M , където:

$$\forall i = 1..n \text{ е в сила } G'(D_i) = \{D_i^j \mid D_i^j \in G(D_i), (D_i^j \cap A_i = \emptyset \vee D_i^j \subseteq A_i)\} \cup \{D_i^j \cap A_i, D_i^j \setminus A_i \mid D_i^j \in G(D_i), D_i^j \cap A_i \neq \emptyset, D_i^j \setminus A_i \neq \emptyset\}$$

По този начин всеки сегмент $D_i^j \in G(D_i)$: $D_i^j \cap A_i \neq \emptyset$ и $D_i^j \setminus A_i \neq \emptyset$ се разделя на два нови сегмента $D' = D_i^j \cap A_i$ и $D'' = D_i^j \setminus A_i$.

Новите сегменти D' и D'' заменят стария сегмент D_i^j в множеството от сегменти $G(D_i)$ на измерението D_i .

Ако резултатът е $G' \equiv G$, казваме че G е вече разширено от M .

Разцепване на правоъгълна област върху сегментация на многомерно пространство)

Нека M е правоъгълна област в пространство S , G е сегментация на S и G е вече разширено от M .

Разцепване на M върху G ще наричаме операцията:

$$split(M, G) = \bigtimes_{i=1}^n \{D_i^j \in G(D_i) \mid D_i^j \subseteq A_i\}$$

За самото изграждане на картата се изпълняват стъпките на описания по-долу алгоритъм.

Алгоритъм за създаване на карта на регулярна разреденост

Нека е дадена карта на регулярна разреденост R_c , дефинирана върху куба

$C : S \rightarrow F$, където

$$S = D_1 \times D_2 \times \dots \times D_n,$$

Разглеждаме R_c представена като обединение на непресичащи се правоъгълни области:

$$L = \{P_1, \dots, P_t\}, \text{ където } R_c = \bigcup_{j=1}^t P_j \text{ и } P_i \cap P_j = \emptyset, i \neq j$$

В началото картата на регулярна разреденост е празна

$$L = \emptyset, R = \emptyset,$$

а сегментацията на многомерното пространство се изразява чрез

$$S: G(D_i) = \{D_i\}$$

При добавяне на ново правило към картата, вече добавените области в картата се разцепват на правоъгълни области, които не се пресичат помежду си.

Нека новото правило описва правоъгълната област

$$B = A_1 \times A_2 \times \dots \times A_n,$$

Тогава се извършва разширение на сегментацията G на многомерното пространство S чрез правоъгълната област B до G' .

За целта за всеки $P_i \in L$ се извършва разцепване на областта P_i върху сегментацията G'

$$\forall P_i \in L : split(P_i, G')$$

и разцепване на правоъгълната област B върху G' .

По този начин B се разделя на множество от непресичащи се правоъгълни области спрямо сегментацията на измеренията в R_c .

$$split(B, G') = \{B_1, \dots, B_m\} = \bigtimes_{i=1}^n \{D_i^j \in G'(D_i) \mid D_i^j \subseteq A_i\},$$

$$B = \bigcup_{i=1}^m B_i, \text{ където } B_i \cap B_j = \emptyset \text{ за } i \neq j$$

$$\forall i, j : P_i = B_j \text{ или } P_i \cap B_j = \emptyset.$$

За да се добави областта B към R_c , се прилага операцията обединение между областите от множеството L' и непресичащите се правоъгълни области, продуцирани от операцията $split(B, G')$.

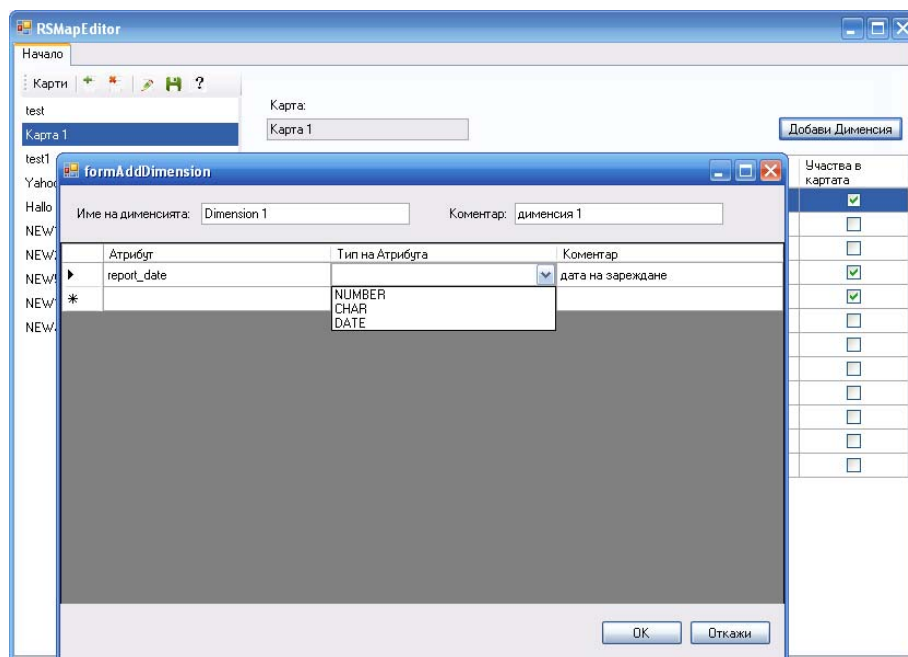
Присвояването $L = L' \cup split(B, G')$ е функционално еквивалентно на изрази $R_c = R_c \cup B$.

Тези стъпки се повтарят за всяко ново правило до изчерпване на правилата.

Графичен редактор за работа с карта на регулярна разреденост

За работа с карти на регулярна разреденост е създадено приложение с удобен графичен интерфейс. Чрез него може да се работи с различни карти на регулярна разреденост и свързаните с тях измерения и атрибути.

Софтуерното приложение поддържа редица функции, като създаване и редактиране на карта на регулярна разреденост, добавяне и редактиране на правила и/или условия към правила, работа с области на куба.



Фигура 3.4 Редактор на карта на регулярна разреденост

Така се осъществяват реалните дейности по работата с картата на регулярната разреденост.

Сред най-важните от тях е реализацията на *метод за откриване на области с регулярна разреденост в даден куб*.

Методът използва картата за операциите по обединение, сечение или разлика между входящата правоъгълна област и карта на регулярна разреденост, като следва посочените в горния алгоритъм стъпки, но не добавя входящата област към картата. По този начин, в резултат на разцепването, се получават области, които или се припокриват изцяло от картата, или са изцяло извън нея.

Редакторът е разработен с помощта на двама дипломанти и докторантката Ина Найденова. В нейната докторска дисертация на тема „Приложение на Data Warehouse системи за разкриване на знания: Разреденост на данните в многомерния аналитичен модел” (защитена успешно на 21.01.2013г.) са представени допълнителни теоретични обосновки на алгоритмите и методите за работа с картата на регулярната рехавост, както и редица резултати от експерименти, подкрепящи ефективността предложената концепция.

По-долу са посочени няколко основни направления на използване на картата, систематизирани в [Naydenova & Kaloyanova et al., 2009].

Приложения на картата на регулярна разреденост

Основната насока на нашите изследвания за регулярната рехавост е свързана с анализ на данните, които постъпват в аналитичната система, но реалното приложение на регулярната разреденост има далеч по-голям обхват.

Картата на регулярната разреденост може да се използва в различни направления – от анализ на данните за многомерния модел с цел подобряване на качеството им до подобрения в потребителския интерфейс на аналитичните системи за крайните потребители.

Подобряване на качеството на данните

Чистотата на данните бе отбелязана като едно от най-важните предизвикателства в процеса на разработка на складовете от данни. Проблемите с качеството на данните обикновено излизат на преден план едва след завършване на ETL процеса. Много често те се дължат на грешките, които се наслагват във времето на експлоатация на системите, които служат като източници на информацията за склада от данни.

Дефинирайки правила, близки до тези на бизнес аналитиците, но в многомерното пространство, картата на регулярна разреденост може да се използва за автоматизиране на проверката на чистотата на данните и откриване на несъответствия между тях.

Допълнителни пояснения на причината за празните клетки в куба поради регулярна и случайна разреденост могат да бъдат текстово въведени и подходящо визуализирани, давайки на потребителите ценна информация относно данните.

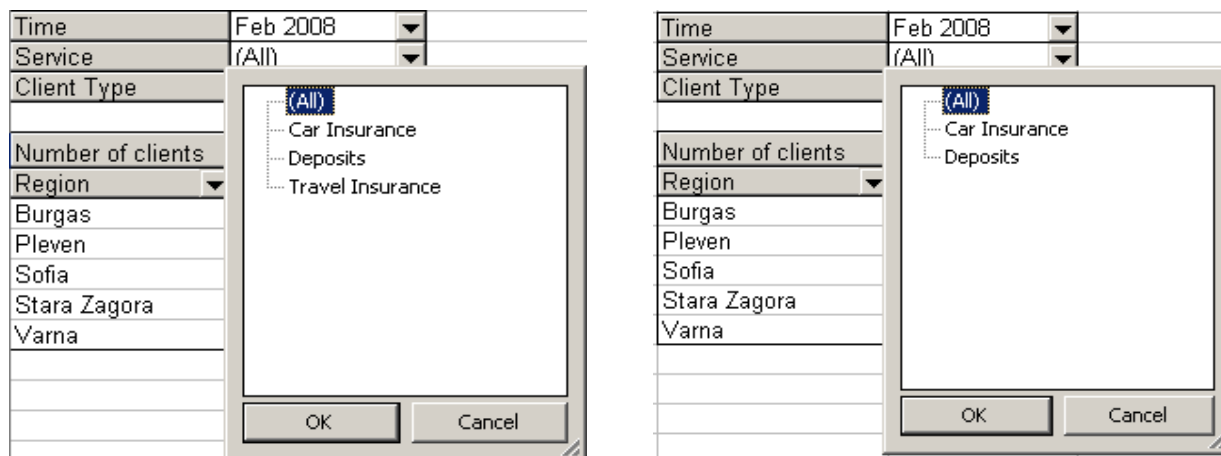
Определянето за кои комбинации от стойности на измеренията не бива да има данни в аналитичната система е ограничение от тип цялостност. Подобно ограничение при транзакционните системи е “not null”, което не позволява на един атрибут да има недефинирани стойности. При многомерния модел чрез ограничението се налага забрана за наличието на определени данни при определени бизнес ограничения.

Ако една клетка от куба принадлежи на област от регулярна разреденост, но в нея има стойност, това означава грешка в данните.

Автоматично избиране на значимите елементи в измеренията

При избиране на конкретни стойности по дадени измерения за дадена справка, на базата на информацията в картата може да се определи дали в останалите измерения има стойности, описващи регулярна разреденост. Тези стойности могат да бъдат изключени при визуализацията.

Фиг. 3.5 демонстрира използването на редактора на картата за автоматичното филтриране на измеренията. Вляво са представени измеренията без филтриране, вдясно – след филтрацията [Naydenova & Kaloyanova et al., 2009].



Фигура 3.5. Автоматично избиране на релевантни измерения

Намаляване на обема на заеманата памет

Отстраняването на повторения на данни в MOLAP среда в адитивни йерархични участъци на неадитивни показатели води до значително намаляване на обема на паметта, заемана от куба. В [Naydenova & Kaloyanova, 2007] е описана схема за обработка на данни в ситуации, при които съществуват йерархични елементи, които имат само един наследник с непразна стойност.

С помощта на картата на регулярна разреденост се откриват йерархичните елементи от този тип. Останалите наследници задължително получават празни стойности поради наличието на регулярна разреденост. Зареждането на такива възли се пропуска, а техните стойности се пресмятат при поискване със стандартните функции (MIN, MAX, AVG), които стандартно не включват в пресмятанията си **NULL** стойности.

Без да се спираме подробно на описания в публикацията алгоритъм, ще приведем само резултати от приложението на картата, които илюстрират ефективността на метода.

Експериментът е проведен върху 6 куба, всеки от които е дефиниран върху 14-мерно пространство и съхранява данни за брой клиенти от бизнес области, свързани със общо застраховане, животозастраховане и пенсионно осигуряване. Дефинирани са по 2 куба за всяка бизнес област - за брой физически лица клиенти на компанията и за брой юридически клиенти.

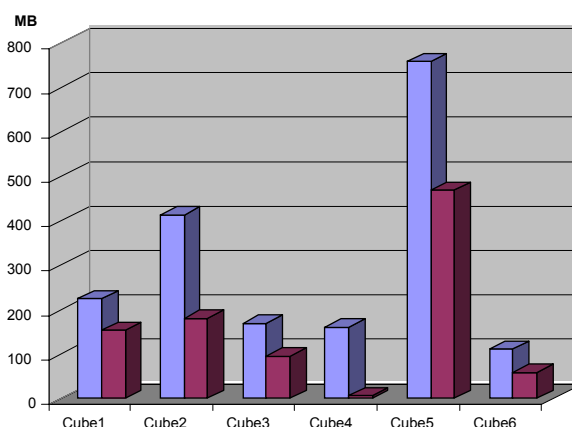
Кубовете съдържат голям брой общи измерения (време, брой години клиент, брой продукти, статус на клиент), специфични бизнес измерения (продукти, организационни единици, канали на продажба) и трети тип измерения, отнасящи се само за частни клиенти (например пол) или само за корпоративни (например отрасли).

Анализираните неадитивни измерения са продукти и организационни единици.

Приложението на метода е върху извадка от данните за даден месец.

Месец - Ноември 2004	Cube1	Cube2	Cube3	Cube4	Cube5	Cube6	Общо
Обем на паметта преди компресията (MB)	224.29	412.76	169.36	160.81	757.02	111.01	1835.25
Обем на паметта след компресията (MB)	153.10	180.28	95.48	5.91	467.73	57.77	960.27
Оптимизация (%)	31.74%	56.32%	43.62%	96.32%	38.21%	47.96%	47.68%

Таблица 3.3. Компресия по 6 куба



Фигура 3.6. Резултати от компресия

Както се вижда от таблица 3.2 и фиг.3.6, резултатите показват намаление на размера на кубовете от 32% до 96% в зависимост от разпределението на данните.

Изводи

В тази глава бяха представени някои резултати от изследвания в областта на многомерния модел на данните.

Анализирани са редица въпроси, свързани с ефективната работа на многомерния модел. Систематизирани са направления на изследване, засягащи ключови елементи на разработката на складове от данни.

Изследвани са проблеми при обединението на данни от различни източници в складовете от данни и са предложени специфични методи, подходящи за използване в процеса на извличане, трансформиране и зареждане на данните в склада.

Правилността на предложения подход е подкрепена чрез разработката на специализиран метод за обработка на информацията за лица, постъпваща от различни източници към склада от данни.

Предложено е решение на един от сериозните проблеми в областта на онлайн аналитичната обработка - разредеността на данните и обработката на кубове с висока размерност.

С въвеждането на концепцията за регулярна разреденост се дава възможност с подходящи структури да се опишат предварително зададени ограничения за наличие на стойности или липса на стойности в клетки от многомерните кубове и тази информацията да се използва за по-ефективно проектирането на кубовете.

Предложеният подход за дефиниране на карта на регулярната разреденост е теоретично обоснован и подкрепен със специално създадени програмни средства, които улесняват практическо му приложение и визуализацията и интерпретацията на данните.

ЗАКЛЮЧЕНИЕ

Настоящият хабилитационен труд „Анализ и проектиране на информационни системи” представя част от научните и научно-приложните резултати на доц. д-р Калинка Калоянова, получени през последните десет години. Те могат да бъдат обобщени по следния начин:

- Дефиниране на подход за систематизирано събиране, анализиране и обобщаване на потребителските изисквания към разработката на информационни системи на база на информацията, получена от потребителите, с цел ефективното им използване при проектирането на системите.
- Осигуряване на ефективната реализация на подхода чрез прилагане на специфичен набор от методи, модели и техники.
- Анализиране и систематизиране на методи за изчистване и обобщаване на входящите данни при изграждане на складове от данни.
- Изследване на влиянието на разредеността на данните при проектирането на многомерните модели в аналитичните системи, отделяне на регулярната (предефинирана) разреденост на данните и създаване на апарат за нейното формално описание.
- Водеща роля при изграждане на систематизиран подход за използване на регулярната разреденост за оптимизация на многомерния модел на базата на карта на регулярната разреденост. Участие при проектирането на софтуерна среда за реализация на подхода и провеждане на експерименти.
- Приложение на описаните подходи при обучението на студенти в областта на информационните системи и свързани области.

Част от извършените изследвания са направени в рамките на няколко проекта към Фонд „Научни изследвания” на СУ”Св. Кл. Охридски”.

При разработката на някои софтуерни продукти, подпомагащи изследователската дейност, са участвали докторанти и магистри от ФМИ.

Плановите за бъдеща изследователска дейност са свързани с усъвършенстване на разгледаните подходи. Всяко от двете направления може да се развива самостоятелно. Възможни са подобрения чрез добавяне на допълнителни авоматизирани средства, подпомагащи моделирането, детайлизиране на отделни стъпки или елементи, както и подобрения в самия процес на реализация.

Друга възможна насока на развитие е във всеки един от разгледаните подходи да се прилагат концепции, заимствани от другия подход. Например, обектно-ориентирани техники като наследяване могат да се приложат при управлението на множество многомерни модели с общи елементи (множество кубове със сходни размерности, които имат леки разлики).

Към подхода за анализиране на потребителските изисквания към аналитичните системи биха могли да се добавят и средства за моделиране на данните, които постъпват в склада от данни.

Отворени за бъдещи изследвания са въпроси, свързани с избор на подходящи методи за извличане на закономерности от данни според спецификата на дадена предметна област и наличните данни в системата.

Натрупването на информация за приложенията на предложените подходи и прилагането на подходящи методи за анализ върху нея ще позволи усъвършенстването им.

ЛИТЕРАТУРА:

- Bernd B., Duttoit A.(2010) Object Oriented Software Engineering with UML, 3 ed, Prentice Hall, 2010.
- Beyer, H., K. Holtzblatt, L. Baker (2004) An Agile User-Centered Method: Rapid Contextual Design, ExtremeProgramming and Agile Methods - XP/Agile Universe 2004, Calgary, 2004, pp. 527-554.
- Beyer, H., K. Holtzblatt (1997) Contextual Design: Defining Customer-Centered Systems, Morgan Kaufmann, 1997.
- Beyer, H., K. Holtzblatt (2009) Representing Work for the Purpose of Design Contextual Design, <http://incontextdesign.com/publications/representing-work-for-the-purpose-of-design/>, 2009
- Bittner K., I. Spence (2002) Use Case Modeling, Addison Wesley, 2002.
- Chaudhuri S., Dayal U. (1997) An overview of data warehousing and OLAP technology. ACM SIGMOD Record, pp.65-74.
- Cockburn A. (2000) Writing Effective Use Cases, Addison-Wesley, 2000.
- Doucet G. (2008) Coherency management: Using Enterprise Architecture to Achieve Alignment, Agility, and Assurance, Journal of Enterprise Architecture (May 2008). Volume 4 Number 2.
- Draheim D. (2010) Business Process Technology: A Unified View on Business Processes, Workflows and Enterprise Applications, Springer, 2010.
- Elmasri R., Sh. B. Navathe, (2006) Fundamentals of Database Systems, 5 Ed., Addison Wesley, 2006.
- Francisco P., M. Kearney (2011) Oracle Exadata and IBM Netezza Data Warehouse Appliance compared, http://www.ibmbigdatahub.com/pdf/Oracle_Exadata_IBMNetezza_Compared_WP_EN.pdf
- Garcia-Molina, H., Ullman, J. D., Widom, J. (2009) Database Systems. The Complete Book, Pearson Education Inc., 2009.
- Han J., Kamber M.(2006) Data Mining: Concepts and Techniques, Second Edition, Elsevier Inc., ISBN 13: 978-1-55860-901-3, 2006.
- Holtzblatt K.(2001) Contextual Design: Experience in Real Life, Mensch & Computer, 2001: 1. Fachübergreifende Konferenz. Stuttgart: B.G. Teubner; 2001, pp 19-22.
- Inmon W. H. (1992) Building the Data Warehouse, QED Technical Publishing Group, Wellesley, Massachusetts, 1992.
- Inmon W. H. (2005) Building the Data Warehouse, 4 Ed., Wiley, 2005.
- Inmon W. H., D. Strauss, G. Neushloss (2008) DW 2.0: The Architecture for the Next Generation of Data Warehousing, Morgan Kaufman, 2008.

Kaloyanova K. (2005) Improving Data Integration for Data Warehouse: A Data Mining Approach, Proceedings of the International Workshop "COMPUTER SCIENCE AND EDUCATION", Borovetz-Sofia, Bulgaria, June, 2005, ISBN 954-535-401-1, pp 39-44.

Kaloyanova K., A. Antonov (2011) Object-oriented approach for developing interactive 3D graphics system in education. Proceedings of 7th Annual International Conference on Computer Science and the Education in Computer Science, Sofia, Bulgaria, July 06-10., 2011, pp. 89-100.

Kaloyanova K. (2012-A) Design from data: how to use requirements for better information system analysis and design, Proc. of the Int. Conference Informatics in Scientific Knowledge, Varna, June, 26-29, 2012, pp. 189-197.

Kaloyanova K. (2012-B) Using the Project Approach in IS Education, Computer Science and Education in Computer Science (CSECS), Boston-New York, July, 2012.

Kaloyanova K. (2013) Successful Practices for Learning Information Systems Development, INTED 2013, Valencia, Spain, March, 4-6, 2013.

Kanabar V., R. D.H. Warburton (2008) MBA Fundamentals Project Management, Kaplan Publishing, 2008.

Kang J.,Yong H., Masunaga Y. (2002) Classification of Sparsity Patterns and Performance Evaluation in OLAP System, IEIC Technical Report, ISSN:0913-5685,vol.102,No.209(De2002 61-84), pp.61- 66, Japan, 2002.

Kaser O., Lemire D. (2003) Attribute Value Reordering for Efficient Hybrid OLAP, Proceedings of the 6th ACM international workshop on Data warehousing and OLAP, pp.1-8,ISBN:1-58113-727-3, 2003.

Kimbal R. (2002) The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd ed., John Wiley & Sons, 2002.

Kozielski, St., Wr. Robert (2009) New Trends in Data Warehousing and Data Analysis, Springer Science+Business Media, LLC, 2009.

Larman G.(2004) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and The Unified Process, 3 Ed., Prentice Hall, 2004.

Maciaszek, L. (2005) Requirements Analysis and Design, 2-nd ed., Addison Wesley, 2005

Martin J. (1983) Managing the Data Base Environment, Prentice Hall, 1983.

Napoli, J.P, **K. Kaloyanova** (2011) An Integrated Approach for RUP, EA, SOA and BPM Implementation, Proceedings of the 12th International Conference on Computer Systems and Technologies, Vienna, Austria, 16-17 June 2011, ACM Digital Library, pp. 63-68.

Naydenova I., **K. Kaloyanova** (2005) Basic Approaches of Integration Between Data Warehouse and Data Mining, Proceedings of the First International Conference On Information Systems & Datagrids, Sofia, February, 2005, ISBN 954-649-761-4, pp 75 -87.

Naydenova I., **K. Kaloyanova** (2006) Some Extensions to the Multidimensional Data Model, ISGT'06: IEEE 2006 John Vincent Atanasoff International Symposium on Modern Computing, October 2006, Sofia, Bulgaria, pp 63-68.

Naydenova I., **K. Kaloyanova** (2007) An Approach Of Non-Additive Measures Compression In Molap Environment, IADIS Multi Conference on Computer Science and Information Systems 2007, Lisbon, Portugal, pp 394-399.

Naydenova I., **K. Kaloyanova**, S.Ivanov (2009) Multi-Source Customer Identification, Third International Conference on Information Systems & Grid Technologies, 28 - 29 May 2009, Sofia, Bulgaria, pp 77-85.

Naydenova I., Z. Covacheva, **K. Kaloyanova** (2009) A model of Regular Sparsity Map Representation, Scientific Journal of Ovidius University of Constantza, Romania, 2009, Vol. 17(3), pp. 197-208.

Naydenova I., **K. Kaloyanova**, G. Georgiev, P. Melkonyan (2009) Data Errors and Relevant Dimension Values Detection with a Regular Sparsity Map, Proceedings of the 4-th Mediterranean Conference on Information Systems – MCIS 2009, (September 25-27, Athens, 2009), ISBN: 978-960-98566-7-6, pp457 – 463.

Niemi T., Nummenmaa J., Thanisch P. (2000) Functional Dependencies in Controlling Sparsity of OLAP Cubes, Springer Berlin / Heidelberg, Volume 1874/2000, pp.199-209, 2000.

[OLAP DML, 2002] Oracle Corporation, Oracle9i OLAP Developer's Guide to the OLAP DML Release 2 (9.2), Part Number A95298-01, March 2002, pp. 3-18.

Potgieter J., OLAP Data Scalability, DM Review Magazine, October 2003.
<http://www.dmreview.com/dmdirect/20031031/7636-1.html>

Papazoglou M., W. van den Heuvel (2007) Service oriented architectures: approaches, technologies and research issues. The VLDB Journal The International Journal on Very Large Data Bases, 16(3):389-415, 2007.

Pottle B. (2008) Oracle Database 11g: OLAP Essentials, Oracle University, D70039GC10, Edition 1.0, December, 2008.

Pressman R.(2009) Software Engineering: A Practitioner's Approach, 7 ed, McGraw-Hill, 2009.

Reddy, M., Pratt, W., Dourish, P., & Shabot, M. (2003) Sociotechnical requirements analysis for clinical systems, Methods of Information in Medicine, N 4, 2003, pp. 437-444.

Rosenberg D, M. Stephens (2007) Use Case Driven Object Modeling with UML, Appress, 2007.

Schach St. (2010) Object-Oriented And Classical Software Engineering, 8 Edition, Mc Graw Hill, 2010.

Shelly G., H J. Rosenblat (2010) Systems Analysis and Design, 8 Edition, Course Technology, Boston, 2010.

Simsion G., Gr. Witt (2004) Data Modeling Essentials, 3 Edition, Morgan Kaufmann, 2004.

Scott W, N. John, M. Vizdos (2005) The Enterprise Unified Process: extending the Rational Unified Process. Prentice Hall PTR, 2005.

Sommerville I. (2010) Software engineering, 9 Ed., Addison Wesley; 2010.

Strauch C.(2010) NoSQL Databases, www.christof-strauch.de/nosql dbs.pdf

Temnenco V., (2010) Using a combined SOA and TOGAF environment for increased productivity: Part 2. Service-oriented architecture from top to bottom, <http://www.ibm.com/developerworks/rational/library/10/soa-togaf-part2-service-oriented-architecture/>

Thilini A., Watson H (2006) Which Data Warehouse Architecture Is Most Successful?, Business Intelligence Journal, Vol. 11, No 1, pp.4-6, 2006.

Topi H, J. Valasich, R.Wright, K.Kaiser, J.F. Nunamaker Jr, J. Sipior, G.J. de Vreede (2010) IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, <http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf>

Найденова И. (2013) Приложение на Data Warehouse системи за разкриване на знания::Разреденост на данните в многомерния аналитичния модел, Дисертация за присъждане на образователната и научна степен Доктор, София, 2013.

Фаулър М. (2007) UML основи, СофтПрес, 2007.

Приложение

Забелязани цитирания на публикациите, въз основа на които е разработен хабилитационният труд

Хабилитационният труд се базира на изследвания и резултати, описани в 12 публикации с номера [1, 2, 4, 5, 9, 11, 12, 16, 17, 19, 20, 24] от „Списък на публикациите за участие в конкурса”. Забелязаните цитирания на тези статии са описани в следващия списък.

Статия [1]

Naydenova I., K. Kaloyanova. Basic Approaches of Integration Between Data Warehouse and Data Mining, Proceedings of the First International Conference On Information Systems & Datagrids, Sofia, February, 2005, pp. 75 -87, ISBN 954-649-761-4

е цитирана в

1. Maguy Medlej. Fouille Dans Les Entrepôts De Données, Mémoire rédigé dans le cadre du Master 2 Recherche en Informatique Spécialité Extraction de Connaissances à partir des Données, Ecole Doctorale De Sciences Cognitives Université Lumière Lyon2, 2007
2. Mamchenco J., I. Sileikiene. Intelligent Data Analysis of E-Learning System Based on Data Warehouse, OLAP and Data Mining Technologies, Proceedings of the 5th WSEAS International Conference on Education and Educational Technology, Tenerife, Canary Islands, Spain, December 16-18, 2006, pp. 171-175
3. Kovacheva Z. Tendencies in Data Warehouse Scope Development, Proceedings of the 6th International Conference on Information Systems and Grid Technologies – ISGT'2012. Sofia, Bulgaria, June, 01-03, 2012. pp. 11-21
4. Савоска Сн. Управуване со информации и одлучување: концепти, методологија и компјутерска подршка, ISBN 978-608-65059-5-0, Гинис, Македонија, 2012

Статия [2]

Kaloyanova K. Improving Data Integration for Data Warehouse: A Data Mining Approach, Proceedings of the International Workshop “Computer Science And Education”, Borovetz-Sofia, Bulgaria, 6-8 June, 2005, pp. 39-44, ISBN 954-535-401-1

е цитирана в

5. Bo Panjiu Liu Peishan. Rules of Assessment Methods and Quality of the Data Mining Systems, International Journal of Computer Integrated Manufacturing Systems, Issue 7, 2009, pp. 1436-1441.

6. Hristov H. Choosing Approach for Data Integration, Proceedings of the 6-th International Conference "Information Systems & Grid Technologies", 1-3 June, 2012, pp. 98-113.

Статия [4]

Naydenova, I., K. Kaloyanova. Some Extensions to the Multidimensional Data Model, ISGT'06: IEEE 2006 John Vincent Atanasoff International Symposium on Modern Computing, 3-6 October 2006, Sofia , Bulgaria, pp. 63-68

е цитирана в

7. Hristov, H. Choosing Approach for Data Integration, Proceedings of the 6-th International Conference "Information Systems & Grid Technologies", 1-3 June, 2012, pp. 98-113, ISSN 1314-4855.
8. Савоска, Сн. Управуване со информации и одлучување: концепти, методологија и компјутерска подршка, Гинис, Македонија, 2012, ISBN 978-608-65059-5-0.

Статия [5]

Naydenova, I., K. Kaloyanova. An Approach Of Non-Additive Measures Compression In Molap Environment, IADIS Multi Conference on Computer Science and Information Systems 2007, Lisbon, Portugal, pp. 394-399

е цитирана в

9. Kovacheva, Z. Tendencies in Data Warehouse Scope Development, Proceedings of the 6th International Conference on Information Systems and Grid Technologies – ISGT'2012. Sofia, Bulgaria, June, 01-03, 2012. pp. 11-21, ISSN 1314-4855.

Статия [9]

Naydenova I., K. Kaloyanova, S.Ivanov. Multi-Source Customer Identification, Third International Conference on Information Systems & Grid Technologies, 28 - 29 May 2009, Sofia, Bulgaria, pp. 77-85

е цитирана в

10. Boyanov, K. The security aspects of the research activities in IICT-BAS, DIMVA 2011-SysSec 1st Project Workshop, 6 July, Amsterdam, Netherlands, pp. 99-101.
11. Kovacheva, Z., Tendencies in Data Warehouse Scope Development, Proceedings of the 6th International Conference on Information Systems and Grid Technologies – ISGT'2012. Sofia, Bulgaria, June, 01-03, 2012. pp. 11-21, ISSN 1314-4855.

Статия [16]

Napoli, J.P, K. Kaloyanova. An Integrated Approach for RUP, EA, SOA and BPM Implementation, Proceedings of the 12th International Conference on Computer Systems and Technologies, Vienna, Austria, 16-17 June 2011, ACM Digital Library, pp. 63-68, pp.63-68, ISBN 978-1-4503-0917-2

е цитирана в

12. S.Savoska, V.Manevska, S.Neshkovska. Usage of Data Warehouse for Data Preparation for the Needs of the State Statistical Office of the Republic of Macedonia and How Was Transparent Data Dissemination Achieved?, NISPAA 2012, Ohrid, 28-29.05.2012.
13. Maneva, N. A Method for Decision Making in Computer Science Research, Proceedings of the 6-th International Conference "Information Systems & Grid Technologies", 1-3 June, 2012, pp. 220-227.
14. Tipografov, M. E. Grigorova. Adaptive Integrated Business Management Systems, , Proceedings of the 6-th International Conference "Information Systems & Grid Technologies", 1-3 June, 2012, pp. 346-351.
15. Dimitrov, V. An Overview of the Department of Defence Architecture Framework (DoDAF), Proceedings of the 6-th International Conference "Information Systems & Grid Technologies" ISGT 2012, 1-3 June, 2012, pp. 139-142.
16. Shoylekova, K., K. Grigorova. Service Oriented Business Architecture. IN: Proceedings of the Eighth International Conference, Internet Education Science, IES -2012, Vinnytsia, Ukraine, 2012, pp. 132-133, ISBN 978-966-641-491-8.
17. Isov, M., K. Grigorova. Arhitecture Vision Mapping between TOGAF and ZACHMAN Arhitecture Frameworks, Proceedings of the Eighth International Conference, Internet Education Science, IES -2012, Vinnytsia, Ukraine, 2012, pp. 122-123, ISBN 978-966-641-491-8.

Статия [17]

Kaloyanova, K., A. Antonov. Object-oriented approach for developing interactive 3D graphics system in education. Proceedings of 7th Annual International Conference on Computer Science and the Education in Computer Science, Sofia/Dobrinishte, Bulgaria, July 06-10, 2011, pp. 89-100, ISSN 1313-8624.

е цитирана в

18. Todorova, M. Implementation of an Approach for Verification of Procedural Programming, GSFT Journal of Computing, vol.2, No.2, June 2012, pp. 70-75.

Статия [19]

Kaloyanova, K. Using the Project Approach in IS Education, Proceedings of 8th Annual International Conference/Workshop on Computer Science and the Education in Computer Science (CSECS), Boston, 5-7 July, 2012, pp. 36-40.

е цитирана в

19. Dimitrov, V. Evolution of Relational Data Model, Proceedings of the International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE-2012), October 5 – 6th, 2012, Sofia, Bulgaria., pp. 365-369, ISBN 978-954-92247-4-0.
20. Todorova M. Applying Program Verification Methods In Software Specialists Education, INTED 2013, Valencia, Spain, pp. 6260-6270, ISBN 978-84-616-2661-8.

Статия [20]

Kaloyanova, K. Design from data: how to use requirements for better information system analysis and design, Proceedings of the International Conference “Informatics in Scientific Knowledge” 2012, Varna, June, 27-29, 2012, pp. 189-197, ISSN 1313-4345

е цитирана в

21. Todorova, M. Applying Program Verification Methods In Software Specialists Education, INTED 2013, Valencia, Spain, pp. 6260-6270, ISBN 978-84-616-2661-8.
22. Todorova, M. Simulation of the Behavior of a System with Social Structure by Means of Generalized Nets, Proceedings of the 13-th International Workshop on Generalized Nets, London, 29 October, 2012, pp. 62-68.
23. Савоска, Сн. Управуване со информации и одлучување: концепти, методологија и компјутерска поддршка, Гинис, Македонија, 2012, ISBN 978-608-65059-5-0.
24. Шойлекова, К., К. Григорова. Изграждане на план, стратегија и бизнис модел на организација от гледна точка на бизнис архитектурата, Научни трудове на РУ”А.Кънчев” – 2012, том 51, серия 6.1, Русе, 2012, pp. 85-89, ISSN 1311-3321.