

МИНИСТАРСТВО ОДБРАНЕ
УНИВЕРЗИТЕТ ОДБРАНЕ У БЕОГРАДУ
ВОЈНА АКАДЕМИЈА

**ПРИЛОГ РАЗВОЈУ BUSINESS
INTELLIGENCE СИСТЕМА НА ПРИМЕРУ
УПРАВЉАЊА УСПЕХОМ СТУДЕНАТА
ВОЈНЕ АКАДЕМИЈЕ**

докторска дисертација

МЕНТОР
редовни професор
др Алемпије Вељовић, дипл. инж.

КАНДИДАТ
мајор
мр Ненад Димитријевић, дипл. инж.

Београд, 2014. године

ЗАХВАЛНИЦА

За успешну реализацију докторске тезе посебну захвалност дугујем ментору професору др Алемпију Вељовићу, и колегама са Војне академије који су својим саветима директно утицали на квалитет овог рада. Велику захвалност дугујем и својој супрузи Ивани, породици и пријатељима који су ме свесрдно подржавали да истрајем у напорима на изради овог рада.

Аутор

САДРЖАЈ

1. УВОД	5
1.1. ПРЕДМЕТ И ЦИЉ ИСТРАЖИВАЊА.....	6
1.2. ХИПОТЕЗЕ.....	7
1.3. НАЧИН ИСТРАЖИВАЊА.....	7
1.4. НАУЧНИ ДОПРИНОС И ОЧЕКИВАНИ РЕЗУЛТАТИ.....	9
1.5. СТРУКТУРА РАДА.....	10
2. BUSINESS INTELLIGENCE СИСТЕМИ	12
2.1. ДОНОШЕЊЕ ОДЛУКА-ЕЛЕМЕНТ ПРОЦЕСА УПРАВЉАЊА.....	12
2.1.1. Процес доношења одлука.....	13
2.1.2. Нивои одлучивања и врсте одлука.....	15
2.2. ПОЈАМ BUSINESS INTELLIGENCE СИСТЕМА.....	17
2.2.1. Информациони системи.....	17
2.2.2. Business Intelligence системи.....	20
2.3. ЕЛЕМЕНТИ BUSINESS INTELLIGENCE СИСТЕМА.....	23
2.3.1. Складиште података.....	24
2.3.2. Вишедимензиона анализа података - OLAP.....	26
2.3.3. Откривање знања у подацима – Data mining.....	32
3. МЕТОДОЛОГИЈЕ И СТАНДАРДИ РАЗВОЈА ИНФОРМАЦИОНИХ СИСТЕМА	35
3.1. ФУНКЦИОНАЛНО И ИНФОРМАЦИОНО МОДЕЛОВАЊЕ.....	40
3.1.1. Стандард за функционално моделовање IDEF0.....	40
3.1.2. Стандард за информационо моделовање IDEF1X.....	50
3.1.3. Веза стандарда IDEF0 и IDEF1X.....	53
3.2. ОБЈЕКТНО-ОРИЈЕНТИСАНО МОДЕЛОВАЊЕ.....	54
3.2.1. UML.....	55
3.2.2. Rational Unified Process.....	63
3.2.3. Microsoft Solution Framework.....	68
3.2.4. Развој вођен моделом (Model-driven development).....	74
3.3. ДИМЕНЗИОНО МОДЕЛОВАЊЕ.....	78

4. РАЗВОЈ ВІ СИСТЕМА	82
4.1. ТРЕНУТНО АКТУЕЛНЕ МЕТОДОЛОГИЈЕ РАЗВОЈА ВІ СИСТЕМА	82
4.2. ПРЕДЛОГ РАЗВОЈА ВІ СИСТЕМА	86
4.2.1. Дефинисање захтева	88
4.2.2. Анализа	92
4.2.3. Дизајн	94
4.2.4. Реализација	98
4.3. ОЦЕНА ПРЕДЛОЖЕНЕ МЕТОДОЛОГИЈЕ РАЗВОЈА ВІ СИСТЕМА	103
5. BUSINESS INTELLIGENCE СИСТЕМ ЗА УПРАВЉАЊЕ УСПЕХОМ КАДЕТА ВОЈНЕ АКАДЕМИЈЕ	107
5.1. ДЕФИНИСАЊЕ ЗАХТЕВА	107
5.1.1. Прикупљање захтева	107
5.1.2. Процена пословног окружења	117
5.1.3. Планирање система	122
5.2. АНАЛИЗА	123
5.2.1. Израда случајева употребе	123
5.2.2. Израда концепта	124
5.2.3. Анализа изворних података	125
5.3. ДИЗАЈН	127
5.3.1. Дизајн складишта података	127
5.3.2. Припрема података	130
5.3.3. Дизајн увођења система	131
5.4. РЕАЛИЗАЦИЈА	134
5.4.1. Израда складишта података	134
5.4.2. Припрема анализе података	135
5.4.3. Израда корисничког интерфејса	143
6. ДОКАЗ ХИПОТЕЗА	154
7. ЗАКЉУЧАК	167
8. ЛИТЕРАТУРА	170
9. СКРАЋЕНИЦЕ	173
10. ПРЕГЛЕД СЛИКА	174

1. Увод

Информациона револуција је условила да успешно пословање буде зависно од присуства информационих технологија у пословним процесима. На самом почетку, идеја о примени информационих технологија се везивала искључиво за повећање продуктивности, ефикасности и ефективности. Тада су информациони системи пројектовани и имплементирани тако да подрже текући начин рада и пословања.[8,12]

Напретком информационих технологија и све већим захтевима модерног пословања, дошло је до развоја система за подршку пословању. Ти системи су као основни циљ имали искоришћавање свих расположивих информација унутар организације, како би руководству омогућили свеобухватно и правовремено одлучивање. Прве системе за подршку одлучивању карактеришу структурирани, периодични управљачки извештаји. Данас, они представљају део сложенијег, динамичнијег и свеобухватнијег концепта који се назива *Business Intelligence*. [8,12]

Управљање на свим нивоима неке организације је кључни елемент који доприноси успеху. Квалитетно стратегијско управљање потребан је услов за успешно пословање. За такав вид управљања потребно је поставити одговарајуће стратегијске циљеве. Успех у постизању тих циљева постиже се на основу правих и правовремених одлука.[48] Неопходни услови за доношење правих и правовремених одлука јесу праве и правремене информације, које се могу добити применом информационих технологија.

Један од неопходних услова за остваривање ефикасног и квалитетног управљања организацијом-институцијом, попут Војне Академије, јесте постојање адекватног и добро организованог информационог система. Једна од функција тог информационог система односи се и на одлучивање. Одлучивање, као предуслов успешном управљању, по својој суштини јесте информациони процес који се реализује кроз трансформацију информација у одлуке, у чему нам могу помоћи *Business Intelligence* (у даљем тексту BI) системи.

Због свега наведеног се као императив поставља развој информационог система који у себи мора укључивати и елементе подршке одлучивању, који ће обухватити све захтеване аспекте пословања једне институције и омогућити интерактиван, брз и флексибилан приступ и различите погледе на податке, као и проналажење скривених законитости у подацима.

Докторска дисертација је настала из жеље ближег дефинисања развоја VI система и примене у процесу управљања. У раду је прво, на основу анализе постојећих стандарда и језика за развој информационих система, дат предлог корака развоја VI система. На основу тако дефинисане методологије извршен је развој VI система за управљање успехом студената Војне Академије. Како је, у току рада на тези, дошло до промене назива полазника Војне академије из студент у кадет, и тексту је коришћен нови назив, односно кадет.

1.1. Предмет и циљ истраживања

Предмет дисертације су *Business Intelligence* системи, а циљ истраживања развој VI система, односно методологија њиховог развоја, и развој VI система за управљање успехом кадета Војне академије. Основни циљ дисертације је да се изврши анализа постојећих методологија за развој информационих система и дефинишу јасни кораци у развоју VI система како би се добио систем који може да обезбеди потребне информације у процесу одлучивања. Као пример ваљаности предложене методологије развоја VI система биће израђен VI систем за управљање успехом кадета Војне академије. Овај систем треба да доносиоцима одлуке на Војној академији да нови квалитет података о успеху кадета.

Основна поставка дисертације везана је за концепт интегралности, који подразумева обједињавање пословних процеса, значајних за одлучивање о конкретном проблему, у VI систем. Допринос развоју VI система, ће се реализовати кроз интеграцију класичних метода функционалног и информационог моделовања са објектно оријентисаним приступом у развоју информационих система. Методологија мора да буде примерена развоју VI система у условима када већ постоји раније акумулирано знање о проблему. Добијени резултати треба да покажу да VI систем, пројектован применом предложене методологије, може бити подршка у одлучивању.

1.2. Хипотезе

Основна хипотеза у дисертацији је да развој VI система, применом класичних и објектно-оријентисаних стандарда за моделовање, димензионог моделовања и интеграционих и аналитичких сервиса система за управљање базама података, омогућава дефинисање и анализу корисничких захтева, дизајн и имплементацију решења за анализу података ради доношења одлука потребних управљању.

Посебне хипотезе су везане за карактеристике које реализовани VI систем мора да има како би одговорио својој намени:

- VI систем мора да је конципиран на аналитичком моделу података – концепт складишта података,
- VI систем мора да омогућава анализу и поређење свих података релевантних за доношење одлуке,
- VI систем мора да буде скалабилан.

1.3. Начин истраживања

Имајући у виду подручје и циљ истраживања користиће се следеће методе, стандарди и језици:

- индуктивни и дедуктивни метод,
- метода посматрања,
- метода моделовања-као најзначајнија метода у развоју информационих система,
- стандард за функционално моделовање IDEF0 за поступак моделовања процеса,
- стандард за информационо моделовање IDEF1X за поступак моделовања података,
- UML као језик за објектно оријентисано моделовање.

Индукција и дедукција су релативно самосталне посебне научне методе. Међутим, оне су и међусобно повезане у јединствену индуктивно-дедуктивну методу. Индукција и дедукција се међусобно допуњују и веома су блиске и као такве могу се примењивати

у истраживачком процесу и процесу научног сазнања уопште чинећи јединствену индуктивно-дедуктивну методу. Индуктивно-дедуктивна метода, подразумева комплекс метода. Индукција подразумева синтезу и генерализацију, а дедукција анализу и специјализацију.[19]

Под појмом **метода посматрања** подразумева се врста методе научног скупљања података непосредним чулним опажајем. Основна карактеристика методе посматрања јесте да се њеном применом непосредно долази до оригиналних и аутентичних података, у реалном времену и без преносилаца који би могли да их деформишу. Она је заснована на научном посматрању и непосредном чулном опажају, односно предмету који је непосредно чулно опазив.[19]

Метода моделовања је специфична врста општих научних метода која се заснива на коришћењу модела у сазнајном процесу ради детаљног сазнавања предмета истраживања. Користи се у свим наукама и научним дисциплинама, нарочито кад је истраживањем тешко или немогуће открити структуру, правилности и функционалне законитости предмета истраживања. До сазнања о предмету истраживања се долази применом аналогije између модела и реалног предмета истраживања.[19]

Основни циљ методе моделовања јесте што тачније и што дубље сазнавање структуре, функционисања или понашања предмета истраживања. Да би се тај циљ остварио потребно је да буду испуњени следећи услови [19]:

- модел мора да буде сличан оригиналу у структури, понашању и резултату функционисања,
- модел мора да представља одређени теоријско-сазнајни или практично реални одраз оригинала и
- модел мора да пружи одређену информацију о предмету истраживања.

Под појмом моделовања подразумева се процес израде модела. **Модел** је коначан резултат процеса моделовања. Важно је ипак још једном напоменути, да процес моделовања не подразумева само израду модела. Процес моделовања је истовремено и процес стицања нових сазнања о предмету истраживања.

Са научног аспекта модел представља упрошћену слику оригинала. То упрошћавање се не односи на оне елементе који се истражују. Напротив, настоји се да они буду што је могуће више верни оригиналу.

Модел се примењује у свим областима људске делатности и без њих се не може замислити ниједно истраживање. У зависности од њихове примене у решавању конкретних проблема, они имају различите функције, а основне су примењена, образовна и истраживачка.

Стандарди IDEF0 и IDEF1X, димензионо моделовање и UML као језик за објектно оријентисано моделовање ће бити детаљније описани у наредним поглављима дисертације.

1.4. Научни допринос и очекивани резултати

У литератури постоји више приступа или методологија за дизајн и развој информационих система. Међутим, када се ради о VI системима не постоји консензус око тога која је од постојећи методологија најбоља, односно постоји свега неколико приступа развоју VI система. Неке од традиционалних методологија и нису директно примењиве за развој VI система.

Научни допринос ове тезе везан је за унапређење методологије развоја VI система. Очекивани резултати везани су за спровођење постављене методологије развоја VI система на конкретном примеру кроз развој VI система за управљање успехом кадета Војне академије и доказивање постављених хипотеза. Такође, очекује се да допринос ове тезе буде и у области управљања успехом кадета Војне академије кроз интеграцију свих расположивих података и утврђивање њихове релације са успехом кадета.

Очекивани резултати ове докторске дисертације су:

- анализа методологија и стандарда за развој информационих система;
- анализа методологија развоја VI система;

- прилог методологији развоја BI система, кроз предлог конкретних фаза и корака развоја;
- функционални и информациони модел система, реализован применом IDEF0 и IDEF1X стандарда и UML-a као језика за објектно оријентисано моделовање;
- складиште података (савремено конципирано на аналитичком моделу података), које треба да омогући прихват и чување свих релевантних података;
- вишедимензиона анализа унетих података и њихово праћење у реалном времену, која треба да руководству пружи све информације у процесу одлучивања;
- скалабилан и поуздан *Business Intelligence* систем, као доказ ваљаности дефинисане методологије и оригинално решење за анализу података потребних управљању успехом кадета Војне академије.

1.5. Структура рада

Рад се састоји из увода, пет поглавља и закључка.

Увод објашњава циљ рада и основне хипотезе, као и очекиване резултате. Због важности коју има, у уводу су кратко описане и методе, стандарди и језици који ће се примењивати у реализацији рада.

Поглавље “*Business Intelligence* системи“ говори о значају доношења одлука, појму и основним елементима *Business Intelligence* система. Посебна пажња је посвећена односу информационих и *Business Intelligence* система. Такође, описани су и појмови складиште података, вишедимензиона анализа података и откривање знања у подацима.

У поглављу “Методологије и стандарди развоја информационих система“ дат је преглед најважнијих методологија које се користе у развоју информационих система. Посебно су обрађене функционално моделовање, објектно-оријентисане методологије и димензионо моделовање. Наведене методологије су, применом индуктивно-дедуктивне и методе посматрања, критички обрађене са аспекта могуће примене у развоју BI система. Овим поглављем је уједно реализовано истраживање наведених методологија.

Поглавље “Развој BI система“ представља одговор на основну хипотезу везану за методологију развоја BI система и заједно са поглављем “Методологије и стандарди развоја информационих система“ представља конкретан научно-истраживачки допринос у овој области и реализацију циља истраживања. У поглављу је дат предлог фаза и корака развоја BI система.

У поглављу “*Business Intelligence* систем за управљање успехом кадета Војне академије“ су, у складу са предложеном методологијом, дефинисани кориснички захтеви, извршена анализа, дизајн и реализација BI система за управљање успехом кадета Војне академије. Ово поглавље уједно представља доказ дела хипотезе који се односи на израду BI система за управљање успехом кадета Војне академије

У оквиру **шестог поглавља “Доказ хипотеза“** посебно су истакнути резултати рада и уједно указано на доказе постављених хипотеза.

У **закључку** су дата завршна разматрања самог рада, указано је на његову могућу примену и потребу наставка научно-истраживачког рада у овој области.

На крају рада су дати литература, преглед скраћеница и слика.

2. Business Intelligence системи

2.1. Доношење одлука-елемент процеса управљања

За већину аутора појмови управљање, руковођење и менаџмент имају исто значење односно представљају синонине. Међутим, за сваки од наведених појмова осим садржајне сличности постоје и одређене различитости. Управљање је свесно усмеравање чланова организације према остварењу постављених циљева организације и подразумева одређивање пословних циљева, рокова у којима се циљеви морају остварити, начина уз помоћ којих ће се циљеви остварити, као и доношење пословних одлука. Управљање институцијом укључује доношење одлука и решавање проблема одлучивања. [48] Информациони систем је тај који обезбеђује податке, информације и знање за потребе доношења одлука и решавање проблема одлучивања и управљања[1].

Доношења одлука, као елемент процеса управљања, представља врло важан део пословања, који се одвија свакодневно и на свим нивоима. Циљ процеса управљања је да доведе систем у жељено стање, па је управљање оријентисано на предвиђање, односно усмерено ка будућим догађајима.

Доношење одлуке је процес утврђивања избора начина решавања неког проблема у складу са циљевима организације и околностима у којима она делује. Врста одлуке зависи од типа проблема и услова одлучивања[21]:

- према *времену* на које се одлуке односе, разликујемо краткорочне, средњорочне и дугорочне;
- према *правима*, нивоу и овлашћењу у одлучивању разликујемо одлуке власника, одлуке главног менаџера, одлуке извршних менаџера, итд.;
- према *важности*, одлуке делимо на стратешке, тактичке и оперативне;
- према *пословној јединици* на коју се одлуке односе, разликујемо одлуке везане за одређени процес или одлуке које се односе на организацију у целини;
- према *структурираности и степену неизвесности*, разликујемо високо структуриране, полуструктуриране и неструктуриране одлуке;

- према *поновљивости* разликујемо репетитивне, адаптивне или иновативне одлуке;
- према *броју доносиоца одлука* имамо појединачне или заједничке одлуке.

У ауторитативним институцијама, попут Војске, постоји јасно дефинисана хијерархија у сфери доношења одлука. Модерне институције усмерене су ка децентрализацији одлучивања где сви запослени имају могућност за доношење неких одлука. Одлуке разликујемо по важности и утицају на пословање, а већину одлука доносе менаџери, људи који имају довољно знања, вештина и способности за такве изазове. Вредност менаџера је да од утемељених података и информација из околине креирају нову информацију, односно створе нову вредност. [21] Њихов је задатак да дају иницијативу и делују на пословни систем тако да га доведу у боље стање. Разлика та два стања је менаџеров допринос, односно вредност коју је он створио.

На основу свега реченог можемо закључити да је одлучивање процес предвиђања и вођења система ка жељеном стању. Бројни су мерљиви и немерљиви параметри који утичу на одлучивање и чине га сложеним.

Многе одлуке су оптерећене ризиком и неизвесношћу, који настају као последица динамичног пословног окружења. Сложеност и обележја проблема који се решава одређују и ниво менаџмента којем су такви проблеми намењени. Пуна извесност отклања сложеност и несигурност. У таквим ситуацијама обично се користе интуиција и латерални приступ, а примена информационих технологија значајно доприноси успешном решавању оваквих проблема. [21]

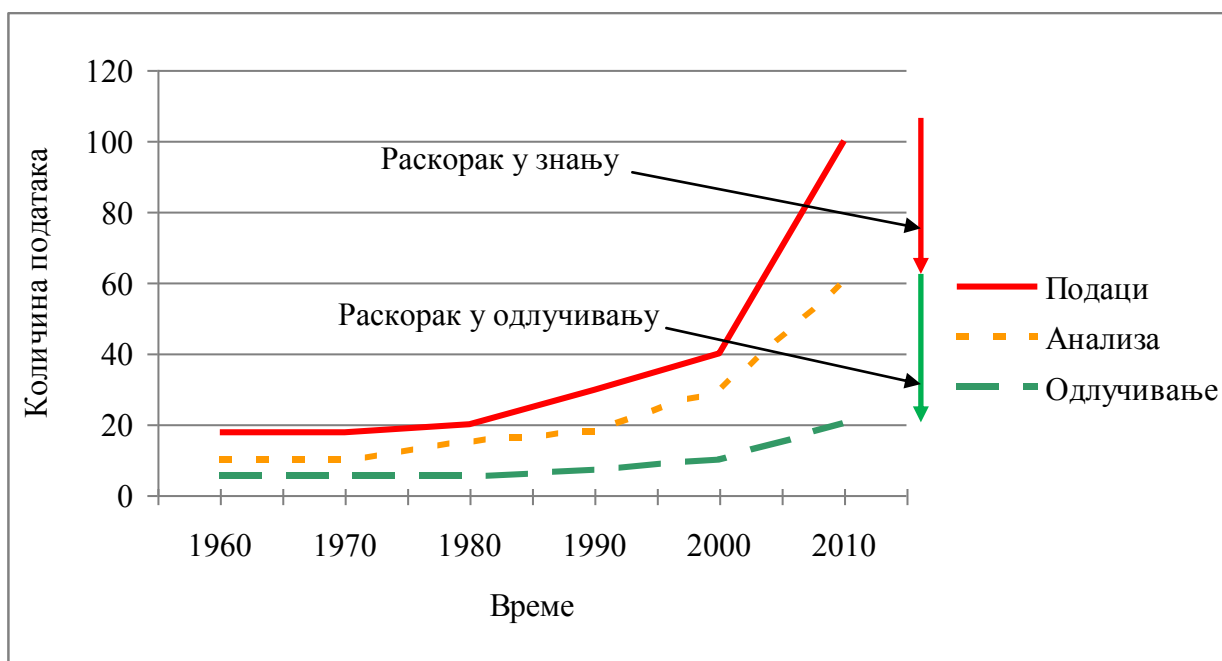
2.1.1. Процес доношења одлука

Процес одлучивања се, најчешће, састоји од следећих фаза:

- идентификација проблема,
- утврђивање и оцене алтернатива,
- избора између алтернатива,
- конкретне акције,
- оцене успешности одлуке.

Успешни менаџери врло често унапред анализирају могуће проблеме и њихова решења, чиме ублажавају или потпуно елиминишу ефекат изненађења појавом проблема. При идентификацији проблема подједнако су важни аналитички и интуитивни приступ. Обично се проблеми прво покушавају идентификовати рационалним и аналитичким размишљањем при чему се најчешће користе квантитативне мере. Након што је проблем дефинисан, следи утврђивање и оцена алтернатива. Постављање хипотезе иницира дијагностику, односно испитивање да ли одабрана хипотеза задовољава критеријуме. [21] При разматрању алтернатива треба узети у обзир и временску димензију одабране опције, односно треба узети у обзир и могући будући развој догађаја. Дакле, алтернативе које у садашњем тренутку изгледају привлачно и дају решење проблема, треба размотрити и са аспекта могућих промена услова у будућности. При анализирању алтернатива могу послужити технике симулације, предвиђања, линеарног програмирања, статистичке методе и слично. Успешна дијагностика води генерисању оптималног решења.

Основа процеса одлучивања су информације и знање. Стварање новог знања на темељу релевантних информација побољшава искуствено поимање неког проблема. [3] Трендове менаџмента треба пратити и непрекидно проучавати, чиме се ствара ново знање и унапређује пословање.

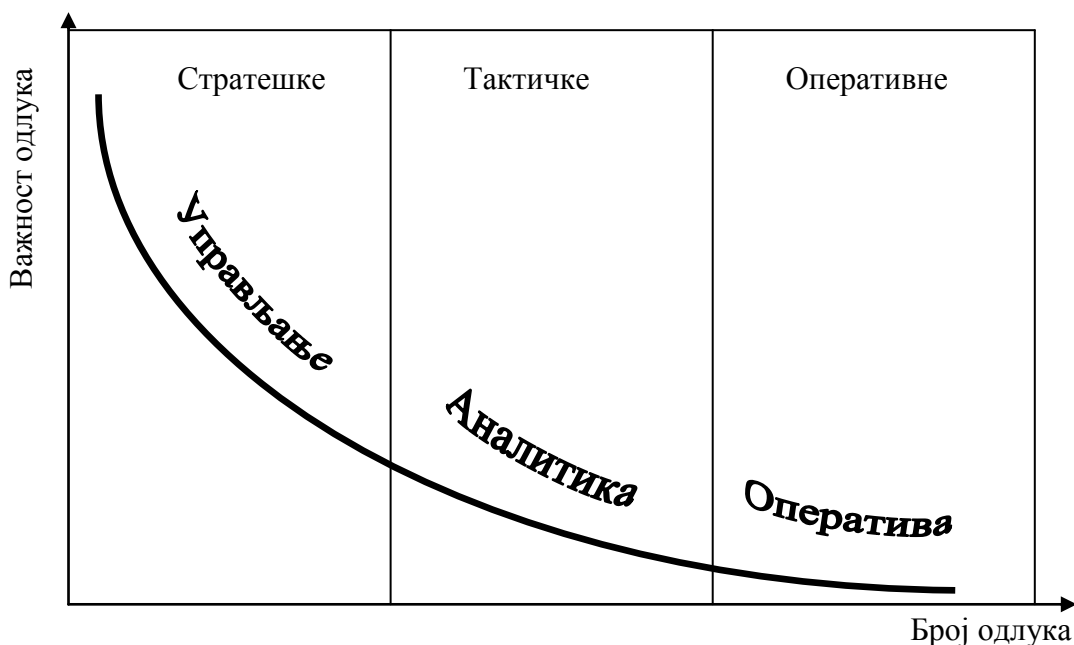


Слика 2.1 – Раскорак у знању и одлучивању [3]

Ако успоставимо везу између количине података које се налазе у информационим системима, могућности њихове анализе и доношења одлука, током времена, долазимо до закључка да енормним растом количине података у последњих петнаест година долази до раскорака који се огледа у смањењу способности доношења одлука трансформацијом информација у знање. ВІ системи представљају једно од решења за превазилажење овог проблема.

2.1.2. Нивои одлучивања и врсте одлука

Развој информационих система који пружају подршку у процесу одлучивања кроз употребу ВІ система, омогућује менаџерима помоћ и подршку у одлучивању чиме се утиче на квалитет пословања и продуктивност. [27] Институције све више прихватају значај информационих система у одлучивању и покушавају да развију информационе системе који ће омогућити доношење бољих и квалитетнијих одлука које су директно усмерене на побољшање ефикасности и продуктивности пословања. Информациони системи и информационе технологије који се користе при одлучивању разликују се с обзиром на временски критеријум одлучивања, нивое примене унутар организације, и будуће кориснике.



Слика 2.2 – Нивои одлучивања [2,3,27]

Три су основна организациона нивоа:

- највиши или стратешки менаџмент-управљање,
- средњи или тактички менаџмент-аналитика и
- најнижи или оперативни менаџмент-оператива.

Нивои унутар организације су у корелацији с временским критеријума одлучивања, тако да су краткорочне одлуке везане за најнижи ниво одлучивања, средњорочне одлуке за средњи (тактички) ниво, а дугорочне одлуке за највиши (стратешки) ниво одлучивања.

Иако су према временској оси одређени нивои одлучивања, врло је значајна динамика процеса одлучивања. Наиме, без обзира на ниво одлучивања, брзина доношења одлуке често је врло значајан елемент одлучивања. Тренутна информација, информација у реалном времену, знатно убрзава пословне процесе и утиче на ефикасност пословања. Сваком нивоу унутар организације припада одговарајући степен агрегираности и развоја информација. Информације се, пењући се кроз хијерархијску пирамиду, синтетизују и сумирају тако да највишем нивоу менаџмента треба пар синтетизованих, али кључних, већ припремљених информација квалитативног обележја, које су настале обрадом информација на нижим нивоима одлучивања.[2]

Оперативни менаџмент користи велике количине детаљних података и информација које дају преглед током неке активности или процеса. Такве одлуке се доносе готово свакодневно, рутинског су типа и понављајуће, па оперативни менаџери имају искуство и знање које је потребно за доношење таквих одлука. Такве одлуке су добро структуриране и лако програмабилне. [2]

Тактички менаџмент-аналитика баве се сложенијим проблемима који захтевају веће искуство. За њих је карактеристичан одређени степен неизвесности, па их зовемо полуструктурираним проблемима. Аналитика поседује одређено искуство и знање о неким аспектима одлучивања, али ипак постоји одређена неизвесност везана за одлуку. Временска димензија таквих проблема је нешто дужа, најчешће средњорочна и они морају да осигурају услове потребне за реализацију одлука највишег нивоа менаџмента. Информације које користе аналитичари су сумиране и синтетизоване и најчешће су у облику сажетих периодичних прегледа, прегледа изузетака или прегледа тренда неког процеса. Информациони систем који подржава одлучивање аналитике, најчешће је

извештајног типа, али треба да има флексибилност и интерактивност да би могао подржавати различите стилове одлучивања. Често су такви системи рађени “по мери“ и подржавају специфичне менаџерске захтеве.

Стратешки менаџмент-управљање доноси одлуке које имају дугорочни утицај на пословање и припадају стратегији институције. Такве одлуке су тешко предвидиве, окренуте будућим догађајима, предвиђањима трендова у делатности, дугорочним проценама и за њих је везан висок степен неизвесности. Оне су неструктуриране и односе се на потпуно нове ситуације, врло ретко се понављају па су за ову врсту одлучивања потребни вештина, искуство, знање и способност процене. Чест је случај да за неструктуриране одлуке недостаје знање или искуство, па је нарочито важно информатички подржати ову врсту одлучивања. Будући да су такве одлуке непредвидиве и карактерише их висока неизвесност, њих је и најтеже информатички подржати. [2]

Управљање ма којом организацијом укључује доношење одлука и решавање проблема одлучивања. [21] Информациони систем је тај који обезбеђује податке, информације и знање за потребе доношења одлука и решавање проблема одлучивања и управљања.

2.2. Појам Business Intelligence система

2.2.1. Информациони системи

Полазећи од тумачења појма система (систем је скуп објеката-ентитета и њихових међусобних веза усмерених ка остваривању заједничког циља), информациони систем (ИС) се може дефинисати као систем у коме се везе између објеката и везе система са околином остварују разменом информација.[16] У ширем смислу информациони систем представља уређени и интегрисани скуп података, процеса, интерфејса, мрежа, технологија и људи који су у међусобној корелацији у циљу подршке и побољшања свакодневних пословних операција и подршке менаџменту у решавању пословних проблема, планирања, управљања, предвиђања, координирања и доношења одлука.[18]

Информациони систем је дакле подсистем пословног система, који прикупља, складишти, чува, обрађује и дистрибуира информације које су важне за пословање.

Информациони систем треба да осигура несметан проток и доступност информација свим спољним и унутрашњим корисницима. Наиме, основни задатак информационог система је да прати пословање институције, аутоматизује ручне поступке, али и подржава одлучивање и нуди напредна и корисна решења која су усмерена на стварање и додавање нове вредности. Информациони систем институције добија информације из различитих извора (спољних и унутрашњих) и обрађује их у нове и употребљиве информације. Вредност информационог система може се посматрати са аспекта генерисања информација корисних за пословање, подршке одлучивању и способности информационог система за стварање и генерисање нових знања помоћу којих институција остварује конкурентску предност.[16,18]

Информациони системи и информационе технологије имају кључни утицај на извођење пословних процеса, без обзира на врсту делатности. Информација која је резултат обраде и функционисања информационог система, представља подлогу за одлучивање. Такве информације могу настати обрадом података из више извора: унутрашњи подаци, подаци које су створили менаџери на свим нивоима и спољни подаци.

Имајући у виду претходно дефинисан појам информационих система долази се до дефиниције пословних информационих система. Пословни информациони системи су дакле информациони системи који подржавају пословне функције и обезбеђују пословну аналитику. Пословни информациони системи се ослањају на пет основних ресурса:[18]

- Људски ресурси – укључују кориснике ИС и оне који развијају, одржавају и рукују системом;
- Хардверски ресурси – укључују све типове “машина“, као што су рачунари у најширем смислу ове речи, и слично;
- Софтверски ресурси – укључују рачунарске програме, приручнике, политику компаније;
- Комуникациони ресурси – укључују мреже и неопходан хардвер и софтвер који их подржава;
- Ресурси података – описују све податке којима организација има приступ, без обзира на њихову форму. Укључује базе података, фајлове и фасцикле.

С обзиром на свеобухватности примене информационих технологија у пословању, врсте и обим информација које информациони системи обрађују, информационе системе начелно можемо разврстати у следеће две групе:

- трансакциони информациони системи и
- системи за подршку одлучивању.

Трансакциони информациони систем пружа информације о свакодневним активностима и пословима који су у домену оперативног нивоа одлучивања. Такве су информације усмерене на решавање добро структурираних проблема који су лако програмабилни и који подржавају рационални приступ одлучивању. Одлучивање у тим случајевима углавном је лишено било какве неизвесности и користи врло једноставне математичке функције (нпр. минимум, максимум, збир). Трансакциони информациони систем ће оперативним менаџерима дати информацију о тренутном стању.

Иако трансакциони информациони систем представља почетну фазу примене и развоја информационих система, не може се занемарити његова улога у одлучивању. Трансакциони информациони систем даје информацију у реалном времену и тиме доприноси динамици одлучивања. Иако је оперативни ниво одлучивања углавном лишен неизвесности, динамика краткорочног одлучивања има врло велику улогу у пословању јер опслужује организацију тренутном информацијом о стању процеса или активности. Подаци о пословним догађајима и трансакцијама смештају се у базама података. Базе података чувају сирове податке и чињенице које се након тога на тражени упит-захтев у истом облику нуде корисницима. Над тим подацима (моделима података) врше се одговарајуће операције, функције (модел функција), што ствара објектне структуре и заједно с извештајима, решењима и корисничким интерфејсом представљају трансакциони информациони систем.[31]

Системи за подршку одлучивању су информациони системи који имају за циљ да подрже пословне процесе. Они подржавају решавање полуструктурираних или неструктурираних проблема који су компетенција виших нивоа менаџмента и њихов циљ је да се омогући квалитетније одлучивање, па их с тог аспекта можемо посматрати као конкурентско “оружје“. Системи за подршку одлучивању су флексибилни и интерактивни, прилагођени кориснику, врло често персонализовани јер подржавају

различите стилове одлучивања. Системи за подршку одлучивању подржавају све фазе одлучивања, почев од фазе формулације проблема, преко фазе пројектовања, фазе избора, па све до имплементације. [22]

Током времена, концепт система за подршку одлучивању је еволуирао, тако да имамо:

- 1960. – системе за израду структурираних извештаја који су омогућавали интерактивну претрагу;
- 1970. – MDS (*Management Decision Systems*) као системе подршке одлучивању;
- 1980. – GDS (*Group Decision Support Systems*) и EIS (*Executive Information Systems*);
- 1990. – развој *Data Warehouse*-а и система за аналитичку обраду података (OLAP) као и визуелног моделовања тих система;
- 2000. – званична појава *Business Intelligence* система.

2.2.2. Business Intelligence системи

Израз *Business Intelligence* први пут је употребљен 1996. године како би означио категорију средстава анализе податка, постављања упита, извештавања, који кориснику у пословним процесима могу помоћи да из огромне количине података синтетизују вредне информације на којима ће заснивати своје пословне одлуке (*The Gartner Group, Септембар 1996*).[1, 53]

Business Intelligence систем као термин се најчешће користи за означавање рачунарске подршке одлучивању. ВІ систем је део информационог система организације наменски развијен да подржи управљање. Управљање захтева свеобухватан и благовремен увид у показатеље функционисања организације како би доношене одлуке биле поуздане и прецизне. Према савременим теоријама менаџмента овај увид треба омогућити што већем броју запослених од којих се, када га добију, може очекивати и већа ефикасност и одговорност за остварене резултате. ВІ технике (*data warehousing, reporting, OLAP, data mining, dashboards* и др.) екстрахују податке из постојећег информационог система и трансформишу их у облик погодан за одлучивање. Имплементација ВІ техника повећава употребну вредност постојећег информационог система организације, услед чега је интересовање за ВІ системе велико и још увек расте.[27, 52]

Пошто не постоји универзална дефиниција појма *Business Intelligence*, различити аутори га дефинишу на различите начине. Једна од најчешће коришћених и општијих дефиниција је следећа: “*Business Intelligence* је такво коришћење података које води ка доношењу бољих пословних одлука. Оно се односи на приступ, анализу и откривање нових могућности”.[3]

Неки аутори *Business Intelligence* дефинишу на следећи начин:[3]

- Larissa T. Moss, Shaku Atre: “*Business Intelligence* није ни производ нити систем. Она представља архитектуру и колекцију интегрисаних оперативних апликација и апликација за подршку одлучивању као и базе података које пословној организацији омогућава лак приступ подацима. “
- Steve Mutfitt: “*Business Intelligence* је начин достављања правих информација у правом формату у праве руке у право време. Дobar систем пословне интелигенције прикупља информације из свих делова организације, анализира их, припрема потребне извештаје и шаље их људима који их требају. На тај начин сваки појединац добија информације скројене према његовим потребама.”
- Ravi Kalakota, Marcia Robinson: “Претварање података у знање је задатак апликација познатих под називом *Business Intelligence*. *Business Intelligence* је скупина нових апликација обликованих тако да могу организовати и структуриране податке о пословним трансакцијама на начин који омогућава анализу корисну у подршци одлучивању и оперативним активностима организације.”
- Len Wyatt: “Када компанија користи податке из својих извршних система при доношењу одлука, то је *Business Intelligence*. *Business Intelligence* системи преузимају податке из извршних система и реструктурирају их тако да корисници могу откривати раније непознате информације о предузећу, пословању, клијентима, пословним активностима или производима.”
- Vinod Badami: “*Business Intelligence* је процес прикупљања расположивих интерних и релевантних екстерних података, и њихова обрада у корисне информације које могу помоћи пословним корисницима при доношењу одлука.”

Свака организација поседује податке о свом пословању најчешће у разним врстама информационих система, документима, пословним књигама и на другим носиоцима података. Процесирање тих података, проналажење скривених релација међу њима, идентификација нових могућности за боље пословање, односно трансформација података у информације и знање, представљају значајан развојни потенцијал сваке организације.

Business Intelligence концепт користи податке да би се дошло до информација и знања потребних за доношење одлука. Ови појмови налазе се у међусобним односима логичке надређености и подређености, односно логичке хијерархије, што је приказано на следећој слици. [2,8]



Слика 2.3 – Логичка хијерархија података, информација и знања[2,8]

Под *подацима* се подразумевају детаљи о разним стварима, догађајима и трансакцијама, који су структурирани, класификовани и меморисани, али најчешће нису организовани на начин да имају неко конкретно значење. Подаци могу бити нумеричког или алфанумеричког типа, фигуре, звукови или слике. У нашем случају разматрамо нумеричке и алфанумеричке податке меморисане кроз разне елементе пословног система организације.

Информације су подаци који су организовани тако да имају конкретно значење за оног ко их прима. Прималац информације интерпретира то значење и из њих изводи закључке. Да би подаци добили неко значење за одређену акцију или одлуку, они се прво морају обработити. Тек овакви подаци могу имати значаја у процесу одлучивања.

Знање се састоји од података организованих на начин да преносе искуство, акумулирано учење и експертско знање а у вези проблема који треба разрешити. Знање представља скуп података процесираних са циљем да извуку критичне импликације и рефлектују искуства из прошлости.

Информације су кључни фактор у доношењу одлука на било ком нивоу, почев од почетног преко оперативног, до стратешког нивоа одлучивања. С обзиром на важност сваке одлуке, информације на бази којих се оне доносе морају бити потпуне, правовремене и прецизне. Међутим, пут од података до информација и коначно до могућности предвиђања будућих пословних потеза (односно откривање знања у тим подацима) није нимало лак.

Обједињујући у себи концепте складишта података, OLAP и Data mining, BI концепт задовољава потребу менаџмента организација за информацијама на свим нивоима управљања: стратегијском, оперативном и основном нивоу. За ниво стратегијског управљања неопходне су информације више општости, у виду збирних прегледа и историјског карактера. Насупрот томе, за ниво основног управљања потребне су специфичне информације уског домена, претежно у виду „сирових“ података, врло актуелних, углавном квантитативне природе и велике прецизности и тачности. Оперативном нивоу управљања неопходне су претежно интерне информације, понекад историјског карактера, односно информације које омогућавају анализирање пословања и предвиђање.

2.3. Елементи Business Intelligence система

Business Intelligence, као концепт, обједињује методологије, технологије и платформе за складиштење података (*Data Warehousing*), OLAP процесирање података (*On-line Analytical Processing*) и откривање „знања“ у подацима (*Data Mining*) које омогућава корисницима креирање корисних управљачких информација из података о пословању који се налазе дисперзовани на различитим трансакционим системима који долазе из различитих интерних и екстерних извора.

Порастом количине података у базама података, растао је и број ентитета и атрибута које је било потребно укључити у процес анализе података, самим тим и у процес доношења одлука. Са циљем да се подржи процес стратешког одлучивања и омогући компанијама да дођу до информација којима ће обезбедити компететивну предност, развијене су процедуре аутоматизоване анализе података, које су омогућиле проналажење скривених законитости у великим скуповима података, односно екстракцију „знања“ из података. Те процедуре, које обједињавају статистичке методе, компоненте вештачке интелигенције, машинског учења и визуализације података, називају се *Data Mining* или откривање знања у подацима.

2.3.1. Складиште података

Централно место целог концепта ВІ система припада складишту података. Складиште података (*Data Warehouse*) је посебно дизајнирана, аналитичка база података у којој се чувају подаци прикупљени из интерних или екстерних извора, и која омогућава сложене, унапред непредвиђене (*ad-hoc*) упите великом броју различитих података.

Концепт складиштења података је складиштење агрегираних, екстрахованих и филтрираних података у базе података, које омогућавају слојевит, вишедимензиони приступ подацима, какав је потребан за доношење одлука највишег стратешког нивоа.

Централизација података је неопходна како би се корисницима омогућили приступ и анализа информација којима је иначе јако тешко прићи у трансакционим базама података.

Складиште података припада вишедимензионим базама података које су израђене на основама димензионог модела. Према идејном творцу W.H. Inmon-у, складиште података представља онај облик базе података којем су својствена следећа четири елемента[39]:

- усмереност предметима (функционалним подручјима),
- садржајна непроменљивост,
- интегрисаност,
- везаност уз време (временска одређеност).

За концепт складиштења података каже се и да је субјективно оријентисан што значи да је фокусиран на одређене теме, тј. организационе активности или процесе одлучивања. У оквиру сваке теме налазе се подаци о оним ентитетима који су битни за доношење специфичне пословне одлуке.

Интегрисаност, складишта података, подразумева да је реч о централизованом бази података која садржи податке из различитих извора унутрашњих или спољашњих. Интегрисаност даље значи и употребу конвенције имена, доменских ограничења, итд.

Временска зависност се огледа у зависности података од времена. Непроменљивост значи да се складиште података не мења перманентно, већ се само периодично допуњује новим подацима из трансакционих система. Овде се као кључне особине издвајају временска димензија и квалитетно разрађен приступ према правилима трансформације пословних, трансакцијских и других података у складиште података.

Постоје два водећа приступа организовању података у складиштима: димензионални (уведен и представљен од стране Ralfa Kimbala [43]) и нормализован (заступан од стране W.H. Inmon-а [39]).

У димензионалном приступу трансакциони подаци су подељени, или у измерене чињенице, тј. нумеричке податке који представљају специфичну вредност, или димензије које садрже референтне информације, дајући свакој трансакцији њен контекст. Основна предност овог приступа је лакоћа управљања и брзина, услед преспајања у димензије. Међутим, истовремено, у случају промене или додавања настају велике потешкоће. Нормализоване табеле се групишу по предметним областима које осликавају дефиницију података. У тој ситуацији, при активности додавања, промене су једноставне, али се губи на перформансама.[37]

За складиштење података већином се израђује димензиони модел, за разлику од трансакционих база података које увек имају релациони модел података темељен на објектима и њиховим односима. Димензиони модел складишта података даје боље могућности визуализације података што је, с обзиром на намену складишта, велика предност. Управо могућност лаке визуелизације података јест кључ разумевања и прихваћања димензионог модела података. [2, 37]

Модел података трансакционих система приказују пословне објекте и њихове међусобне односе који произлазе из пословног процеса. Дводимензионални табеларни модели трансакционих система прикладнији су за праћење односно управљање пословним процесом, а димензиони модели складишта података за извештавање и анализу о резултатима пословних процеса.

Делови архитектуре складишта података су:

- Извори података (*Data sources*)
- ETL процеси (*Extraction Transformation Loading*)
- Модел базе података (логички и физички)
- OLAP сервер, коцка
- Meta подаци (*Meta data*)
- Складиште оперативних података (*Operational Data Storage*)
- Data martovi (*Data Marts*)
- Алати за извештавање и анализу (*Reporting and Analytical tools*)

2.3.2. Вишедимензиона анализа података - OLAP

On-line Analytical Processing (OLAP) омогућава брз и флексибилан приступ подацима и представља надградњу складишта података. OLAP је намењен on-line анализама и извештавањима, за разлику од продукционих система намењених ажурирању база података и обради трансакција (*On-line Transaction Processing – OLTP*).

Трансакционе базе података (OLTP), као доскорашње примарне имплементације информационих система, омогућавале су унос података и постављање упита над тим подацима, генерисање извештаја и дијаграма. Како је количина података у базама података почела нагло да расте, свако генерисање сложених упита доводило је у питање основну намену трансакционе базе података – а то је унос података, што се манифестовало кроз успорен рад. У исто време појавио се проблем добијања квалитетних извештаја из OLTP система. Разлози за ову чињеницу су следећи:

- Анализира се велика количина података,
- Анализа дуго траје,
- Резултати су компликовани за приказивање,

- Обавезно је присуство лица са информатичким знањима,
- Велики напор за оперативни систем рачунара.

Због наведеног проблема OLTP система, следећи логичан корак био је раздвајање свакодневних оперативних процедура од захтева за анализом података и развоја алата који би омогућили анализу добијених резултата и самим тим подржали (са информатичке тачке гледишта) процес доношења одлука. Тако је настао концепт складишта података, као засебно дизајниране вишедимензионе базе података чија је основна намена била да издвоји из трансакционе базе са циљем да те податке учини доступним за даљу анализу коришћењем, пре свега, OLAP технологија које су донеле са собом и нови концепт вишедимензионих упита.

Табела 2.1. Основна разлика OLTP и OLAP обраде података.[22]

OLTP	OLAP
Обрада података	Анализа података
Велик број трансакција	Комплексни упити
Mb-Gb података	Gb-Tb података
Сирови подаци	Сумарни и прерађени подаци
Мноштво различитих корисника	Доносиоци одлука и аналитичари
Ажурирани подаци	Историјски подаци

За разлику од SQL упита, који се постављају над трансакционом базом и који дају одговор на питање - шта се налази у бази података, OLAP иде корак даље и улази у сферу значења података и даје одговоре на питања да ли су неке претпоставке које су издвојене из базе података истините. Односно, OLAP ствара серију хипотетичких законитости и веза између података и користи системе упита да их потврди или оповргне.

OLAP приступ омогућио је свеобухватан приступ подацима, организујући их и приступајући им кроз димензије које репрезентују пословне факторе као што су време, географски региони или класификације итд. То је омогућило да се подаци представе у пословном контексту, односно да се дође до информација које ће подржати процес

доношења одлука. Основну предност коју је донео OLAP концепт у односу на SQL упите, јесте могућност интерактивног испитивања различитих пословних фактора кроз генерисање вишедимензионих упита. Основни елементи OLAP система су [22]:

- Складиште података – OLAP база, као база потребних информација,
- OLAP сервер, за управљање и манипулацију ускладиштеним подацима,
- веза-интерфејс, према кориснику и према другим апликацијама,
- алати за управљање и подешавање.

OLAP база података је дефинисана следећим компонентама:

- Нумеричке мере – мере су вредности података или чињенице које корисници анализирају. Примери мера су износ продаје, трошкови продате робе итд.
- Димензије – димензије представљају пословне категорије које обезбеђују контекст нумеричким мерама. Димензијама ОЛАП-а је лакше навигирати него димензијама шеме звезде.
- Коцке – коцке комбинују све димензије и све мере у један концептуални модел.

OLAP димензије садрже следеће хијерархијске елементе:

- Димензије – организовани нивои и чланови у структури дрвета.
- Нивои – група чланова димензије који имају исто значење.
- Чланови – свака дискретна вредност у димензији.

Коцка је структура података која мере групише по нивоима и хијерархији сваке димензије коју желите да анализирате. Коцке комбинују више димензија, попут времена, географског положаја и производних линија, са сумарним подацима, попут података о продаји или стању залиха. Коцке у строгом математичком значењу и нису „коцке“, јер им странице не морају обавезно бити једнаке. Ради се, међутим, о прикладној метафори за један сложени концепт.

Мера је скуп вредности у коцки заснованих на колони у табели са чињеницама коцке, а које су најчешће нумеричке вредности. Мере представљају централне вредности у

коцки које су унапред обрађене, сакупљене и анализирани. У уобичајене примере спадају продаја, профит, приходи и трошкови.

Члан је ставка у хијерархији која представља једно појављивање података или више њих. Члан може да буде јединствен или нејединствен. На пример, године 2007. и 2008. представљају јединствене чланове на годишњем нивоу временске димензије, док јануар представља нејединствене чланове на месечном нивоу зато што у временској димензији може бити више јануара уколико она садржи податке за више година.

Израчунати члан је члан чија је вредност израчуната помоћу израза за време извршавања. Вредности израчунатог члана понекад се могу добити из вредности других чланова. На пример, вредност израчунатог члана „Профит“ може да се одреди одузимањем вредности члана „Трошкови“ од вредности члана „Продаја“.

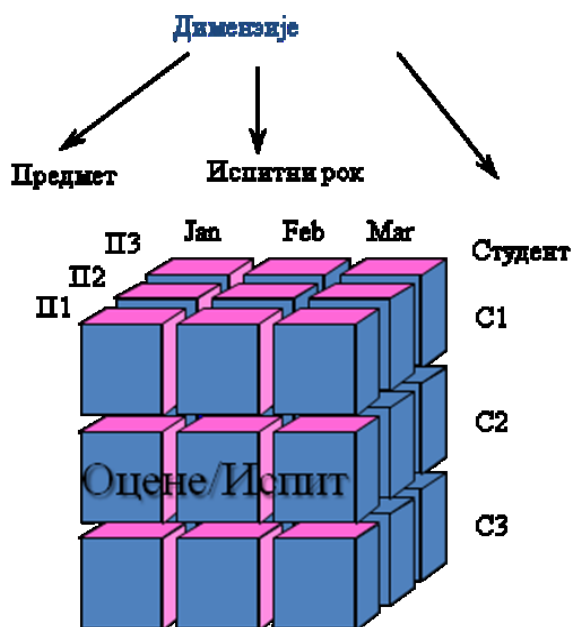
Димензија је скуп једне или више организованих нивоа хијерархије у коцки које корисник разуме и користи као базу за анализу података. На пример, географска димензија може да укључи нивое за земљу/област, државу/покрајину или град. Временска димензија може да укључи и хијерархију са нивоима за годину, квартал, месец и дан. У извештају изведене табеле или извештају изведеног графикана свака хијерархија постаје скуп поља која се може проширити и смањити да би открили ниже или више нивое података.

Хијерархија представља логичку структуру стабла у којој су организовани чланови неке димензије, и то тако да сваки члан има један надређени члан и ниједан подређени члан или неколико њих. Подређени члан налази се на суседном нижем нивоу у хијерархији који је директно повезан са тренутним чланом. На пример, у хијерархији „Време“ која садржи нивое квартал, месец и дан, јануар је подређени члан квартала¹. Надређена вредност је најчешће консолидација вредности свих подређених чланова. На пример, у хијерархији „Време“ која садржи нивое квартал, месец и дан, квартал¹ је надређен члану јануар.

Унутар неке хијерархије подаци могу да буду организовани у више и ниже **нивое** детаља, попут нивоа година, квартал, месец и дан у хијерархији „Време“.

Основне карактеристике које поседује сваки OLAP алат су: вишедимензионалност, *drill-down/up*, ротација као и више начина за преглед информација.

Вишедимензионалност је основна карактеристика OLAP алата, која омогућава вишедимензионалну анализу, тј. даје могућност кориснику да параметре пословања види и анализира у пресеку димензија које описују те параметре. Људима је природно да посматрају пословне појаве кроз димензије. Ако се појава прати у три димензије, реч је о коцки, а у више димензија о хиперкоцки. Свакој димензији коцке припада један параметар појаве. Свака тачка у коцки има тачно одређене вредности сваке посматране димензије. На следећој слици приказан је један пример вишедимензионалне OLAP коцке.



Слика 2.4 – OLAP коцка

Drill-down/up - представља аналитичку технику која омогућава кориснику да се креће по нивоима агрегације података, тј. од највишег нивоа до одговарајућих детаља података на најнижем нивоу. Ова функционалност је могућа за димензије које поседују хијерархију. У складишту података такође морају постојати одговарајући агрегати дефинисани за ту хијерархију, како би ова операција била брза и ефикасна.

Ротација је карактеристика OLAP алата, која омогућава поглед на податке из обрнуте перспективе. OLAP системи излаз најчешће приказују у пресеку величина представљених X и Y осом. Заменом величина на X и Y оси добија се ротирани приказ излазних информација.

Кад се говори о архитектури OLAP-а, тренутно постоје:

- вишедимензиони OLAP (MOLAP);
- релациони OLAP (ROLAP);
- хибридни OLAP (HOLAP).

MOLAP и ROLAP се разликују по начину физичког чувања података. Код MOLAP система подаци се чувају у вишедимензионој структури, а у случају ROLAP система подаци се чувају у релационим базама података.

MOLAP (*Multidimensional OLAP*) базе података имају ограничење физичке величине скупа података, са којима могу да раде, и ограничење на број димензија. Да би се вршила било каква анализа, потребно је прво учитати податке у вишедимензионе структуре, при чему се врше разни прорачуни како би се креирале агрегације и попунили подаци. Предност MOLAP система је што обезбеђује одличне перформансе система, нарочито када се ради са већ израчунатим подацима (агрегацијама). Недостатак MOLAP система је тешкоћа додавања нових димензија.

ROLAP (*Relation OLAP*) системи приступају подацима директно из складишта података и раде са релационим базама података. Ови системи могу да раде са великим скуповима података. Чим се одреди извор података, може се започети са анализом. Такође, код ROLAP система не постоје ограничења по питању броја димензија која постоје у случају MOLAP система.

Најбоље решење представља HOLAP (*Hybrid OLAP*), чији алати могу приступати и релационим и вишедимензионим базама података. Циљ коришћења HOLAP алата јесте да се искористе предности MOLAP алата (кратко време одзива система и аналитичке могућности) и ROLAP алата (динамички приступ подацима).

Битно је истаћи да HOLAP није прост збир MOLAP-а и ROLAP-а, већ је заправо ROLAP који има могућност извршавања врло сложених SQL наредби. Дакле, циљ је био да се задрже све предности ROLAP-а, али да се при томе додају и неке нове могућности за рад са вишедимензионим базама података.

2.3.3. Откривање знања у подацима – Data mining

Еволуцијом традиционалне економије у економију знања, информација је постала кључни ресурс, а крајњи корисник чинилац који намеће услове пословања. У оваквом окружењу где је сировина све мање важна, а све више се продаје интелигенција, знање постаје кључни фактор успешног пословања, а управљање знањем незаобилазна компонента савременог управљања.

Постоји више дефиниција ове технике. По једној *Data Mining* је нетривијална екстракција имплицитних, претходно непознатих и потенцијално корисних информација из ускладиштених података. Друга дефиниција говори да је *Data Mining* процес тражења веза и шаблона који постоје у великим базама података, и који су скривени међу великом количином података. *Data Mining* се може дефинисати и као процес подршке у одлучивању у ком се траже одређене правилности међу подацима у бази. [15, 18, 41, 52]

Подаци сами по себи не могу генерисати знање. Они представљају основну форму информације, којом је потребно управљати, проверавати, откривати и тумачити како би се дошло до знања скривених у њима. Откривањем облика, трендова и аномалија у великој количини ускладиштених података, представља један од највећих изазова информатичког доба.

Data Mining се данас користи у два различита домена: у домену предвиђања и домену откривања облика или законитости у подацима.

У домену предвиђања *Data Mining* алгоритми се користе у сврху предвиђања понашања посматраног ентитета, особе или објекта на бази задатих параметара и доступних података.

У домену откривања знања у подацима, *Data Mining* алгоритми омогућавају проналажење облика или изузетака и девијација у подацима који нису очигледни и одмах уочљиви.

Три основна „стуба“ *Data Mining*-а су: технике, подаци и моделовање. Технике *Data Mining*-а омогућавају идентификовање правила у великом броју података. Неке од техника-алгоритама *Data Mining*-а су:[18]

- Стабло одлучивања (*Decision Trees*) – метод за класификацију и предвиђање. Коришћењем серије питања и правила за категоризацију података, могу се предвидети да ће извесни типови података имати већи или мањи значај;
- Правила асоцијације (*Association Rules*) – овај алгоритам помаже у идентификовању релација између различитих елемената. Он групише по сличности, односно користи се за проналажење групе података који се најчешће заједно налазе у једној трансакцији;
- *Naïve Bayes* – овај алгоритам се заснива на Bayes-овој теореме са претпоставкама о независности променљивих код различитих елемената података;
- Секвенционално откривање образаца (*Sequential Pattern Discovery*) ова техника је слична техници асоцијације, с тим што повезује догађаје временски и одређује у каквој су вези;
- Класификација (*Classification*) – посматра понашања и атрибуте предетерминисаних група;
- Кластеринг (*Clustering*) – техника кластеринга омогућава груписање записа података који су слични на основу секвенци претходних догађаја;
- Предвиђање (*Forecasting*) – техника предвиђања се посматра кроз два аспекта:
 - регресивна анализа (*Regression Analysis*) користи познате вредности података за предвиђање будућих вредности или будућих догађаја који се заснивају на историјским трендовима и статистици;
 - откривање временске секвенце (*Time Sequence Discovery*) се разликује од регресионе анализе у томе што предвиђа само временски зависне вредности података;
- Неуронске мреже (*Neural Nets*) – као што човек учи на основу искуства тако може и рачунар. Неуронске мреже моделују неуронске везе у људском мозгу и на тај начин симулирају учење;
- Текст *Mining* – овај алгоритам анализира неструктуриране текстуалне податке.

Већина *Data Mining* техника-алгоритама захтева велику количину података како би се изградио модел којим би се успешно вршила класификација, процењивање, предвиђање и слично. Подаци се данас складиште на начин који оптимизује перформансе задатка због којих се складишти. *Data Mining* захтева “чишћење” тих како би се добили подаци јединственог формата. *Data Mining* алгоритми захтевају вишеструку обраду над великом количном података, што је незамисливо извршити ручно. Прилог томе иде чињеница да су рачунари добрих перформанси данас више него приступачни.

Први *Data Mining* програми су настали у владиним и универзитетским лабораторијама. Убрзо је почела њихова комерцијална употреба. Захваљујући бар кодовима и другим врстама скенера које аутоматски прикупљају и памте рачуне купаца, и супермаркети постају информациони “брокери”. Кад год купац обави куповину, база се ажурира новим подацима. У интересу супермаркета је да се из тих података генеришу одређена правила закључивања везана за то: који производи се купују заједно, како их аранжирати на рафовима. Такве информације прикупљене од крајњих потрошача су од велико значаја и за произвођаче. [18, 41, 52]

Data Mining је дакле значајна компонента при анализи података. Добро пројектована база података садржи историјске записе о трансакцијама. Примена *Data Mining* алата на ове податке може омогућити доношење квалитетнијих одлука у будућности. *Data Mining* је процес откривања шаблона и правила закључивања из великог скупа података. Најчешћи задаци су класификација, предвиђање, груписане по сличности, одређивање кластера и профила.

3. Методологије и стандарди развоја информационих система

Методологија развоја информационих система би требала да подразумева физичку имплементацију животног циклуса развоја који укључује[3,18,45]:

- активности за сваку фазу,
- индивидуална и групна правила за сваку активност,
- стандарде квалитета за сваку активност и
- алате и технике које треба да буду употребљене за сваку активност.

Методологија би требала да обезбеди конзистентан и репродуктиван приступ који може бити примењен на све пројекте, у исто време смањујући ризик од грешака.[18] Том приликом се ствара комплетна и конзистентна документација за тренутне и будуће пројекте. Имајући у виду чињеницу да се пројектни тимови мењају, методологија омогућава да они тимови који настављају рад својих претходника, то ураде брзо и лако.

Проблем у описивању рада пословног система је што тешко могу да се користе природни језици, због двосмислености. С друге стране, прецизан опис преко формалних језика је неразумљив за већину људи. Из тог разлога потребна је техника која ће организовати природне језике на тај начин да се елиминише двосмисленост и омогући ефикасна комуникација и разумевање. Показало се да је поступак моделовања и једна од најефективнијих техника за разумевање и једнозначну комуникацију између пројектаната и корисника.[16]

Суштински, моделовање је пројектовање система пре имплементације. Пословни проблеми су често веома комплексни што може да омета комуникацију између развојног тима и корисника. Моделовање омогућава да се сложени проблеми разјасне и самим тим олакша њихово разумевање.

Моделовањем се елиминишу детаљи, чиме се умањује комплексност система који се анализира. У процесу моделовања се користе графички симболи (нпр. геометријске фигуре) да би се омогућио сликовити приказ (слика вреди више од хиљаду речи).

Поред графичког приказа потребно је дати и прецизне дефиниције које се појављују у моделу и одговарајући текст који има улогу средства комуникације.

Овакав приступ захтева одређени ниво апстракције, којом се врши контролисано искључивање детаља, односно извлаче се заједничке карактеристике у описивању неког система. Тако је на вишим нивоима апстракције систем описан јасније, а на нижим детаљније.

Стање система се моделује како би се идентификовали[3,18,45]:

- захтеви;
- проблеми и ризици;
- информације које недостају.

Моделовање је присутно у већини инжењерских дисциплина, а сваку од њих карактерише и постојање специфичних језика за моделовање. У области развоја софтвера, моделовање се изводи класичним приступом (израда модела података и модела процеса) или објектно - оријентисаним приступом (објектни модели).[16]

За поступак моделирања развијени су посебни алати, као што су на пример VPwin за функционално моделовање (подржава IDEF0 standard), затим ERwin за моделовање података (подржава IDEF1X стандард), *Rational Rose* који подржава UML, односно објектно моделовање, и многи други. Ови алати, стандарди и технике омогућавају:

- анализу и пројектовање информационих система;
- израду документације;
- бољу комуникацију унутар тима;
- управљање великим и сложеним пројектима.

Основни разлог за појаву методологија развоја софтверских система, самим тим и информационих система, је велики број неуспешних пројеката. Наиме развој нових софтвера се убрзао почетком 70-их година прошлог века, пре свега за војне примене. Тад се софтвер развијао без јасних смерница-корака развоја, по принципу “само нека ради“. Разлог томе је био и у чињеници да су софтвери који су развијани били

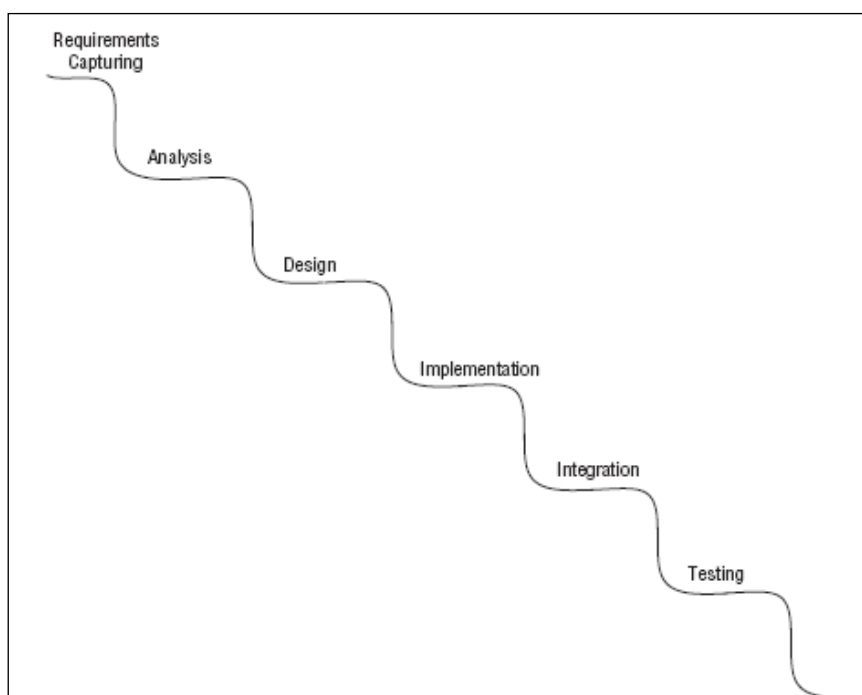
релативно једноставни. Иако је овакав приступ довео до неуспеха великог броја пројеката, на жалост и данас имамо примере оваквог развоја по систему “кодирај па поправљај“. [3]

Данас, постоји више методологија развој софтвера, односно информационих система. Већину тих методологија можемо класификовати на следећи начин[3,46]:

- Прескриптивне:
 - o модел водопада (сукцесивне активности, строго дефинисани улази и излази, све активности морају бити реализоване),
- Инкремантале:
 - o *Rapid Application Development* модел (модификација модела водопада у више мањих појединачних циклуса, рад у више тимова, развојни циклус 60-90 дана, развој “у једном даху“),
- Развојно-еволутивне:
 - o прототипски развој (почиње се са иницијалним развојем прототипа, корисник врло брзо може да “види“ изглед будућег система, дизајн и спецификација се раде на крају),
 - o спирални модел (развој у више циклуса, сваки циклус се реализује као модел водопада, свака наредна итерација-циклус додаје нову функционалност),
- *Unified Process* (Итеративни и инкрементални развој, користи најбоље особине већине методологија, тимски рад),
- Агилни развој (више итерација, мањи ризик смањењем времена за развој, свака итерација мини пројекат за себе, интерактиван рад корисника и инжењера, тимски рад):
 - o *Extreme Programming* (XP),
 - o *Adaptive Software Development* (ASD),
 - o *Dynamic Systems Development Method* (DSDM),
 - o *Scrum*,
 - o *Crystal*,
 - o *Feature Driven Develop*,
 - o *Agile Modeling* (AM).

Раније је доминантно место имао такозвани модел водопада (слика 3.1). У тој методологији развоја софтвера, основне активности за развој софтвера (то јест, анализа, пројектовање, имплементација, тестирања и имплементације) пратиле су једна другу на строго секвенцијалан начин. То значи да следећа активност не почиње док се претходна не заврши. [16, 17]

Дакле, модел водопада обухвата фронтални развој софтвера. Прво, су сви ресурси (особља, време и пажња) усмерени на анализу. Када су сви захтеви у потпуности завршени (за цео систем), почиње фаза дизајна. Када је систем у потпуности дизајниран, до најситнијих детаља, креће фаза имплементације. Након што је систем имплементиран врши се његово тестирање у целини. Коначно, када је систем тестиран интегрално, може се сматрати да је уведен у употребу.[46]

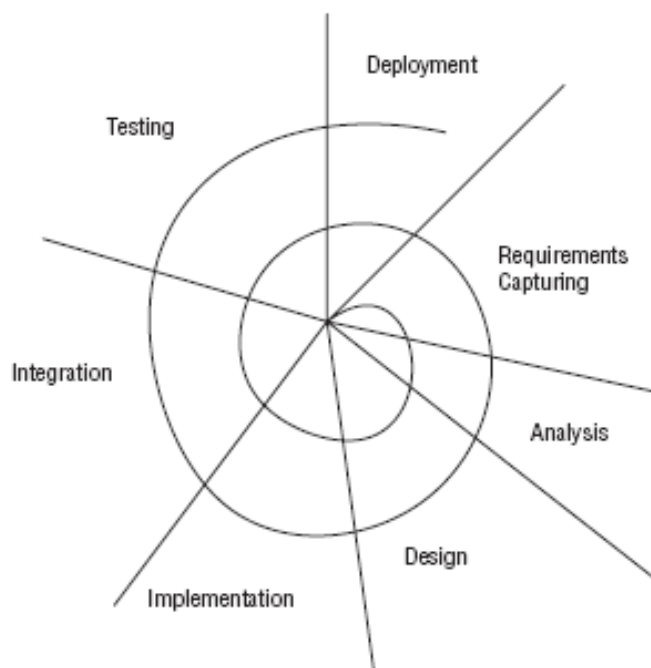


Слика 3.1 – Модел водопада [16]

Испоставило се да је ова методологија неефикасна и ризична, јер је описани фронтални приступ развоју софтвера често водио до дезинтеграције софтвера. Наиме, комплексни софтверски системи су развијани у деловима, на пример, различити људи у различитим условима развијају различите делове. У моделу водопада, ови делови су интегрисани тек по завршетку. [46]

Међутим, ова интеграција носи веома велики ризик, јер је бројне делове сложеног система врло тешко одмах уклопити. Оваква ситуација је често доводила до пада пројекта и, што је још горе, то се дешава у завршним фазама пројекта, када је већина система већ имплементирана. Другим речима, грешке се откривају касно, када је већ много уложено у имплементацију. Дакле, уочавање грешака је веома скупо, јер то понекад захтева поновно покретање целог процеса од самог почетка, што може да потраје у недоглед. Наравно, за врло једноставне системе, овај модел је изгледа савршено могућ, али је врло често нереалан и осуђен на неуспех, имајући у виду сложеност постојећих система.[46]

Због наведених разлога модел водопада је замењен новијим, “агилнијим“ и мање ризичним моделом-методологијом, који се зове спирални модел, приказан на слици 3.2. [16]. У овом моделу, све основне активности за развој софтвера (анализа, пројектовање, имплементација, тестирање и увођење) се итеративно понављају у циклусима, при чему сваки циклус примењује све ове активности на мањи део система вршећи доградњу функционалности.



Слика 3.2 – Спирални модел [16]

На овај начин, систем расте корак по корак из циклуса у циклус. Овај модел такође омогућава да се у пројекту идентификују и реше евентуални проблеми што је пре могуће. Поред тога, повећана је и флексибилност тако да је лакше изаћи у сусрет новим

захтевима или одлуци о потреби за променом дизајна. Методологија која је настала на спиралном моделу развоја позната је под именом *Rational Unified Process* (RUP).

У даљем тексту, биће више речи о методологијама и стандардима који су директно утицали на развој методологије развоја BI система у овом раду.

3.1. Функционално и информационо моделовање

Историјски гледано током шездесетих и седамдесетих година Douglas T. Ross развио је технике моделовања познате као SADT (*Structured Analysis & Design Technique*). Прихватајући SADT технику авијација САД је развила SADT као део ICAM (*Integrated Computer Aided Manufacturing*) програма током касних 70-тих, које су добиле назив технике IDEF (*Integation DEFinition*). Циљ ICAM програма је био да се побољша производна продуктивност применом рачунара. Учесници у изградњи ICAM програма су увидели све предности коришћења IDEF технике, јер текстуални опис не представља ефикасан начин за документовање процеса и података. [5,20]

У раним 90-тим, IDEF Users Group, у кооперацији са NIST-ом (*National Institutes for Standards and Technology*), отпочео је рад на IDEF0 стандарду за моделовање процеса и IDEF1X(*eXtend*) као техници за информационо моделовање (семантичко моделовање података), публикујући их 1993. године. Ови стандарди су под покровитељством IEEE (*Institute of Electrical and Electronics Engineers*) приквашени и као ISO стандарди. [5,20]

Један од циљева моделовања је и да се развију технологије које ће омогућити логичку и физичку интеграцију мрежа хардверски и софтверских веома различитих конфигурација. Управо техника IDEF моделовања то омогућава.

3.1.1. Стандард за функционално моделовање IDEF0

IDEF0 [5,20,22] је стандард који дефинише језик моделовања, одговарајућа правила и технике за развој графичке презентације система или организације. Употреба овог стандарда омогућава конструкцију модела, који се састоји од системских функција (активности, процеси операције), функционалних веза и података (информација или објеката). Намена оваквих модела је да подрже интеграцију система. Овим стандардом

се тако омогућује техника моделовања која је независна од CASE (*Computer-Aided Software Engineering*) метода и алата који се могу наћи на тржишту.

Техника моделовања зацртана овим стандардом има следеће карактеристике: [5,20,22]

- Генеричност (за анализу система са различитом наменом и сложености)
- Ригорозност и прецизност (за производњу коректног, употребљивог модела)
- Концизност (да омогући разумевање, комуникацију и вредновање)
- Концептуалност (за представљање функционалних захтева а не физичку организациону имплементацију)
- Флексибилност (да подржи више фаза у животном веку пројекта).

Предлаже се коришћење ове методе за пројекте који:

- Захтевају технику моделовања за анализу, развој, реинжењеринг, интеграцију и аквизицију информационих система,
- Уграђују технике моделовања у анализу процеса пословања или методологију софтверског инжењеринга.

Стандард се састоји из нормативног и информативног дела. Обавезно је поштован нормативног дела, док информативни део треба посматрати као сугестије које је пожељно поштовати.

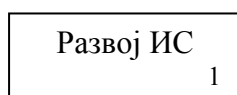
IDEF0 може да се користи за моделовање широког спектра система. Код нових система, ова техника може да се користи за дефинисање захтева и задавање функција, а затим да се пројектује решење које ће задовољити захтеве и изводити функције. За постојеће системе, IDEF0 може да се користи за анализу функција које систем изводи и за снимање механизма (начина) на које се оне извршавају.

Резултат примене IDEF0 технике на неки систем је модел који се састоји од хијерархијске серије дијаграма, текста и речника који су међусобно повезани. Две основне компоненте моделовања су функције (представљене правоугаоницима) и подаци или објекти који повезују те функције (представљени стрелицама).

IDEF0 је техника моделовања заснована на комбинацији графике и текста приказаних на један организациони и систематски начин да појачају разумљивост, подрже анализу, обезбеде логику за потенцијалне промене, специфицирају захтеве, или подрже ниво пројектовања система и активности интеграције. IDEF0 модел се састоји из хијерархијског низа дијаграма који степенасто приказују нивое детаљно описаних функција и њихових интерфејса унутар система. Постоји три типа дијаграма: графички, текстуални и речник. Графички дијаграми дефинишу функције и функционалне односе између правоугаоника и синтаксе и семантике стрелица. Текстуални дијаграми и речник обезбеђују збирне информације као подршку графичком дијаграму.

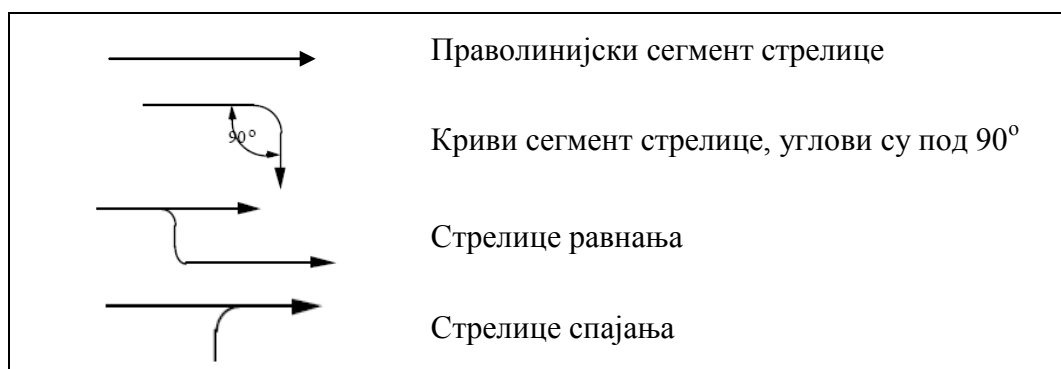
IDEF0 је инжењерска техника за реализацију и управљање потребама анализе, предностима анализе, дефиницијом захтева, функционалном анализом, пројектовањем система, руковођењем, и као таква је основа за континуирано побољшање самог система. IDEF0 модели обезбеђују “скицирање“ функција и њихових интерфејса које морају бити да омогућавају доношење системских инжењерских одлука тако да буду логичке, прихватљиве, интегрбилне и успешне. IDEF0 модели одсликавају како се системске функције међусобно односе и раде као што нацрт-скица производа рефлектује како се различити делови производа слажу заједно.

Структурне компоненте и елементи језика и правила која дефинишу односе између њих се означавају као језичка синтакса. Компоненте IDEF0 синтаксе су правоугаоници и стрелице, правила и дијаграми. Правоугаоници представљају функције, дефинисане као активности, процеси или трансформације. Стрелице представљају податке или објекте повезане са функцијама. Правила дефинишу како се користе компоненте, а дијаграми обезбеђују формат за посматрани модел речима и графички. Формат такође обезбеђује основе менаџмента конфигурације модела. Правоугаоник обезбеђује опис онога што се дешава у пројектној функцији. Типични правоугаоник је приказан на слици 3.3. Сваки правоугаоник ће имати име и број унутар граница правоугаоника. Име ће бити активни глагол или фраза која описује функцију. Сваки правоугаоник на дијаграму ће садржати број у доњем десном углу. Бројеви правоугаоника се користе да би препознали субјект у датом тексту.



Слика 3.3 – Синтакса правоугаоника

Стрелица се састоји из једног или више сегмената линије, са главом стрелице на једном крају. Као што је приказано на слици 3.4., сегменти стрелице могу бити прави или криви (са луковима под 90° повезаним хоризонталним и вертикалним деловима), и могу имати разгранату конфигурацију (рачвање или спајање). Стрелице не представљају ток или секвенцу као у традиционалном моделу тока процеса. Стрелице садрже податке или објекте који се односе на функције које се реализују. Функције које примају податке или објекте су ограничене на доступне податке или објекте.



Слика 3.4 – Синтакса стрелице

Семантика представља значење синтаксних компоненти језика и изражава тачност интерпретације. Интерпретација указује на појединости као што су нотација правоугаоника и стрелица и функционални однос интерфејса.

Како IDEF0 подржава моделовање функција, име правоугаоника ће бити глагол или глаголска фраза, као нпр. „развој“, која даје опис функције која правоугаоник представља. Пример функције „Развој“ трансформише неконтролисана у контролисана делове. Након давања имена правоугаонику врши се инкорпорација стрелица (уређење оријентације по странама правоугаоника).

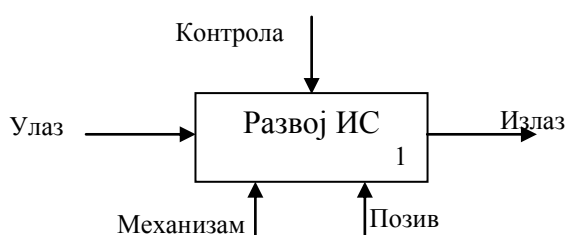
Стандардна терминологија се користи ради осигурања егзактне комуникације. Значења правоугаоника се описно именују глаголима или глаголским фразама и групишу и раздвајају у декомпонованим дијаграмима. Значења стрелица се групишу и раздвајају у дијаграмима а сегменти стрелица се означавају лабелама које су именице или именичне фразе. Лабеле сегмената стрелица су прописане и ограничавају значење својих сегмената само на примену одговарајућих података и објеката које сегмент стрелице графички приказује. Значења стрелица се даље изражава преко синтакси рачве и споја.

Свака страна правоугаоника има стандардно значење у односима правоугаоник-стрелица. Страна правоугаоника на коју се везује стрелица одређује улогу стрелице. Стрелице које улазе са леве стране правоугаоника су Улази.

Улази се трансформишу или троше функцијом у циљу добијања излаза. Стрелице које улазе са горње стране су Управљачке стрелице (Контроле). Управљачке стрелице специфицирају захтеване услове функције за добијање одговарајућег излаза. Стрелице које излазе са десне стране су Излази. Излази су подаци или објекти добијени функцијом.

Стрелице повезане са доње стране правоугаоника приказују Механизме. Стрелица нагоре приказује она средства која подржавају извршавање функције. Друга средства могу бити наслеђена од родитеља правоугаоника. Стрелице механизма усмерене надоле су стрелице позива. Стрелице позива омогућавају заједничко коришћење између модела или између делова истог модела. Позивни правоугаоник обезбеђује детаљ за правоугаоник који позива. Стандардне позиције стрелица су приказане на слици 3.5.

Информације подршке које се тичу функције и њеног циља биће назначене у тексту који се односи на дијаграм. Дијаграм може и не мора имати придружен текст. Када се користе акроними, скраћенице, кључне речи или фразе, потпуно дефинисани термини се дају у речнику.

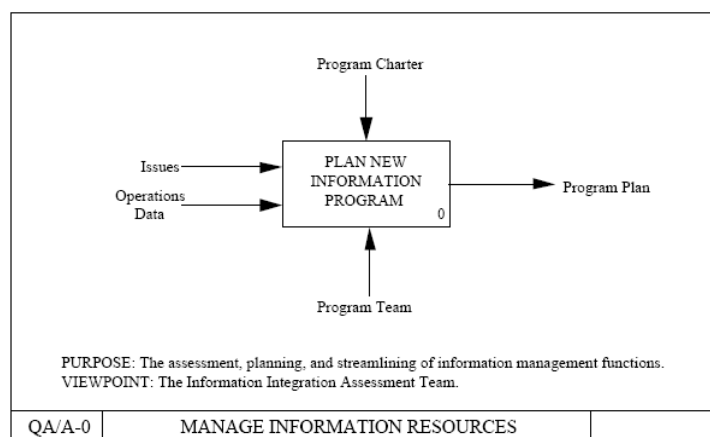


Слика 3.5 – Позиције и улога стрелица[5]

Правоугаоници описују функције које показују шта мора бити испуњено. Име функције ће бити активни глагол или глаголска фраза. Стрелице идентификују податке или објекте потребне функцији или пак, произведене функцијом. Свака стрелица треба бити означена лабелом са именицом или именичном фразом.

Као што је раније речено, IDEF0 модели су састављени од три типа информација: графички дијаграм, текст и речник. Ови типови имају један узајамни однос. Графички дијаграм је главна компонента IDEF0 модела, садржи правоугаонике, стрелице, интерне везе правоугаоник/стрелица и одговарајуће односе. Правоугаоници представљају сваку главну функцију субјекта. Ове функције су изломљене или декомпоноване на више детаљних дијаграма, тако да је субјект описан на неопходном нивоу да подржи циљеве одговарајућег пројекта. Дијаграм највишег нивоа обезбеђује најопштији и најапстрактнији опис субјекта приказаног моделом. Овај дијаграм је праћен низом деце дијаграма који пружају више детаља о субјекту.

Сваки модел имаће контекст дијаграм највишег нивоа, у коме се субјект модела описује као један правоугаоник са својим ограниченим стрелицама. Овај дијаграм се зове А-0 дијаграм. Стрелице на овом дијаграму су повезане са околином субјекта. Како овај један правоугаоник представља цео субјект име правоугаоника је опште. Ово важи и за интерфејсе стрелица које такође представљају комплетан скуп спољних интерфејса субјекта. А-0 дијаграм такође подешава подручје модела или границе и оријентацију. Пример А-0 дијаграма је дат на слици 3.6. [5]

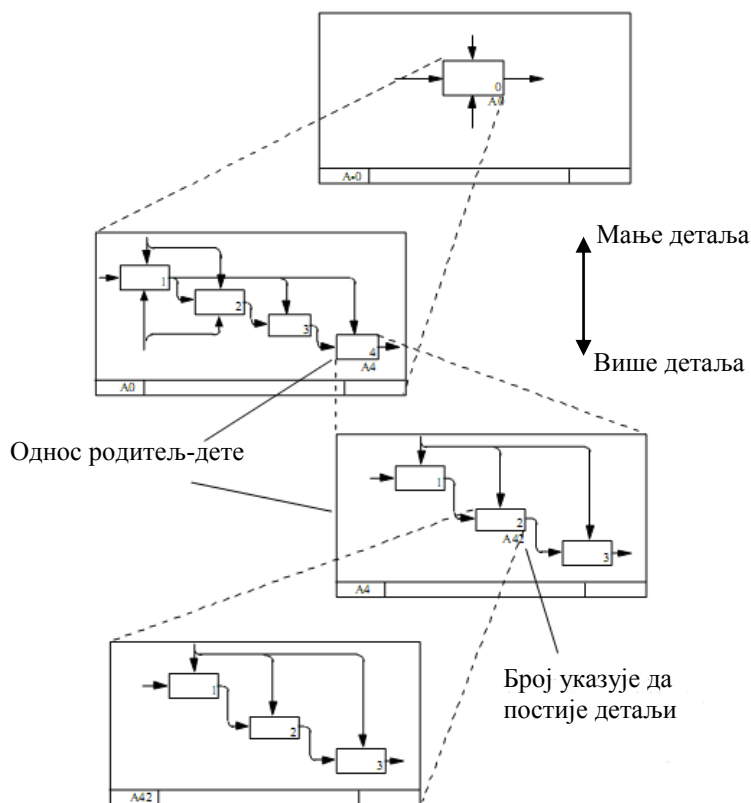


Слика 3.6 – Дијаграм највишег нивоа [5]

А-0 контекст дијаграм такође приказује кратке изјаве о тачки гледишта и циљу модела, што помаже вођењу и ограничавању креирања модела. Тачка гледишта одређује шта се може видети унутар контекста модела, и из које перспективе и угла. У зависности од аудиторјума могу бити усвојене различите тачке гледишта што говори о различитим аспектима субјекта. Ствари важне са једне тачке гледишта могу чак и да се не појаве у моделу приказаном са друге тачке гледишта истог субјекта.

Изјава о циљу изражава разлог зашто се модел израђује и стварно одређује структура модела. Најважнији елемент долази први у хијерархији, као целокупна функција највишег нивоа декомпонована у подфункционалне делове, који се даље декомпонуюју све док се сви релевантни детаљи целе тачке гледишта адекватно не изложе. Свака подфункција се моделира посебним правоугаоником, са родитељ правоугаоникима детаљно описаним преко деце дијаграма на следећем нижем нивоу. Сви деца дијаграми морају бити унутар подручја контекст дијаграма највишег нивоа.

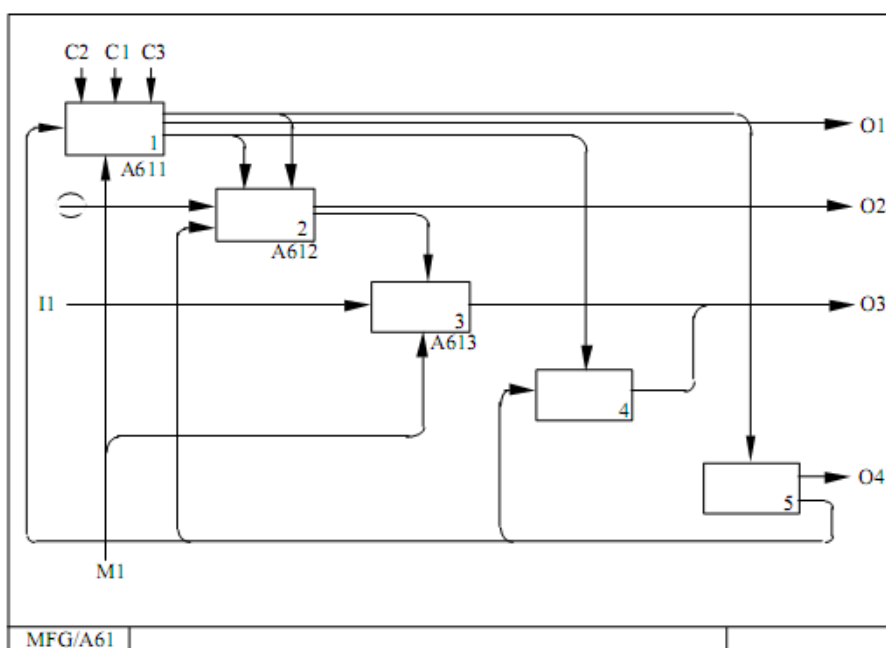
Свака појединачна функција приказана у дијаграму највишег нивоа се може декомпоновати на своје главне подфункције стварањем деце дијаграма. Редом, свака од ових подфункција се може декомпоновати стварајући други, нижег нивоа дете дијаграм. На датом дијаграму неке од функција, ниједна од функција или све функције се могу декомпоновати. Сваки дете дијаграм садржи децу правоугаонике и стрелице што обезбеђује додатне детаље о родитељ правоугаонику. Дете дијаграм које проистиче из декомпозиције функције обухвата исто подручје као и родитељ правоугаоник кога описује. Тако, дете дијаграм се може посматрати као унутрашњост родитељ правоугаоника. Ова структура је показана на слици 3.7. [22]



Слика 3.7 – Декомпозициона структура [22]

Родитељ дијаграм садржи један или више родитељ правоугаоника. Сваки обичан (не контекст) дијаграм је уједно дете дијаграм док по дефиницији детаљно описује родитељ правоугаоник. Тако дијаграм може бити родитељ дијаграм и дете дијаграм. Слично, правоугаоник може бити и правоугаоник родитељ (детаљно дат дете дијаграмом) и дете правоугаоник (појављује се на дете дијаграму). Примарни однос у хијерархији је између родитељ правоугаоника и дете дијаграма који га описује. Чињеница да је дете правоугаоник детаљно описано, и зато је такође родитељ правоугаоник, се исказује преко израза за референцирање детаља (DRE). DRE је кратак код написан испод доњег десног угла родитељ правоугаоника који указује на његов дете дијаграм. DRE ће попримити један од следећих облика:

- Хронолошки створен број назван "С-број" који јединствено идентификује посебну верзију дете дијаграма.
- Број стране дете дијаграма у публикованом документу у коме се појављује модел.
- Број чвора који се односи на дете дијаграм. Ако постоје вишеструке верзије дете дијаграма, посебна верзија се не може специфицирати.
- Број напомене модела чији текст одређује услове за избор посебне верзије дете дијаграма.



Слика 3.8 – Референцирање детаља[20]

Слика 3.8 илуструје коришћење чворних бројева као DRE. На слици, присуство DRE испод правоугаоника 1,2,3 указује да су они детаљно описани одређеним дета дијаграмима.

Једна од најважнијих карактеристика IDEF0 као концепта моделовања је да он степенасто уводи веће и веће нивое детаља кроз структуру дијаграма гранајући модел. На овај начин, остварена је комуникација која омогућава читаоцу да са добро ограниченим темама учи са сваког дијаграма.

IDEF0 модел почиње представљањем целог субјекта као једне јединице, правоугаоника са спољним стрелицама у ограниченим условима повезаним са функцијом и ресурсима ван субјекта. Овај правоугаоник се назива главни правоугаоник модела. (Има чворни број А0). Док главни правоугаоник IDEF0 модела представља субјект као целину, описно име у правоугаонику је уопштено. Исто важи за спољне стрелице модела, које представљају комплетан скуп спољних граничних услова субјекта као целине, укључујући приступ механизмима који носе све перформансе.

Дијаграм у коме се појављује А0 главни правоугаоник представља контекст модела и назива се контекст дијаграм. Минимални контекст за модел је специјални контекст дијаграм са чворним бројем А-0. А-0 контекст дијаграм има само један правоугаоник А0 главни правоугаоник, са означеним стрелицама, и такође текстуалне дефиниције Тачке гледишта и Циља модела. (А-0 дијаграм нема ICOM кодове или заграде на неповезаним крајевима стрелица).

Понекад, међутим, у циљу обезбеђивања комплетнијег приказа средине контекста модела, опциони А-1 контекст дијаграм је такође представљен. У А-1 контекст дијаграму, А-0 правоугаоник је један од три до шест правоугаоника, па се као резултат добија потпун дијаграм родитељ (са три до шест правоугаоника) за главни правоугаоник модела где чворови А1 до А6 остају деца прве генерације. У случају где се користи А-1 контекст дијаграм, и даље постоји А-0 контекст дијаграм. Овај А-0 дијаграм још увек има један А0 главни правоугаоник, са означеним спољним стрелицама и текстуалним дефиницијама Тачке гледишта и Циља модела.

Контекст дијаграми су дијаграми који имају чворне бројеве облика "A-n" (са минус знаком), где је n веће или једнако нули. Обично, неконтекст дијаграми немају минус знак у њиховим чворним бројевима. Број главног правоугаоника модела је увек нула. Број правоугаоника 0 треба да се појави на задатом A-0 контекст дијаграму модела и такође на опционом A-1 контекст дијаграму где заузима место једног од правоугаоника (1 до 6) A-1 родитељ дијаграма. Тако је A-0 увек број чвора родитељ правоугаоника и дете дијаграм за цео модел и увек се детаљно описује правоугаоницима са бројевима чворова A-1, A-2, A-3, до највише A-6.

Са само једним правоугаоником, A-0 је прави контекст дијаграм али није прави родитељ дијаграм. Прави дијаграми имају три до шест правоугаоника. Родитељски контекст је онај који обезбеђује или именује контекст за дијаграм на месту правог родитељ дијаграма. Родитељски контекст A-0 дијаграма је захтевани A-0, ако не постоји A-1 контекст дијаграм. Ако постоји A-1 контекст дијаграм, A-1 је прави родитељ A0 дијаграма. Родитељски контекст A-0 контекст дијаграма је увек "највише (на врху)".

На крају можемо закључити да када се користи на систематски начин, IDEF0 омогућује: [5,22]

- Извршење функционалне декомпозиције и дизајна на свим нивоима, за систем састављен од људи, машина, материјала, рачунара и информација;
- Стварање документације паралелно са извођењем ревизије стандарда ISO 9000:2000 као и са реинжењерингом пословних процеса;
- Бољу комуникацију између аналитичара, пројектног тима, корисника и менаџера;
- Омогућује дискусију тимова да би се постигло међусобно разумевање;
- Омогућује управља великим и сложеним пројектима;
- Обезбеђује елементе потребне за информационо моделовање (IDEF1X).

Софтверска реализација IDEF0 стандарда, који се користи у овом раду, је BPwin(*Bussines Process Windows*) фирме LogicWorks.

3.1.2. Стандард за информационо моделовање IDEF1X

Други стандард који је IDEF *User Group* дефинисала је IDEF1X техника за информационо моделовање. Информационо моделовање представља апстрактно виђење реалног система тј. то је поједностављено представљање реалног система преко скупа објеката(ентитета), веза између објеката и атрибута објеката. Информационо моделовање је појам који је дефинисан у оквиру IDEF1X методологије и дефинише одговарајући модел података. [5,20,22]

IDEF1X је семантички богат моделар података треће генерације који је реализован у оквиру софтвера ERwin CASE алата. ERwin(*Entity Relationships for Windows*) CASE алат омогућује генерисање у неки од система за управљање базама података (СУБП). Мора се нагласити појам генерисање а не програмирање јер ERwin омогућује да се директно креирају табеле, везе, атрибути и сва ограничења која су се некада програмирала.

Поступак израде IDEF1X модела садржи следеће кораке: [5,20,22]

- дефинисање независних ентитета;
- дефинисање зависних ентитета;
- дефинисање веза.

Независни ентитет је објекат који има бар једну особину која га може једнозначно идентификовати, што значи да независни ентитети имају властиту идентификацију (не зависе од других ентитета). Графички се независни ентитети приказују правоугаоницом у који се уписује назив типа ентитета. [5,20,22]

Зависни ентитети су ентитети чија идентификација и егзистенција зависе од другог или других ентитета.

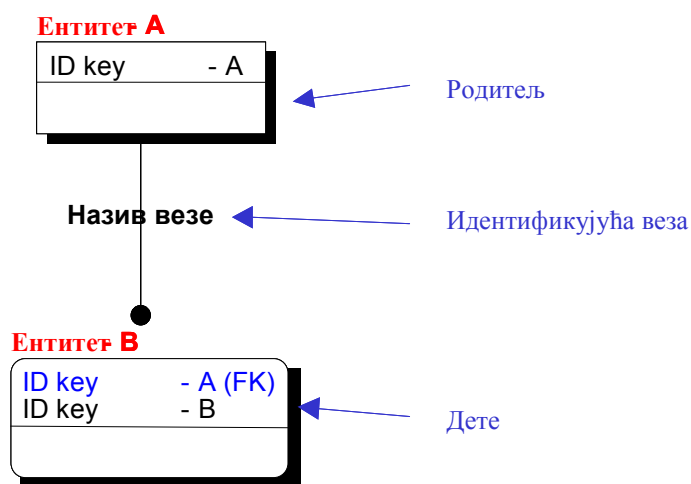
Веза је асоцијација између два или више ентитета, односно веза представља однос који постоји између објектата, у реалности или апстрактно. Ентитет од кога веза полази зове се “родитељ”, а ентитет ка коме је веза успостављена назива се “дете”. [5,20,22]

Постоје следећи типови веза: [5,20,22]

- идентификујућа, која ентитет дете идентификује са ентитетом родитељ;
- неидентификујућа, не идентификује дете преко идентификатора родитеља и
- веза категорије тј. везе према подтипovima.

Идентификујућа веза има овај назив зато што су кључеви ентитета “родитеља“ део идентитета ентитета “дете“, односно да је ентитет “дете“ завистан од ентитета “родитеља“ преко идентификатора. Имајући у виду наведено следи да ако се инстанца ентитета “дете“ идентификује преко везе са ентитетом “родитељ“, онда се веза дефинише као идентификујућа веза и сваки примерак ентитета “дете“ је повезан са најмање једном инстанцом ентитета “родитељ“. [5,20,22]

Идентификујућа веза је приказана пуном линијом и повезује ентитет родитеља са ентитетом дете са тачком на страни ентитета дете (слика 3.9).



Слика 3.9 – Идентификујућа веза[5, 20]

Код идентификујуће вези ентитет “родитељ“ има свој независни примарни кључ (ID key -A), а ентитет “дете“ има сложени кључ који се састоји од сопственог кључа (ID key -B) и пренесеног кључа родитеља (ID key -A(FK)). Дакле, инстанце ентитета “родитељ“ су независне, а инстанце “ентитета“ дете се не могу идентификовати без идентификатора ентитета “родитељ“. [5,20,22]

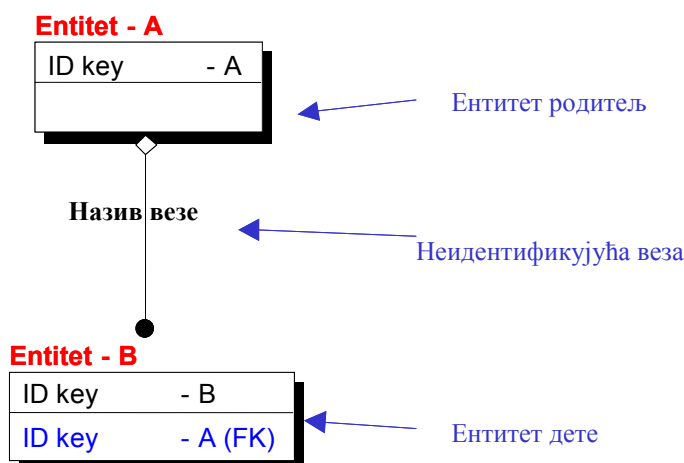
Ако се свака инстанца ентитета “дете“ може јединствено идентификовати без знања везе са инстанцом ентитета “родитељ“, онда се таква веза дефинише као **неидентификујућа веза**.

Неидентификујуће везе су приказане испрекиданом линијом која повезују ентитет “родитељ“ и ентитет “дете“, са већом црном тачком на страни ентитета дете.

Неидентификујућа или такозвана слаба веза зависи од начина дефинисања кључева од “родитеља“ ка “детету“. Могуће су две варијанте:

- обавезна неидентификујућа веза и
- необавезна (опциона) неидентификујућа веза.

Ако је веза обавезна, гледано од “родитеља“, онда постојање инстанце “дете“ зависи од постојања инстанце “родитељ“ (слика 3.10).



Слика 3.10 – Неидентификујућа веза [5, 20]

Случај егзистенцијалне и идентификационе зависности у CASE алату *ERwin* се обележава симболом ромба или дијаманта. Ромб може постојати само у славим везама (пошто је јака веза у оквиру примарног кључа који не може да има *NULL* вредност).

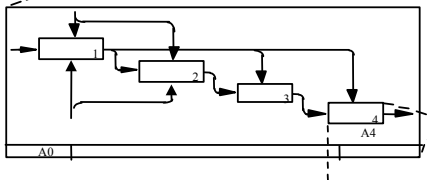
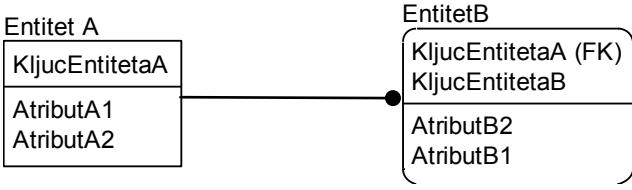
Веза категорије је дефинисана за хијерархијску везу између надређеног генеричког ентитета који садржи заједничке особине подређених ентитета категорије. Овај тип везе се дели на: [5,20,22]

- комплетни сет категорије или потпуне структуре кад је затворен скуп ентитета категорије и
- некомплетан сет категорије или непотпуну структуру када није затворен скуп ентитета категорије.

3.1.3. Веза стандарда IDEF0 и IDEF1X

С обзиром да модел процеса приказује детаље неког процеса, а модел података објекте за реализацију тог процеса, може се успоставити веза између тих модела. [22]

Табела 3.1. Веза између модела процеса и модела података [22]

Модел процеса	Модел података
	
<p>Правоугаоници представљају активности, дефинисане као функције, процеси и трансформације.</p>	<p>Именовано подручје представља скуп ентитета потребних да се обави наведена активност. Релациона линија између ентитета представља везу између два ентитета и описује однос између ентитета за обављање активности.</p>
<p>Стрелице представљају носиоце информација за обављање активности.</p>	<p>Ентитет је сваки конкретан или апстрактан објекат који постоји, постојао је, или може постојати, укључујући и везе између таквих објеката.</p>
<p>Подаци који се преносе стрелицама до активности немају посебно обележје, али се подразумева да су садржане у стрелицама.</p>	<p>Атрибут је именована особина ентитета.</p>

3.2. Објектно-оријентисано моделовање

Објектна парадигма је резултат еволуције софтверске технологије, а не револуције. Она се грана на концепте и принципе из процедуралних и других парадигми које су признате као успешне у софтверској пракси изградње. Ти концепти су признати и коришћени и раније у програмирању, али често нису подржани од стране процедуралних парадигми и језика. С друге стране, они су експлицитно промовисани у основним објектно-оријентисаним (ОО) концептима и принципима, и директно подржани од стране ОО програмирања, језика за моделовање и метода. [16,31] Најважнији концепти и принципи ОО парадигме су:

- Апстракција (укључујући апстрактне типове података)
- Енкапсулација
- Наслеђивање
- Полиморфизам.

Апстракција је чин који одбацује појединачне делове неке појаве до оног нивоа детаља који се и даље може схватити. Апстрактни тип података се дефинише као:

- Тип, који специфицира групу инстанци која постоји у време извршења програма
- Кохерентна заједница података и операција који раде над подацима
- Део програма који има своју интерну имплементацију у једном делу, састоји се од структуре података и имплементације операција и интерфејса са друге стране, садржи идентификаторе операција (имена и аргументи операција) и споља видљиву семантику
- Представа апстракције из домена проблема или програмског решења.

Енкапсулација је принцип где је споља видљив и приступачан интерфејс и скривена, недоступна имплементација, може бити јасно идентификована на софтверским компонентама.

Наслеђивање је концептуални однос између две врсте апстрактних типова података који могу бити изражени чињеницом да потомак врсте имплицитно поседује све карактеристике (структуру и понашање) претка, осим ако су експлицитно искључени.

Наслеђивање често (али не увек) садржи подтип основног типа, при чему је потомак такође подтип претка основног типа. Ово значи да су инстанце подтипа такође и инстанце претка основног типа података.

Полиморфизам је концепт који омогућава да различите класе пружају различите имплементације за исту операцију. Инстанце тих класа ће реаговати на призивање исте операције другачије, у складу са имплементацијама за своју класу. Специјалан (и најважнији) случај полиморфизма долази када подтип замењује рад пружајући различите примене наслеђене операције. Полиморфизам омогућава да понашање одговара типу инстанце који се активира чак кроз инстанцу којој је приступљено као општој.

Операција је спецификација услуга које се могу затражити од одређене инстанце или апстракције. Имплементација операције пружа понашање које се манифестује у складу са конкретним захтевима.

Веза објектно-оријентисане парадигме са развојем различитих информационих система је преваходно постигнута коришћењем објектно-оријентисаних програмских језика за имплементацију понашања (или тзв пословна логика) на основне релационе базе података којој се приступа преко перзистентног слоја података који мапира објекте и релације. Овај приступ је делимично заменио употребу четврте генерације програмских језика који се директно уклапају у релационе парадигме. У садашњој фази техничког развоја, овај широко коришћен приступ пати од дисконтинуитета у развоју узрокованих непотпуним или неформалним везама објекта са релационом парадигмом. Овај проблем се може превазићи коришћењем парадигме развоја информационих система на основу модела коришћењем извршног UML–а.

3.2.1. UML

UML (*The Unified Modeling Language*) је стандардни језик који се користи за моделовање софтвера. Предложен је средином деведесетих, а стандардизован 1997. године. То је језик опште намене чији је циљ моделовање свих врста софтверских система)[4].

UML није, међутим, потпуно формални језик. То значи да његова семантика није дефинисана на недвосмислен начин у свим својим елементима. Из тог разлога, UML не може да се користи као језик на исти начин као и традиционални програмски језици, јер би у том случају спецификација софтверског система била двосмислено тумачена од стране рачунара. Да би се такав језик користио мора имати формалну, недвосмислену семантику, то јест, јединствено тумачење сваког свог концепта који би требало да има *run-time* ефекте.

UML је графички језик који укључује стандардизовану графичку нотацију која се користи за креирање апстрактног модела система, која се назива UML модел. Сачињен је од интегрисаног скупа дијаграма. Примена UML-а помаже у остваривању следећих задатака[16]:

- Спецификација;
- Визуализација;
- Пројектовање архитектуре;
- Развој;
- Симулација и тестирање;
- Документација.






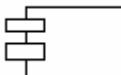

Један од примарних циљева у пројектовању самог UML-а је био да се корисницима обезбеди визуелни језик за моделовање тако да они могу да развијају и размењују моделе, као и да се интегришу најбоља знања.

У UML -у постоје четири врсте ствари (*things*)[16]:

- Ствари структуре – статички делови модела који репрезентују концептуалне или физичке елементе (именице);
- Ствари понашања – динамички делови модела који репрезентују понашање кроз простор и време (глаголи);
- Ствари груписања – организациони делови модела;
- Ствари анотације – описни делови модела, коментари који се примењују на било који документ




Статички делови UML модела су дати у следећој табели [6].

Табела 3.2. Статички делови UML модела [6]

Име	Симбол	Опис
Класа		Опис скупа објеката који деле исте атрибуте, операције, везе и семантику. Имплементира један или више интерфејса.
Интерфејс		Колекција операција које описују сервисе класе или компоненте.
Сарадња		Дефинише интеракцију и удружује улоге и друге елементе тако да раде заједно и обезбеђују колаборативно понашање.
Корисник		Спољашњи ентитет који комуницира са системом, обично особа.
Случај употребе		Опис скупа секвенци акција које систем изводи да би извршио неки захтев корисника.
Компонента		Физички и заменљиви део система који обезбеђује реализацију скупа интерфејса
Чвор		Физички елемент који постоји у време извршавања и представља рачунарски ресурс – има меморију и могућност процесирања.

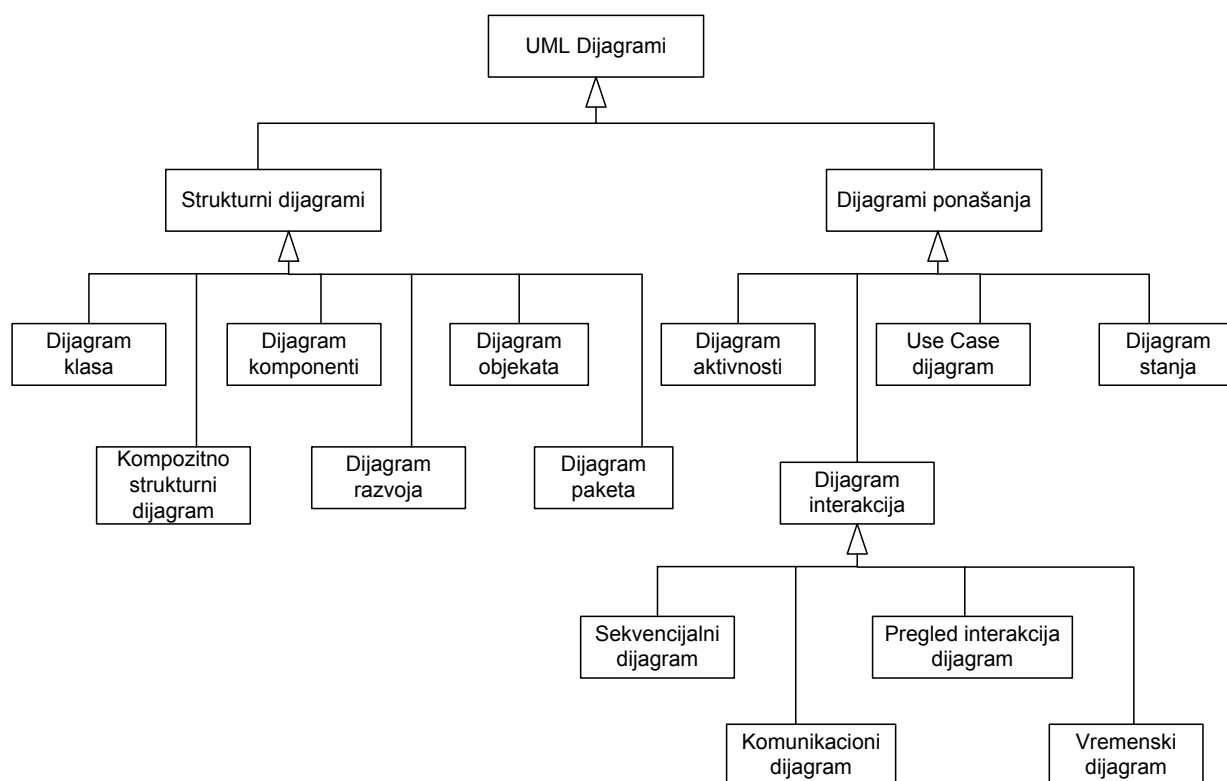
Релације које се могу јавити између ствари су дате у следећој табели.

Табела 3.3 Релације UML модела

Име	Симбол	Опис
Зависност		Семантичка релација између независне и зависне ствари. Независна ствар утиче на семантику зависне. Усмерење – из зависног случаја.
Генерализација		Објекти специјализованих елемената (дете) представљају замене за објекте генерализованих елемената (родитељ). Врх стрелице на родитељу.
Реализација		Семантичка релација између класификатора, где један класификатор специфицира уговор који други класификатор гарантује да ће испунити.

Име	Симбол	Опис
Асоцијација	$0..1$ * radi radj	Структурна релација која описује скуп веза којим се поставља веза између објеката.

У оквиру UML 2.0 идентификовано је тринаест типова дијаграма. Категоризација дијаграма је приказана на наредној слици [6].



Слика 3.11 – Хијерархијска подела UML 2.0 дијаграма [6,51]

Дијаграми случајева употребе омогућавају крајњим корисницима да разумеју систем. Приказују поглед корисника на функционисање система (шта систем ради, а не како систем функционише). Развој дијаграма случајева употребе дефинише се следећим активностима:

- Дефинисањем учесника;
- Дефинисањем случајева употребе;
- Дефинисањем типова веза између учесника и случајева употребе;
- Израдом дијаграма случајева употребе.



Слика 3.12 – Дијаграм случај употребе

Корисник је човек који користи систем, док је учесник специфична улога коју корисник има у комуникацији са системом. Учесник је особа или вештачки ентитет (софтвер или систем) који учествује у случају употребе.

Учесника је могуће идентификовати на основу одговора на следећа питања:

- Ко ће користити основну функционалност система (примарни учесници)?
- Ко треба да управља, администрира и одржава систем (секундарни учесници)?
- Коме ће бити потребна подршка система у обављању дневних задатака?
- Којим уређајима систем треба да управља?
- Са којим другим системима дотични систем треба да буде у вези?
- Ко или шта је заинтересован за резултате које систем производи?

Случај употребе – дефинише функционалност система са становишта учесника – шаблон понашања делова система. Питања за учесника – идентификују случајеве употребе:

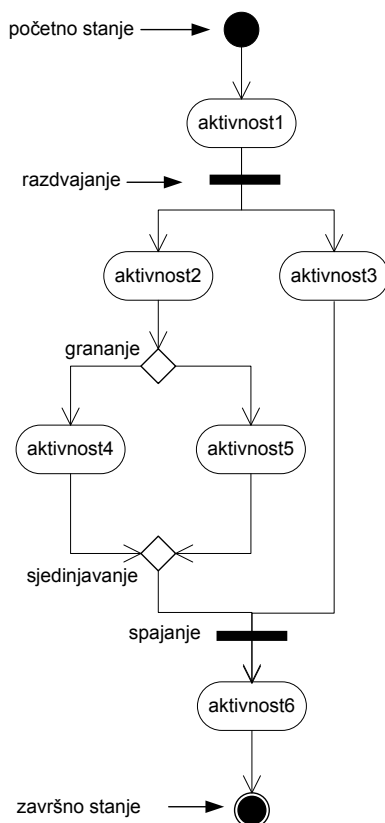
- Које функције учесник захтева од система – шта учесник треба да ради?
- Да ли учесник треба да ажурира неке информације у систем?
- Да ли учесник треба да буде обавештен о догађајима у систему?
- Да ли свакодневни рад учесника може да се поједностави кроз нове функције система?

Везе које се успостављају у дијаграму случајева употребе:

- Асоцијација (*Association*)
 - Асоцијација између случајева употребе типа <<include>>
 - Асоцијација између случајева употребе типа <<extend>>
- Генерализација (*Generalization-Inheritance*)
- Зависност (*Dependency*)

Дијаграми активности (*Activity Diagram*) служе за истраживање и описивање тока активности, приказивање акција операција у класи, слично као традиционални дијаграм тока програма. Могу се користити за описивање пословних процеса, радног тока у контексту организације итд.. Радни ток (тј. ток активности) може да буде једноставна операција, као што је увођење реда у неком систему уредности, или може да буде и сложенији, као што је контролисање производње и њен развој.

Дијаграм активности приказује ток активности кроз систем. Дијаграми се читају одозго на доле и имају гранање и раздвајање за описивање услова и паралелних активности. Раздвајање се користи када се јави већи број активности у исто време. На слици 3.13 након активности *aktivnost1* следи раздвајање, што значи да се активности *aktivnost2* и *aktivnost3* одвијају паралелно у истом времену. Након активности *aktivnost2* јавља се гранање, које се приказује симболом одлуке (празан ромб). Гранање описује које активности ће да се одиграју након одговарајућег услова. Сва гранања на крају се завршавају у истој тачки која представља сједињавање и означава крај гранања. Након сједињавања све паралелне активности се комбинују и спајају пре завршног стања.

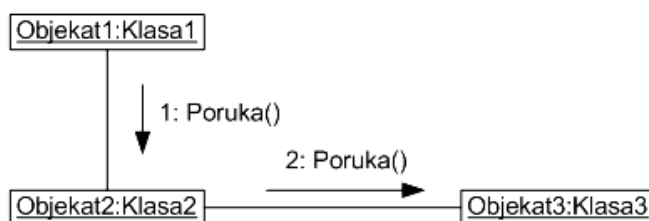


Слика 3.13 – Пример дијаграма активности[5]

Интеракциони дијаграми моделују понашање Use Case-ова описујући начин како група објеката интерагује у циљу завршења неког задатка. У дијаграме интеракција спадају дијаграм сарадње и дијаграм секвенци или секвенцијални дијаграм. Ови дијаграми биће описани у овом поглављу.

Дијаграм сарадње показује везу између објеката и редослед порука које се прослеђују између њих. Ови дијаграми помажу да се лакше прикажу сложеније интеракције и да се покажу везе између објеката који сарађују.

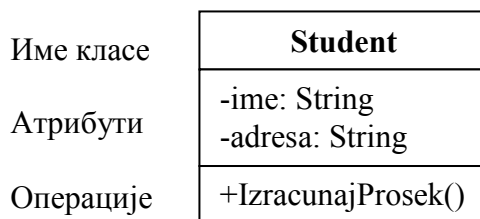
Стрелице између објеката приказују поруке које се прослеђују између њих. Бројеви поред порука представљају секвенцу бројева. Како назив предлаже, бројеви приказују секвенцу порука која се прослеђује између објеката. За једноставније примере може да се користи секвенца бројева 1.,2.,3. итд., док за сложеније се прелази на 1,1.1,1.2,1.2.1,...итд. На слици 3.14 приказани су основни елементи: објекти и поруке које се прослеђују између њих.



Слика 3.14 – Основни елементи дијаграма сарадње[5]

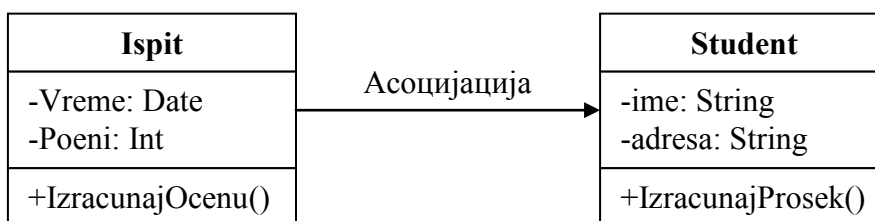
Секвенцијални дијаграми се користе за моделовање редоследа, тј. секвенце, догађаја који настају у току међусобне интеракције између објеката у систему. Објекти при том могу да буду: организационе јединице, компаније, рачунари, људи, процеси или пак неке механичке ствари. Секвенцијални дијаграми обично описују секвенце порука између више објеката, где су редослед и време порука детаљно описани.

Дијаграми класа се широко користе да опишу типове објеката у систему и њихове везе. Ови дијаграми користе следеће елементе: класе, пакете и објекте. Дијаграми класа описују три различита погледа при пројектовању система и то концептуални поглед на систем, спецификацију и имплементацију. Класа се састоји од имена, атрибута и операција. На следећој слици приказан је пример класе.



Слика 3.15 – Пример класе

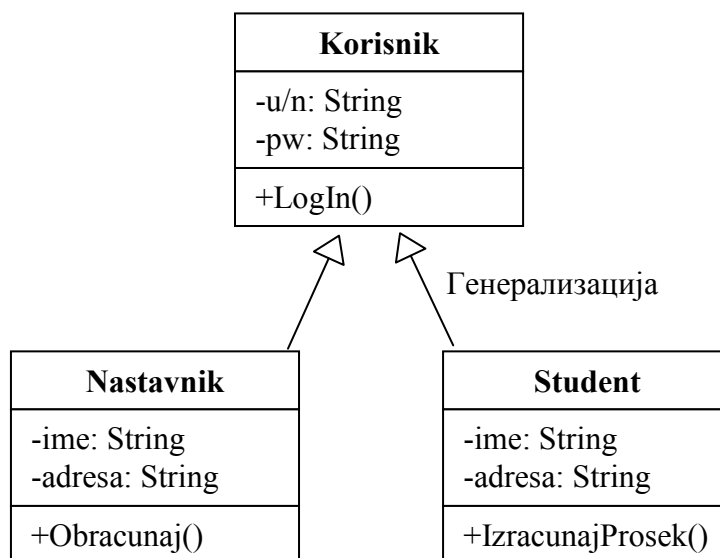
Дијаграми класа такође приказују везе као што су наслеђивање, асоцијативност и друге. На следећој слици приказан је пример асоцијативности.



Слика 3.16 – Пример асоцијативности

Веза асоцијативност је веза која се највише користи код ових дијаграма. Ова веза приказује везу између инстанца класа. На пример, класа Ispit је асоцирана класом Student. Вишеструкост асоцијације назначавача број објеката који учествује у вези.

Следећа веза која се широко примењује је генерализација. Генерализација се користи када су две класе сличне у основи, али се разликују у неким важним детаљима. На следећој слици приказан је пример генерализације.



Слика 3.17 – Пример генерализације[4,18]

На приказаном примеру генерализације класе *Nastavnik* и *Student* имају неке сличности као што су име и адреса, али имају и своје атрибуте и операције. Класа *Korisnik* је генералнија од класа *Nastavnik* и *Student*.

Пре почетка цртања дијаграма класа потребно је размотрити три различите перспективе система који се представља: 1) концептуалну перспективу, 2) спецификацију, и 3) имплементацију. Потребно је фокусирати се на једну перспективу и видети како раде све заједно. Када се пројектују класе потребно је размотрити атрибуте и операције које ће да садржи. Затим је потребно да се види како ће инстанце класа интераговати са осталим. Ово су први кораци у развоју дијаграма класа. Користећи ове основне технике могуће је развити комплетан поглед система.

3.2.2. Rational Unified Process

Rational Unified Process (RUP) је методологија која развоја софтвера, заснована на UML-у. Сматра се да је RUP “водич“ за ефикасно коришћење UML-а. Ова методологија омогућава јасно додељивање задатака и одговорности унутар развојног тима. Циљ методологије је да обезбеди развој високог квалитетног софтвера који ће задовољити потребе крајњих корисника, у оквиру предвиђеног буџета и у оквиру предвиђених рокова. Суштина RUP-а је израда и доградња модела, а не велика количина докумената у папирном облику.[7]

Rational Unified Process методологију је развила и унапређује је фирма *Rational Software*. Приликом развоја методологије водило се рачуна да она обезбеди максималну продуктивност приликом развоја софтвера, обезбеђујући да чланови тима користе исти језик, исти процес и исти поглед на развој софтвера. RUP је подржан и са доста алата који аутоматизују велики део самог процеса моделовања. У исто време он је прилагодљив тако да га могу користити мали развојни тимови, али и велике софтверске фирме. RUP се темељи на обезбеђивању коришћења најбоље праксе сваком члану тима. Неке од основних предности RUP-а су:[7]

- Итеративни развој софтвера;
- Управљање захтевима;
- Коришћење архитектуре засноване на модулима;

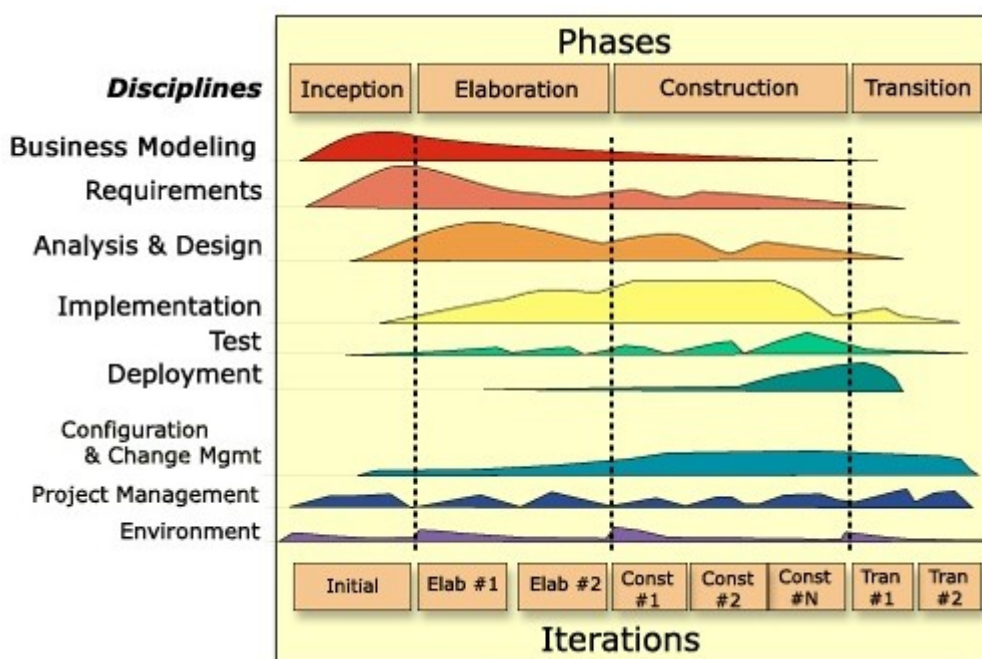
- Визуелни модел софтвера;
- Провера квалитета софтвера;
- Контрола измена-верзија софтвера.

Постоји девет фаза-корака у развоју софтвера-система применом RUP-а. Шест од њих можемо сматрати “инжењерским“ и уједно кључним корацима и то су: [7]

1. Моделовање пословних процеса;
2. Моделовање захтева корисника;
3. Анализа и дизајн;
4. Имплементација;
5. Тестирање;
6. Увођење.

Три можемо сматрати “помоћним“ корацима и они подразумевају:

1. Управљање пројектом;
2. Управљање конфигурацијом и променама;
3. Управљање окружењем.



Слика 3.18 – Фазе и дисциплине RUP-а [7]

И ако имена шест основних корака могу да вас подсети на фазе у традиционалној “водопад“ методологији развоја, треба имати у виду да се фазе итеративног процеса разликују и да се појединачни кораци стално понављају у току животног циклуса. У ствари RUP методологију чини 9 корака који се повремено понављају у различитим фазама и различитог интензитета.

Моделовање пословних процеса.

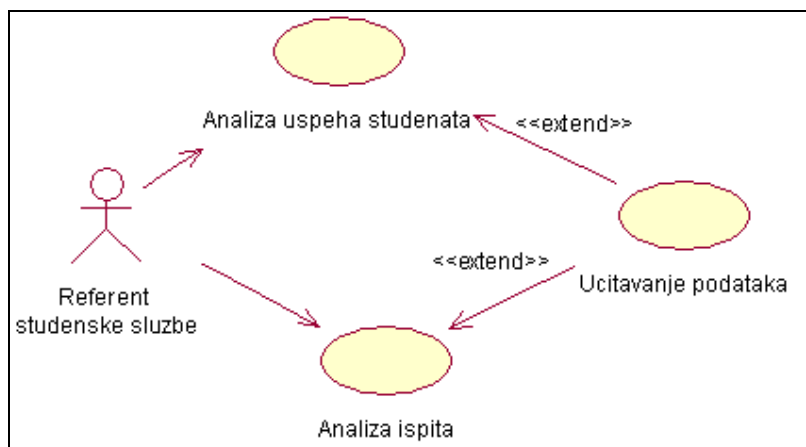
Један од великих проблема, који доводи до додатног ангажовања, је недостатак квалитетне комуникације између развојног тима и корисника. Ово води ка томе да се захтеви корисника не користе у довољној мери у раду развојног тима и обрнуто. RUP управо омогућава превазилажење овог проблема обезбеђујући јединствени језик за кориснике и програмере, обезбеђујући им могућност да заједнички креирају модел пословних процеса који су предмет будуће аутоматизације. [7]

Моделовање пословних процеса се врши такозваним пословним случајевима употребе (*Business Use Case*). На овај начин се управо омогућава јасна слика онога што треба аутоматизовати. Пословни случајеви употребе се додатно анализирају како би се схватили принципи моделованих пословних процеса.

Моделовање захтева корисника

Циљ моделовања захтева корисника је да се опише шта будући систем треба да ради. То омогућава међусобан договор корисника и програмера, што доводи до ефикаснијег рада на развој система. У овом процесу се креира документ – модел који представља визију будућег система. У оквиру тог модела се идентификују корисници будућег система у виду људи корисника и других система корисника који могу да буду у интеракцији са нашим системом.

Такође, идентификују се случајеви коришћења који енкапсулирају будуће понашање система. Случајеви коришћења се развијају на основу изражене потребе корисника и тако обезбеђују повећану важност овог модела. На следећој слици приказан је пример једног модела случајева употребе.



Слика 3.19 – Пример Use Case модела

Сваки од случајева употребе у моделу се описује детаљно. Опис случаја употребе, израдом дијаграма активности, описује како ће систем комуницирати са корисником и који ће операције при том извршавати. Овако дефинисан модел случајева употребе користи се у свим наредним корацима развоја система.

Анализа и дизајн

Циљ анализе и дизајна је да покаже како ће систем на крају бити реализован. Овај корак у развоју се заснива на функционалностима дефинисаним у моделу случајева коришћења који су, наравно, настали на основу захтева корисника. Резултат анализе и дизајна је модел који представља дизајн – изглед будућег система. Дизајн служи као апстракција будућег кода система, односно, дизајн показује будућу структуру- модуле система.

Модел дизајна се најчешће састоји од класа структурираних у пакете и подсистема са дефинисаним интерфејсима који репрезентују будуће елементе у имплементацији. Овај модел такође описује и како ће те класе међусобно сарађивати у реализацији захтева корисника.

Имплементација

Основни циљеви имплементације су:

- Да дефинише организацију кода , односно, слојеве имплементације;
- Да имплементира класе и објекте у конкретне програмске датотеке;

- Да обезбеди тестирање развијених компоненти система;
- Да интегрише делове у један целовит систем.

Систем се, у суштини, реализује имплементацијом његових компоненти. RUP омогућава и описује како већ развијене компоненте поново користити, или како имплементирати нове компоненте и, при том, обезбедити да се систем лако одржава и надграђује.

Тестирање

Сврха тестирања је:

- Да верификује интеракцију између објеката;
- Да верификује интеграцију свих компоненти система;
- Да верификује да су сви захтеви корисника узети у обзир;
- Да идентификује аномалије у систему пре његовог увођења у употребу.

Итеративност коју RUP процес доноси омогућава да се тестирање практично врши од самог почетка пројекта. Ово омогућава да се евентуалне грешке уоче најраније могуће што значајно смањује трошкове развоја пројекта. Кроз тестирање се обезбеђује провера поузданости, функционалности и перформанси система. Тестирање се, кад год је то могуће, аутоматизује, што додатно олакшава рад на пројекту.

Увођење

Сврха увођења је да, најуспешније могуће, прати појаву верзија система и обезбеди њихову дистрибуцију кориснику. Активности у оквиру ове фазе су:

- Праћење верзија система;
- Дистрибуција система;
- Инсталирање система;
- Обезбеђивање помоћи корисницима;
- Планирање и извођење бета тестова;
- Миграцију података.

И ако је увођење једна од последњих фаза, многе од наведених активности се могу реализовати и раније у току развоја система. То омогућава додатну ефикасност и повећава квалитет будућег система.

Управљање пројектом

Управљање пројектом је, у ствари, проналажење баланса између постављених циљева, ризика који прате пројекат и ограничења у развоју система. Крајњи циљ је да се обезбеди да будући корисници буду задовољни. Чињеница да доста пројеката не доживи успешан завршетак говори о значају ове активности.

Управљање пројектом треба да обезбеди:

- Флексибилно радно окружење за развој система;
- Практичну помоћ за планирање избор програмера и надзор над пројектом;
- Управљање ризицима.

Управљање конфигурацијом и променама

У оквиру ове активности обезбеђује се контрола над деловима система које развијају засебни тимови или програмери. Циљ је да се обезбеди избегавање повећања трошкова и конфузија до које може да дође оваквом организацијом развоја система.

Ова активност обезбеђује и праћење зашто, кад и од стране кога су поједини делови система мењани. Такође, овом активношћу се обезбеђује и управљање евентуалним променама захтева корисника и њиховом утицају на развој система.

Управљање окружењем

Сврха управљања окружењем је да обезбеди потребне услове и подршку развојном тиму. Ова активност, практично, представља неку врсту водича за развој и имплементацију система.

3.2.3. Microsoft Solution Framework

Microsoft Solution Framework (MSF) је настао као синтеза најбољих аспеката разних методологија и огромног искуства компаније Microsoft у управљању сопственим пројектима. MSF доноси флексибилан и скалабилан приступ у планирању, дизајну, развоју и увођењу успешних ИТ решења који подмирује потребе организације и пројектног тима било које величине. MSF обезбеђује скуп модела, правила и смерница за пројектовање и развој пословних решења на начин који осигурава да се свим

елементима пројекта (људи, процеси, алати) успешно управља. MSF процесни модел комбинује најбоље принципе модела водопада и спиралног модела. Дисциплине MSF -а су Управљање ризицима, Управљање тимом и Управљање процесима.[18]

Управљање ризицима (*Risk Management*) – подржава проактивно управљање ризицима, континуалну оцену ризика и одлучивање током животног циклуса пројекта. Пројектни тим континуално процењује, надгледа и активно управља ризицима. Управљање ризицима дефинише шест корака током којих тим управља текућим ризицима, планира и извршава стратегије управљања ризицима[18]:

- Идентификовање ризика – применом *Brainstorming*-а могу се идентификовати сви потенцијални ризици.
- Анализа ризика – према процени вероватноће догађаја ризика и његовог утицаја на систем, ризици се сортирају према приоритету.
- Планирање ризика – користе се информације добијене анализом ризика како би се формулисали планови, стратегије и акције. За сваки ризик се процењује његов утицај на исход пројекта, наводе се начини његовог умањења и кораци које треба спроводити уколико до ризика дође.
- Праћење ризика – надгледа се статус одређених ризика.
- Контролисање ризика – процес извршавања планова акција и њиховог извештавања.
- Учење из ризика – формулишу се научене лекције како би се то знање поново употребило у сличним случајевима код будућих пројеката.

Главни постулат **управљања тимом** је да се установе јасна задужења за одређене задатке на пројекту, али и подељена одговорност за успех пројекта. Тим ради тако да оствари једну визију, а чланови тима у томе равноправно учествују. У оквиру тима, свачија улога једнако доприноси и једнако је одговорна за успех пројекта.[18]

MSF управљање тимом наглашава значај јасних улога, обавеза и циљева појединих чланова за успех пројекта. Флексибилност MSF модела даје могућност прилагођавања домену пројекта, величини тима и способностима чланова тима. Једна улога не мора да буде једна особа, односно неколико људи може припадати истој улози, док појединац може учествовати у више од једне улоге. [18]

MSF модел одређује шест различитих улога од којих свака има јасно дефинисане обавезе и циљеве и то[18]:

1. *Управљање програмом (Program Management)*. Циљ ове улоге је да испоручи решење у оквиру дефинисаних ограничења пројекта. Ова улога покрива следеће функционалне области:

- Управљање пројектом - праћење и управљање буџетом и мастер планом пројекта, управљање ризицима на пројекту, управљање расподелом ресурса и комуникацијом у пројектом тиму, праћење напретка на пројекту и управљање извештавањем.
- Квалитет процеса - управљање нивоом квалитета процеса и препорука побољшања.
- Администрација - имплементирање дефинисаних процеса и подршка лидерима тимова у коришћењу тих процеса.

2. *Управљање производом (Product Management)*. Циљ ове улоге је да обезбеди задовољење захтева клијента. Улога управљања производом сарађује са улогом управљања програмом и улаже напоре за установљивање заједничке визије пројекта свих учесника на пројекту. Ова улога покрива следеће функционалне области:

- Пословна вредност - анализа пословног проблема, постављање пословних циљева, дефиниција и одржавање пословне оправданости (*Cost Benefit*) пројекта, дефиниција и мерење пословне вредности који пројекат генерише, дефиниција потребних аналитика.
- Маркетинг - управљање маркетиншким и PR порукама.
- Адвокат клијента - дефинисање и управљање заједничком визијом пројекта и решења, управљање очекивањима клијента и комуникацијом са клијентом.
- Планирање производа - прикупљање, анализа и давање приоритета пословним захтевима клијента; утврђивање мерних јединица за пословне захтеве и критеријума успешности; утврђивање плана постављања различитих верзија производа – када ће бити постављање нове верзије, да ли пуна верзија или само део, најновија верзија или не и сл.

3. *Развој (Development)*. Циљ ове улоге је да пројектује и развије решење у складу са дефинисаним спецификацијама. Развојна улога учествује у изради архитектуре решења

и функционалне спецификације, затим ствара решење, и припрема развојно и тестно окружење. Ова улога покрива следеће функционалне области:

- Предлог стратегије, архитектуре решења и функционалности апликације
- Консултације о технологији - евалуација и валидација технологија; активно учешће у развоју и прегледу функционалне спецификације
- Процена постојећих извора података
- Имплементација архитектуре и дизајна (мапирање архитектуре предузећа са архитектуром за имплементацију решења; имплементација физичког дизајна решења)
- Развој/прилагођавање апликације; имплементација потребних аналитика
- Моделовање података, базе података и складишта података.
- Развој инфраструктуре (развој потребних карактеристика, развој упутства за постављање апликације; преглед кода и тестирање у сарадњи са носиоцима улоге тестирања)

4. *Тестирање (Test)*. Циљ ове улоге је да одобри верзију за постављање, након идентификовања и решавања свих проблема везаних за квалитет решења. Ова улога покрива следеће функционалне области:

- Планирање тестирања (развијање приступа који ће се користити и плана тестирања; учествовање у постављању мерила квалитета; развој тестних спецификација)
- Пројектовање тестирања (развој и одржавање аутоматизованих тест случајева, алата и скриптова; извођење тестова како би се утврдио статус развоја решења)
- Извештавање о тестирању (обезбеђивање података тиму о квалитету производа, праћење свих грешака и осигурање њиховог разрешавања пре постављања верзије)

5. *Управљање постављањем (Release Management)*. Циљ ове улоге је да обезбеди несметано постављање верзија производа. Сарађује са ИТ сектором клијента. Ова улога покрива следеће функционалне области:

- Инфраструктура (развој планова инфраструктуре предузећа клијента, политике и процедура; управљање набавкама хардвера и софтвера; изградња тестног и прелазног окружења које одговара продукционом окружењу)
- Подршка у процесу постављања (обезбеђивање примарног контакта и подршке ИТ корисницима; управља SLA уговорима са клијентима; обезбеђује решавање инцидентата и проблема)
- Операције (подешавање параметара система, база података и сл)
- Логистика (управљање логистиком)

6. *Искусство корисника (User Experience)*. Циљ ове улоге је да побољша ефективност решења за кориснике. Ова улога покрива следеће функционалне области:

- Приступ (обезбеђење укључивања захтева корисника за приступ у дизајн решења)
- Заступање корисника пред пројектним тимом
- Упутство за обуку и коришћење; дефинисање и развој система подршке
- Обука корисника и Help desk.

Управљање процесима води кроз редослед активности везаних за пројекат и представља животни циклус пројекта. Управљање процесима је флексибилно и може да се прилагоди пројектовању и развоју широког спектра комплексних пројеката. MSF управљање процесима преставља општи скуп активности које су подељене у фазе и представља итеративан модел који се може применити на развој и увођење комплексних ИТ решења.[18]



Слика 3.20 – MSF управљање процесима [18]

MSF управљање процесима се састоји од пет различитих фаза:[18]

1. *Предвиђање (Envisioning)*. Предвиђање се може дефинисати као прављење грубог описа циљева и ограничења пројекта. Сврха фазе предвиђања је да направи заједничку визију пројекта за све учеснике на пројекту и пројектни тим. За време фазе предвиђања тим дефинише циљ, сврху, оквире, захтеве корисника, ризике, претпоставке, пројектна ограничења и правила и предлаже могуће решење. Крајњи резултат ове фазе је пројектни документ фазе предвиђања.

2. *Планирање (Planning)*. За време планирања тим припрема детаљну функционалну спецификацију решења и припрема планове рада, процену трошкова и распоред за различите резултате. Тим такође припрема и план комуникације са учесницима на пројекту, као и план управљања ризицима. Крајњи резултат ове фазе је пројектни документ фазе планирања.

3. *Развој (Developing)*. За време фазе развоја, пројектни тим прави решење. Крајњи резултат ове фазе је решење спремно за тестирање.

4. *Стабилизација (Stabilizing)*. За време фазе стабилизације тим обавља интеграцију, учитавање и тестирање решења. За време ове фазе, решење прелази из стања у коме су све карактеристике завршене као што је дефинисано у функционалној спецификацији за ову верзију, у стање у коме се испуњавају дефинисани нивои квалитета. Крајњи резултат ове фазе је решење спремно за увођење у пословање.

5. *Постављање (Deploying)*. За време ове фазе, тим уводи технологију решења, смешта компоненте, стабилизује увођење, преноси пројекат оперативцима и подршци на страни клијента и добија одобрење пројекта од клијента. Крајњи резултат ове фазе је оперативно решење.

У MSF постоје и такозвани маркери. Маркери су временске тачке на пројекту у којима је потребна одређена провера актуелног стања и поређење са планираним. Маркери су углавном везани за испоруку дефинисаног резултата или прибављање одобрења

(интерног или екстерног) ради наставка пројекта. Маркерима контролише управљање процесима у MSF-у.

3.2.4. Развој вођен моделом (Model Driven Development)

Развој вођен моделом је приступ у подизању нивоа апстракције који се успешно користи више од десет година. Његова основна премиса је да користи моделе уместо (само) кода у развоју конкретног софтвера. Модели су углавном нелинеарне форме, за разлику од кода који је инхерентно линеаран. Нелинеарно значи да се модели састоје од елемената који су међусобно повезани на начин који је слободнији него једноставна секвенца у којој сваки елемент може да има (највише) два суседна елемента. Из тог разлога, модели обично користе визуелне ознаке, као што су дијаграми, уместо чистог текста.[16]

Развој вођен моделом је дисциплина која обухвата програмирање модерног софтвера заснованог на моделима, користећи језике за моделовање и алате. Међутим, појмови и принципи ове дисциплине имају корене у другим областима инжењеринга.

Друге дисциплине инжењеринга, које су много старије и зрелије него софтверско инжењерство (нпр. грађевински, механички или електротехнички инжењеринг), захтевају да дизајнери новог система прво израде поједностављену представу самог система како би анализирали карактеристике будућег система пре него што се изгради. Таква поједностављена представа система се зове модел система а процес његовог стварања се назива моделовање.

Модели могу бити изражени у апстрактним појмовима, као што су, на пример, математички модели, шеме или технички цртежи. С друге стране, модели могу бити физички, обично мале копије направљене од пластике, дрвета или других материјала. Модели дозвољавају развојном тиму да експериментише са дизајном и тестира његове карактеристике пре него што систем буде заиста изграђен. Такав приступ смањује ризик од неисправне конструкције реалног система, са потенцијално катастрофалним или скупим последицама.

Не постоји ниједан разлог због чега софтверско инжењерство не би следило ове примере добре праксе других инжењерских дисциплина и експлоатисало моделе и

моделовање у изградњи сложених софтверских система. Једна кључна разлика је у томе што у софтверском инжењерству, “материјал” од кога су модели направљени могу бити исти као крајњи систем. Модел може бити формална и недвосмислена спецификација која се може интерпретирати на другом софтверском систему или може бити претворен у форму у којој се може извршавати на одређеним уређајима(рачунарима). У том случају, овај модел се назива извршним и представља прави систем у изградњи.

Модел је поједностављена представа стварног света. Направљен је да пружи боље разумевање сложених система који се развијају. Модели сложених система се израђују, јер се такви системи не могу схватити у потпуности. Моделовањем се постижу четири основна циља:[16]

- Модели помажу да се сагледа систем какав је стварно или какав дизајнери желе да он буде (*визуелизација*);
- Модели дозвољавају дизајнеру да опише структуру и понашање система (*спецификација*);
- Модели дају шаблоне који су водич у изградњи система (*конструкција*);
- Модел документује дизајнерске одлуке које су донете роком развоја (*документација*).

У основи, програми креирани у традиционалним програмским језицима могу бити третирани као модели, јер описују софтвер на формалан начин. Међутим, програми креирани у традиционалним програмским језицима се не сматрају моделима, због следећег:[16]

- Модели се углавном састоје од апстрактних појмова, а програми се баве детаљима конкретне имплементације;
- Модели су често некомплетни и непрецизни, посебно у раним фазама анализе, мада их ово не спречава да се извршавају. Програми су коначан облик развоја софтвера;
- Модели су углавном описани у визуелном (дијаграми) облику у комбинацији са текстуалним деловима, док су традиционални програмски језици углавном текстуалне форме.

Дисциплина софтверског инжењерства у којој су модели централни употребни предмети развоја, који се користе за изградњу система, заједно са дизајнерским одлукама и генерисањем других употребних предмета дизајна, назива се Развој вођен моделом (*Model Driven Development* – MDD).

За изградњу модела, програмери морају да знају речник који се користи за моделовање. Речник за моделовање (тј. скуп концепата, заједно са семантиком ових концепата, њихове особине, везе и синтаксу) мора имати облик дефиниције језика за моделовање. Коришћењем одређеног језика за моделовање, програмери састављају реченице на том језику, стварајући моделе. Дакле, језик за моделовање је кључни алат на располагању програмерима за изградњу модела, јер нема сврхе у прављењу модела без разумевања његовог значења. Дакле, језик за моделовање пружа могућност за комуникацију између програмера.

Да би био успешан, језик за моделовање мора да буде пажљиво избалансиран да би ујединио различите, често супротне захтеве.

С једне стране, језик за моделовање мора бити довољно једноставан, лако разумљив и употребљив моделарима. Ако би језик за моделовање био сувише сложен да би био разумљив, не би био широко прихваћен од стране софтверске заједнице. Поред тога, веома је пожељно да корисници који постављају захтеве разумеју језик за моделовање, или барем део који се користи у спецификацији захтева. Ово својство може значајно смањити ризик од неспоразума између корисника и моделара у тумачењу захтева.

С друге стране, језик за моделовање не би требало да буде превише једноставан или сувише конкретан. Ако језик за моделовање не би био довољно општи, не би било могуће покривати све случајеве који се јављају у реалном свету.

Поред тога, језик за моделовање мора бити довољно апстрактан и да концептуално буде близу домена проблема. Језик за моделовање који није довољно апстрактан пати од велике концептуалне удаљености од домена проблема. У том случају, развој захтева велике менталне напоре, јер моделари морају вршити стално концептуално мапирање из домена проблема у модел. Другим речима, језик мора бити довољно изражајан. Дакле, моделовање и трошкови развоја су у директној вези са могућношћу апстракције

проблема. Ако је језик довољно изражајан, модели се састоје од мањих реченица, а програмери ће уложити мање труда да одреде систем у изградњи и обрнуто.

Поред тога, језик за моделовање треба да омогући израду неформалних, непотпуних или неконзистентних модела јер процес моделовања у раним фазама дефинисања захтева и концепта има потребу за таквим моделима. С друге стране, језик за моделовање треба да буде једноставан и довољно примитиван како би се прецизно користила семантика која омогућава трансформацију модела у извршну форму. Високо апстрактни модел се обично трансформише у нижи облик који може бити интерпретиран од стране других софтверских система или се даље трансформише у још нижи облик који се на крају извршава на конкретном уређају (рачунару).

Да би се могао директно извршавати, језик за моделовање мора имати прецизну и потпуно формалну семантику. Такође, и модел који би требало да буде извршен, мора бити формалан, недвосмислен и доследан. То су најважније карактеристике језика моделовања које би требало да су употребљиве и корисне за моделовање свих сложених софтверских система у целини, као и информационих система.

Конкретне нотације језика моделовања и имплементације морају бити текстуалне или визуелне. Текстуални језици омогућавају да направимо међусобно повезане моделе. Иако текстуалне моделе, као што су програми или скрипте или реченице написане у текстуалном језику, могу програмери да виде визуелно у две димензије, они су у суштини секвенцијални јер су трансформисани и анализирамо их као низове карактера.

Насупрот томе, визуелни језици преузимају израду модела у две или чак три димензије, најчешће користећи дијаграме у комбинацији са текстом. Они су обично сликовити дијаграми, који дају много више описа а лакше их је и прихватити него текстуалну форму. То је разлог зашто је језик за моделовање са дијаграмима популаран за апстрактно моделовање.

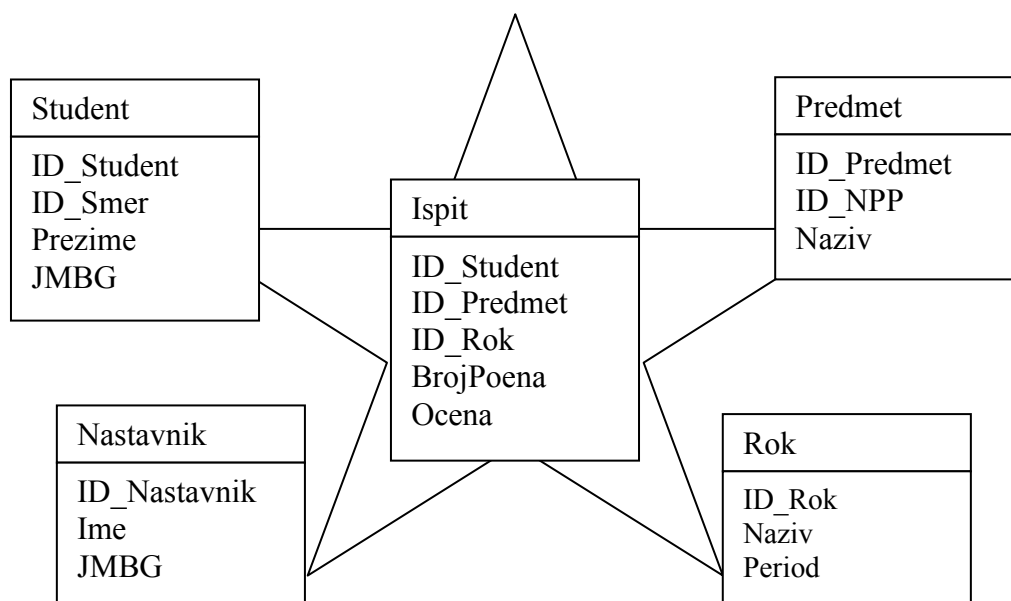
Међутим, није увек тачно да су визуелни језици више употребљиви него текстуални. Неке ствари могу бити много ефикасније изражене у текстуалном облику него у визуелном. На пример, за дефинисање једноставне петље или једноставно додавање два оператора, текстуални облик може бити много једноставнији и концизнији од

еквивалентне графичке нотације. Укратко, текстуални и језици са дијаграмима имају своје предности и недостатке. Обично је њихова комбинација најефикаснија за примену у моделовању.

3.3. Димензионо моделовање

Иако димензионо моделовање не представља у пуном смислу методолошки приступ, можемо га сматрати *de facto* индустријским стандардом у развоју аналитичких база података, односно информационих система за подршку у одлучивању. Димензионо моделовање одликују јасно дефинисани артефакти и кораци у развоју самог модела. Из наведених разлога у даљем тексту ће бити више речи о тим артефактима и о поступку израде димензионог модела.

Да би се превазишли проблеми при раду, за велике упите у складишту података, користимо димензионе моделе. Приступ тродимензионалном моделовању обезбеђује начин да се побољшају перформансе упита за збирне извештаје без утицаја на интегритет података. Међутим, то повећава трошак за додатни складишни простор. Димензиона база података обично захтева много више простора него релациона база података. Међутим, са смањењем трошкова простора за складиштење, овај проблем постаје мање значајан.[2]



Слика 3.21 – Шема звезде [2]

Димензиони модел се такође често назива *шема звезде*. Овај тип модела је врло популаран у складиштењу података, јер може да пружи много боље перформансе упита, посебно на веома великим упитима, од E/R модела. Међутим, она је такође од велике користи јер се лакше разуме. Састоји се, обично, од табела чињеница, са бројним другим табелама (табеле димензија) које је окружују и које садрже описне податке. Када је нацртана подсећа на звезду па је зато и добила то име. Наредна слика приказује пример шеме звезде.[2]

Димензионо моделовање је техника логичког дизајна чији је циљ презентација података у облику који обезбеђује високе перформансе система ради вршења анализе података. У димензионом моделовању структуре података су тако организоване да описују мере и димензије. Мере су нумерички подаци смештени у централној, такозваној, табели чињеница. Димензије су стандардни пословни параметри који дефинишу сваку трансакцију. Димензије се смештају у табеле непосредно, или преко друге табеле димензије, повезане са табелом чињеница.

Табела чињеница садржи нумеричке вредности онога што се мери. На пример, чињенична вредност од 80 може да значи да је број поена на испиту био 80. Свака табела чињеница садржи кључеве повезане са димензионом табелом. Ови кључеви се називају *страни кључеви* у табели чињеница. [2] Табеле чињеница обично садрже мали број колона. У односу на табеле димензија, табеле чињеница имају велики број редова.

Информације у табели чињеница имају следеће карактеристике:

- Нумеричка су и користе се за генерисање агрегација и збирова
- Вредности би требало да буду нумеричке како би се могло вршити њихово сумирање
- Све чињенице морају се директно односити на димензије. Ово омогућава приступ додатним информацијама из табеле димензија.

Табеле димензија садрже детаљне податке о чињеницама. То омогућава пословним аналитичарима да боље разумеју податке и извештаје. Табеле димензија садрже описне информације о нумеричким вредностима у табели чињеница. То јест, садрже атрибуте чињеница. На пример, табеле димензија за примену анализе испита могу да садрже

атрибуте као што су временски период, предмет и наставник. Пошто су подаци у табели димензија денормализовани, обично имају велики број колона. Табеле димензија обично садрже знатно мање редова података него табела чињеница. Атрибути у табели димензија се обично користе као редови и колоне у извештају или упиту приказа. На пример, текстуални описи на извештају потичу из атрибута у табели димензија. [2]

Димензије често могу бити хијерархијски организоване. Сваки хијерархијски ниво наставља се другим хијерархијским нивоом. На пример, унутар димензије време дани се настављају недељама а недеље кварталима. Хијерархије су изузетно битне за димензионо моделовање као основа за функције „поглед у дубину“ и „поглед навише“. Поглед у дубину је процес којим захтевамо детаљнији поглед на податке. Поглед навише би представљао збирни поглед (на већем нивоу) на податке. На слици су приказане хијерархије за потребе складишта података и тест апликације „Анализа испита“.



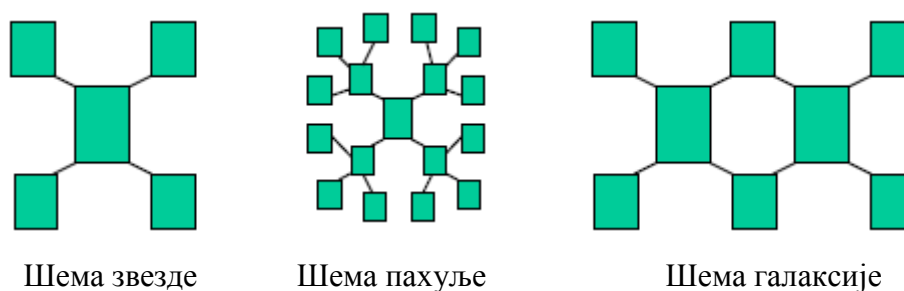
Слика 3.22 – Хијерархија димензија Студент, Предмет и Време

Димензиони елемент је специјална категорија података која репрезентује појединачни ниво у хијерархији димензија. За сваки ниво хијерархије постоји један димензиони елемент. Тако, за димензију студент могу бити димензиони елементи: класа, специјалност и смер. У нашем моделу димензиони елемент класа представља најнижи ниво у хијерархији димензије студент, док је смер највиши ниво у хијерархији.

Димензиони атрибути описују одређене димензионе елементе. На пример, презиме је атрибут који описује димензиони елемент студент, назив предмета је атрибут који описује димензиони елемент предмет. Димензиони атрибути допуштају корисницима да изврше категоризацију нехијерархијских података.

У зависности од представе димензија на моделу говоримо о нормализованом или денормализованом моделу. Постоје три основне врсте димензионих модела, а то су:

- Модел звезде
- Модел пахуље
- Модел галаксије.



Слика 3.23 – Врсте димензионих модела [2]

Код **модела звезде** шема имају једну табелу чињеница и више табела димензија. Димензиона табела није денормализована. Основне предности шеме звезде су што омогућава дефинисање сложених више-димензионих података у виду једноставног модела, смањује број физичких веза које се морају процесирати при задавању упита чиме се постиже побољшање перформанси система и омогућава проширење складишта података уз релативно једноставно одржавање. Велика мана шеме звезде је што се повећава редувантност података. [2]

Даља нормализација и проширење табела димензија, у шеми звезде, имају за резултат формирање **модела пахуље**. Модел пахуље се користи, када се уклањају колоне ниске кардиналности из табела димензија, како би раздвојили нормализоване табеле које онда поново повезујемо на основну димензиону табелу. [2]

Модел галаксије је димензиони модел који се састоји од више табела чињеница, спојених преко димензија. [2]

4. Развој ВІ система

Раније су се информациони системи развијали, а да се није водило рачуна о њиховој међусобној интеграцији. Сваки од тих система је имао јасно дефинисане границе и најчешће је био дизајниран да решава један изолован пословни проблем. Класична методологија је била примењивана за развој ових “статичких система“. За развој информационих система, који не подразумевају каснију интеграцију, на пример, класична методологија водопада је била апсолутно примењива у њиховом развоју. Она је пружала довољно смерница за планирање, дизајн и реализацију појединачних система. На жалост, овакве традиционалне методологије не предвиђају стратешко планирање, анализу свих пословних процеса, процењивање нових технологија и управљање верзијама система. [17]

За разлику од статичких појединачних система, ВІ системи су динамички и врло тешко се могу развијати „у једном даху“. Функционалности и подаци у ВІ системима се морају анализирати у више итеративних корака, што врло често доводи до нових захтева у развоју следеће верзије система. ВІ системи омогућавају доношење одлука на основу података из свих пословних процеса. Као такви, они се стално унапређују на основу повратних информација од корисника. Имајући ово у виду постојеће методологије развоја информационих система су најчешће неадекватне или неодговарајуће за развој ВІ система.

4.1. Тренутно актуелне методологије развоја ВІ система

У аутору доступној литератури се не може наћи превише методологија које су намењене искључиво развоју ВІ система, већ се за развој често користе методологије, односно, кораци развоја складишта података. Тако је још крајем деведесетих на Војнотехничкој академији израђено је више стручних и научних радова везаних за развој складишта података. Карактеристични радови из тог периода су дипломски рад Тот Ивана из 1999. године под насловом “*Користићење елемената data warehouse-a на примеру годишње анализе ВТА ВЈ на нивоу Смера*” и дипломски рад Димитријевић Ненада из 2001. године под насловом “*Примена елемената data warehouse-a и визуелног програмирања у ВТА ВЈ*“.

Управо је у раду “Примена елемената Data Warehouse–а и визуелног програмирања у ВТА ВЈ” дат предлог интеграције различитих концепата, односно интеграција функционалног и информационог моделовања са објектно-оријентисаним, за развој складишта података. Предложени кораци развоја дат су на следећој слици. [9]



Слика 4.1 – Шематски приказ развоја складишта података

На основу наведених корака у дипломском раду је успешно реализована и клијент сервер апликација за анализу испита путем упита.

Када се ради о развоју BI система, Larissa T. Moss и Shaku Atre су 2003. године дали предлог развоја BI система по следећим корацима (BI Roadmap): [17]

- Оправданост (*Justification*) – процена пословних потреба;
- Планирање (*Planning*) – развој стратешких и тактичких планова који указују на то како ће се пројекат развијати и уводити у систем;
- Анализа пословања (*Business Analysis*) – детаљна анализа пословних процеса како би се разумели пословни захтеви за потенцијалним решењем;
- Пројектовање (*Design*) – пројектовање решења;
- Изградња (*Construction*) – реализација(имплементација) производа;
- Увођење (*Deployment*) – имплементација и продаја завршног производа.



Слика 4.2 – Фазе развоја BI система [17]

Наведени кораци развоја инсистирају на тимском раду, уз инкременталан и итеративна приступ. О свеобухватности приступа, али уједно и о захтевности, говори и чињеница да је за реализацију пројекта BI система потребно реализовати преко 900 појединачних задатака јасно структурираних у и груписаних у поједине фазе развоја. [17]



Слика 4.3 – Кључни кораци развоја BI система према [17]

Наведена методологија је намењена пре свега сложеним пројектима BI система и захтева ангажовање значајних људских ресурса, пре свега са експертским знањима о проблему који се решава. Иако се сама методологија издиже изнад појединачних техника развоја она ипак сугерише њихову примену за реализацију појединих задатака. На основу свега реченог можемо сматрати да се ради о методологији која садржи јасне фазе, процедуре и правила развоја BI система.

Још једну методологију развоја BI система проналазимо 2004. године код компаније ORACLE. Ова методологија инсистира на следећим корацима развоја: [25]

- Идентификовање захтева крајњих корисника;
- Идентификовање извора података;
- Дизајн модела података;
- Израда складишта података;
- Генерисање збирних података;
- Припрема приступа подацима;
- Дистрибуција клијентског софтвера и документације;
- Израда и дистрибуција извештаја.

Наведеном методологијом су дефинисани кораци наизглед независни од решења кроз које ће BI систем бити реализован. Међутим, у суштини наведени кораци су прилагођени управо BI решењима компаније ORACLE и омогућавају развој BI система различите сложености. Чак шта више, може се констатовати да се ради више о предлогу корака развоја BI система него о методологији јер не садржи потпуно дефинисане процедуре и правила развоја, већ се више ослања на технике и алате компаније ORACLE.

Слична ситуација је и са развојем BI система које препоручује компанија Microsoft. Наиме у својој препоруци, како реализовати BI систем, Microsoft говори искључиво о својим алатима за BI, не нудећи начелан приступ независан од техничког решења. У свом приступу Microsoft сугерише следеће кораке развоја [27]:

- Планирање решења и инфраструктуре;
- Дизајн и израда складишта података;
- Дизајн и израда екстракције, трансформације и учитавања изворних података;
- Дизајн и израда аналитичких модела;
- Дизајн и израда извештаја;
- Дизајн и израда извештаја у Excel-у и SharePoint-у;
- Оптимизација решења и планирање даљих операција.

Јасно је дакле да се и у овом случају ради више о корацима развоја него о методологији. Основна предност Microsoft приступа је примена на пројекте мањег обима у ограниченом времену за њихову реализацију[27].

4.2. Предлог развоја VI система

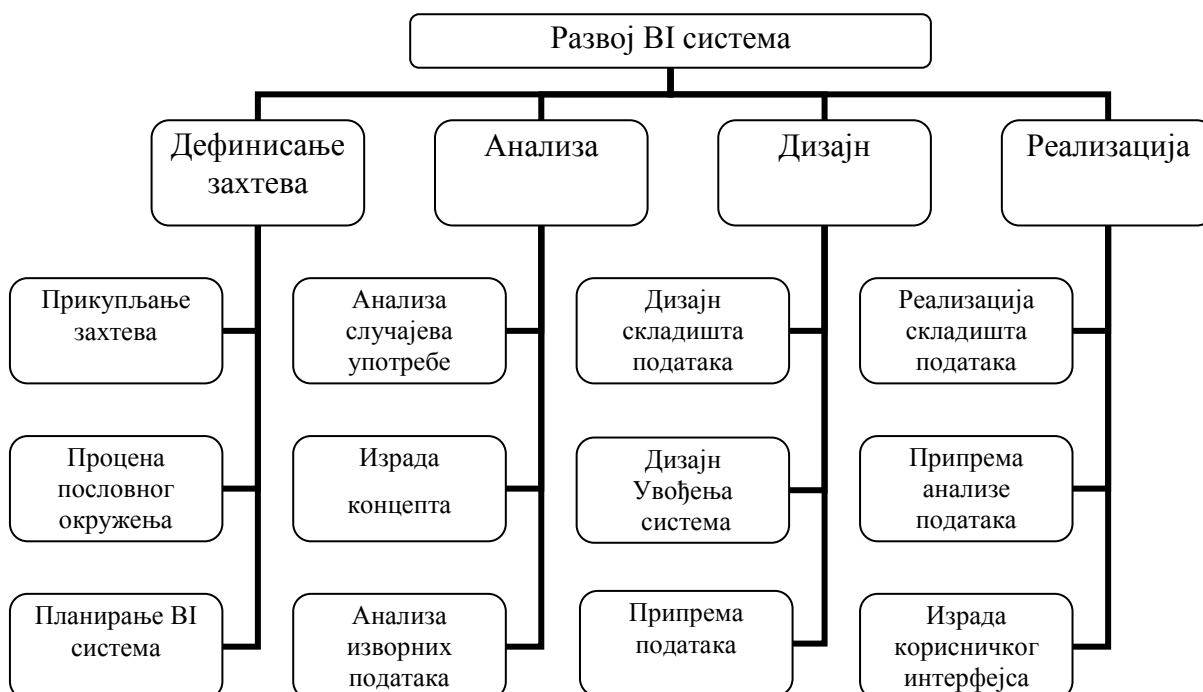
На основу до сада наведеног, може се закључити да методологија развоја VI система мора да буде флексибилна и да омогућава отпочињање развоја у било ком од њених корака, што зависи од тога да ли VI систем већ постоји или се развија нови. Методологија треба да омогући развој VI система кроз више подпројеката који појединачно пролазе све предвиђене фазе. Потребно је предвидети и могућност паралелне реализације појединих фаза – корака и тако додатно повећати ефикасност развојног тима. Такође, методологија ће инсистирати на примени UML-а у моделовању.

Приликом израде методологије одлучено је да се, поред интеграције функционалног и информационог моделовања и објектно оријентисаног приступа, прилагоде и кораци развоја и тако обезбеди методологија примерена пројектима мањег обима, у условима кад су време и остали ресурси ограничени. Ослонац у методологији биће заснован и на претходном искуству, односно максималном искоришћењу знања која су раније акумулирана. Управо ће то бити примењено код интеграције поменутих метода, односно развојном тиму се омогућава да примени или функционални и информациони метод или објектно оријентисани, или њихову комбинацију у зависности од знања са којима аналитичари и програмери располажу.

Методи функционалног и информационог моделовања, анализирани у трећем поглављу, потенцирају две врсте приступа развоју информационих система, и то одозго на доле и одоздо на горе. Објектно оријентисани развој, кроз израду пословних и системских случајева употребе, омогућава приступ одозго на доле, а кроз израду модела концепта и дизајн базе података омогућава реализацију приступа одоздо на горе. Такође, и један и други приступ се заснивају на изради модела, који је централни артефакт развоја. То је посебно истакнуто у развоју вођеним моделом који у свом раду потенцира Д. Милићев[16].

Заједнички садржалац корака раније описаних методологија развоја информационих система и методологија развоја VI система су фазе: дефинисање захтева, анализа, дизајн и имплементација[10], а централни артефакт модел и моделовање. Управо те кораке препознају, и на њима инсистирају, Larissa T. Moss и Shaku Atre [17] и ORACLE [25].

Имајући у виду досадашње излагање, предложићемо развој BI система заснован на темељу раније дефинисаних корака развоја складишта података [9], односно интеграцији стандарда за функционално и информационо моделовање IDEF0 и IDEF1X, објектно оријентисане парадигме засноване на UML-у и димензионог моделовања као основе за израду аналитичких база података. Методологија нема за циљ енкапсулацију примене појединих метода и стандарда, већ напротив њихово паралелно или коришћење у складу са тренутном потребом, односно знањима са којима располаже развојни тим. На следећој слици дата је структура предложене методологије.



Слика 4.4 – Шематски приказ развоја BI система

Као што је већ речено предложена методологија развоја BI система представља синтезу више метода, стандарда и методологија, и састоји се од следећих фаза:[10]

- Дефинисање захтева;
- Анализа;
- Дизајн;
- Реализација.

Дефинисањем захтева треба да сагледамо систем и уочимо потребе корисника. Дефинисањем захтева обавља се и идентификација система. У овом кораку ће се паралелно користити класични методе и објектне примене UML-а.

Анализа треба да омогући, за претходно дефинисане пословне активности, израду предлога решења система и као крајњи циљ има опис појединих функционалности кроз израду дијаграма интеракције. У овом кораку UML ће имати примат.

Дизајн формира коначну структуру будућег система кроз дефинисање модела класа, базе података (применом UML-а) и припрему података за учитавање у будући систем.

Реализација представља последњи корак постављене методологије и представља круну целокупног процеса. Конкретне “последнице“ овог корака су складиште података, упити, OLAP коцке и *Data Mining* алгоритми реализовани кроз конкретне апликације корисничког интерфејса.

4.2.1. Дефинисање захтева

Дефинисање захтева представља почетни процес у циклусу развоја BI система. Дефинисање захтева је уједно и веза између класичног приступа развоју дефинисаног IDEF0 методологијом и објектно оријентисаног приступа дефинисаног RUP методологијом преко дијаграма случајева употребе и дијаграма активности. Дефинисањем захтева обављена је идентификација система. На следећој слици дата је могућа структура Дефинисања захтева.



Слика 4.5 – Шематски приказ Дефинисања захтева

4.2.1.1 Прикупљање захтева

Постоји више техника за прикупљање информација. Неке од њих су и :

- Посматрање;
- Интервју;
- Истраживање.

Посматрање је техника која подразумева директно посматрање како појединац обавља свој посао, у свом радном окружењу. Да би ова техника била ефикаснија особу коју посматрате би требало мотивисати да у току рада објашњава разлоге обављања појединих задатака. Посматрање може бити:

- Пасивно - подразумева посматрање корисника и пасивно слушање свих објашњења везаних за обављање посла;
- Активно – поред посматрања, постављају се и питања помоћу којих вам корисник објашњава догађаје и активности.

У току посматрања пожељно је прикупити и документа и екранске форме тренутног решења. Овом техником се могу увидети и неки елементи тренутног решења као и процеси који изазивају незадовољство код корисника.

Посматрање има и својих ограничења. На пример, с обзиром да се информације прикупљају у току редовног обављања посла, можда нећете имати довољно времена да поставите сва питања како би разјаснили све детаље.

Интервју је техника прикупљања података која се најчешће примењује код процеса у оквиру којих се задаци одвијају релативно брзо, код схватања пословних процеса из угла менаџера и о код упознавања са аспектима пословања који се не могу директно посматрати, због тога што су аутоматизовани. Квалитет прикупљених информација ће зависити од вештине вођења интервјуа, као и од компетентности особе са којом се води интервју.

Увек треба имати на уму да је сврха интервјуа истраживање, а не да оцењивање и критика тренутног стања. Из тог разлога треба користити јасне и концизне реченице, а не давати сопствено мишљење у оквиру питања. Такође, треба избегавати дуга и сложена питања која изискују више од једног типа информација.

Истраживање подразумева прикупљање детаљних и статистичких података. Информације које се могу прикупити техником истраживања су: организациона структура, правила и пракса које олакшавају обављање задатака, посебне потребе за хардвером и софтвером и слично.

4.2.1.2 Процена пословног окружења

Највећи изазов код ВІ пројеката је квалитет извора података. Анализа података се разликује од фазе анализе система у оквиру које се користи традиционална методологија. Активности које се одвијају током анализе података су усмерене ка разумевању и кориговању постојећих неслагања у пословним подацима, независно од било ког метода системског пројектовања или имплементације. Стога, анализа података је пословно, а не системски оријентисана активност.

Процена пословног окружења се може реализовати кроз:

- Прикупљање корисничких (*User-Driven*) захтева;
- Прикупљање изворних (*Source-Driven*) захтева;
- Израду модела пословних случајева употребе.

Прикупљање корисничких захтева је поступак који се базира на дефинисању захтева истраживањем функција које корисник има у својој организацији или којима тежи. Ово је корак који мора да обухвати све пословне процесе. Веома је битно да се утврди ток података/информација из сваке организационе јединице. Ова активност се реализује израдом модела функција, тј. да се посредством IDEF0 методологије изведе функционално моделовање информационог система.

Прикупљање изворних захтева представља апстрактно виђење реалног система, тј. то је поједностављено представљање реалног система преко скупа објеката (ентитета), веза између објеката и атрибута објеката. За прикупљање изворних података користи се методологија одоздо нагоре, односно раније описани IDEF1X стандард.

Развој објектно оријентисаних језика и алата омогућио нам је да процену пословног окружења реализујемо и **израдом пословних случајева употребе**. Овакав приступ као резултат даје општи модел функционисања система заснован на објектној парадигми.

4.2.1.3 Планирање система

У реализацији задатака било које врсте, важна активност је добро планирање те реализације по одређеним корацима. Добро управљање неким пројектом је предуслов реализације самог пројекта. Стихијски приступ реализације пројекта би довео до великих пропуста у реализацији, а вероватно би утицао и на успешан завршетак пројекта. Квалитетно управљање пројектом у смислу планирања обезбеђује добру реализацију проблема, а може да доведе и до смањења трошкова пројекта.

Планирање развоја ВІ система обухвата многе активности, које се јављају и при реализацији било ког другог пројекта. Оно се састоји од следећих активности:

- Креирање пројектног плана,
- Дефинисање ресурса;
- Дефинисање развојног тима.

Креирање пројектног плана

Пројектним планом се врши дефинисање захтева које треба реализовати, временски рокови у оквиру којих те задатке треба реализовати и потребни ресурси. Другачије речено, пројектним планом се одређује шта, до када, ко и уз помоћ којих ресурса треба да уради. Корекције плана треба радити само у критичним случајевима кад је у питању реализација самог пројекта. Треба се трудити да се предвиђени рокови прецизно поштују. Поштовање рокова је основни предуслов за успешну реализацију постављеног циља.

Дефинисање ресурса

Потребно је напоменути да је при планирању за развој ВІ система потребно дефинисати техничке услове. При томе се мисли на техничку инфраструктуру која укључује хардвер, софтвер, DBMS системе, оперативне системе, компоненте мреже, складишта мета података, уређаје и друго.

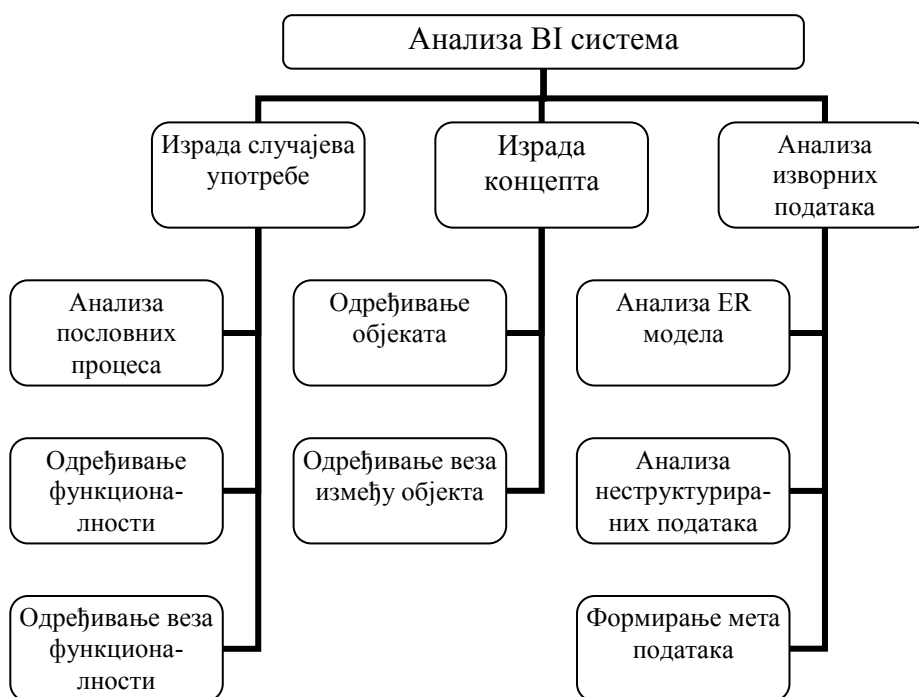
Дефинисање развојног тима

Развојни тим се начелно састоји из сталног и проширеног дела. Стални део тима мора бити доступан 100% времена од почетка до краја развоја система. Оптимална величина сталног тима је 4 или 5 чланова, кога чине: Руководилац развоја, Представник

корисника-послодавца, Аналитичар пословних процеса са ИТ знањима и Техничар из ИТ одељења корисника. Проширени део тима није фиксног бројног стања што зависи од потребе за конкретним знањима у развоју ВІ система. Њега обично чине програмери и администратори са конкретним знањима потребним за поједине фазе пројекта.

4.2.2. Анализа

Анализа дефинише објекте садржане у реалном систему, њихову међусобну сарадњу и у први план ставља истраживање проблема тј. налажење и описивање објеката (концепата) у домену проблема, не дајући одговоре на питање како су решења дефинисана. Овај корак је критична фаза јер је потребно уочити који се све објекти појављују у реалном систему и специфицирати атрибуте и интеракцију између тих објеката. Крајњи циљ анализе је опис појединих функционалности кроз израду дијаграма интеракције. На следећој слици дата је могућа структура Анализе ВІ система.



Слика 4.6 – Шематски приказ Анализе ВІ система

4.2.2.1 Израда случајева употребе

Случајеви употребе описују функције система из перспективе корисника на начин који је њима разумљив. Преко случајева употребе се описује низ догађаја који се дешавају када корисник користи систем да би довршио одређени процес. Сваки случај употребе

треба да се односи на јединствени мерљиви задатак или циљ. Случајеви употребе се састоје од елемената који се налазе унутар система и који су одговорни за функционалност и понашање система.

Модел случајева употребе помаже учесницима на пројекту да се сложе око граница и функционалности система. Веома је важно да се прате захтеви корисника, резултати тих захтева-задатака и могући изузеци. Случајеви коришћења описују учеснике (*Actor*), објекте (*Objects*) и акције (*Actions*) који достижу циљеве у систему.

4.2.2.2 Израда концепта

Након идентификовања захтева корисника и дефинисања визије и граница система, креира се концептуално решење које описује како развојни тим намерава да реализује захтеве корисника. Фокус је на концепте, а не на детаље решења.[16]

Концептуални модел је дакле модел реалног система који представља целокупан његов информациони садржај. Изградња концептуалног модела и спецификација апликација су међусобно условљени. У неким приступима се концептуални модел изграђује директно, на бази пословног модела система, а објекти концептуалног модела служе за спецификацију апликација. У другим приступима се концептуални модел развија на основу спецификације апликација, односно структуре улаза и излаза апликације. Као алат за концептуално моделовање у конвенционалним приступима се најчешће користи модел објекти-везе. Код објектно-оријентисаног приступа, за приказ концептуалног модела користи се дијаграм класа.[16]

4.2.2.3 Анализа изворних података

Један од важнијих циљева код ВІ система је испорука чистих, интегрисаних и усклађених података. Основни извори података за концепт ВІ система су оперативни (транзакциони) тзв. OLTP (*On-Line Transaction Processing*) подаци. Анализа изворних података се сматра кључним елементом и одузима 80% времена, јер је потребно дефинисати одговарајућа правила за преузимање података из изворних података. Овај поступак се врши анализом раније дефинисаних ER модела, али и могућих неструктурираних података.

Анализа података омогућава да се приступи формирању мета података. Мета подаци су подаци о подацима. Они су за складиште података исто што и каталожна картица за библиотеку. Помоћу њих се идентификује садржај и локације података у складишту података. Мета подаци представљају и везу складишта податка и апликације која складиште користи.

Мета подаци треба да садрже листу информација о томе шта се у складишту података налази. Ова листа треба да садржи моделе, димензије, факта, мере коришћене у складишту података.

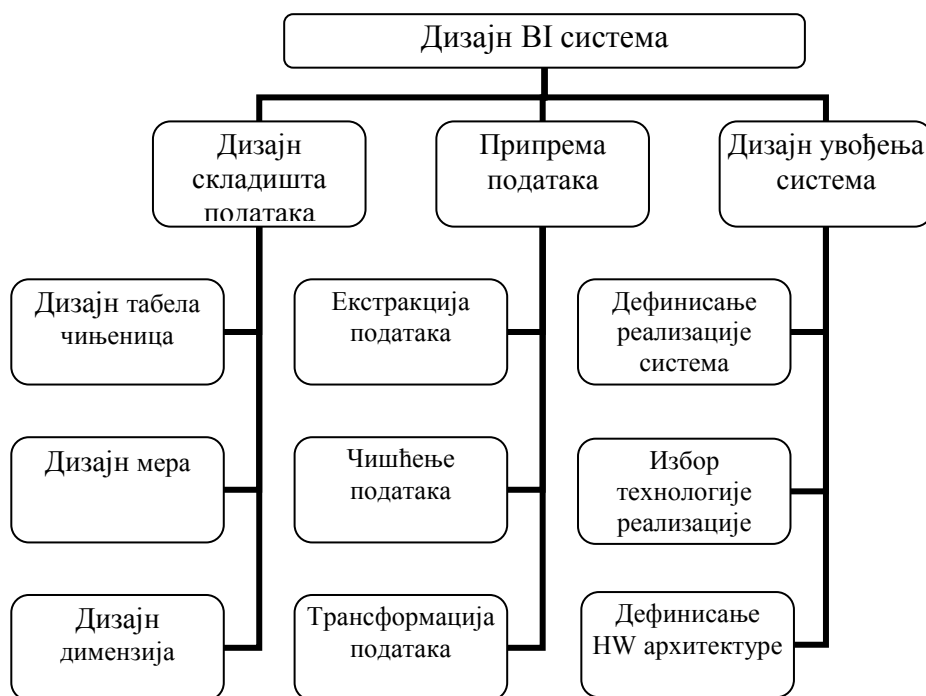
За сваки модел, мета подаци треба да садрже име, дефиницију и сврху модела. Мета подаци о моделу треба да садрже и листе димензија, факата и мера које се користе, а и име особе од које корисници складишта података и апликације могу да добију одређене информације.

Мета подаци о димензијама поред основних података (име, дефиниција...) треба да садрже податке о хијерархијама везаним за ту димензију, правила промене, фреквенцију коришћења, атрибуте, факта и мере везане за ту димензију. Мета подаци о фактима треба поред основних да укључе и податке о фреквенцији коришћења, мерама и димензијама са којим су у вези. Мета подаци о мерама треба да садрже податке о типу податка мере, домену, фактима и димензијама са којима су мере у вези и о правилима извођења мере.

4.2.3. Дизајн

Дизајн је последњи корак моделовања ВІ система. У оквиру дизајна се приказује како ће решење бити имплементирано, идентификују се технологије за развој решења и обезбеђује основа за реализацију.

Дизајн у ствари описује компоненте, сервисе и технологије решења са аспекта развојног тима. Дизајн не подразумева кодирање решења, већ омогућава да се креирају детаљне спецификације компоненти за развој решења као и да се идентификују технологије које се могу користити за развој решења. На следећој слици дата је могућа структура Дизајна ВІ система.



Слика 4.7 – Шематски приказ Дизајна ВІ система

4.2.3.1 Дизајн складишта података

Имајући у виду карактеристике складишта података дизајн у ствари подразумева израду димензионог модела. Димензионо моделовање је техника чији је циљ презентација података у облику који обезбеђује високе перформансе система ради вршења анализе података. Производ ових активности је дакле димензиони модел. Основу за израду димензионог модела представљају раније дефинисани мета подаци.

У димензионом моделовању структуре података су тако организоване да описују мере и димензије. Мере су нумерички подаци смештени у централној, такозваној, табели чињеница. Димензије су стандардни пословни параметри који дефинишу сваку трансакцију. Димензије се смештају у табеле непосредно, или преко друге табеле димензије, повезане са табелом чињеница.

4.2.3.2 Припрема података

Припрема података се врши на основу раније одређеног извора података, извршене анализе података и захтева корисника. Припрема се врши одређеним екстракционо/ трансформационим алатима кроз екстракцију, чишћење, и трансформацију података. Резултат ових активности треба да буду подаци који ће нам омогућити израду ETL сервиса који ће омогућити учитавање података у складиште.

Екстракција података

Подаци који ће се користити у складишту података морају се екстраховати из операционих система (база података у оквиру неког система) који садрже те податке. Подаци се иницијално екстрахују у процесу креирања складишта података, а касније се на основу одређених процедура врши додавање нових података у складиште података. Екстракција података је врло једноставна операција ако се потребни подаци налазе у једној релационој бази, али може да буде и веома комплексна операција ако су подаци смештени у вишеструким хетерогеним операционим системима. Циљ процеса екстракције података је да све потребне податке, у погодном и конзистентном формату, припреми за читавање у складиште података.

Пре процеса екстракције требало би проверити да ли у бази података из које вршимо екстракцију нема логичких грешака (нпр. у табели Кадет нема кадета који се налази у табели положених испита). Овакве грешке би пре екстракције требало уклонити коришћењем процедура за проверу грешака.

Постоји могућност да се не може утврдити евентуално постојање логичких грешака. То се дешава у ситуацијама када се екстракција врши из више извора података. Приликом екстракције из више извора података може се јавити и проблем неконзистентности података услед различитог означавања истих појмова (називи држава се могу скраћено означавати са три или са два симбола).

Чишћење података

Због проблема који се приликом екстракције података јављају, подаци добијени екстракцијом се морају „очистити“. Чишћење података подразумева: проверу постојања логичких грешака, „побољшање“ података, елиминисање осталих грешака. Провера логичких грешака укључује:

- Проверу вредности атрибута;
- Проверу атрибута у контексту осталих података у реду;
- Провера атрибута у контексту редова друге табеле која је повезана;
- Провера веза између редова исте или повезаних табела (провера пренесених кључева).

„Побољшање“ података је процес чишћења којим се тежи да подаци добију пуно значење. Пример за ово су подаци о именима и адресама. Често су ти подаци смештени на више места у бази и временом постају несинхронизовани. Овим процесом се тежи да се такве ситуације разреше.

Трансформација података

Трансформација података је критичан корак у развоју складишта података. У оквиру процеса трансформације врши се последња припрема података пре учитавања. Процес трансформације може да се уради и непосредно пре учитавања података у складиште коришћењем ETL сервиса. Типична трансформација података укључује:

- Превођење поља са више имена у једно поље;
- “Разбијање“ поља са датумом у посебна поља за годину, месец и дан;
- Превођење поља са једном репрезентацијом у другу (нпр. са 1 и 0 у ДА и НЕ);
- Креирање и додавање кључева за табеле димензија.

По завршеној трансформацији постоје сви услови да се приступи генерисању мета података.

4.2.3.3 Дизајн реализације система

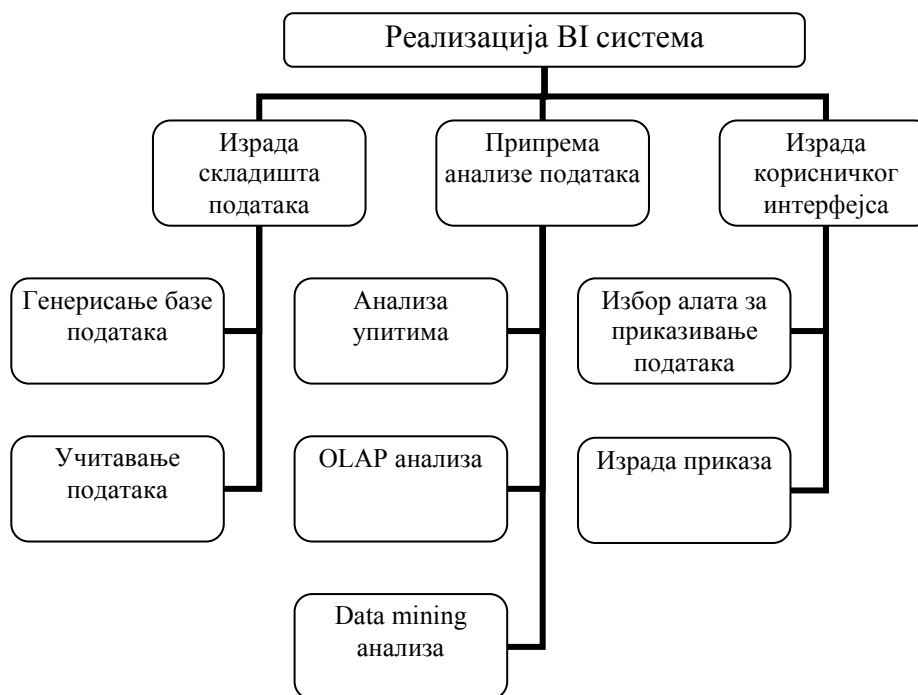
Модел реализације система (*Deployment Model*) је дијаграм који придружује апликације и њихове сервисе на топологији сервера. Сврха модела је да омогући развојном тиму и тиму пуштања решења у рад да пројектују и планирају топологију сервера и њихову конфигурацију.

Ово је последњи корак у оквиру активности дизајна, који подразумева дефинисање структуре хардверских компоненти које треба да омогуће практичну реализацију система. Ова структура се у UML нотацији описује дијаграмом размештаја (*Deployment Diagram*). Дизајн увођења система се реализује кроз:

- Дефинисање реализације система;
- Избор технологије реализације;
- Дефинисање хардверске архитектуре.

4.2.4. Реализација

У оквиру реализације врши се пресликавање модела, дефинисаних у оквиру дизајна, у аналитичку базу података, припрема за анализу података и израда корисничког интерфејса. На следећој слици дата је могућа структура Реализације BI система.



Слика 4.8 – Шематски приказ Реализације BI система

4.2.4.1 Израда складишта података

Израда складишта података представља активност која је логичан наставак процеса дизајна складишта података. У оквиру ове активности треба генерисати складиште података и извршити учитавање података.

Активност генерисања складишта података се врши коришћењем SQL језика. Наиме, алат у коме се врши креирање физичког модела омогућава непосредно генерисање складишта или преко генерисања SQL кода преко такозваних DDL (*Data Definition Language*) датотека. Након генерисања базе-складишта података може се приступити учитавању података. У току учитавања могу се евентуално извршити још неке трансформације, мада би са трансформацијама података требало завршити пре учитавања због проблема конзистентности базе. Поступак се врши у следећим корацима:

- Избор базе података која представља извор података;

- Одређивање складишта података у које учитавамо;
- Избор онога што се жели копирати (табеле са подацима или се прави упит за копирање);
- Избор табела из којих се учитавају подаци и у које се учитавају подаци;
- Покретање учитавања.

4.2.4.2 Припрема анализе података

Све што је урађено у оквиру процеса Дефинисања захтева, Анализе и Дизајна треба да омогући анализу података. Велики значај анализе утицао је и на развој различитих техника за анализу података. Неке од тих техника су упити и извештаји генерисани упитима, вишедимензиона-OLAP анализа и *Data Mining*. Ове технике се користе како би се приказали резултати разних упита, анализирали подаци из различитих углова и откривали шаблони и правила међу постојећим подацима.

Анализа упитима је процес који обухвата постављање одређеног питања, добијања одговарајућих података из складишта података на основу тог питања, трансформације тих података у одговарајући облик и њихово приказивање у захтеваном формату. За ову технику се каже да је „вођена аналитичарем“ (*Driven by Analyst*), што значи да одговор директно зависи од упита (питања) аналитичара.

Анализа упитима се изводи кроз следеће кораке:

1. Дефинисање упита – превођење питања и хипотеза из реалног система у формат упита који се могу користити за подршку у одлучивању;
2. Приступ подацима и добијање података – добијање тражених података, на основу SQL команди за дефинисане упите;
3. Извођење прорачуна над подацима – надградња претходно дефинисаних упита одговарајућим операцијама над подацима и поновно извршавање упита;
4. Припрема извештаја – на основу упита генерисање извештаја;
5. Приказивање извештаја – презентација података у форми извештаја крајњем кориснику на начин који му одговара, преко екранских форми или штампањем на папиру.

Вишедимензиона - OLAP анализа је популаран начин проширења могућности упита. Овакав начин анализе замењује анализу преко великог броја упита. Вишедимензиона анализа омогућава кориснику да врши анализу по великом броју независних елемената, који чине систем који се анализира, и да прегледају податке који су међусобно доста комплексно повезани. Крајњи корисници нису увек заинтересовани за исти ниво детаља података, тако да је вишедимензионом анализом омогућено да се ниво детаља динамички мења. Корисник, на пример, може да гледа податке који се односе на цео смер, класу или на конкретног кадета у зависности од потреба анализе.

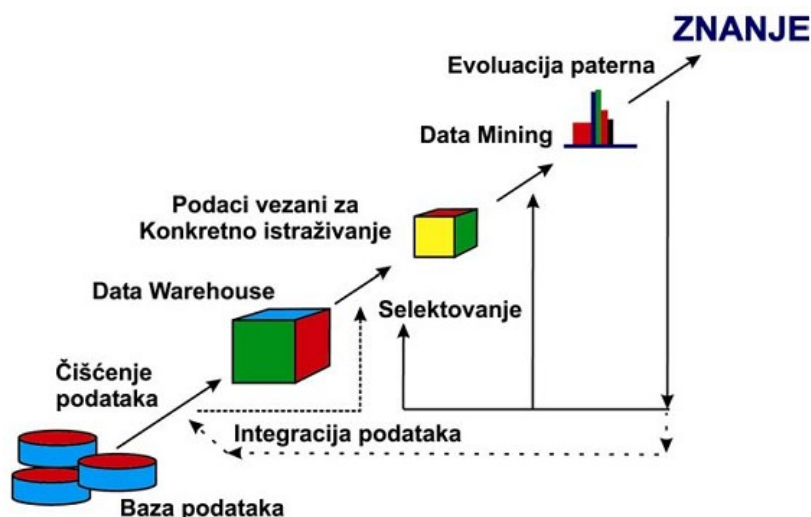
Најпогоднији начин за вршење вишедимензионе анализе је примена OLAP-а (*Online Analytical Processing*). OLAP представља начин организовања великих база података. Подаци се организују на начин који је најпогоднији за анализу од стране корисника. Коришћењем OLAP-а подаци се могу организовати по нивоима детаља, разврставањем у одређене категорије (креирање хијерархије димензија).

Први корак у креирању OLAP коцке над релационом базом је одређивање извора података и креирање упита над тим подацима за добијање елемената коцке. Следећи корак је креирање саме коцке.

Data Mining је релативно нова техника за анализу података и веома је различита од упита и OLAP-а. Каже се да је то техника за откривање података (вођена подацима), односно шаблона и правилности које се међу подацима појављују. Ова техника се изводи применом специфичних алгоритама за анализу података.

Data Mining је дуготрајан и сложен процес који се састоји од седам основних фаза [15]:

- Пречишћавање података;
- Интеграција података;
- Избор података;
- Трансформација података;
- Data Mining;
- Евалуација узора;
- Презентација знања.



Слика 4.9 – Процес Data mining-a [15]

Веома често базе података садрже и податке који нису тачни, потпуни, прецизни или не поседују довољан ниво конзистентности да би били употребљени у сврхе анализе. Прва фаза у процесу *Data Mining* има улогу да открије такве податке и елиминише их из даље анализе. Није редак случај да се користе више различитих система за управљање базама података. У том случају посебан проблем се јавља приликом интеграције података садржаних у различитим системима. Потребно је веома много пажње и искуства да би се ови подаци представили у одговарајућем облику. У фази избора података издвајају се “атомски подаци” које садрже информације релевантне за посматрану појаву. У фази трансформације изабрани подаци се превode у најпогоднији облик за примену *Data Mining* анализа. Када су подаци трансформисани, примењују се *Data Mining* технике које генеришу узор, који се касније процењују и тако се долази до драгоцених сазнања о посматраној појави.

4.2.4.3 Израда корисничког интерфејса

За већину корисника најважнији део је управо кориснички интерфејс који за њих представља саму апликацију. Добро дизајниран кориснички интерфејс у највећем броју случајева, обезбеђује успех и прихватљивост пословне апликације. Пре отпочињања израде требало би се одредити за алат којим ће се приступати аналитичким подацима. [11]

Да би интерфејс био ефикасан треба размотрити следеће карактеристике:

- Интуитиван дизајн –тако да корисници интуитивно могу да га разумеју;
- Оптимално коришћење простора екрана – планирати количину информација која ће бити приказана на једном екрану;
- Одговарајући изглед - обратити пажњу на боје, на пример за поруке грешке користити црвену;
- Лакоћа навигације – за прелазак на поља користити Tab, стрелице и пречице;
- Контролисана навигација – одржати редослед приступања компонентама.

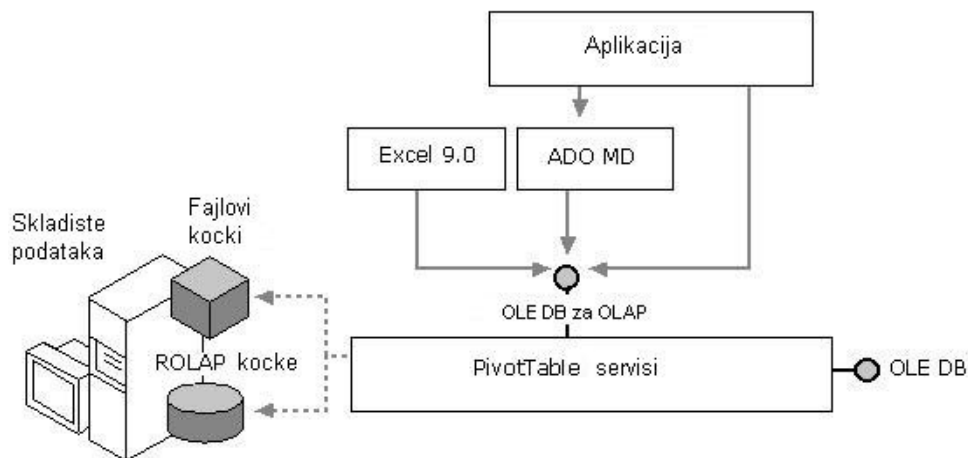
Након што се одлучите за дизајн корисничког интерфејса, почињете да креирате њихов прототип. Последњи корак је њихова валидација у односу на захтеве, анализу и дизајн система.

Кориснички интерфејс трансакционе базе података везан је за дефинисање менија, екранских форми извештаја. Екранске форме су основни тип објеката у већини СУБП и треба да омогуће кориснику представљање података из базе и унос података у базу. Форме у себи могу имати велики број других објеката (контрола).

Кориснички интерфејс OLAP анализе треба да омогући једноставно добијање графичких излаза на основу којих ће руководство доносити пословне одлуке, а све то на основу раније формиране коцке и спреге са Pivot табелама Excelu.

На следећој слици су приказана два начина приступа подацима у OLAP коцкама:

- коришћењем Microsoft Excela-a или
- израдом посебне апликације, применом такозваних ADO механизма. [8,9]



Слика 4.10 – Приступ OLAP коцкама [8,9]

Кориснички интерфејс дела система који се односи на *Data Mining* треба да омогући једноставно прелажење из једног алгоритма у други и упоређивање добијених резултата, при том користећи све предности уграђени елемената у изабрани алат за управљање системом.

4.3. Оцена предложене методологије развоја VI система

Постоји неколико дефиниција речи методологија, многи аутори чак не праве разлику између методологије и појединачних метода развоја информационих система. Оно што је заједничка дефиниција за већину аутора је да се под методологијом подразумевају фазе, процедуре, правила, технике и алати за развој информационих система.

Као такву сматрам да методологију треба да чине: структура која представља водич о томе шта и кад треба урадити и технике које омогућавају да се уради оно што је потребно. Такође, примена формализоване методологије у развоју информационих система би требала да нам обезбеди структурираност појединачних задатака, тимски рад, контролу пројекта, контролу људских ресурса и да неизвесност смањи на најмању могућу меру.

Избор модела и методологије развоја свих информационих система, па самим тим и VI система, зависи од [3]:

- обима пројекта;
- природе проблема, односно расположивог знања о проблему;
- расположивог времена;
- расположивих ресурса.

Када се говори о обиму пројекта већина постојећих методологија је намењена пре свега развоју комплетног VI система великих организација. Као такве, те методологије су тешко примењиве на пројекте мањег обима, када се применом VI парадигме жели решити један конкретан проблем. Последица је да решавање тог проблема траје исувише дуго уз трошење значајних ресурса, што често доводи до неуспеха.

Такође, природа проблема значајно утиче на избор методологије. Наиме, није све једно да ли су за прикупљање захтева и њихову анализу потребна додатна специфична знања. Имајући у виду да су пројекти развоја система временски ограничени, то значајно може да утиче на могућност реализације свих корака постојећих сложених методологија развоја ВІ система.

Као што је наведено, управо је расположиво време значајан фактор приликом избора методологије, односно приликом доношења одлуке о њеном квалитету и могућности примене. Постојеће методологије инсистирају на великом броју фаза и корака и самим тим је њихова примена ограничена у ситуацијама када нам време није савезник.

Расположиви ресурси су такође значајан фактор који утиче на методологију развоја. Када говоримо о ресурсима мисли се пре свега на људе и њихова знања, доступне софтверске алате и рачунарску опрему. Кад се ради о људима подједнако су битна знања информатичара-програмера и наручиоца-будућег корисника. Постојеће методологије инсистирају на постојању великог броја уско стручних лица што није увек могуће обезбедити или би значајно утицало на цену пројекта.

Како би проверили ваљаност предложене методологије за потребе развоја ВІ система извршена је анализа по следећим факторима:

- Примена на ВІ систем мањег обима;
- Примена када постоји значајно знање о предмету ВІ система;
- Примена у ограниченом времену за реализацију пројекта;
- Утицај расположивих ресурса;
- Јасна структура корака и технике реализације.

Ради јасније процене резултата анализе одлучено је да се испуњеност наведених фактора оцењује оценама „1“ (минималан позитиван или максималан негативан утицај), „3“ (постоји утицај али није пресудан) и „5“ (минималан негативан или максималан позитиван утицај).[28]

Све описане методологије су засноване на принципима инкременталности и итеративности и омогућују да се конкретни кораци у реализацији прилагођавају у зависности од обима пројекта. Ова констатација се донекле не односи на *BI Roadmap* јер предвиђа значајан број задатака који воде до коначног циља. Такође, из истих разлога постојање претходног знања о предмету *BI* система позитивно утиче на примену методологије. Можемо закључити да *BI Roadmap* заслужује оцену „3“ за примену у пројектима мањег обима, а све методологије заслужују највишу оцену по оба критеријума.

Када се говори о времену потребном за реализацију пројекта *Microsoft* је својим алатима омогућио да се пројекат реализује довољно брзо. Остале методологије су зависне од времена али не пресудно управо због примењених објектно оријентисаних принципа, односно инкременталности и итеративности. Можемо закључити да *Microsoft* заслужује максималну оцену, а остале методологије оцену „3“.

Утицај расположивих ресурса на примену наведених методологија такође није пресудан иако пре свега људски ресурси представљају значајан дефицит за сваки пројекат. Међутим, како *ORACLE* решења, на којима се заснива њихова методологија, још увек имају изузетно високу цену можемо закључити да *ORACLE* заслужује минималну а остале методологије оцену „3“.

Како *Microsoft*, за разлику од осталих, за развој *BI* системима не инсистира на методологији већ на алатима можемо закључити да код *Microsoft*-а постоји јасна структура технике реализације али не и корака реализације. Можемо закључити да *Microsoft* заслужује оцену „3“, а остале методологије највишу оцену.

Табела 4.1. Поређење методологија развоја BI система.

Критеријум	<i>BI Roadmap</i>	<i>ORACLE</i>	<i>Microsoft</i>	Предложена методологија
Примена на <i>BI</i> систем мањег обима	3	5	5	5
Примена када постоји значајно знање о предмету <i>BI</i> система	5	5	5	5

Критеријум	BI Roadmap	ORACLE	Microsoft	Предложена методологија
Примена у ограниченом времену за реализацију пројекта	3	3	5	3
Утицај расположивих ресурса	3	1	3	3
Јасна структура корака и технике реализације	5	5	3	5
ЗБИР ОЦЕНА	19	19	21	21

Имајући у виду чињеницу да су методологије са којима је вршено поређење већ доказане у пракси и да су по њима реализовани многи пројекти можемо закључити и да је предложена методологија ваљана за развој BI система и да представља и најбољи избор за реализацију пројекта мањег обима када већ постоји искуство везано за предмет BI система.

5. Business Intelligence систем за управљање успехом кадета Војне академије

Методологија развоја BI система, предложена у претходном поглављу, има за намеру да покаже да је могуће унапредити и поједноставити начин интеграције постојећих метода развоја информационих система и тако омогући израда BI система помоћу кога се могу добити корисне информације за управљање пословним процесима. Као пример за доказ ваљаности методологије, одлучено је да се реализује BI систем за управљање успехом кадета Војне академије. Како фокус није управљање целим пословним системом ВА већ анализом успеха студената, предложена методологија највише одговара оваквом обиму пројекта. Предложена методологија ће обезбедити компромис између чињеница да постоји и раније искуство у овој области и да су остали ресурси ограничени.

5.1. Дефинисање захтева

Дефинисање захтева представља почетни корак у циклусу развоја BI система. Сам поступак дефинисања захтева је итеративан и инкременталан и одвија се у сталној сарадњи између пројектанта и корисника. Итеративан је јер се одвија у више циклуса подједнаке структуре, а инкременталан је јер се свака итерација ослања на производе претходне, додавајући нове функционалности.

5.1.1. Прикупљање захтева

Прикупљање захтева представља битан моменат у развоју BI система. Ова активност за пројектанта представља информисање и учење, односно упознавање са потребама и жељама корисника, како би донео праве закључке. Као што је раније дефинисано, прикупљање захтева се може реализовати посматрањем, интервјуом и истраживањем. У нашем случају смо применили све три технике.

У сврху прикупљања корисничких захтева дефинисана су питања за интервју са менаџментом Војне академије. Испред Начелника и Декана Војне академије одговоре на питања је формулисао пк Ненад Димитријевић. Добијени су следећи одговори и ставови:

- Да ли се на ВА управља успехом кадета?

Да наравно, посебно у оквиру поступка евалуације.

- Како се одвија процес управљања успехом кадета ВА?

На основу Закона о Високом образовању и закључака савета за Високо образовање се реализују анализе успеха на основу којих се предлажу мере за унапређење квалитета образовања на ВА.

- На основу којих информација се тренутно управља успехом кадета ВА?

Податке о успеху кадета води кадетска служба и то су подаци из испитне пријаве и записника о полагању испита. Такође, посебно се анализира из ког предмета је највећи број не положених испита а из ког су сви кадети положили. Пролазност се рачуна у процентима, а анализира се и број освојених ЕСБП бодова. Сви извештаји се раде тако да дају преглед података по класи и студијском програму збирно за цели академију.

- Које су тренутно мере (просек, број положених испита и слично) за оцену успешности кадета?

Да управо су то мере на основу којих се цени успешност кадета. Анализе се раде поређењем броја пријављених, изашлих и кадета који су положили одређени испит.

- Да ли постоји потреба за коришћењем додатних информација у процесу управљања успехом кадета ВА?

Да наравно, свака додатна информација која ће унапредити процес анализе је добродошла.

- Како се подаци о испитима прикупљају, обрађују и презентују доносиоцима одлуке?

Податке о успеху кадета води кадетска служба. До сада су се подаци водили у Excel базама података, а тренутно се у функцију уводи нови информациони систем.

- Која лица-органи ВА учествују у овом процесу?

У овом процесу учествују кадетска служба, секретаријат и деканат ВА.

- Који ниво детаља информација је потребан руководству ВА за доношење одлука у овом процесу?

У зависности од нивоа одлучивања треба да буде и ниво података. Руководству свакако иду сумарни подаци, а не појединачни о сваком кадету понаособ.

- Колико често се врши анализа успешности студирања кадета ВА?

Анализа се врши након сваког испитног рока, школске године и по потреби.

У оквиру прикупљања захтева, посматрано је и како референт Одсека за питања кадета и слушалаца врши анализу испита и успеха и уједно је са њима обављен и интервју. Интервју сам извршио са начелником Одсека за питања кадета и слушалаца, потпуковником Домазет Урошем, и са референтом у одсеку госпођом Војиновић Босом. Од њих сам сазнао да је процес анализе испита и успеха сложен и да захтева доста времена. Сазнао сам по којим елементима се сама анализа врши, који су подаци неопходни за анализу и ниво детаља тих података. Такође, сазнао сам да се у наредном периоду очекује увођење аутоматизованог система за вођење података о испитима и кадетима.

Подаци о испитима кадета се тренутно уносе и чувају у Excel датотекама, по семестру, класама и студијским програмима. Изглед табеле је дат на наредној слици. Као што се може видети за сваког кадета се воде подаци за појединачни предмет. Кадет има могућност да испит полаже три пута (1,2,3) редовно или пред комисијом (К). У табели се води и просек, број не положених испита (број слабих), укупан просек и број освојених ЕСПБ бодова. Анализа успеха се врши по свим елементима из табеле.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X				
1	ЈЕДИНИЦА	СТУДИЈСКИ ПРОГРАМ	КЛАСА	Ред. бр.	ПРЕЗИМЕ И ИМЕ СТУДЕНТА	Предмети												ПРОСЕК СТУДЕНАТА	БРОЈ СЛАБИХ	УКУПАН ПРОСЕК	Број ЕСПБ бодова							
2						Предмет1				Предмет2				Предмет3														
3						1	2	3	К	1	2	3	К	1	2	3	К					1	2	3	К			
4						1	2	3	К	1	2	3	К	1	2	3	К					1	2	3	К			
5	1. КАДЕТСКА БРИГАДА	Назив	53. информатика	1																								
6				2																								
7				3																								
8				4																								
9				5																								
10				6																								
11				7																								
12				8																								
13				9																								
14				10																								
15	ПРОСЕЧНА ОЦЕНА																											

Слика 5.1 – Изглед табеле Преглед полагања испита

Паралелно са претходним табелама подаци се уносе у матичну књигу, како би се израчунао општи успех и припремили подаци за штампање дипломе. У оквиру матичне књиге се воде следећи подаци:

- број индекса
- презиме
- име оца
- име
- класа
- смер
- ЈМБГ
- евиденциони број
- у војној школи од
- датум рођења
- место рођења
- општина рођења
- година коју је понављао
- држава рођења
- држављанство
- националност
- социјално порекло
- општина упућивања у војну школу
- врста средње школе и седиште
- успех постигнут у средњој школи
- школска година у којој је завршио претходно образовање
- просечна оцена
- општи успех
- разлог отпуштања
- датум отпуштања
- број наредбе
- отпуштен у школској години
- датум дипломирања

- датум почетка школовања
- датум завршетка школовања
- број дипломе.

Поред наведених података током интервјуа сам сазнао како се прати процес селекције кандидата за упис на Војну академију (Акт СЈР инт бр 90-84. од 12.02.2010. године). На слици 5.2 је приказан изглед табеле Селекција.

О кандидату се воде идентификациони подаци; жеље у вези студијског програма (смера); завршена школа и успех у средњој школи; резултат систематског прегледа; резултати психолошког тестирања (оцене од 1-5, кандидат задовољава само са оценама 4 и 5), математике и физичке провере; и сума свих освојених поена. Резултат систематског прегледа и психолошког тестирања су елиминационе природе. У суму бодова улази успех из средње школе 40%, резултат математике 45% и резултат физичке провере 15%. На основу суме се прави ранг листа и одлучује о упису.

Редни број	Група	Презиме	Име оца	Име	Избор смера			Завршена школа	I разред	II разред	III разред	IV разред	Σ Средња школа
					I	II	Остало						
1,													0,00
2,													0,00
3,													0,00
4,													0,00
5,													0,00
6,													0,00
7,													0,00
8,													0,00
9,													0,00
10,													0,00

Редни број	Група	Презиме	Име оца	Име	Психолошко тестирање					Систематски	Број бодова из математике			Поени ФВ	Бодови ФВ	Σ
					Г	Е	НМ	СЗ	П		Шифра	Математика	Шифра			
1,															0,00	0,00
2,															0,00	0,00
3,															0,00	0,00
4,															0,00	0,00
5,															0,00	0,00
6,															0,00	0,00
7,															0,00	0,00
8,															0,00	0,00
9,															0,00	0,00
10,															0,00	0,00

Слика 5.2 – Изглед табеле Селекција

У току рада на дисертације дошло је и до увођења информационог система на Универзитету одбране. Информациони систем Универзитета одбране (ИСУО) је систем за праћење, администрирање и организовање наставе и наставних процеса на високошколским установама и развијен је на Open Source технологијама последње генерације. Основне функционалности ИС Универзитета одбране су¹:

- подршка организовању и извођењу наставе на основним академским, дипломским академским (мастер), специјалистичким академским, докторским академским и интегрисаним академским студијама
- подршка организовању и извођењу наставе на свим облицима студијама
- праћење и евидентирање комплетног досијеа студената
- планирање плаћања школарина (могућност евидентирања плаћања на рате) и забрана пријаве испита и осталих акција
- подршка организовању испитних рокова, условљавања пријаве испита уз могућност пријаве испита преко Интернета и електронског плаћања испита
- подршка студијским програмима по Болоњском процесу и комплетно имплементираним одредбе новог закона о високом образовању
- подршка старим наставним плановима и програмима, као и прелазним режимима
- студентска анкета и могућност забране пријаве испита или уписа наредне године у случају непопуњене анкете
- штампање диплома, додатака дипломи као и осталих уверења и извештаја за студенте
- штампање и израда статистика уписа, успеха студената, пролазности на испитима као и осталих статистика непоходних за управу факултета, универзитет или Министарство просвете
- студентски стандард (пријаве за студентске домове, кредите, стипендије и ГСП повластице)

У оквиру овог система воде се подаци о кадетима приказани на следећој слици. Ови подаци одговарају онима који су се водили до увођења новог система.

¹ за анализу је коришћен јавно доступни материјал [26] који потпуно одговара ИСУО

Слика 5.3 – Изглед форме Студент [26]

Подаци са претходних школовања који се воде у систему су приказани на наредној слици.

Г. з.	Т. с.	Установа
	С	Виша архитектонска школа

Слика 5.4 – Подаци са претходног школовања [26]

Када је реч о резултатима са испита, они се уносе са добијених записника и пријава и могу се видети на следећој слици.

Матични подаци	Резултати положених испита	Резултати неположених испита
Уписи	+	
Предрачун школарине		
Уплата		
Прати		
Преглед пријава		
Забрана пријаве		
Резултати испита		
Преглед обавеза		
Списак		
Обавештења		
Кратак преглед		
Завршни рад		
Диплома		
Уверења		
Молбе		
Ментор		
Еквивалентирање		

#	Предмет	Рок	О.	Наставник	Е.	Б.	Дат. пол.
1	001ЛФ - Лабораторијске вежбе из Физике	фебруар - 2008/09	10	Предраг Осмокровић	2		26.02.2009.
2	001Ф1 - Физика 1	септембар - 2008/09	6	Јован Цветић	5		25.08.2009.
3	001ПКР - Практикум из коришћења рачунара	јануар - 2008/09	9	Бошко Николић	2		03.02.2009.
4	001ЕЈ1 - Енглески језик 1	јануар - 2008/09	10	Милош Ђурић	2		30.01.2009.
5	001ЛОЕ - Лабораторијске вежбе из Основа електротехнике	септембар - 2008/09	7	Дејан Тошић	2		08.09.2009.
6	001ЕЈ2 - Енглески језик 2	јун - 2008/09	10	Милош Ђурић	2		30.06.2009.
7	001УМ - Увод у менаџмент	јун - 2008/09	10	Дарко Надић	2		18.06.2009.

Слика 5.5 – Изглед форме Резултати положених испита [26]

За потребе анализе успешности студирања из наведеног информационог система се могу добити извештаји по следећем:

- назив предмета;
- број пријава;
- број изашлих;
- излазност у процентима;
- број положили;
- број положили у процентима;
- структура оцена;
- просек
- број ЕСБП бодова.

У суштини ради се о стандардним статичким извештајима по унапред дефинисаним параметрима који се не могу динамички мењати. Проблем који постоји је и тај што нови систем не омогућава аутоматску анализу по класама односно модулима.

Истраживање сам извршио тако што сам прикупио све јавно доступне податке који се односе на надлежности и задатке Војне академије. У даљем тексту су дати је преглед прикупљених података.

Војна академија је високошколска јединица са својством правног лица која изводи акредитоване студијске програме у пољу друштвено-хуманистичких и техничко-технолошких наука.[24]

Мисија Војне академије је: “Системом војног образовања изградити идентитет професионалног официра, часног, оданог, оспособљеног и спремног за интелектуалне и етичке изазове официрског позива у служби отаџбине, Републике Србије.”[24]

Циљ школовања на Војној академији је образовање и оспособљавање кадета и официра за одговарајуће дужности у роду-служби (у складу са мисијама и задацима Војске) и стицање знања неопходних за даље усавршавање.[24]

Војна академија је образовно-научна установа на којој се: [24]

- стиче високо образовање на основним студијама;
- организују последипломске студије;
- организује последипломско стручно усавршавање;
- школују слушаоци за резервне официре и
- организују основна, примењена и развојна истраживања значајна за подизање квалитета високошколске наставе, оспособљавање научног и наставног кадра и увођење кадета у научни рад.



Слика 5.6 – Шема организације Војне академије [24]

Секретаријат обавља следеће послове и задатке:[24]

- командовање, планирање ангажовања, реализација задатака, изградња функционалне способности;
- остварује кадровску, мобилизацијску и правну функцију;

- обједињује командну функцију ради одржавања потребног нивоа оперативних и функционалних способности и ефикасног руковођења и командовања;
- обезбеђује функционисање система логистичке подршке ВА;
- спроводи краткорочно и дугорочно планирање опремања и набавки, реализацију набавки и пројектовање потреба развоја и плана расхода и прихода;
- реализује планирање, организовање, координацију и контролу система телекомуникација, информатике, криптозаштите информација, канцеларијског пословања, архивирање документације и војнопоштанског саобраћаја у ВА;
- обезбеђује развој, планирање наставе и управљање системом обуке у Војној академији, послове из надлежности статуса кадета и слушалаца и реализацију подршке у наставној материјалној бази;
- спроводи стручно и оперативно обављање послова и задатака из области цивилно- војних односа, међународне војне сарадње и морала;
- спроводи финансијско управљање и контролу над управљањем и коришћењем средстава, материјално пословање и контролу материјалног и финансијског пословања и
- реализује библиотечку делатност за потребе система одбране.

Деканат је организациона целина Војне академије одговорна за организацију, реализацију и контролу наставног процеса на свим нивоима школовања и усавршавања, развој студијских програма и планирање као и реализацију и праћење научно-истраживачке делатности. [24]

Управа за кадрове Министарства одбране расписује конкурс за пријем кандидата из грађанства за кадете Војне академије. Кандидати треба да испуњавају опште и посебне услове конкурса. Општи услови конкурса су: [24]

- да су држављани Републике Србије;
- да су здравствено способни за школовање, што утврђује надлежна војно-лекарска комисија;
- да се против њих не води кривични поступак или поступак због кривичног дела за које се гони по службеној дужности, односно да нису осуђивани за таква дела казном затвора у трајању дужем од шест месеци или казном малолетничког затвора као и
- да нису у брачној или ванбрачној заједници.

Посебни услови конкурса су: [24]

- да су рођени 1991. године или касније;
- да су завршили четворогодишњу средњу школу или да похађају IV разред.

Кандидати који испуњавају опште и посебне услове конкурса за пријем за кадете Војне академије полажу пријемни испит из области математике, подлежу провери физичке способности, психолошкој и медицинско-здравственој процени.

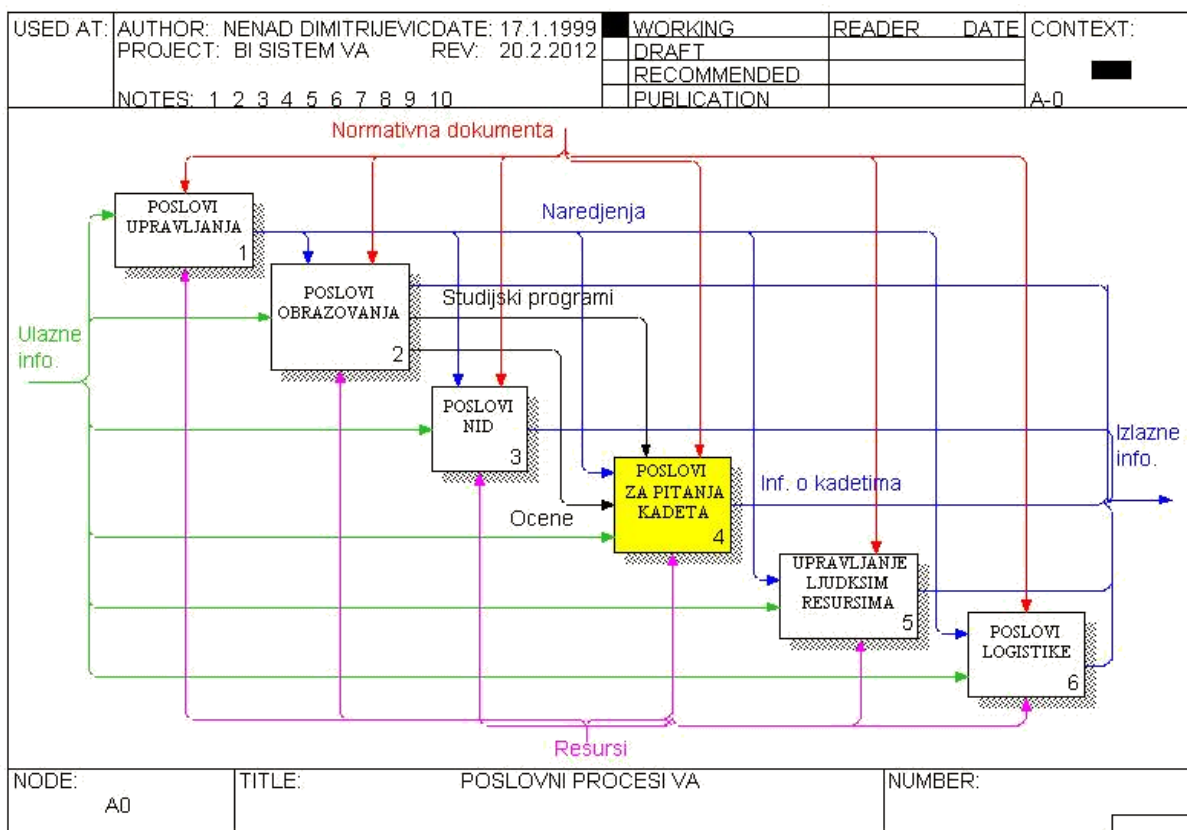
Кандидати пријаву подносе територијалном органу Регионалног центра Министарства одбране. Уз пријаву прилажу: [24]

- оверену фотокопију уверења о држављанству Републике Србије;
- оверену фотокопију извода из матичне књиге рођених;
- потврду надлежног органа да се против њих не води кривични поступак или поступак због кривичног дела за које се гони по службеној дужности, односно да нису осуђивани за таква дела казном затвора у трајању дужем од шест месеци или казном малолетничког затвора;
- оверене фотокопије сведочанства свих разреда и фотокопију дипломе средње школе (за кандидате који су завршили четворогодишње школовање), односно, сведочанства завршених разреда и оверен препис оцена са полугодишта четвртог разреда (за кандидате којима је школовање у току) и
- изјаву да нису ожењени-удате, односно да не живе у ванбрачној заједници и да немају деце, коју попуњавају у територијалном органу Регионалног центра Министарства одбране приликом конкурисања.

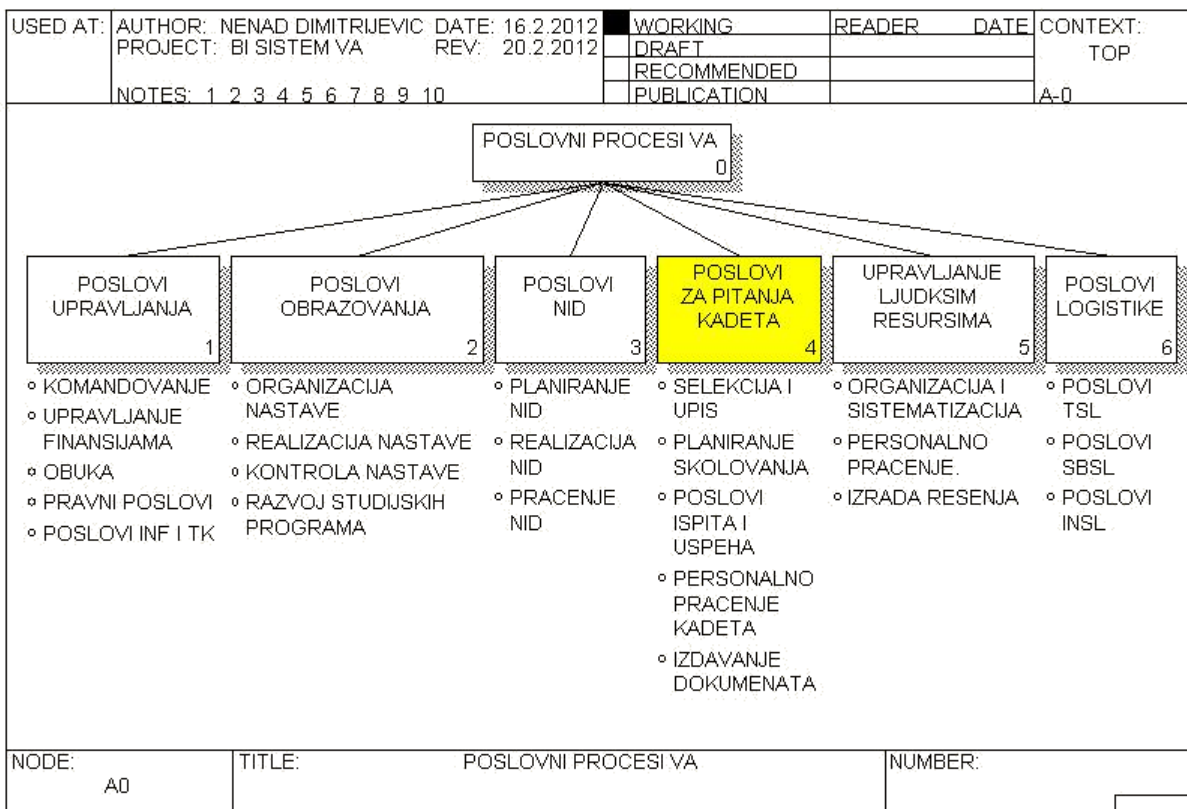
5.1.2. Процена пословног окружења

Како је раније (током 1999. године), за потребе извођења наставе на предмету Развој информационих система, извршено прикупљање корисничких и изворних података и израђени одговарајући модели, у даљем тексту даћемо њихов приказ.

На сликама 5.7. и 5.8. је приказани декомпозициони дијаграм и стабло активности пословних процеса на Војној академији. На сликама можемо приметити да су управљање, образовање, НИД, питања кадета, људски ресурси и логистика послови којима се бави Војна академија. Нама ће у фокусу бити пословни процес који се односи на питања кадета и њега ћемо даље декомпоновати.



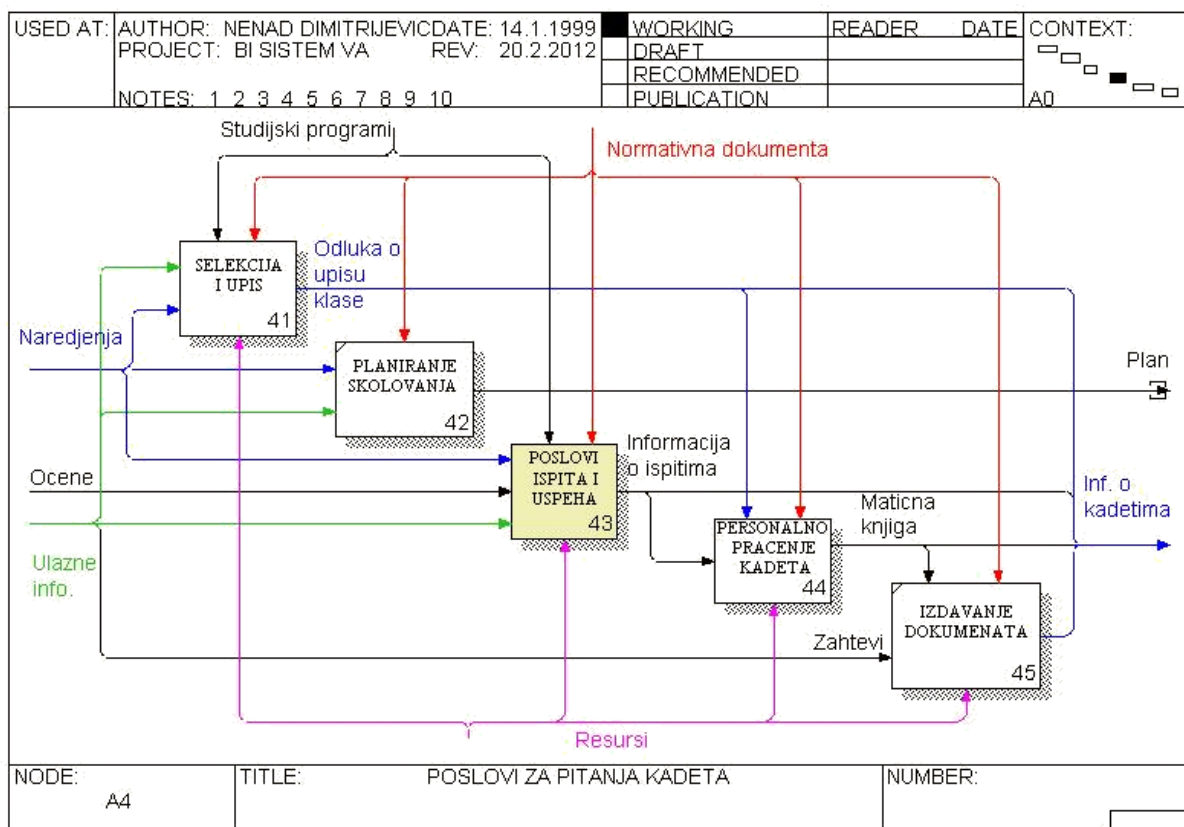
Слика 5.7 – Декомпозициони дијаграм пословних процеса на ВА



Слика 5.8 – Стабло активности на ВА

Даљом декомпозицијом послова за питања кадета добија се дијаграм на слици 5.9. Овај дијаграм показује да: селекција и упис, планирање школовања, послови испита и успеха, персонално праћење кадета и издавање докумената су у ствари послови за питања кадета. Овим пословима се иначе бави Одсек за питања кадета и слушалаца.

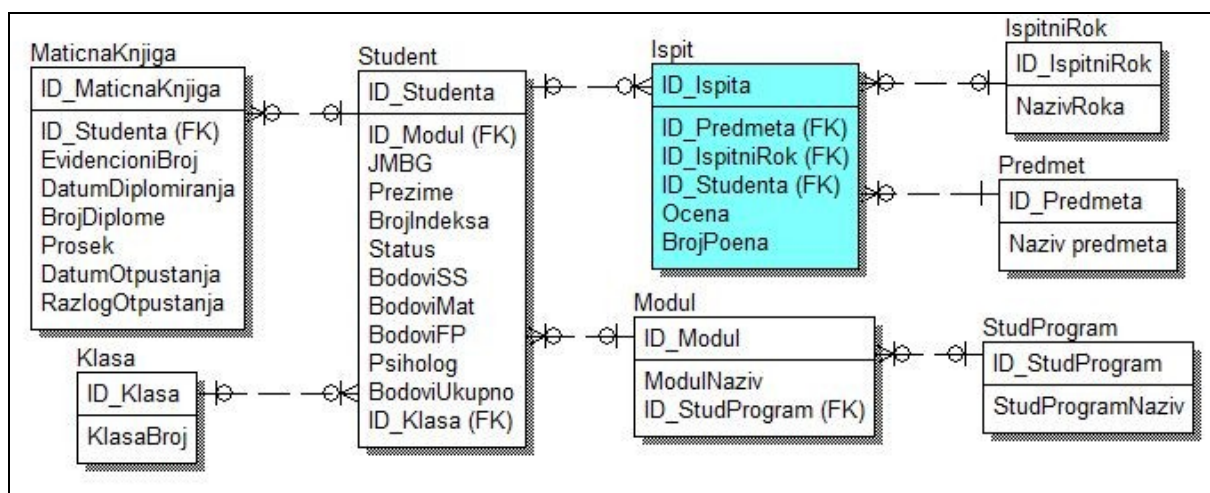
На дијаграму се види да су улазне информације везане за послове испита и успеха: наређења, оцене и остале улазне информације. Овај посао се одвија на основу студијских програма и других нормативних докумената. Излаз овог посла су информације о испитима, односно информације о успеху кадета.



Слика 5.9 – Декомпозициони дијаграм послова за питања кадета

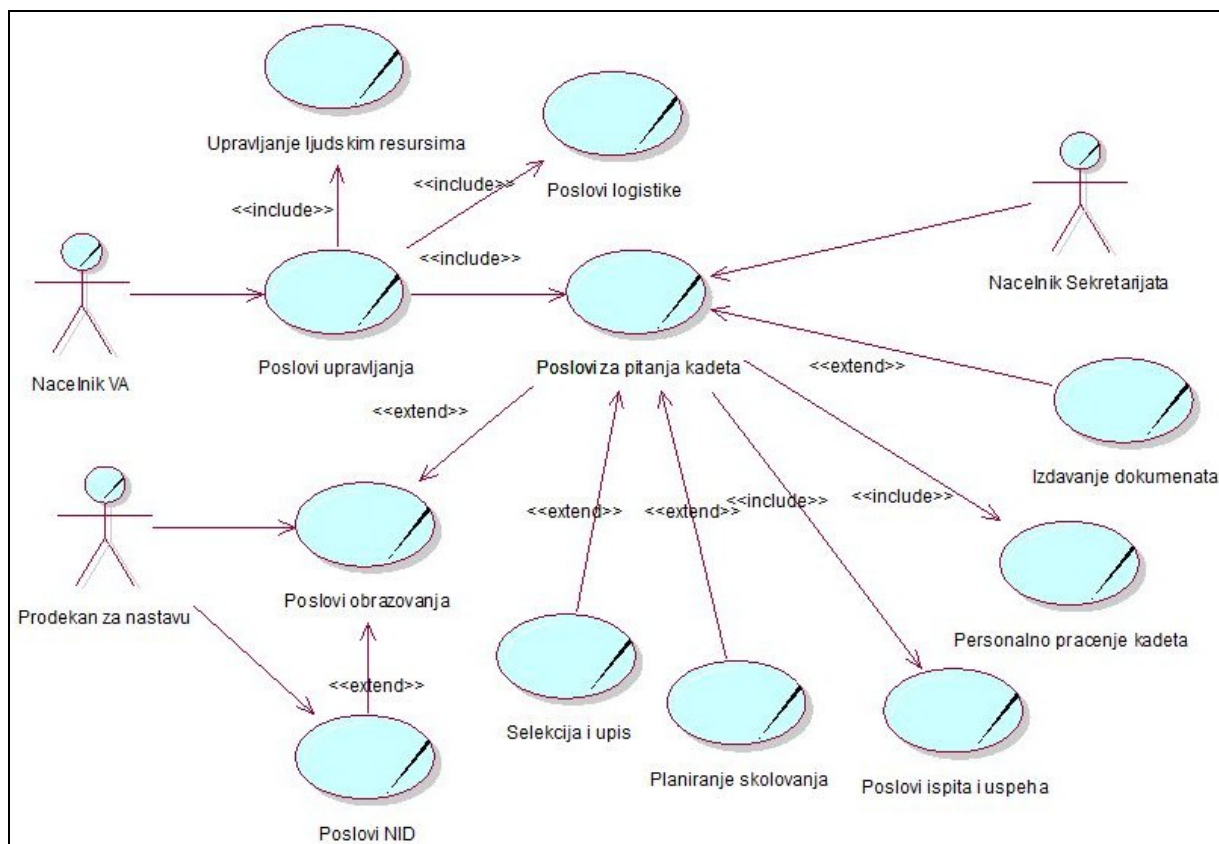
Овако дефинисаним декомпозиционим дијаграмом смо јасно дефинисали границе нашег система, односно одредили смо фокус нашег будућег рада.

Прикупљање изворних података је извршено израдом IDEFX1 модела у ERwin-у, који је приказана на наредној слици. Модел је “директна последица“ Excel табела које се користе за послове испита и успех кадета и атрибута из структуре базе података у новом информационом систему.



Слика 5.10 – IDEFX1 модел изворних података

Даља декомпозиција послова, применом IDEF0 стандарда, није рађена па је то искоришћено као добра прилика да се, у научно истраживачке сврхе, процена пословног окружења изврши и израдом модела пословних случајева употребе.



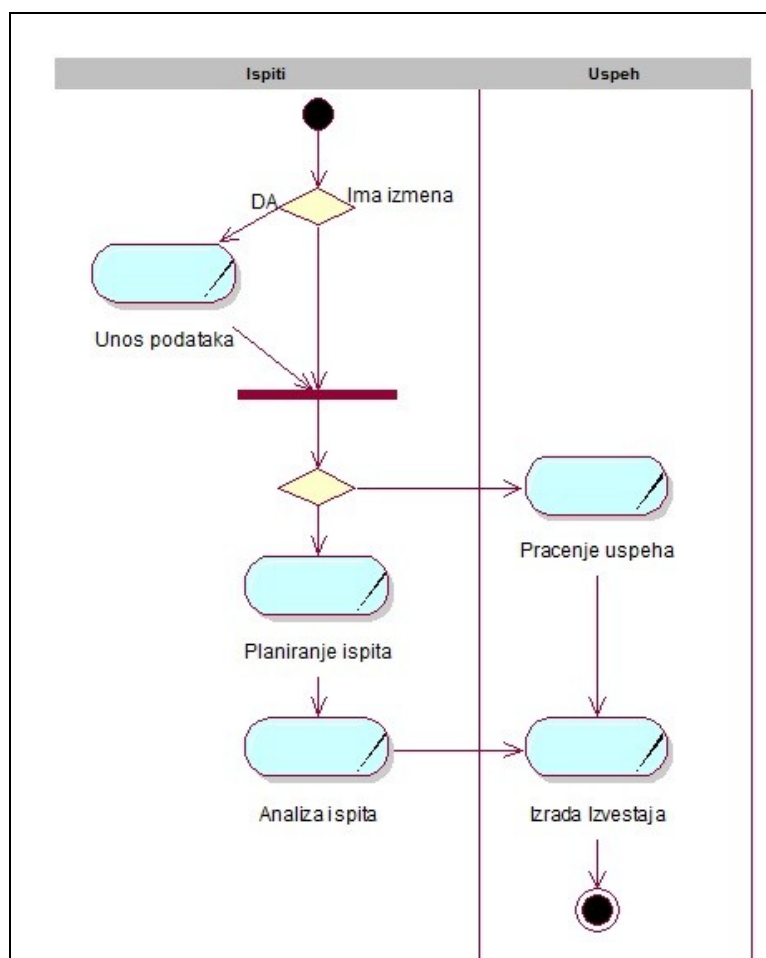
Слика 5.11 – Модел пословних случајева употребе на ВА

На моделу су приказани основни послови на Војној академији и послови који се односе на питања кадета. Можемо видети да већину пословних процеса покреће начелник Војне академије.

Послови везани за питања кадета су укључени у послове управљања, а проширују послове образовања. Њих покреће начелник Секретаријата Војне академије, али посредно и начелник Војне академије и продекан за наставу.

Питања кадета у себе укључују персонално праћење кадета и послове испита и успеха, а проширују се по потреби: селекцијом и уписом, планирањем школовања и издавањем докумената.

Како би омогућили анализу пословног случаја употребе Послови испита и успеха (који је предмет нашег интересовања) извршена је израда дијаграма пословних активности.



Слика 5.12 – Дијаграма пословних активности Послови испита и успеха

Дијаграм пословних активности Послови испита и успеха приказује конкретне активности које референт у Одсеку за питања кадета и слушалаца реализује. Можемо видети да се на почетку увек проверава да ли треба унети нове податке, а онда се бира да ли се реализују активности у вези испита или успеха. На крају се врши израда потребних извештаја.

5.1.3. Планирање система

За потребе израде пројектног плана коришћен је алат Microsoft Project. Прво је извршена анализа структуре задатка. Ова анализа обухвата одређивање списка активности, листе међусобне зависности активности и процену трајања активности. У Табели 5.1. је дат преглед активности које треба реализовати у оквиру израде нашег VI система.

Табела 5.1. Списак активности, њихове међусобне зависности и времена трајања.

Рбр	Активности	Трајање (данима)	Почетак	Завршетак	Претходне активности
1	Дефинисање захтева	9			
2	Набављање литературе	3			1
3	Дефинисање захтева	7			1
4	Анализа захтева	10			2,3
5	Дизајн решења VI система	2			4
6	Израда складишта података	3			5
7	Учитавање података	7			5,6
8	Припрема анализе података	7			5,6
9	Израда корисничког интерфејса	8			7,8

На овај начин се дефинишу рокови у оквиру којих се може ценити степен реализације VI система. Праћењем реализације постављених активности омогућава се уочавање евентуалних проблема и правовремено реаговање ради њиховог отклањања.

За потребе израде BI система за управљање успехом кадета Војне академије од ресурса се морало обезбедити: рачунар са оперативним системом, SQL Server 2008, Microsoft Office 2007, CASE алати (BPWin, ERWin, Rational Rose) и опширна литература из ове области.

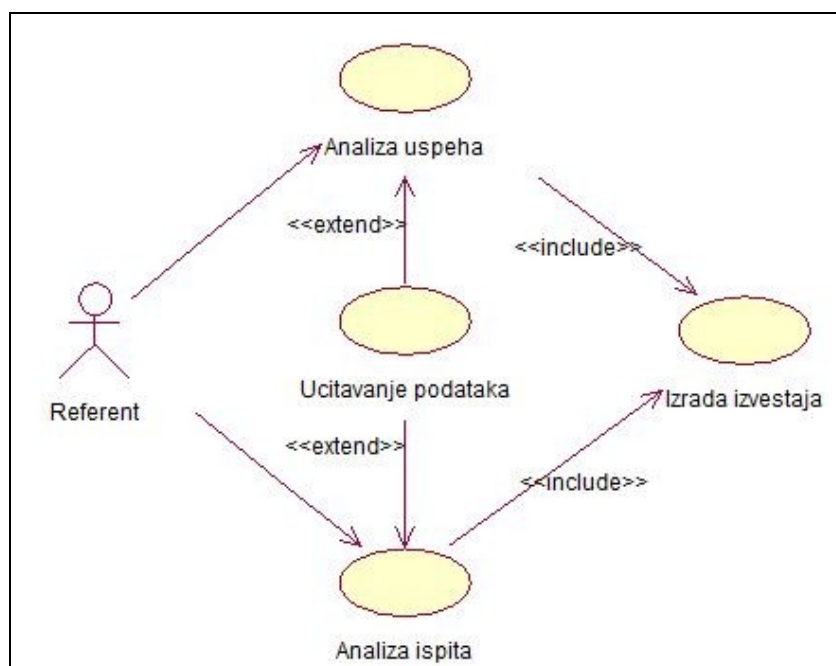
У изради BI система није формиран пројектни тим из разлога што се не ради о пројекту у пуном смислу речи, већ о научно истраживачком пројекту који за циљ има потврђивање дефинисаних хипотеза, а не израду комплетног система.

5.2. Анализа

На основу прикупљених захтева и процене пословног окружења извршена је израда случајева употребе, концептуални модел и анализа изворних података.

5.2.1. Израда случајева употребе

На основу анализе модела пословних случајева употребе и дијаграма пословних активности који описују поједине пословне случајеве употребе, сачињен је модел случајева употребе. На њему су дефинисане функционалности будућег система и њихове међусобне везе. Као функционалности су издвојене: Анализа успеха, Анализа испита, Израда извештаја и Учитавање података.



Слика 5.13– Модел случајева употребе за анализу успеха

Случај употребе **Анализа успеха** енкапсулира активности које корисник система мора да уради да би извршио сагледавање успеха студената

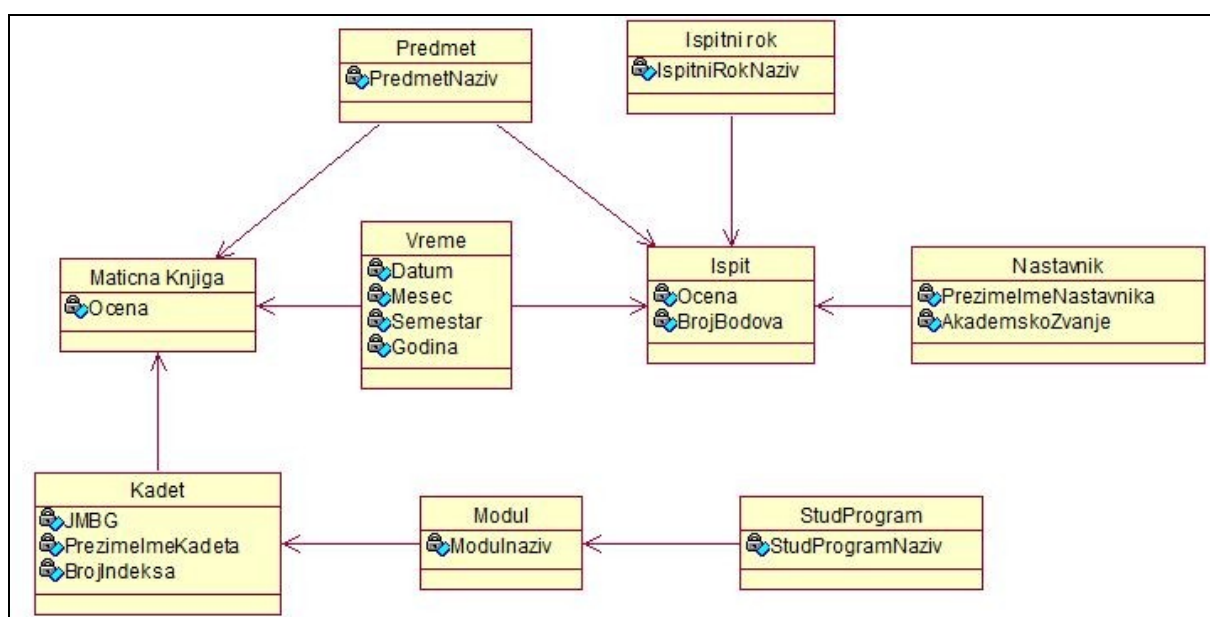
Случај употребе **Анализа испита** има за циљ да обезбеди информације о пролазности на испитима по различитим параметрима.

Случај употребе **Израда извештаја** је практично резултат ангажовања и енкапсулираће израду прегледа и дијаграма управљању Војне академије.

Случај употребе **Учитавање података**, проширује Анализу испита и успеха, и има за циљ да у позадини реализује пуњење складишта података селектованим подацима из OLTP система и других извора података.

5.2.2. Израда концепта

У оквиру активности дефинисања концепата треба на основу претходно дефинисаних захтева и спознаје система одредити објекте(концепте) који се јављају у систему и сваки од њих описати одређеним атрибутима. Реализацијом, методологијом постављених активности, формиран су концепти као подршка за реализацију функционалности случајева употребе. Објекти и њихове везе су одређени на основу анализе података о испитима и кадетима који се воде у Excel датотекама.



Слика 5.14 – Модел концепта решења

5.2.3. Анализа изворних података

Анализа изворних података је извршена на основу података о испитима и кадетима који се воде у Excel датотекама и ИСУО. Као резултат те анализе су генерисани мета подаци за ВІ систем за управљање успехом кадета Војне академије:

- Мета подаци модела
 - Име: ВІ систем за управљање успехом кадета Војне академије
 - Дефиниција: Овај модел садржи податке о успеху кадета.
 - Сврха: Управљање успехом.
 - Контакт особа: Димитријевић Ненад – пројектант.
 - Димензије: Кадет, Време, Предмет.
 - Факта: Испит.
 - Мере: Оцена.
 - Мета подаци димензија
 - Име: Кадет
 - Дефиниција: Кадет је особа која је припадник ВА. Кадет полаже испите.
 - Алиас: /
 - Хијерархија: Кадет припада класи, класа модулу, а модул студијском програму.
 - Правила промене: Нови кадет се додаје као нови ред у табели кадета.
 - Фреквенција коришћења: Често
 - Атрибути: ID Кадета, Презиме, Име, Број индекса, Датум уписа, Статус
 - Факта: Испит
 - Мере: Оцена
 - Поддимензије: Класа, Модул
-
- Име: Време
 - Дефиниција: Време полагања испита.
 - Хијерархија: Школска година-Испитни рок-Датум полагања
 - Правила промене: Додавање новог датума се аутоматски реализује.
 - Фреквенција коришћења: Често

- Атрибути: ID Времена, Датум, ID Испитног рока, Школска година.
 - Факта: Испит
 - Мере: Оцена
 - Поддимензије: Испитни рок.
-
- Име: Предмет
 - Дефиниција: Предавања и вежбе из одређене области
 - Хијерархија: /
 - Правила промене: Нови предмет се додаје као нови ред у табели.
 - Фреквенција коришћења: Често
 - Атрибути: ID Предмета, Назив предмета, ID Наставника.
 - Факта: Испит
 - Мере: Оцена
 - Поддимензије: Наставник.
- Мета подаци о фактима
 - Име: Испит
 - Дефиниција: Облик показивања знања из одређеног предмета.
 - Фреквенција коришћења: Често
 - Пренесени кључеви: ID Кадета, ID Времена, ID Предмета.
 - Факта: Испит
 - Мере: Оцена
 - Димензије: Кадет, Време, Предмет
 - Мета подаци о мерама
 - Име: Оцена
 - Дефиниција: Бројчана вредност показаног знања на испиту.
 - Фреквенција коришћења: Често
 - Тип податка: smallint
 - Домен: 5-10
 - Факта: Испит
 - Димензије: Кадет, Време, Предмет

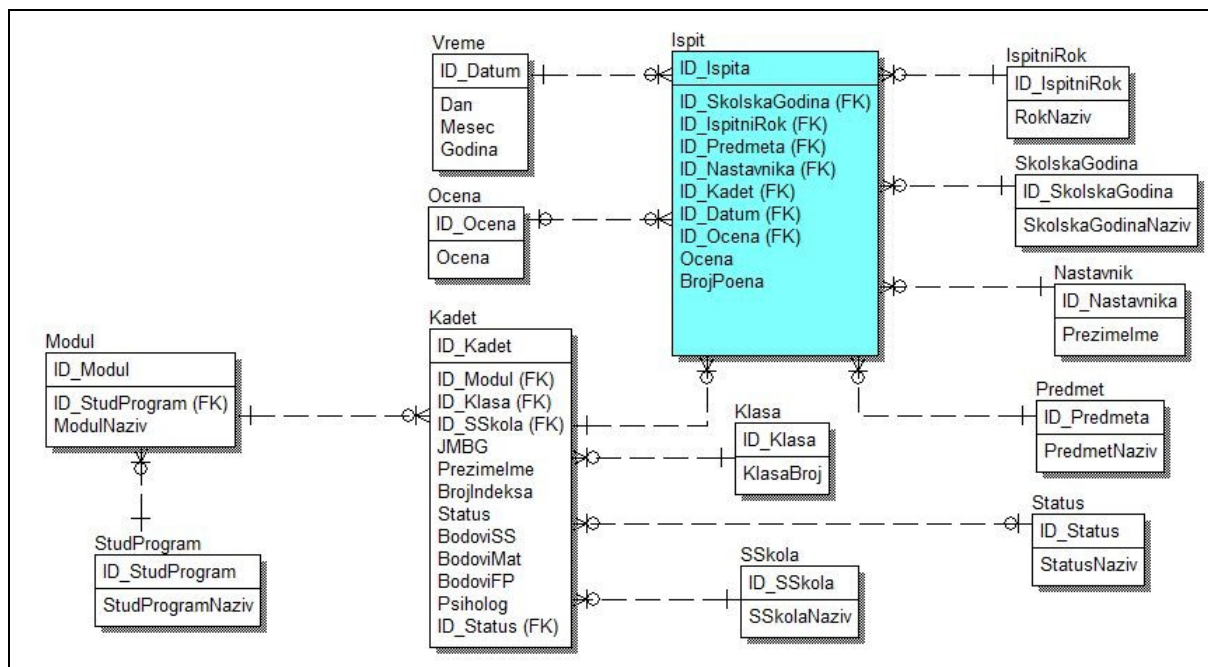
Овако дефинисани мета подаци, заједно са дефинисаним концептима, представљају основу за дизајн складишта података и ETL сервиса.

5.3. Дизајн

У оквиру Анализе смо извршили идентификацију проблема, али нисмо дали одговор на питање како решити проблем. Дизајн треба да одговори на то питање, логичким и физичким декомпоновањем система на мање целине и притом изврши спецификацију тих целина. У претходној фази смо извршили припрему података у оквиру које смо одредили мета податке, на основу тога у оквиру Дизајна треба извршити израду димензионог модела складишта података.

5.3.1. Дизајн складишта података

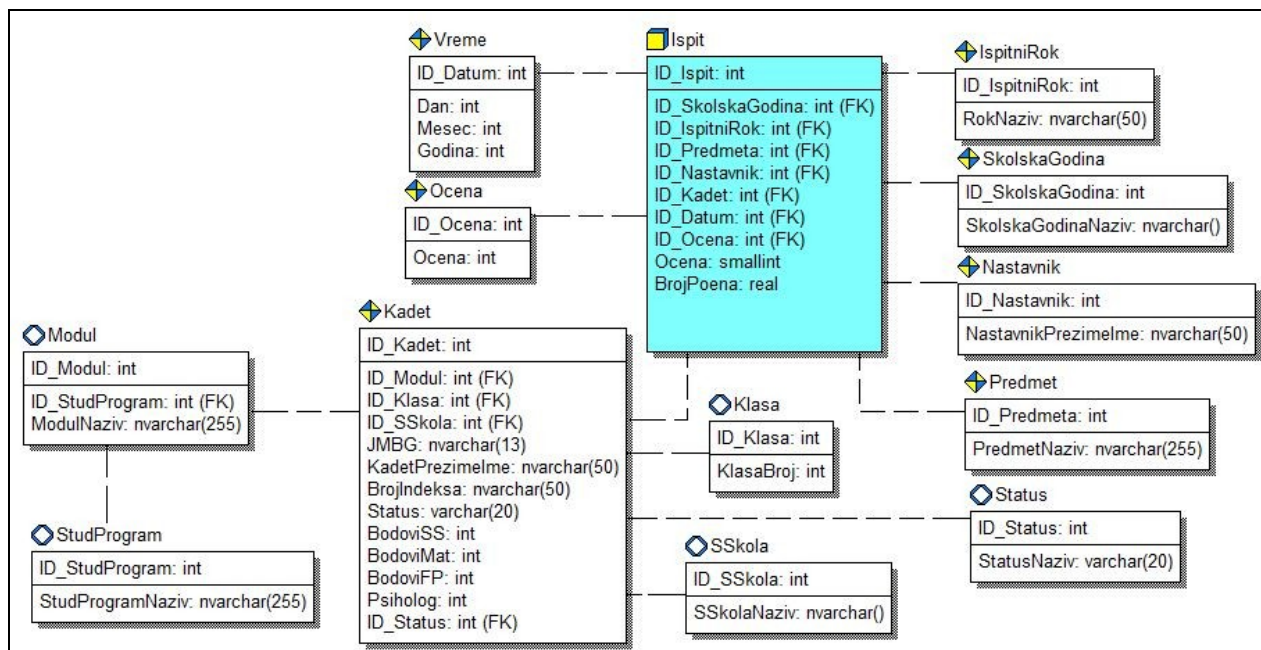
Основу за израду димензионог модела представљају мета подаци генерисани у оквиру фазе Анализе и активности анализа података. На основу тих података врши се дефинисање хијерархија, елемената и атрибута, нормализација и денормализација и дефинисање агрегација. Производ ових активности су модели приказани на следећим сликама. На слици 5.15 приказан је логички модел података, а на слици 5.16 димензиони модел складишта података.



Слика 5.15 – Логички модел складишта података

Наведени модели су израђени коришћењем CASE алата за моделовање ERwin.

На димензионом моделу јасно можемо разликовати табеле чињеница (ознака жуто плаве коцкице), табеле димензија (жуто плави ромб) и табеле хијерархија (плаво оивичени ромб).

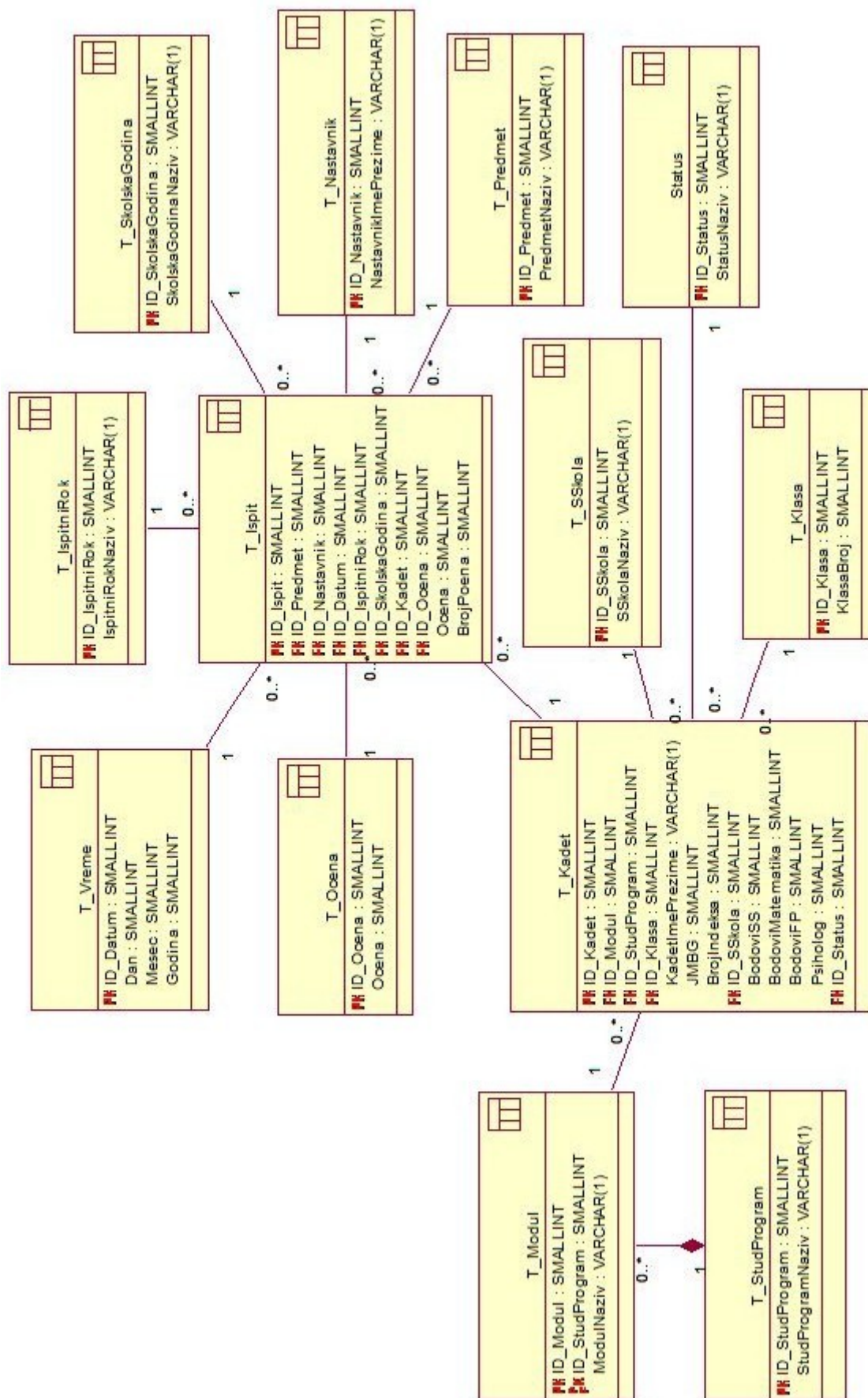


Слика 5.16 – Димензиони модел складишта података

На основу дефинисаног димензионог модела врши се израда физичког модела складишта података. Креирање физичког модела складишта података је поступак превођења логичког модела података у физички модел складишта података. Физички модел складишта података преводи димензиони (логички) модел складишта података и том приликом долази до конвертовања:

- ентитета из модела података у табеле физичке базе података,
- атрибута у колоне, у одговарајућим табелама, и
- кандидата за кључеве ентитета у примарне кључеве у табелама.

У нашем случају за израду физичког модела је коришћен алат за објектно моделовање *Rational Rose*. Овако је још једном направљена веза између класичних и објектно оријентисаних метода за развој информационих система. На слици 5.17. је приказан физички модел шеме складишта податка. Тако у оквиру једне *Rational Rose* датотеке имамо формиране све моделе од почетка до краја моделовања нашег BI система.

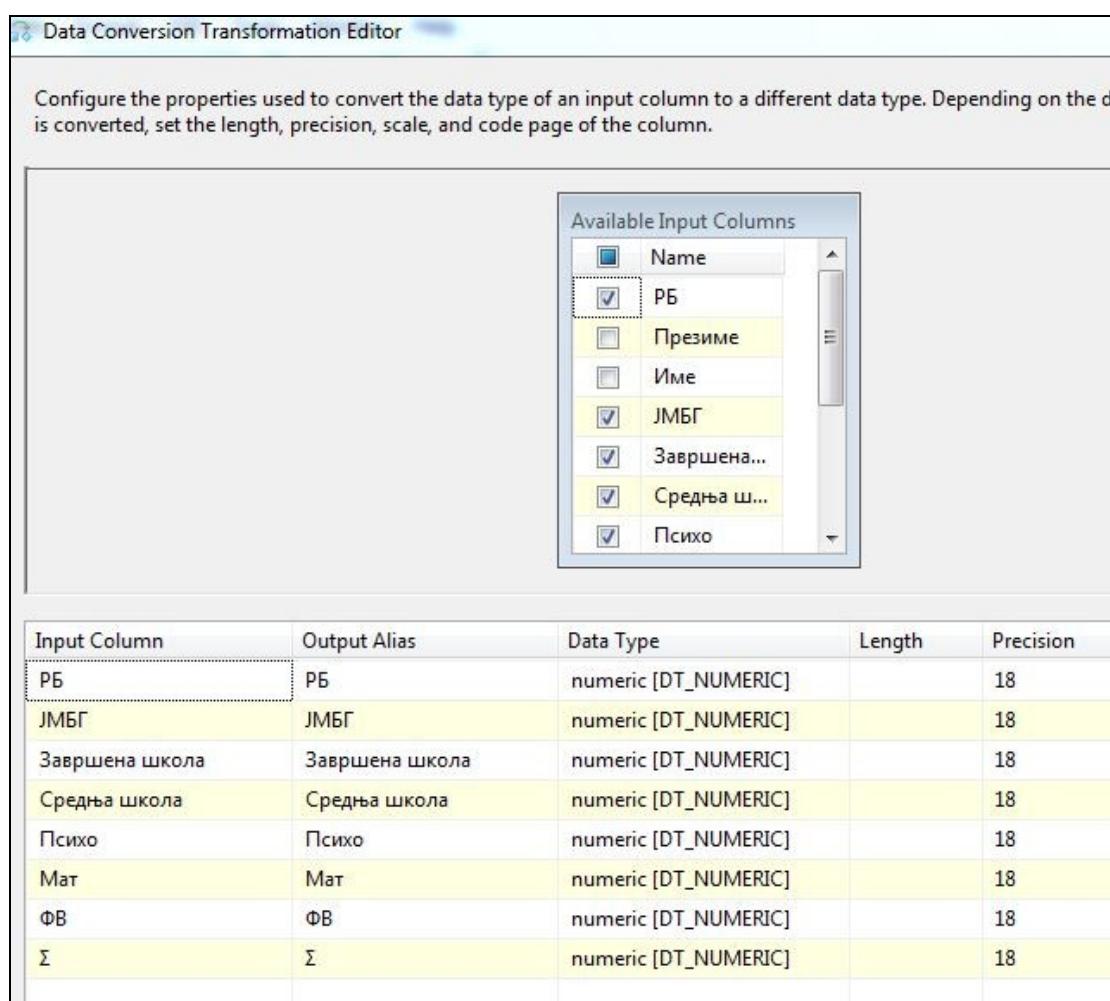


Слика 5.17 – Физички модел складишта података

5.3.2. Припрема података

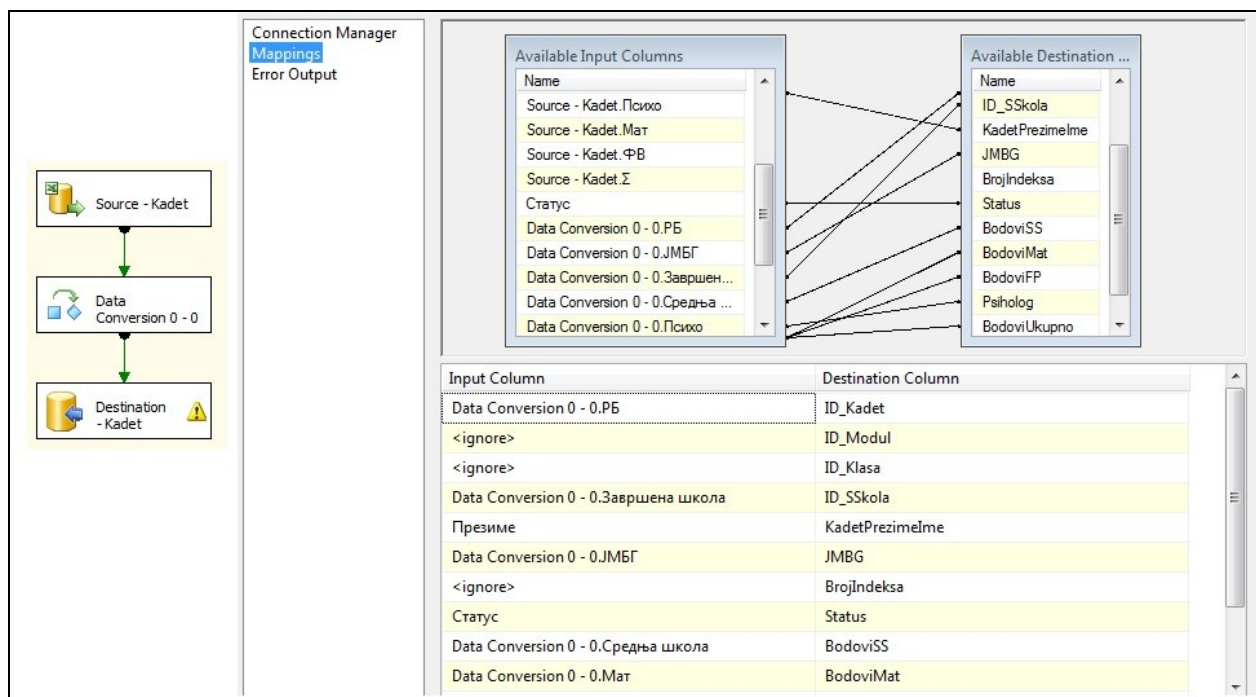
У оквиру корака Припрема података је извршена трансформација имена и презимена у један заједнички слог и извршена конверзија Excel типова података у типове који одговарају систему за управљање базама података. Ова активност је извршена у више итерација пре и након реализације складишта података у оквиру SQL Server-а. Управо је ова итеративност и основна предност објектног приступа у развоју.

На следећој слици је приказана једна од екранских форми, у оквиру SQL Server Integration Services алата, за конверзију података.



Слика 5.18 – Припрема података

У оквиру једне од итерација извршена је и израда ETL пакета који ће нам касније омогућити учитавање података у складиште. Изглед овог пакета и шеме мапирања података између изворне и циљне табеле приказан је на слици 5.19.



Слика 5.19 – Мапирање података и ETL пакет за учитавање

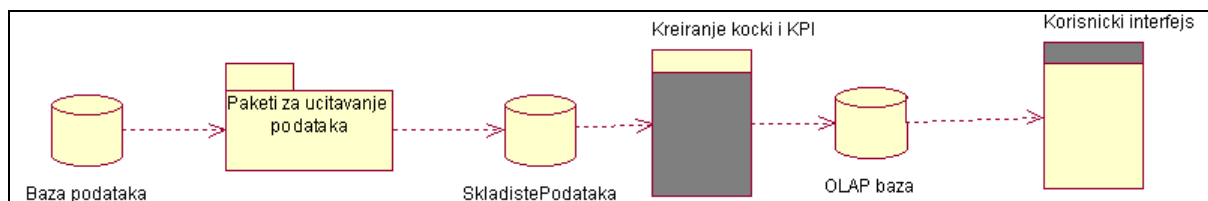
5.3.3. Дизајн увођења система

Као што смо раније дефинисали дизајн увођења система се реализује кроз:

- Дизајн реализације система;
- Избор технологије реализације;
- Дефинисање хардверске архитектуре.

Дизајн реализације система

Код формирања структуре система водило се рачуна да он буде вишеслојан. Оваква структура омогућава поделу одговорности за предвиђене функционалности система а самим тим повећава се скалабилност и поузданост самог система. Основна идеја приликом дизајна реализације је била вишеслојност у приступу бази, односно складишту података. На следећој слици приказана је структура реализације система.



Слика 5.20 – Дизајн реализације система

Избор технологије реализације

Непосредно пре дефинисања хардверске архитектуре и реализације система потребно је изабрати технологију реализације, односно систем за управљања базама података и системе за анализу података.

У вези система за управљање базама података одлучено је да то буде *Microsoft SQL Server 2008*, јер има многе особине које поједностављују процес инсталирања, развоја, управљања и коришћења база података. Оно што је најбитније *SQL Server 2008* садржи алате који олакшавају рад са складиштима података. *SQL Server 2008* нуди и анализу кроз комплетан и интуитивни скуп *Data Mining* алата.[23]

Microsoft SQL Server 2008 аналитички сервис доноси унапређене концепте, као што су скалабилност, напредни аналитички алати и *Microsoft Office* интероперабилност. Кроз унапређену скалабилност повећава се продуктивност програмера. Рад са аналитичким сервисима се одвија кроз *Business Intelligence Development Studio* који је заснован на *Visual Studio* развојном окружењу и омогућава адекватну подршку за дизајн, развој, сарадњу и оптимизацију. Све ово омогућује повећану ефикасност и продуктивност. [23]

Скалабилност *SQL Server 2008* аналитичких сервиса се огледа у могућности да подрже огромне базе података са више хиљада корисника. *SQL Server 2008* има и могућност креирања различитих мета података тако да омогућује креирање локализоване аналитичке базе података на језику корисника.

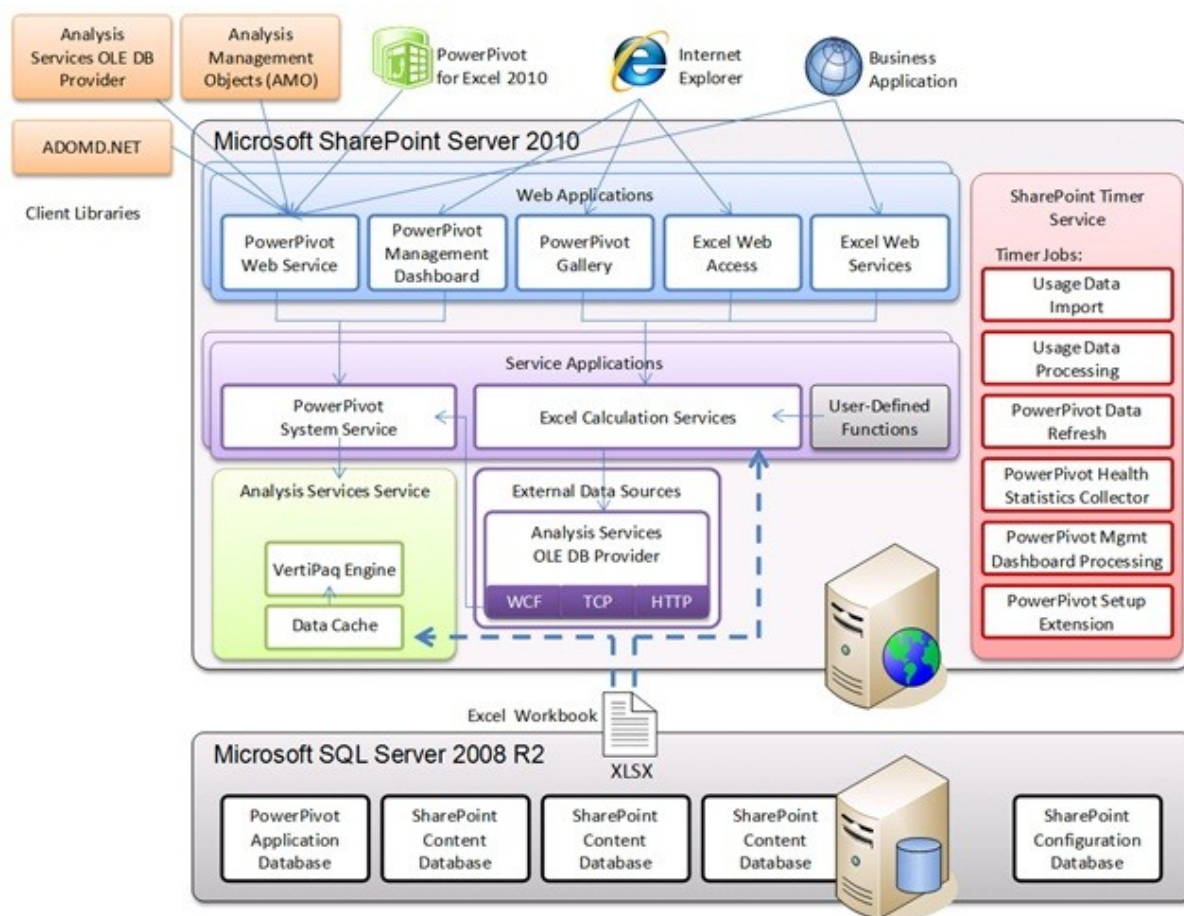
Додатна предност *SQL Server*-а 2008 је и веза са *Microsoft Excel*-ом. Наиме, *Microsoft* је креирао посебан *Data Mining Add-in* за *Office 2007*. Ови додаци су дизајнирани тако да одговарају крајњем кориснику и дају могућност вршења напредних анализа директно у *Excel*-у, посредно користећи *Data Mining* могућности *Microsoft SQL Server 2008* Аналитичких Сервиса.

У ове сврхе су развијене две засебне компоненте:

- *Data Mining* клијент за *Excel* који омогућава креирање и управљање комплетним *Data Mining* пројектом из *Excel*-а;
- *Table Analysis Tools* за *Excel* омогућавају коришћење аналитичких *Data Mining* сервиса за анализу података сачуваним у *Excel* -у.

На основу наведеног можемо закључити да *Microsoft SQL Server 2008* аналитички сервиси омогућавају лаку надградњу постојећих система. Подаци су доступнији корисницима јер приступ подацима на било који начин није довољан. Корисницима требају познати алати и окружења. *SQL Server 2008* аналитички сервиси омогућавају додатну интероперабилност са *Office* пакетом кроз *Data Mining* проширења.

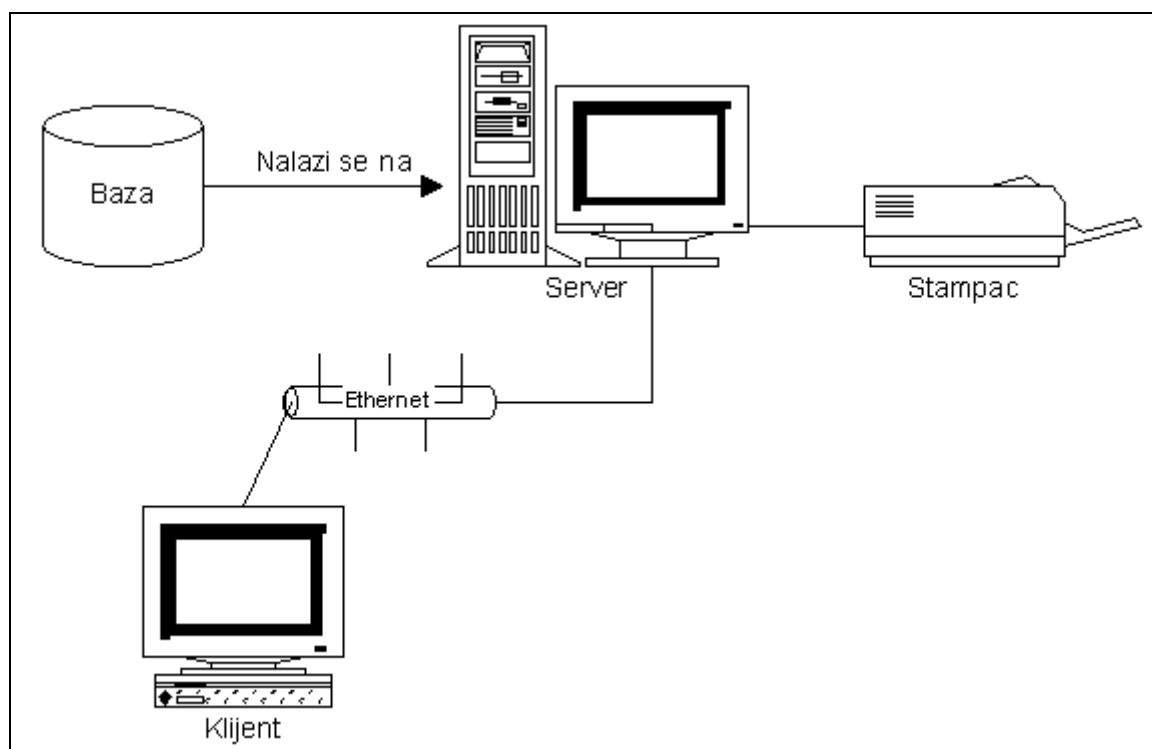
На следећој слици је приказан комплетан скуп *Microsoft SQL Server 2008 Business Intelligence* решења.



Слика 5.21 – *Microsoft SQL Server 2008 Business Intelligence* [23]

Дефинисање хардверске архитектуре

За разлику од постојећег окружења, ново би подразумевало одређена хардверска проширења, као и промене у архитектури подсистема. Следећа слика представља могући изглед хардверске структуре за потребе BI система за управљање успехом кадета Војне академије.[9]



Слика 5.22 – Дијаграм хардверске архитектуре [9]

Оваквом организацијом хардверске архитектуре се још једном подстиче идеја вишеслојности у приступу израде BI система.

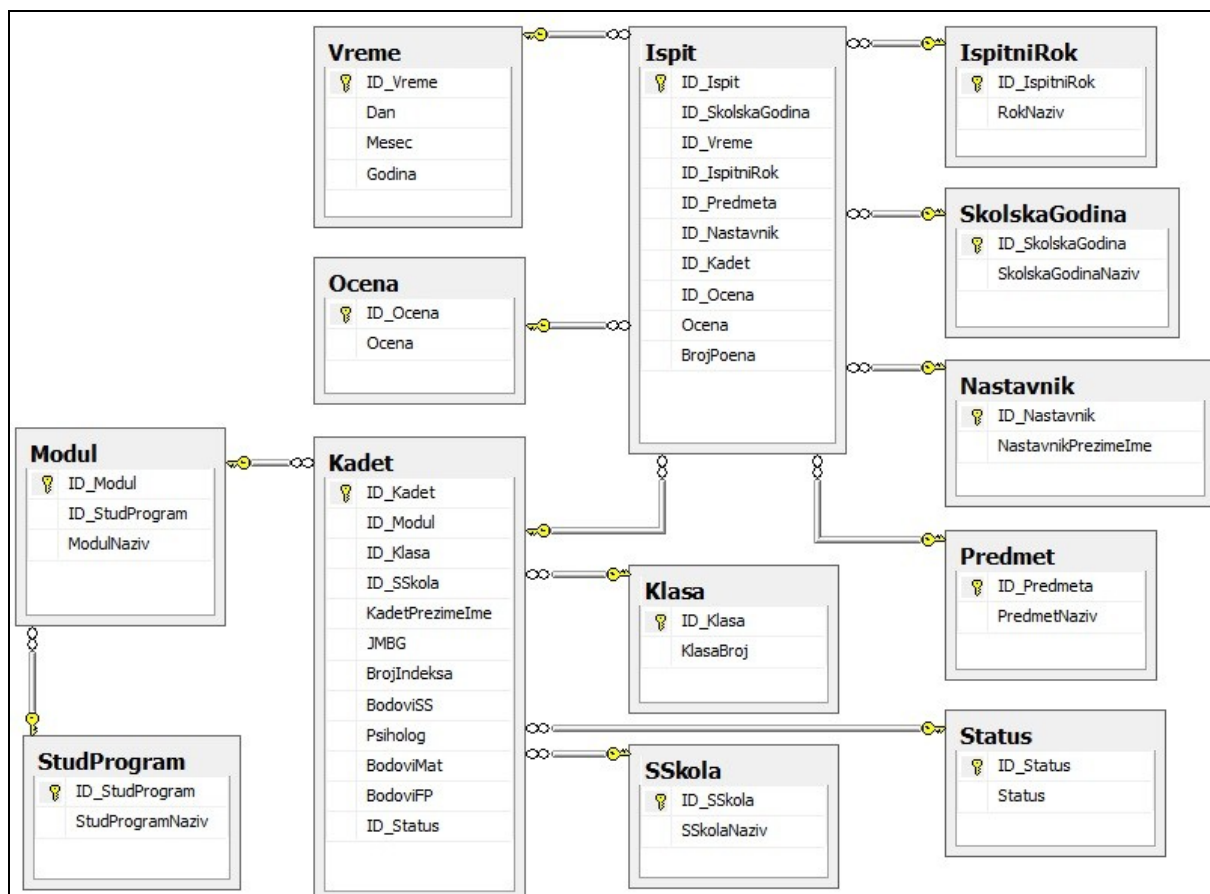
5.4. Реализација

Реализација представља последњу фазу дефинисане методологије и огледа се у “оживљавању” система на основу претходно дефинисаних захтева, анализе и дизајна.

5.4.1. Израда складишта података

Генерисање складишта података изведено је у *SQL Server 2008*-у, коришћењем CASE алата. Наиме, алат у коме је извршено креирање физичког модела (*Rational Rose* и *ERwin*) омогућава аутоматско генерисање кода преко такозваних DDL (*Data Definition Language*) датотека за релационе базе података. Физички модел се тако преводи у SQL код који се извршава на СУБП и тако креира база.

На слици 5.23. приказана је физичка реализација складишта података.



Слика 5.23 – Физичка реализација складишта података у SQL Server-у 2008

Након генерисања базе, следи **учитавање података**. У ту сврху су коришћени интеграциони сервиси SQL Server-а и раније дефинисани пакети за учитавање података. Дакле, учитавање података је извршено покретањем ETL пакета раније приказаног на слици 5.16. Како је било потребно постојање и додатних података који нису постојали у Excel табелама извршено је њихово “случајно“ аутоматско генерисање. Коришћењем случајно генерисаних података обезбеђена је и заштита конкретних података о кадетима обзиром да се садржај дисертације јавно публикује.

5.4.2. Припрема анализе података

Све што је урађено у оквиру фаза Дефинисања захтева, Анализе и Дизајна треба да омогући BI систем за анализу података. Велики значај анализе података условио је и развој различитих техника за анализу података. Неке од тих техника су упити и извештаји генерисани упитима, вишедимензиона - OLAP анализа и *Data Mining*.

Израда упита

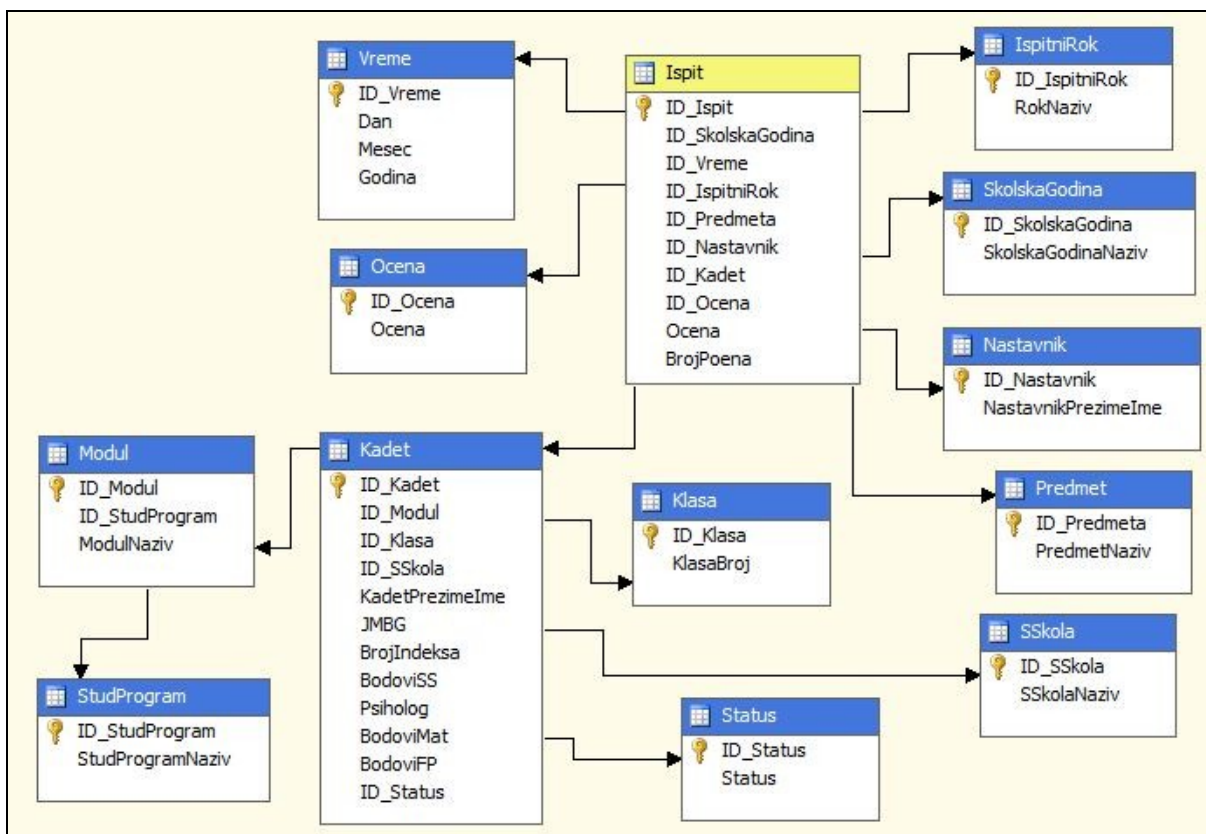
Анализа упитима је процес који обухвата постављање одређеног питања, добијања одговарајућих података из складишта података на основу тог питања, трансформације тих података у одговарајући облик и њихово приказивање у захтеваном формату. За ову технику се каже да је "вођена аналитичарем" (*Driven by Analyst*), што значи да одговор директно зависи од упита (питања) аналитичара. Ова врста анализе је већ реализована кроз постојећи информациони систем и Excel датотеке тако да јој није посвећена посебна пажња у овом раду.

Вишедимензиона - OLAP анализа

Вишедимензиона анализа је популаран начин проширења могућности упита. Овакав начин анализе замењује анализу преко великог броја упита. Вишедимензиона анализа омогућава кориснику да врши анализу по великом броју независних елемената, који чине систем који се анализира, и да прегледају податке који су међусобно доста комплексно повезани. Крајњи корисници нису увек заинтересовани за исти ниво детаљности података, тако да је вишедимензионом анализом омогућено да се ниво детаљности динамички мења.

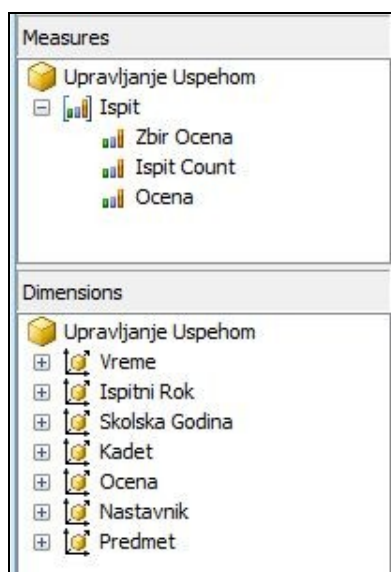
Најпогоднији начин за вршење вишедимензионе анализе је OLAP који у ствари представља начин организовања великих база података на начин који је најпогоднији за анализу од стране корисника. Коришћењем OLAP -а подаци се могу организовати по нивоима детаљности, разврставањем у одређене категорије (креирање хијерархије димензија). OLAP структуре података се називају коцкама, јер комбинација неколико димензија подсећа на геометријски облик коцке.

Креирање OLAP коцки може да се изврши коришћењем аналитичких сервиса *SQL Server 2008* (скуп алата који омогућавају израду OLAP база података и манипулацију над њима). [23] Први корак у креирању OLAP коцке је одређивање извора података (у нашем случају складишта података) и креирање погледа на те податке, а након тога се креира и сама коцка. OLAP коцке садрже два типа података: мере, које представљају нумеричке вредности показатеља интересантних за анализу, и димензије, које представљају категорије које користите да бисте организовали ове мере. На слици 5.24 приказана је OLAP коцка за потребе BI система за управљање успехом студената Војне академије.



Слика 5.24 – OLAP коцка

На слици 5.25 приказане су дефинисане мере и димензије. Како оцена представља кључни податак за потребе анализе, тако су и мере дефинисане израдом “збира оцена“ и самом “оценом“. Генерисана је и мера “број испита“ како би дељењем “збира оцена“ добили просек положених испита кадета.



Слика 5.25 – Преглед мера и димензија OLAP коцке

Додатно су израђене мере за број “палих“ кадета и “просек“ по испиту. Ове мере су израђене као засебни прорачуни. Екранске форме за израду ових мера су приказане на наредним сликама.

The screenshot shows the configuration window for a measure named 'Prosek'. The 'Name' field contains 'Prosek'. Under 'Parent Properties', the 'Parent hierarchy' is set to 'Measures'. The 'Expression' field contains the following DAX formula: `sum([Ocena].&[6]:[Ocena].&[10], [Measures].[Zbir Ocena])/sum([Ocena].&[6]:[Ocena].&[10], [Measures].[Ispit Count])`. Under 'Additional Properties', the 'Format string' is '#,##0.00;-#,##0.00', 'Visible' is 'True', 'Non-empty behavior' is 'Zbir Ocena', and 'Associated measure group' is 'Ispit'.

Слика 5.26 – Мера “Просек“

The screenshot shows the configuration window for a measure named 'Pali'. The 'Name' field contains 'Pali'. Under 'Parent Properties', the 'Parent hierarchy' is set to 'Measures'. The 'Expression' field contains the following DAX formula: `([Measures].[Ispit Count], [Ocena].[Ocena].&[5])/([Measures].[Ispit Count], [Ocena].[Ocena].[All])`. Under 'Additional Properties', the 'Format string' is 'Percent', 'Visible' is 'True', 'Non-empty behavior' is 'Ispit Count', and 'Associated measure group' is 'Ispit'.

Слика 5.27 – Мера “Пали“

За потребе израде мера “пали“ и “просек“ коришћен је MDX (*Multi Dimensional eXpressions*) језик који је специјализован за рад са вишедимензионим подацима.

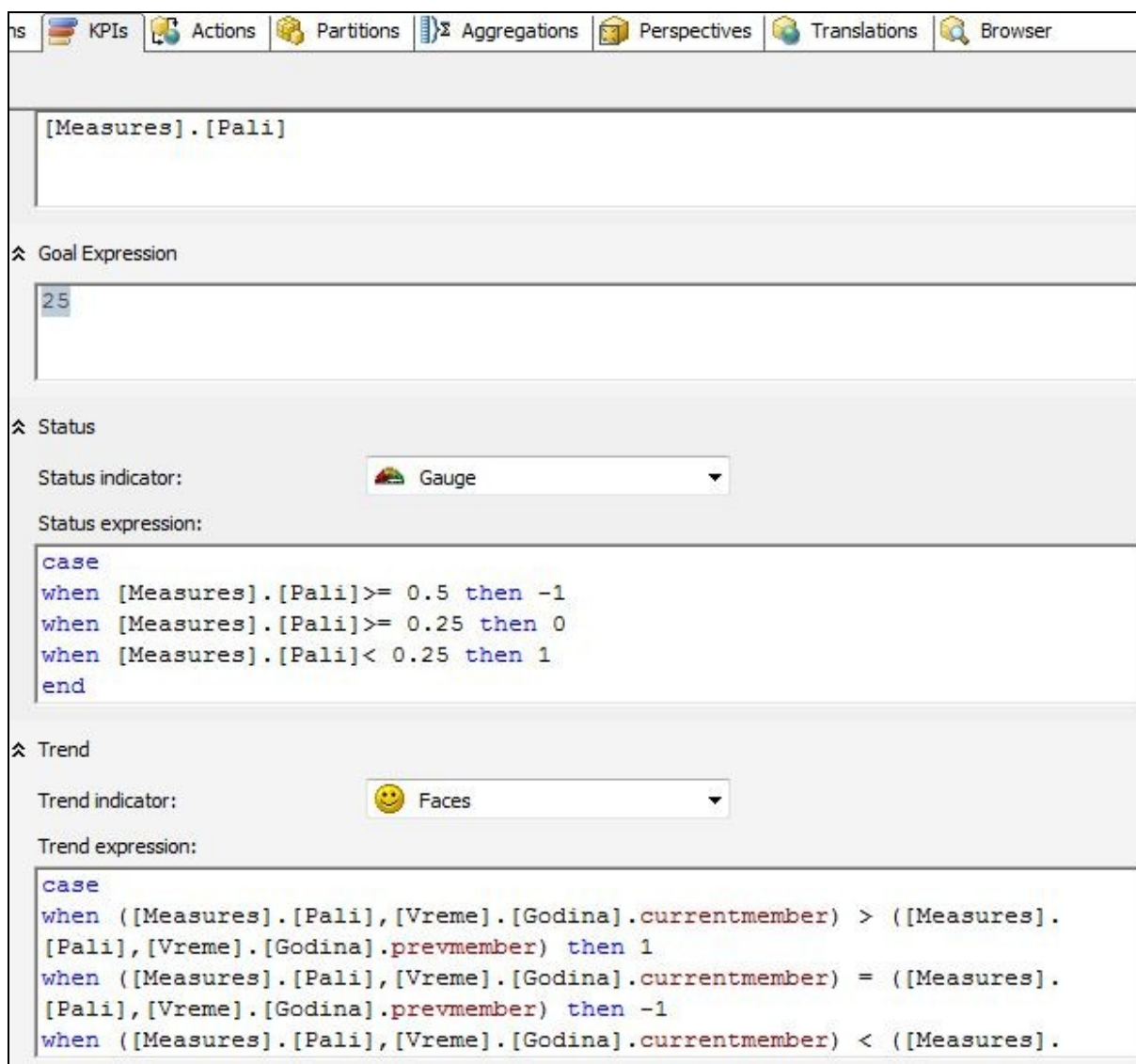
На основу дефинисане коцке, мера и димензија приступило се изради показатеља перформанси (*Key Performance Indicators* - KPI). KPI је посебно израчуната мера која је дефинисана на серверу и дозвољава да пратите “кључне параметре учинка“, укључујући статус (Да ли се тренутна вредност поклапа са пројектованом?) и тренд (како се вредност мења с временом?). Поред наведеног, KPI може дефинисати препоручену графику, као што је семафор да покаже „добро-зелено, просечно-жуто, лоше-црвено,“ кад се приказује статус или тренд коришћењем усмерених стрелица (стрелица на горе-раст, стрелица на доле-пад и хоризонтална стрелица – без промене). Слика 5.28 показује пример дефинисања KPI “Просек“ За дефинисање израза везаних за статус и тренд користи се MDX.

The screenshot shows a software interface for defining a KPI. The top navigation bar includes icons for 'KPIs', 'Actions', 'Partitions', 'Aggregations', 'Perspectives', 'Translations', and 'Browser'. The main configuration area is divided into several sections:

- Name:** A text input field containing 'Prosek'.
- Associated measure group:** A dropdown menu with 'Ispit' selected.
- Value Expression:** A text area containing the MDX expression: `[Measures].[Prosek]`.
- Goal Expression:** A text area containing the value: `8`.
- Status:** A dropdown menu for 'Status indicator' with 'Traffic light' selected.
- Status expression:** A text area containing a case statement:

```
case
when [Measures].[Prosek]>= 8 then 1
when [Measures].[Prosek]< 8 and [Measures].[Prosek]> 7 then 0
when [Measures].[Prosek]< 7 then -1
end
```

Слика 5.28 – Дефинисање KPI “Просек“



Слика 5.29 – Дефинисање KPI “Пали”

На слици 5.30 су приказани реализовани KPI који омогућавају да се управљање успехом врши посматрањем статуса или тренда, без прегледа бројчаних података. Уколико постоји потреба, конкретним подацима се приступа кроз Pivot табеле.

Display Structure	Value	Goal	Status	Trend
Pali	13.43%	25		
Prosek	7.74	8		

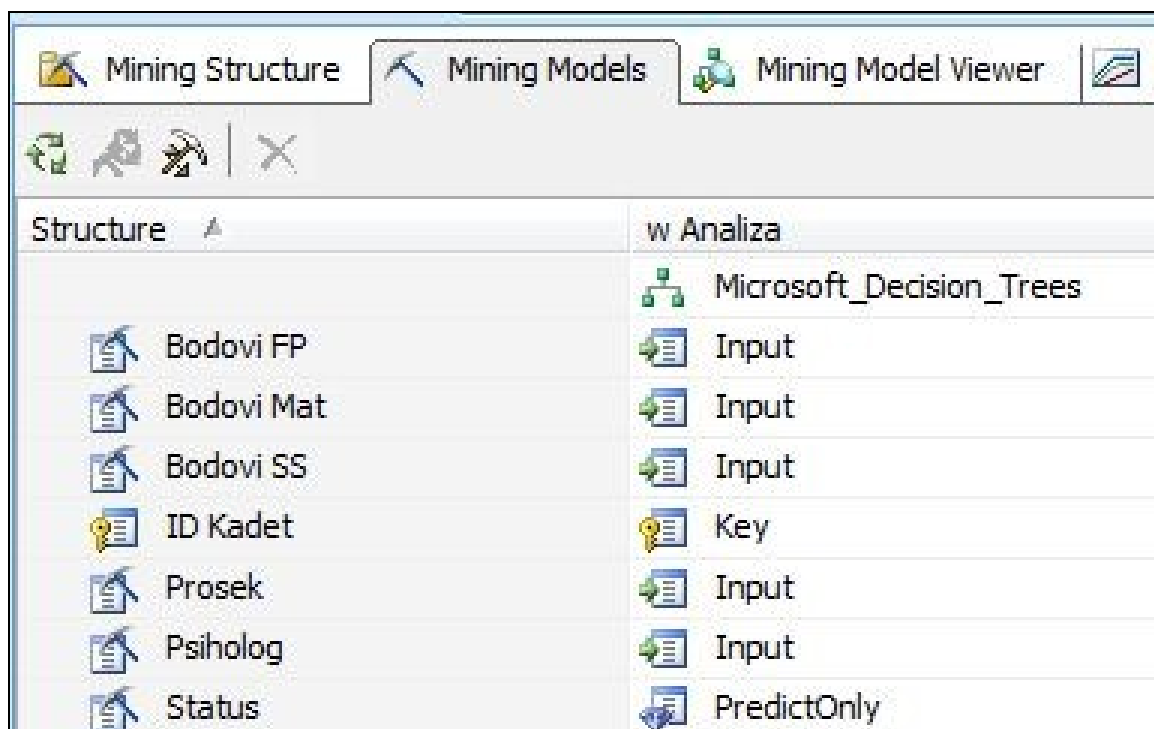
Слика 5.30 – Приказ дефинисаних KPI

Анализа података организованим у OLAP коцке може да се врши коришћењем *PivotTable* (динамичка табела са обједињеним подацима из нека базе података) сервиса који омогућавају приступ подацима у OLAP коцкама. Пивот табеле представљају веома моћан и користан алат у Microsoft Excel-у.

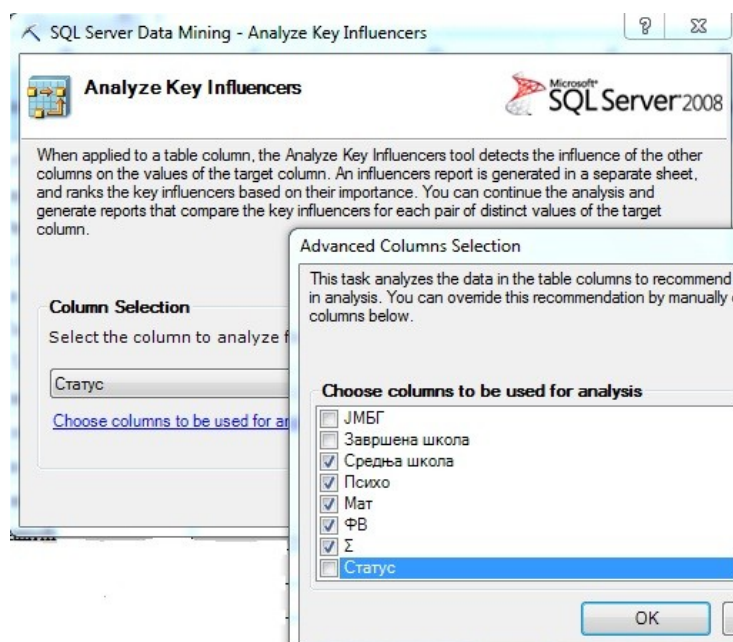
Креирање пивот табела нам омогућава сумирање и анализу веома великог броја података и приказ информација према нашим потребама. Разлог због кога се ове табеле називају „пивот табеле“ лежи у могућности превлачења поља која садрже нама битне информације као и стварање унакрсних поља. Уколико желимо да прикажемо податке који изражавају суму или број понављања, такве информације добијамо унакрсном анализом поља из листе аналитичких података.

Data Mining

За потребе *Data Mining* анализе извршна је израда одговарајућег модела који ће за улазне параметре са уписа кадета и просека током студија вршити одређене анализе. У нашем случају вршиће се анализа утицаја дефинисаних параметара на то да ли ће кадет дипломирати или бити отпуштен са Војне академије.

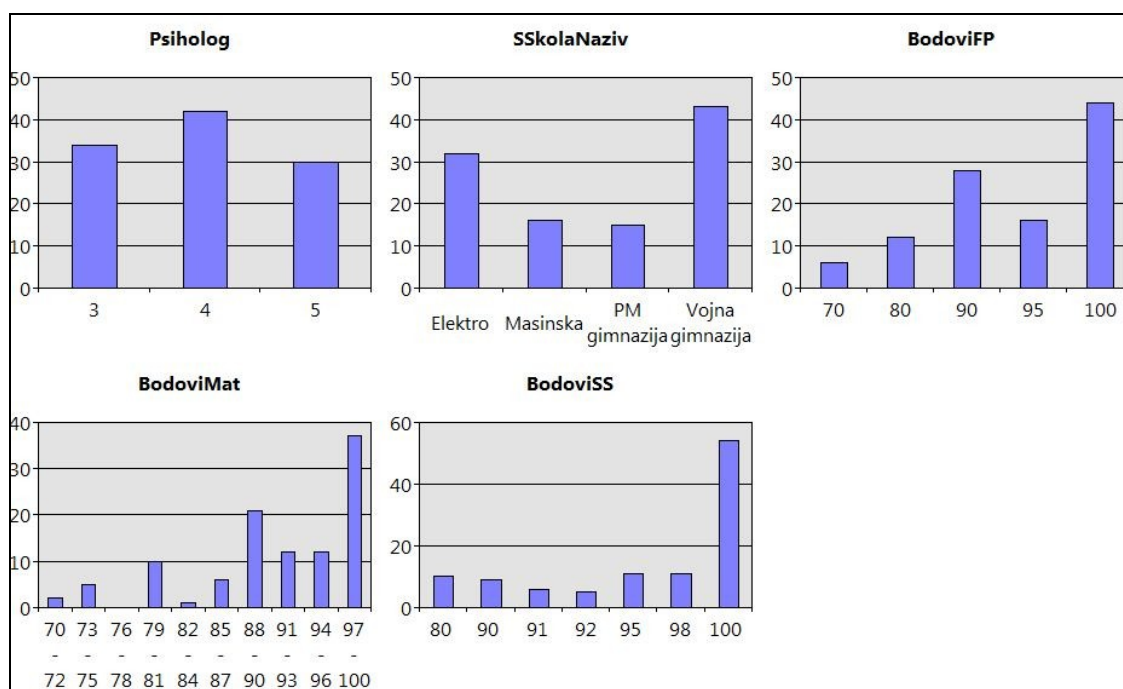


Слика 5.31 – Data mining модел



Слика 5.32 – Дефинисање параметара за Data mining

На претходним сликама је приказан изглед екранских форми за потребе дефинисања *Data Mining* модела. Као техника анализе коришћен је алгоритам “Стабло одлучивања“. На основу дефинисаног *Data Mining* модела могуће је генерисати и одређене графиконе који представљају преглед параметара који улазе у анализу. Изглед ових графикона је дат на слици 5.33.



Слика 5.33 – Графикони настали израдом *Data mining* модела

5.4.3. Израда корисничког интерфејса

Ова активност треба да обезбеди конзистентан, флексибилан интерфејс који, пре свега, испуњава захтеве корисника и организационе стандарде. Кориснички интерфејс је за крајњег корисника слика нашег система и веома је важно да буде флексибилан и да рефлектује стил рада корисника.

Избор алата за приказивање података

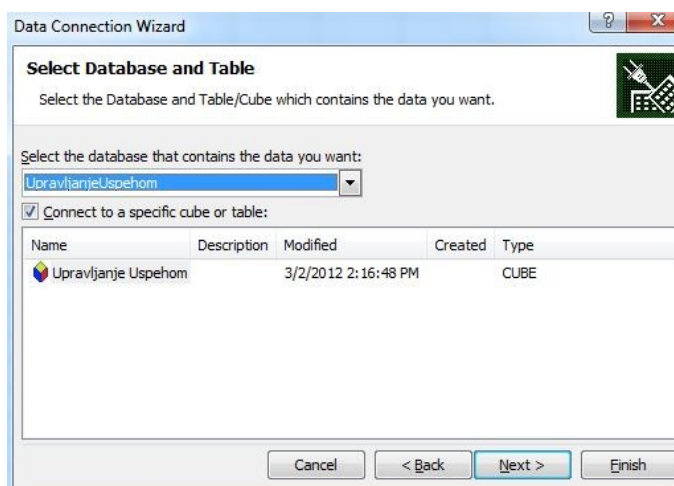
Анализа података организованим у OLAP коцке може да се врши коришћењем Pivot табела (динамичка табела са обједињени подацима) који омогућавају приступ подацима у OLAP коцкама. У нашем случају, одлучено је да се користи *Microsoft Excel* јер је то алат чија је основна намена анализа података (Microsoft класификација). Excel можемо сматрати потпуно функционалним клијентом, интероперабилним са *SQL Server 2008* аналитичким сервисима. Функционалност Excel –а се огледа у следећем:

- омогућава приступ подацима смештеним у OLAP коцкама. Excel омогућава Pivot табеле којима је могуће представити вишедимензионе податке кориснику. Такође, кориснику је омогућена даља обрада тих података. Све се то обрађује на серверу, а резултати су приказани и на серверу и на клијенту;
- омогућава и напредне могућности аналитичких сервиса као што су KPI и *Data Mining* (инсталирањем одговарајућих додатака).

Израда приказа

Као што смо рекли, анализа података, организованих у OLAP коцке, се врши изградом такозваних Pivot табела. Процес израде Pivot табеле у Excel-у, за приступ OLAP коцкама, врши се у следећим корацима:

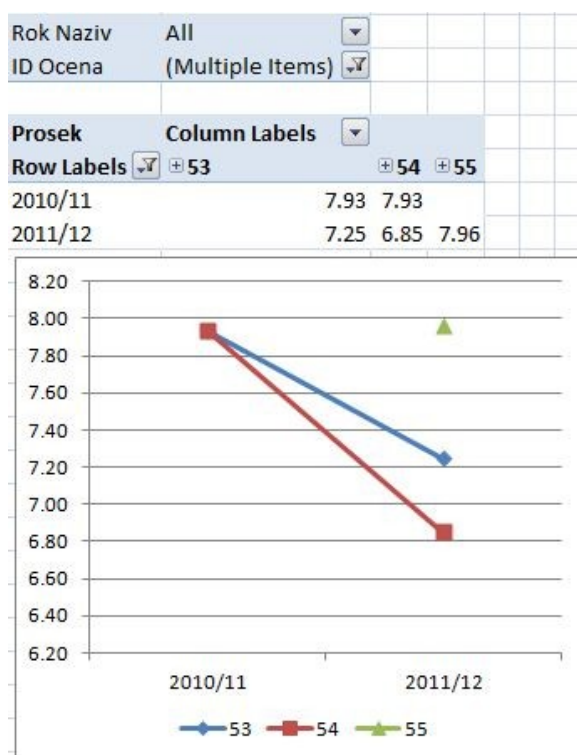
1. Одређивање локације података – бира се спољни извор података,
2. Повезивање са спољним извором података – наводи се пут до сервера OLAP коцке,
3. Формирање Pivot табеле – превођење димензија и података OLAP коцке у одговарајућа поља Pivot табеле (поље података, реда, колоне и стране).



Слика 5.34 – Приступ OLAP коцки

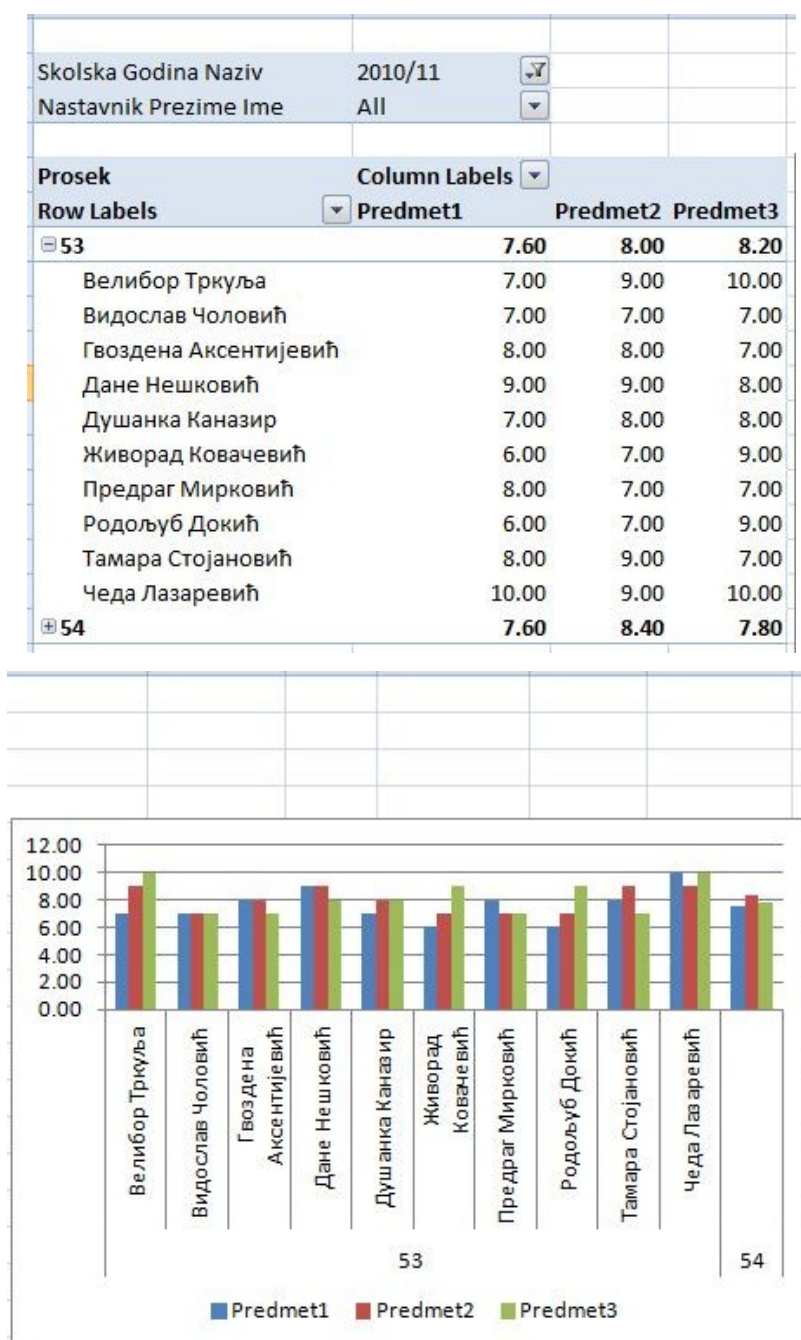
У даљем тексту биће приказане могућности вршења анализа у Excel-у, на основу дефинисане OLAP коцке, а за потребе управљања успехом кадета Војне академије.

На слици 5.35 је приказана Pivot табела која приказује кретање просека класе по школским годинама. На основу приказаних података управљање Војне академије може да закључи да је просек 53. и 54. класе смањен и да треба предузети мере на унапређење стања.



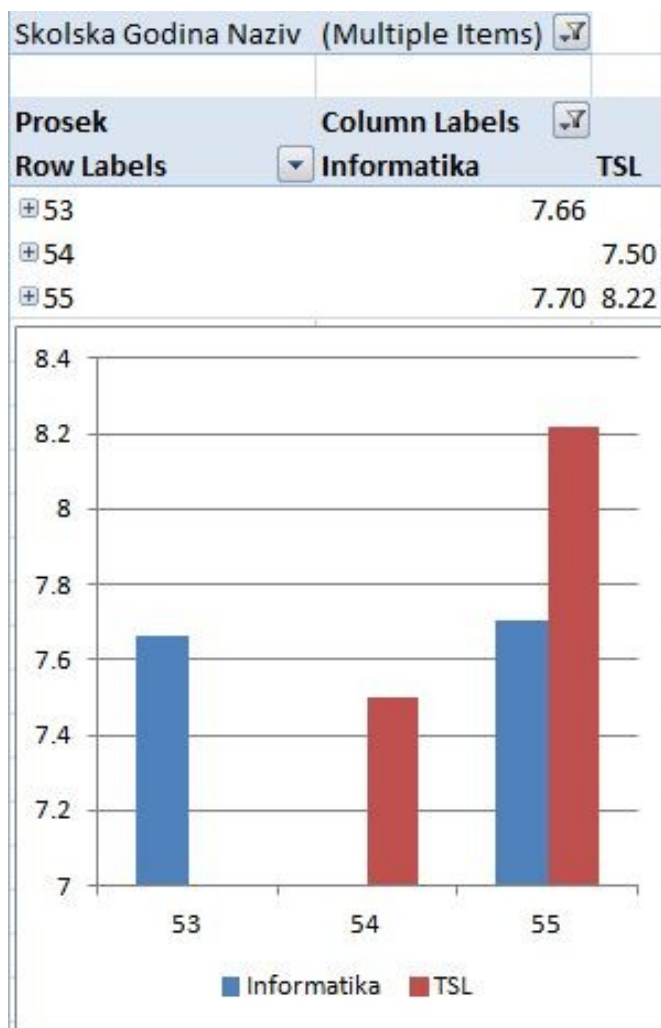
Слика 5.35 – Pivot табела Просек по школским годинама

На слици 5.36 је приказана Pivot табела која приказује кретање просека класе по појединим предметима. На основу приказаних података управљање Војне академије може да закључи да је просек на Предмету1 на граници прихватљивог (ако је то рецимо 7,5) и да треба предузети мере за унапређење стања. Коришћењем могућности које Pivot табеле пружају могуће је и детаљније сагледати оцене појединих кадета унутар класе за поједине предмете.



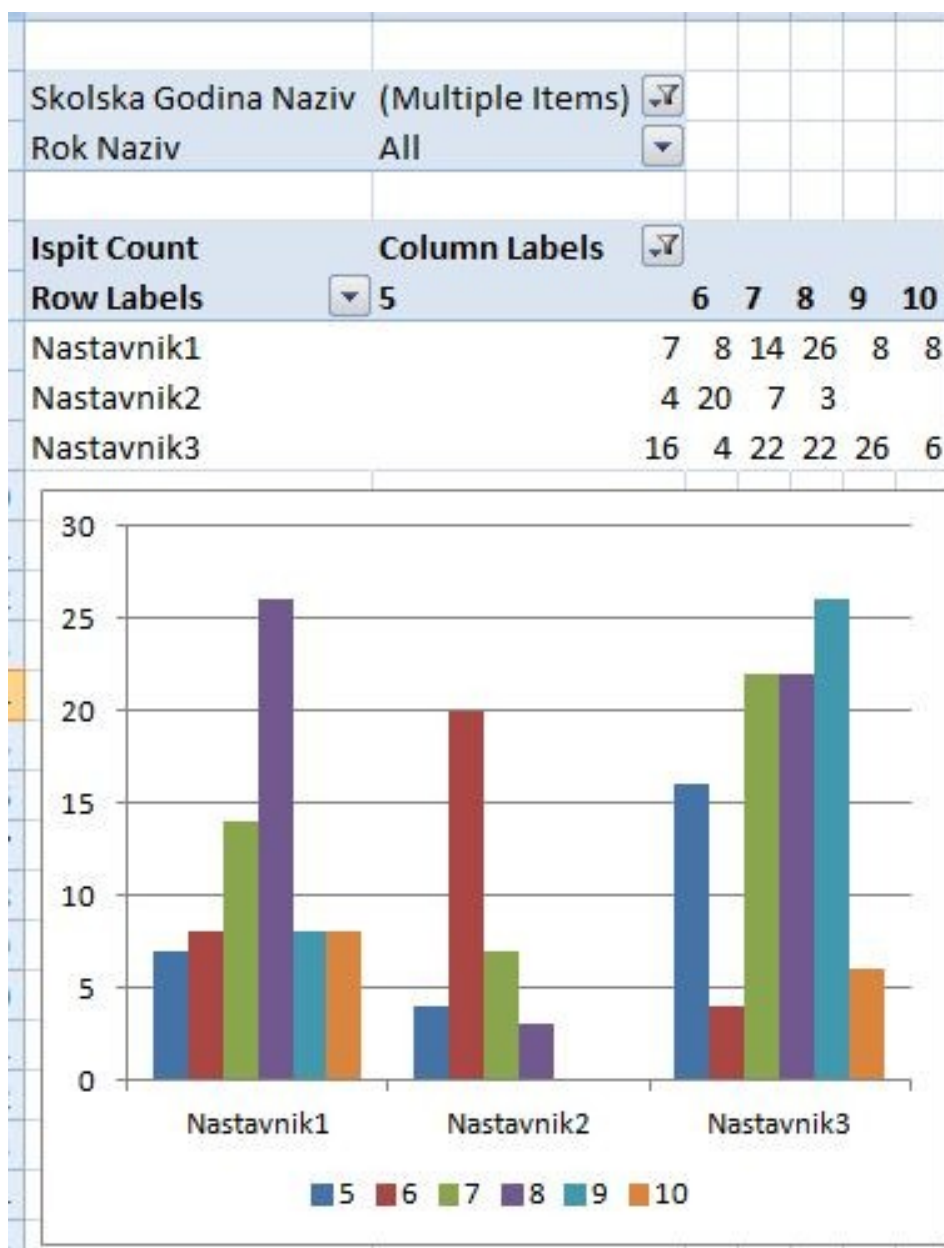
Слика 5.36 – Pivot табела и дијаграм Просек по предмету

На слици 5.37 је приказана Pivot табела која приказује кретање просека по класама и модулима. На основу приказаних података управљање Војне академије може да закључи да је просек 53. класе информатике већи у односу на 54. класу ТСл, а да је у 55. класи просек за модул ТСл знатно повећан и налази се на очекиваном нивоу.



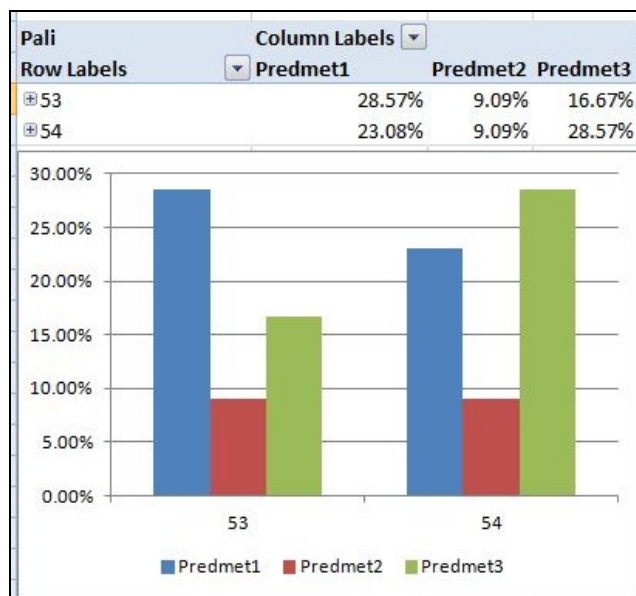
Слика 5.37 – Pivot табела Просек по класама и модулима

На слици 5.38 је приказана Pivot табела која приказује које оцене дају поједини наставници. На основу приказаних података управљање Војне академије може да закључи да Наставник2 уопште не даје оцене 9 и 10. На основу овог податка може се вршити даља анализа и утврдити разлог за овакво стање.



Слика 5.38 – Pivot табела Оцене по наставнику

На слици 5.39 је приказана Pivot табела која приказује проценат кадета који нису положили испите из појединих предмета. На основу приказаних података управљање Војне академије може да закључи да 53. класа има проблеме са полагањем Предмета1, а 54. класа са полагањем Предмета3. На основу овог закључка доноси се одлука, на пример за који предмет треба организовати додатни рад у појединим класама (модулима).



Слика 5.39 – Pivot табела Пало по предмету

На слици 5.40 је приказана Pivot табела за анализу успеха кадета. На овој табели су поред података приказани и KPI за праћење стања.

Skolska Godina Naziv	All			
Rok Naziv	All			
Column Labels	Prosek	Prosek Status	Prosek Trend	Broj ESBP
Row Labels	2011	2011	2011	2011
55	8.84	🟢	↑	0
54	9.02	🟢	↑	32
53	8.78	🟢	↑	35
Велибор Тркуља	9.00	🟢	↑	3
Видослав Чоловић	8.47	🟢	↑	4
Гвоздена Аксентијевић	8.00	🟢	↑	4
Дане Нешковић	8.86	🟢	↑	5
Душанка Каназир	6.50	🔴	↓	3
Живорад Ковачевић	7.00	🟢	↑	3
Предраг Мирковић	6.00	🔴	↓	2
Родољуб Докић	7.00	🟢	↑	3
Тамара Стојановић	9.21	🟢	↑	3
Чеда Лазаревић	8.56	🟢	↑	5
131		🔴	↓	
128		🔴	↓	
127		🔴	↓	

Слика 5.40 – Pivot табела за анализу успеха

Управљање Војне академије на основу претходне табеле може и визуелно да прати стање и по потреби предузима мере. На основу жутог светла за статус (просек између 7,00 и 8,00) и црвене стрелице за тренд (просек се смањио у односу на прошли период), код 53. класе потребно је предузети мере да се стање поправи.

Даљом анализом се могу установити детаљни разлози за овакво стање, односно врши се анализа по кадетима унутар класе. На овај начин се врло брзо долази до закључка који су кадети узрок смањења успешности (просека) класе.

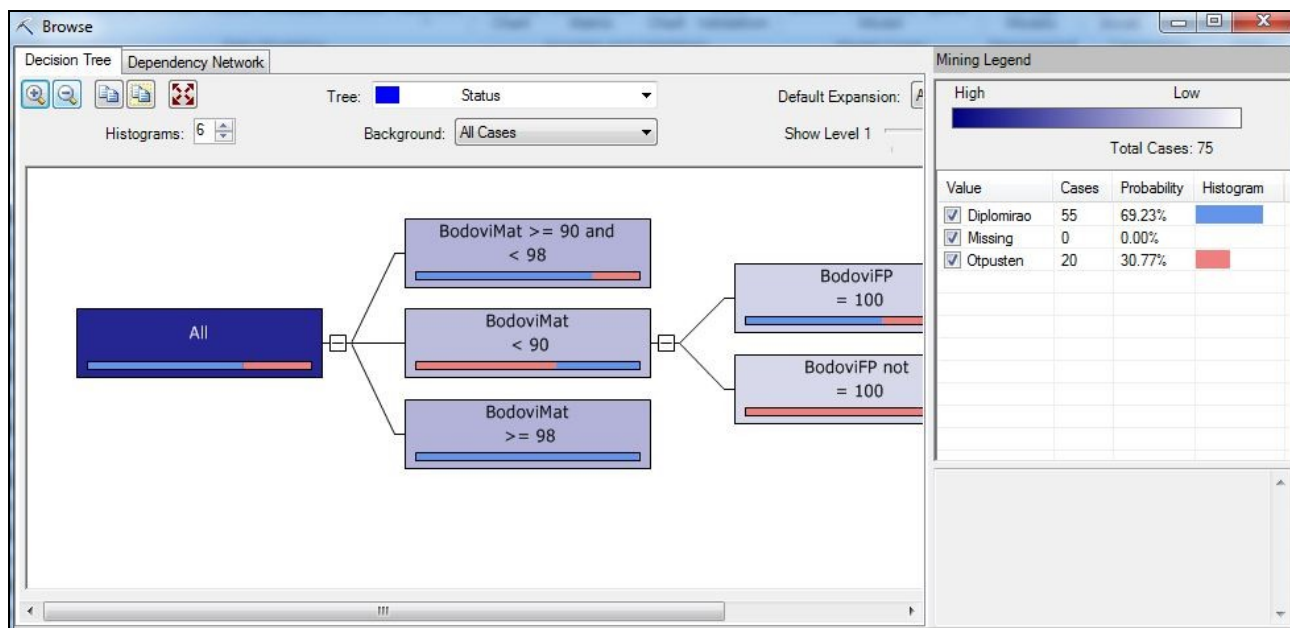
На слици 5.418 је приказана Pivot табела која приказује проценат кадета који нису положили испите по испитним роковима. На основу зеленог светла за статус (процент мањи од 20%) може се закључити да је све у реду. Међутим, црвена стрелица за тренд (повећава је проценат кадета који не полажу испите) долазимо до закључка да ипак треба предузети одређене мере за унапређење стања.

Predmet Naziv		All				
Godina		All				
Column Labels						
Април			децембар			
Row Labels	Pali Status	Pali Trend	Pali	Pali Status	Pali Trend	Pali
55			52.94%			6.67%
54			47.37%			
53			28.57%			
129						
128						
127						

Слика 5.41 – Pivot табела за анализу испита

Приказ *Data Mining* анализе података је реализован кроз *Data Mining* алате SQL сервера и MS Excel-а. Од мноштва алгоритама и алату за *Data Mining* анализу одлучено је да се као пример користе Стабла одлучивања и кластеровање.

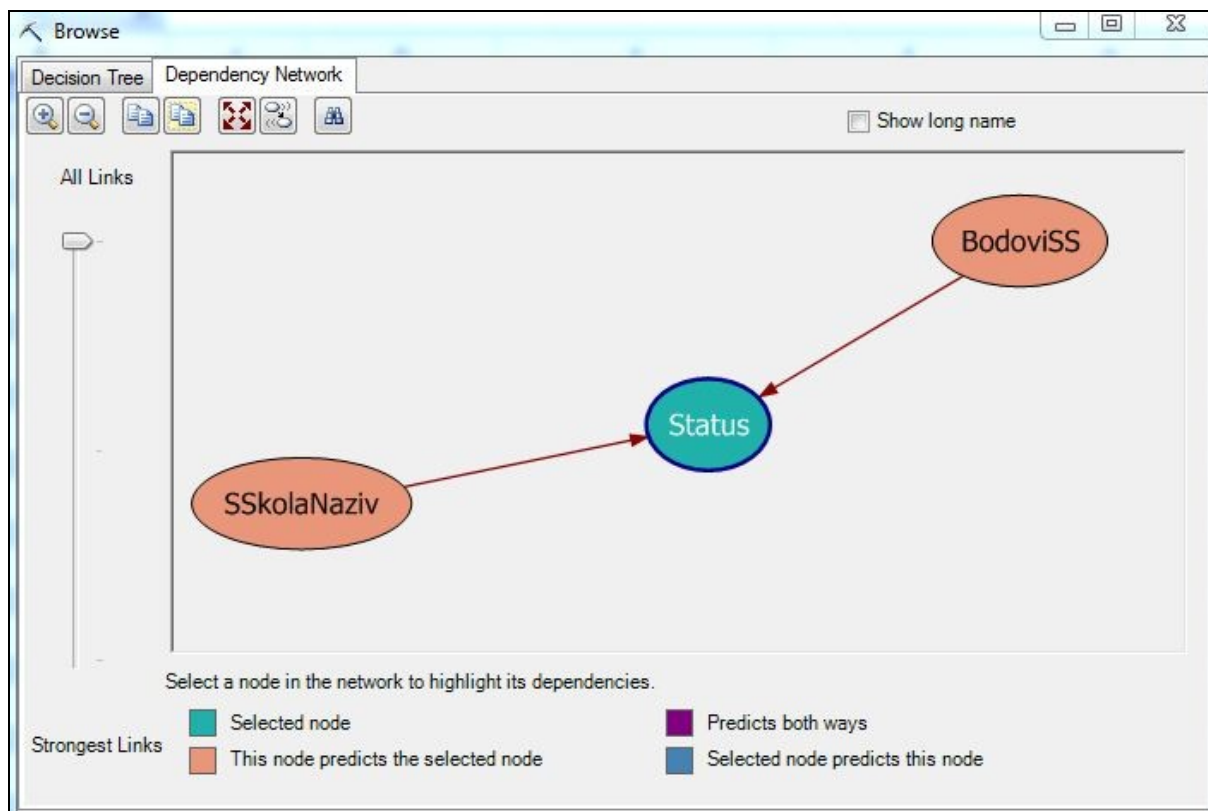
Слика 5.42. показује да највећи утицај на успешност студирања има број бодова који је кандидат освојио на тесту из математике. Наиме, сви кадети који су имали више од 98 бодова су дипломирали. На дијаграму се може видети и утицај резултата физичке провере јер су сви кадети који су имали мање од 90 бодова из математике али су имали 100 бодова из физичке провере, такође, дипломирали.



Слика 5.42 – Data Mining “Стабло одлучивања”

На основу ових података управљање Војне академије може да од самог почетка студирања предузима мере како би повећали успешност дипломирања кадета који имају предиспозиције за лошије резултате.

Други пример *Data Mining* анализе података су алгоритми мрежне зависности. На слици 5.43. приказан је дијаграм који нам говори да утицај на статус (да ли ће дипломирати или не) кадета има средња школа (Врста школе и број бодова) коју је кадет завршио. (Подаци се разликују у односу на слику 5.42 из разлога што је коришћена различита полазна база података)



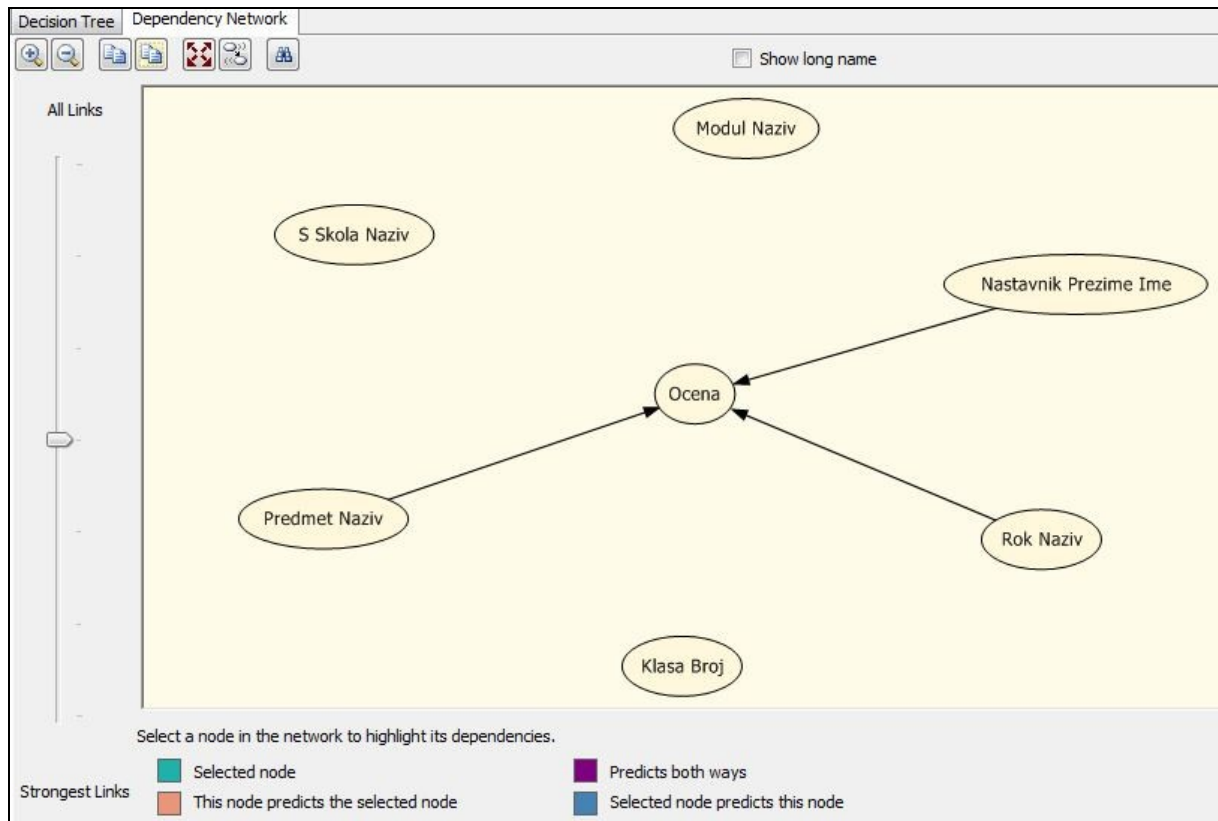
Слика 5.43 – Зависност статуса кадета

Померањем клизача на левој страни слике може се утврдити и који од ова два параметра има већи утицај на коначни исход.

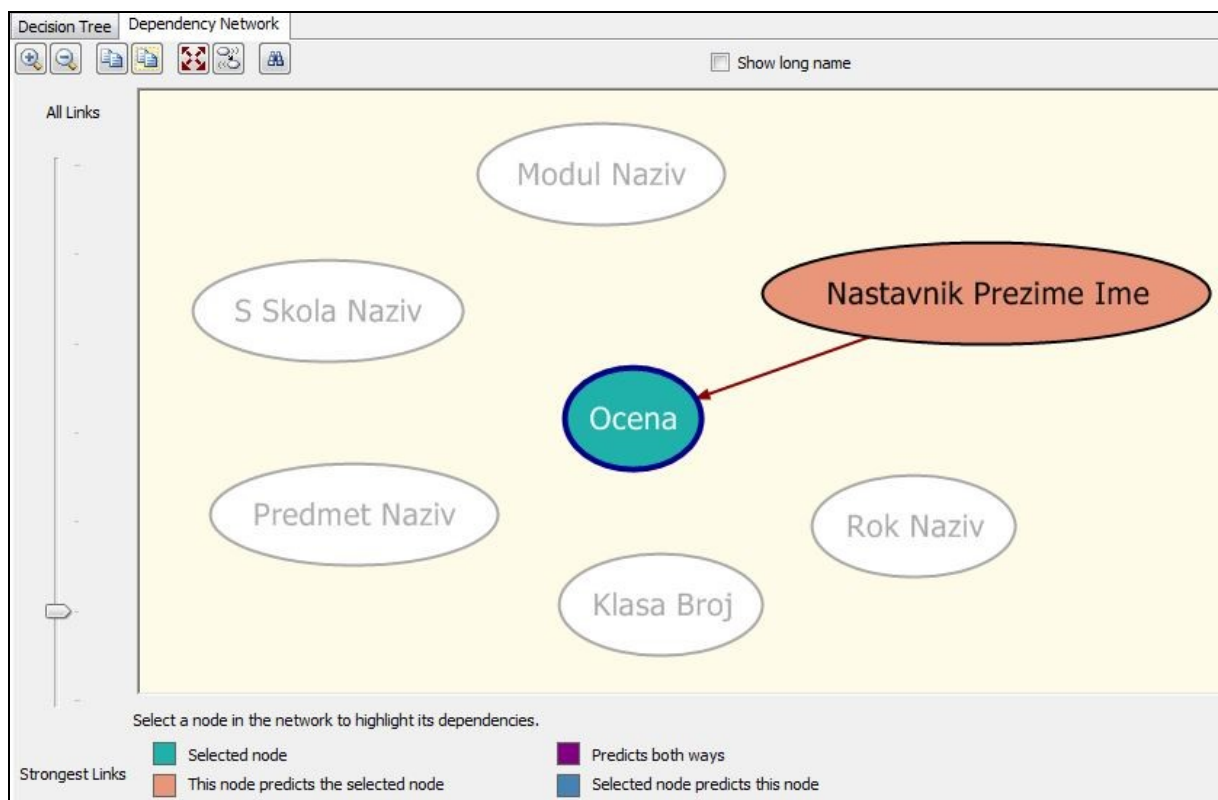
На слици 5.44. приказан је дијаграм који нам говори да утицај на оцену (коју ће оцену кадет добити на испиту) кадета имају следећи параметри:

- Завршена средња школа;
- Модул школовања;
- Предмет;
- Класа;
- Испитни рок;
- Наставник.

Померањем клизача на левој страни слике може се утврдити и који од наведених параметра има већи утицај на коначни исход, односно оцену кадета. На слици 5.45. можемо видети да највећи утицај на оцену има Наставник који врши оцењивање.

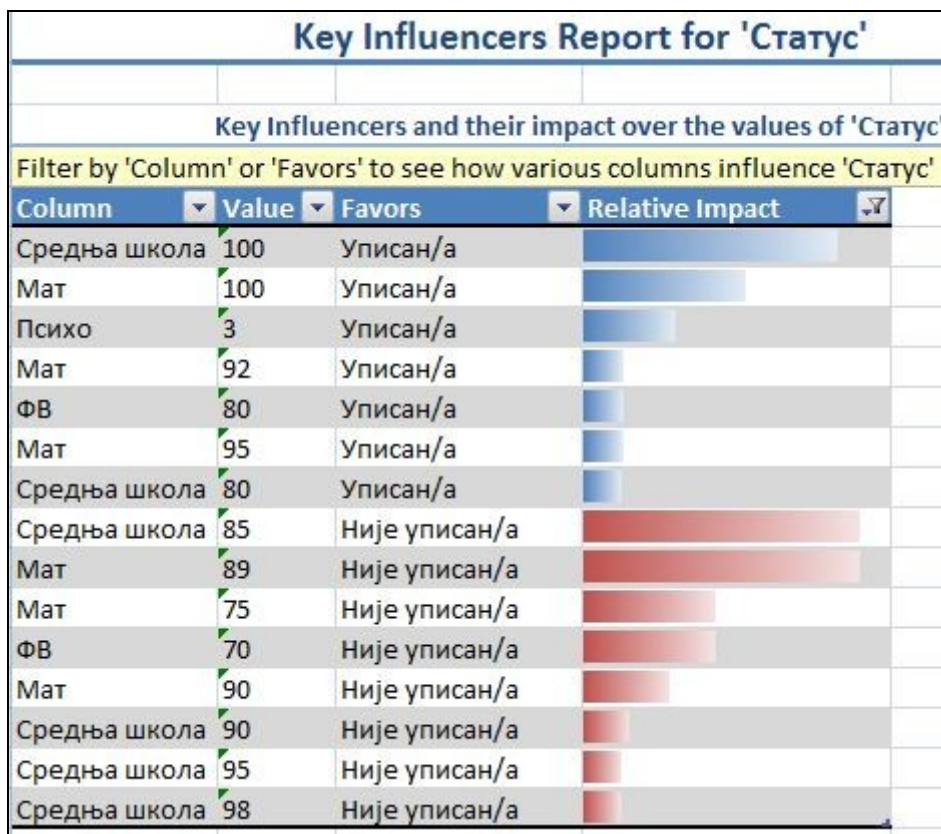


Слика 5.44 – Зависност оцене на испиту



Слика 5.45 – Зависност оцене на испиту

Data Mining анализа података је вршена и над подацима са уписа кадета на Војну академију. Основа за ову анализу је алгоритам стабла одлучивања. Ова анализа показује који су од улазних параметара пресудили приликом уписа. Анализом слике 5.45 долази се до закључка да су за упис најбитнији бодови из средње школе и резултати теста из математике.

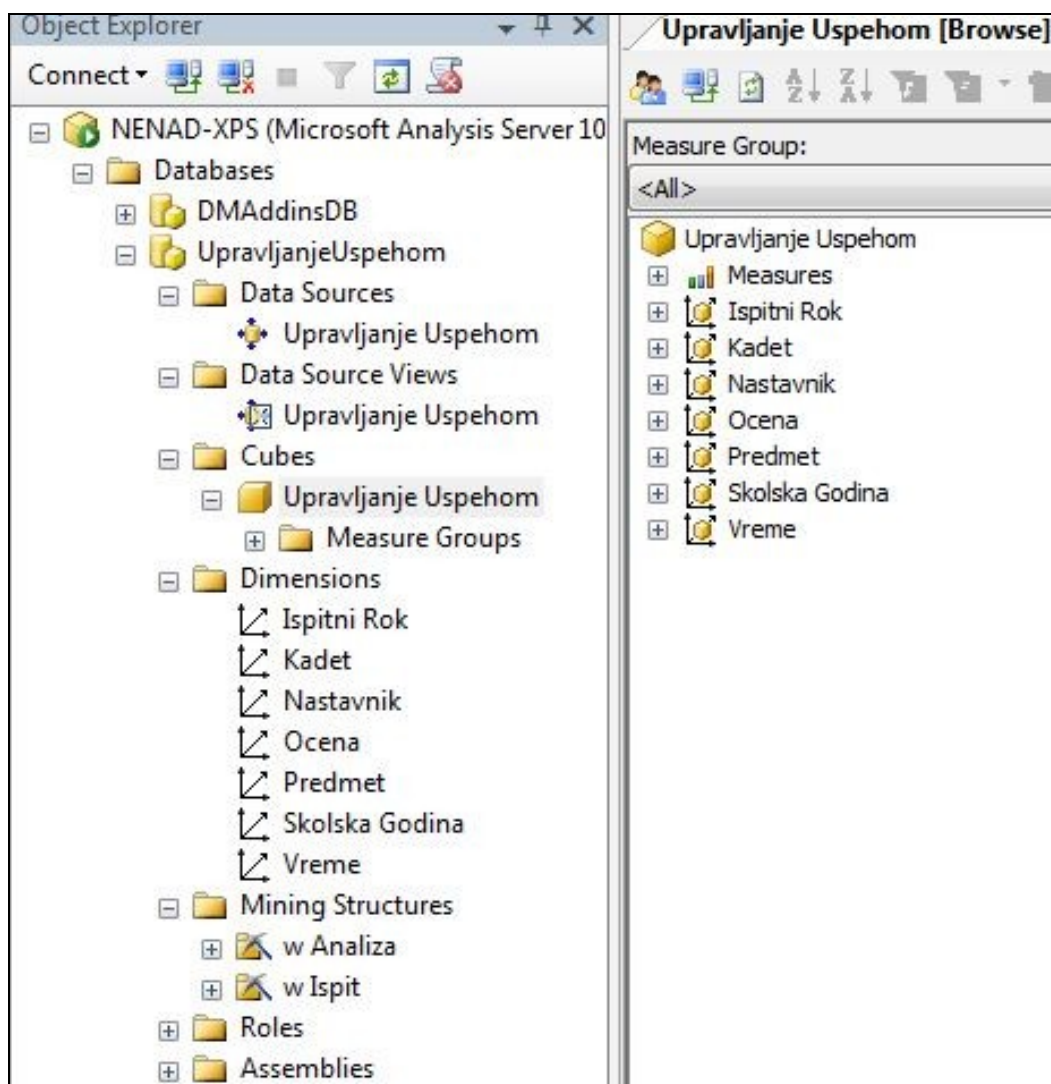


Слика 5.46 – Утицај појединих параметара на упис кадета на ВА

Приказане Pivot табеле и *Data Mining* анализа су само део од могућег скупа аналитичких алата које је могуће применити у управљању успехом кадета Војне академије. Примена конкретног алата односно параметара анализе зависи само од тренутних потреба за информацијама у процесу доношења одлуке од стране управљања Војне академије.

6. Доказ хипотеза

У оквиру реализације BI система за управљање успехом кадета Војне академије извршена је израда складишта података (слике 5.15, 5.16 и 5.23) које је савремено конципирано и засновано на аналитичком моделу података. На слици 6.1. је приказана структура аналитичке базе података која је аутоматски креирана на основу складишта података и дефинисане OLAP коцке. Оваква структура нам омогућава вршење предвиђених анализа података. Ове чињенице су уједно и **доказ посебне хипотезе да BI систем мора да је конципиран на аналитичком моделу података, односно концепту складишта података.**



Слика 6.1 – Аналитичка база података

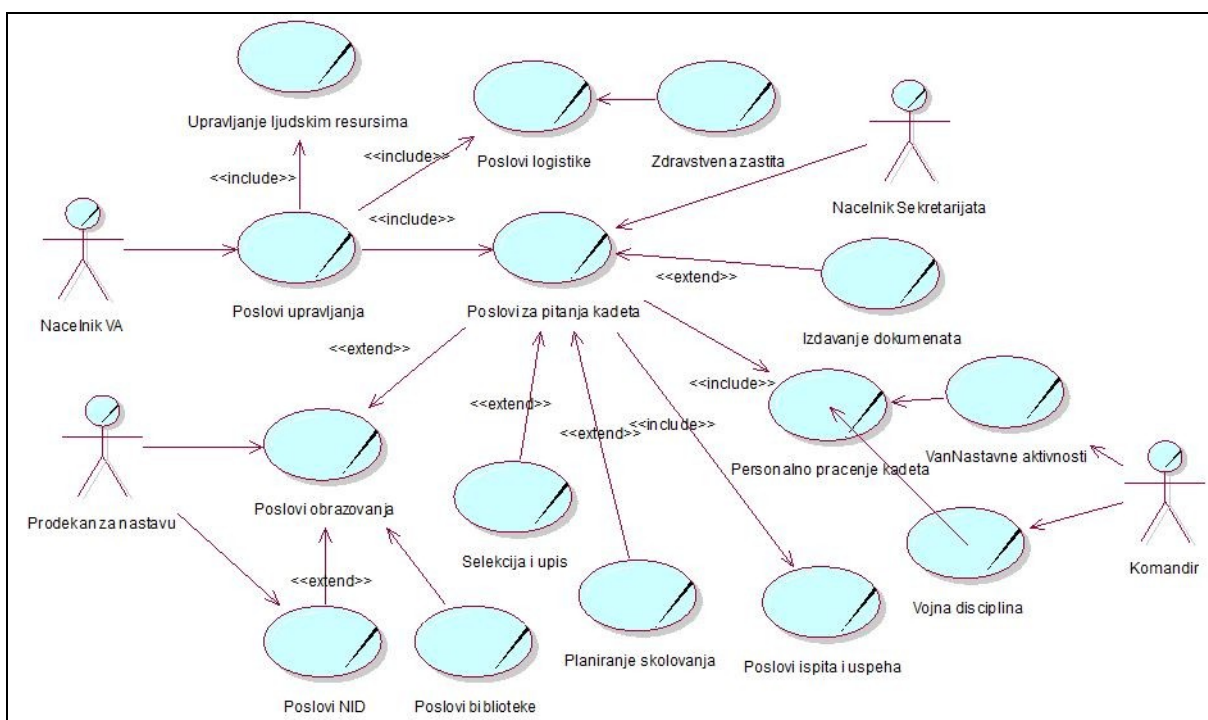
У ВІ систем за управљање успехом кадета Војне академије су укључени сви параметри из изворних система. На моделу OLAP коцке (слика 5.24 и 5.25) се може видети да се успех студената прати по више димензија истовремено. Тако дефинисане димензије су директна последица процене пословног окружења и анализе изворних података. У табели 6.1. је приказан преглед података који о кадетима постоје на Војној академији, који се тренутно користе у вршењу анализа и података које користи реализовани ВІ систем. Као што може да се примети одлучено је да се за анализу користе и подаци који се не налазе само у студентској служби већ код командира (Стимулативне и дисциплинске мере), лекара (поштеде), у библиотеци (број коришћених књига). На овај начин се жели обезбедити укључивање свих параметара из система и интеграција података из различитих извора.

Табела 6.1. Преглед података коришћених за анализу

Подаци о кадетима који се воде на ВА	Подаци који се тренутно користе за анализу успеха	Реализовани ВІ систем
Оцене са испита	Оцене са испита	Оцене са испита
Просек	Просек	Просек
Број положених	Број положених	Број положених
Број неположених	Број неположених	Број неположених
ЕСБП бодови	ЕСБП бодови	ЕСБП бодови
Успех из средње школе	-	Успех из средње школе
Ранг на пријемном испиту	-	Ранг на пријемном испиту
Завршена средња школа	-	Завршена средња школа
Наставник који је реализовао испит	-	Наставник који је реализовао испит
Испитни рок	Испитни рок	Испитни рок
Време реализације испита	-	Време реализације испита
Успешност дипломирања	-	Успешност дипломирања
Пол кадета	-	Пол кадета
Број поштеда	-	Број поштеда
Ван наставне активности	-	Ван наставне активности
Коришћење библиотеке	-	Коришћење библиотеке
Дисциплинске мере	-	Дисциплинске мере

На основу наведених чињеница, очигледно је да реализовани систем пружа могућност свеобухватне анализе и доношење одлука заснованог на доступним улазним параметрима. Оваквом поставком података је **доказана и хипотеза да VI систем мора да омогућава анализу и поређење свих података релевантних за доношење одлуке.** Додатак овој тврдњи представљају и Pivot табеле и Data mining анализа (слика 5. 35 до 5.45) реализоване у оквиру израде корисничког интерфејса.

Као што смо већ рекли, у раду је коришћен шири скуп података, из различитих извора, од онога које менаџмент Војне академије тренутно користи за праћење успеха кадета. Поред доказа хипотезе о поређењу свих релевантних података, коришћење ширег скупа података биће искоришћено за доказ хипотезе о скалабилности. Како је предложена методологија заснована на принципима инкременталности и итеративности, и најбољој пракси коју доноси RUP, у проширивању основног скупа података кренуло се додатном анализом пословних случајева употребе.



Слика 6.2 – Проширени модел пословних случајева употребе

На слици 6.2. приказани су пословни случајеви употребе, на којима треба уочити проширења која се односе на: Послове библиотеке, Војну Дисциплину, Ван Наставне активности и Здравствену заштиту.

На основу наведених проширења јасно је да случај употребе „Учитавање података“ са лике 5.13 мора да обезбеди читавање података из више извора, односно да постоји потреба поновног спровођења итерације која се односи на анализу изворних података.

Када говоримо о подацима везаним за коришћење библиотеке, велика предност је постојање информационог система за послове библиотеке на Војној академији, односно подаци о коришћењу књига су структурирани и релативно једноставно им је приступити.



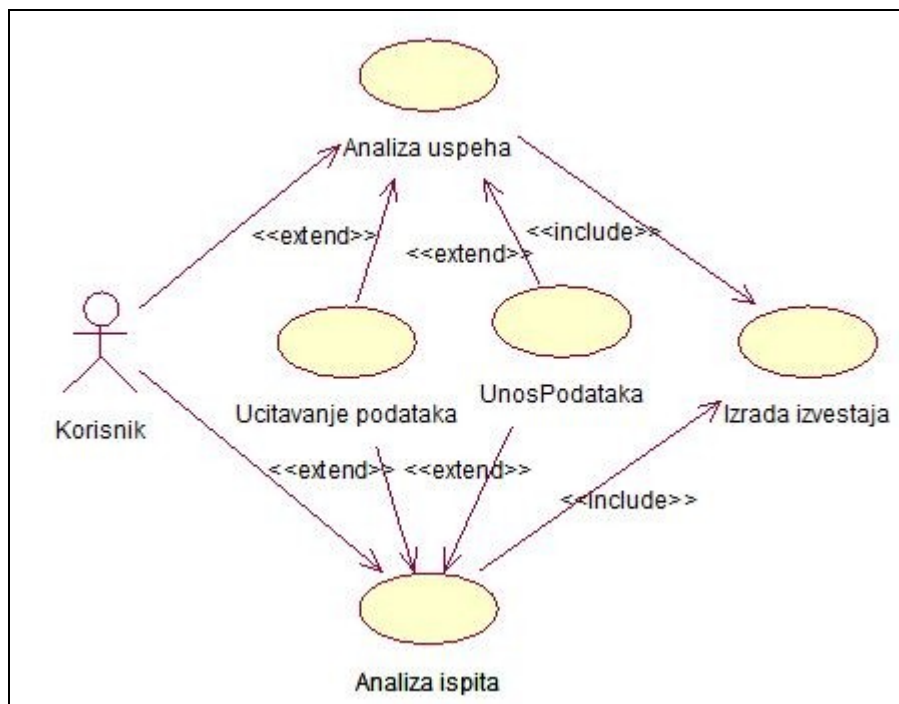
Слика 6.3 – Модел изворних података за задужење у Библиотеци

На основу слике 6.3 можемо закључити да се у информационом систему библиотеке на Војној академији налазе сви потребни подаци на основу којих, израдом једноставног упита, можемо доћи до податка колико је књига у одређеном периоду користио поједини корисник (у нашем случају за Својство=Кадет).

Када се ради о подацима о Броју поштеда, Дисциплинским мерама и Ван наставним активностима ситуација је потпуно другачија. Наиме наведени подаци су делимично структурирани али на нивоу папира, односно евиденција лекара и командира. Тако на пример податке о поштедама са наставних активности можемо наћи у здравственом картону кадета, здравственој књижици и дневнику реализације наставе.

Подаци о дисциплинским мерама се налазе у неформалним персоналним картонима кадета и у рапортној књизи. Највећи проблем се јавља са подацима о Ван наставним активностима. Наиме ти подаци се налазе код руководиоца појединих секција на Војној академији и делимично у неформалним персоналним картонима кадета које воде командири.

Због наведене ситуације постоји потреба доградње предложеног решења како би омогућили обухват додатних података. Једна од могућности је постојање посебног случаја употребе, модула будућег система, за ручни унос података у складиште.

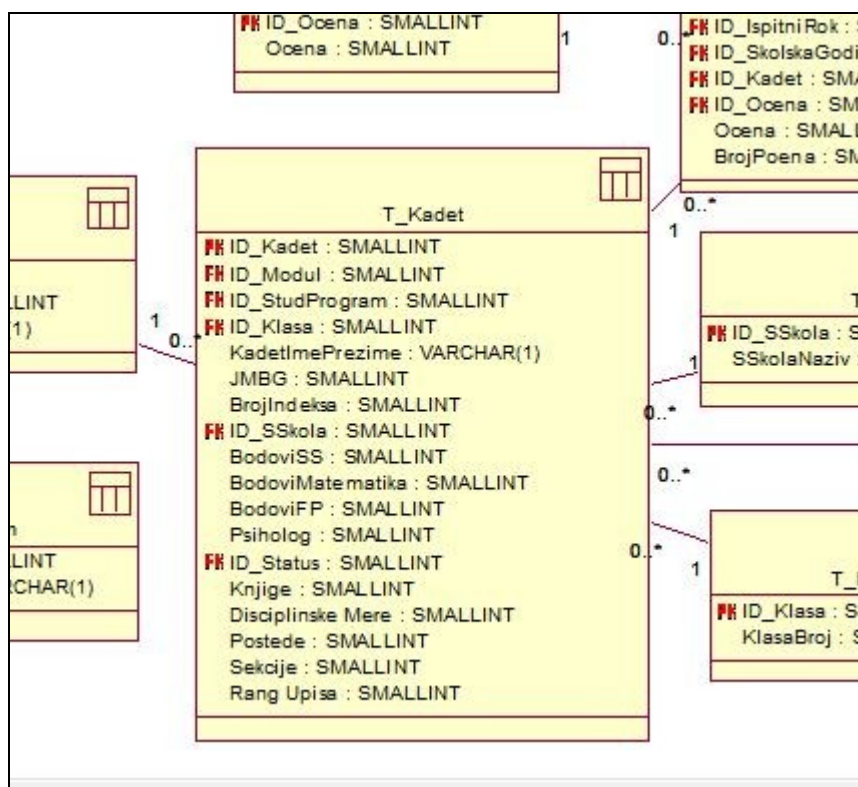


Слика 6.4 – Проширени модел случајева употребе

Након анализе изворних података, потребно је обезбедити интеграцију тих података у систем кроз редизајн складишта података и припрему података за учитавање. Иначе, интеграција података из различитих извора је тренутно један од најактуелнијих праваца истраживања у области пословних информационих система. Циљ ових истраживања је развој технологија и алата за интеграцију података из различитих извора. Извори података могу бити конвенционалне базе података, или Веб странице, текстуалне датотеке, електронска пошта и остали типови извора информација.

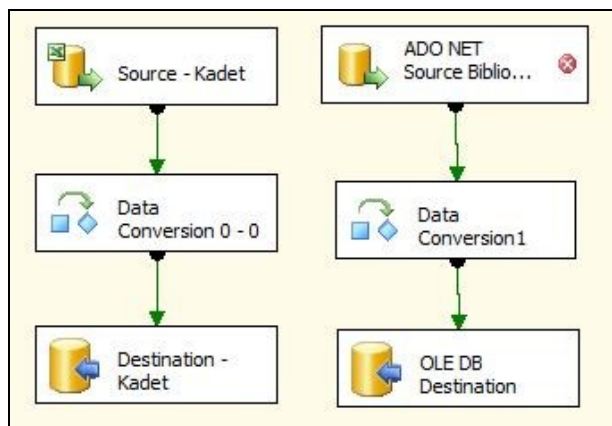
Када говоримо о редизајну складишта података он је извршен изменама у табели Кадет, тако да она садржи проширени скуп следећим колонама:

- Књиге, садржи податке о броју коришћених књига у библиотеци,
- Дисциплинске мере, ДА за постојање дисциплинских НЕ за стимулативне мере,
- Поштеде, број поштеда у одређеном временском периоду,
- Секције, број секција чији је кадет члан.



Слика 6.5 – Редизајнирана табела „Кадет“

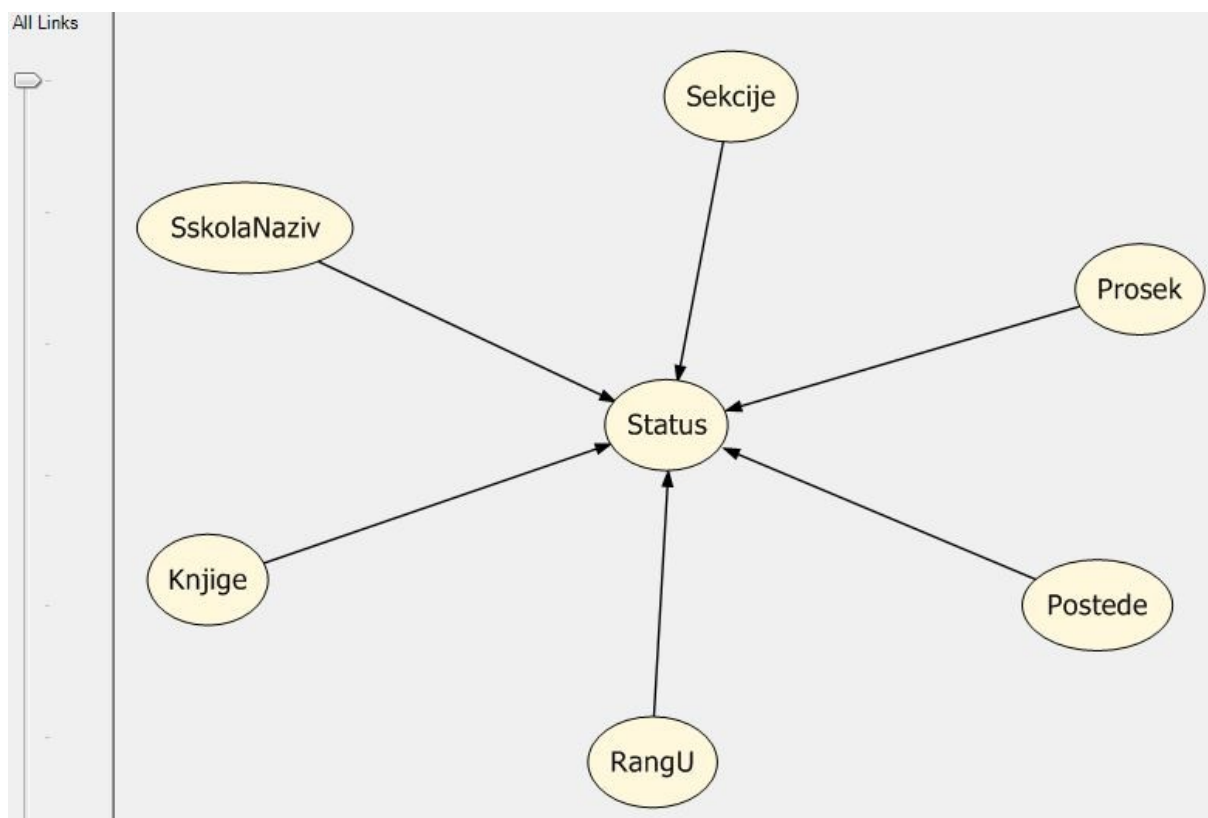
Како је за реализацију BI система за управљање успехом кадета Војне академије изабран је *SQL Server 2008*, искоришћене су све његове предности приликом интеграције односно учитавања података. Наиме, *Microsoft SQL Server Integration Services* пружају скалабилну платформу за интеграцију података са изузетним *Extract, Transform, Load (ETL)* и интеграционим могућностима, омогућавајући једноставније управљање подацима из различитих врста извора.



Слика 6.6 – Проширени пакет за припрему података

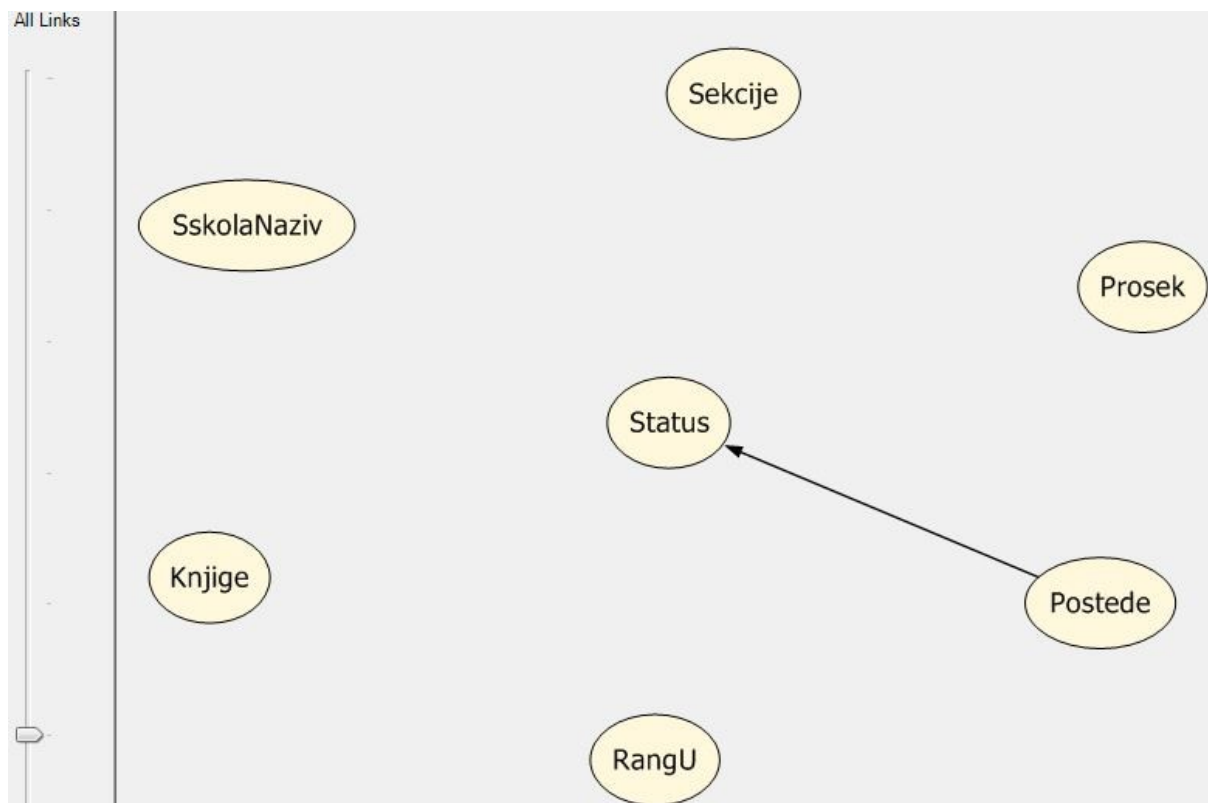
На слици 6.6. је приказан проширени пакет за припрему података, који је надграђен у делу који се односи на учитавање података из информационог система Библиотеке на Војној академији. Као кључна веза приликом интеграције података је искоришћен ЈМБГ који једнозначно одређује кадета и омогућује повезивање података из више извора. ЈМБГ је искоришћен и као веза ка подацима о Поштедама, Дисциплинским мерама и Ван наставним активностима, међутим као проблем остаје чињеница да се ти подаци не налазе у електронском облику тако да је њихово аутоматизовано учитавање у складиште практично немогуће. Како могућност остаје њихов ручни унос или претварање у електронски облик (XML, Flat File) и накнадно учитавање применом интеграционих сервиса *Microsoft SQL Server*-а.

Како би ипак обезбедили анализу утицаја и ових података на успех кадета, одлучено је да се, у научно-истраживачке сврхе, они аутоматски генеришу на основу података за 126. класу Смера Службе информатике из периода 2001-2003 година. На овај начин је извршено генерисање података за 10.000 кадета Војне академије. Након тога је применом *Data Mining* алгоритма мрежне зависности извршена анализа утицаја, свих података о кадету, на успешност дипломирања и просек студија.



Слика 6.7 – Зависност успешности дипломирања

Након проширивања скупа димензија који представљају одређене чињенице о кадету, одмах је уочљиво да се зависност статуса кадета (да ли ће дипломирати или не) од тих чињеница променио у односу на ситуацију на слици 5.43. Наиме, поред повећања броја фактора који утичу на дипломирање, највећа разлика је у томе што највећи утицај има број поштеда током школовања.



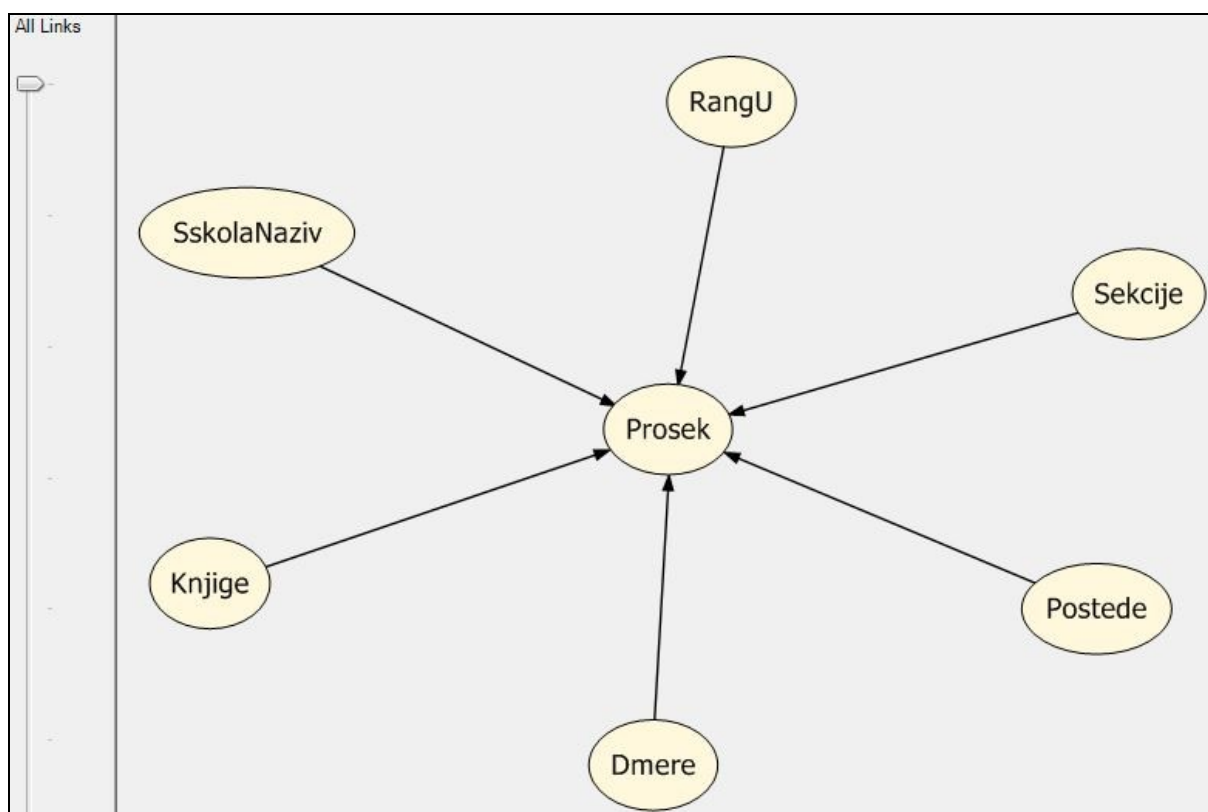
Слика 6.8 – Зависност успешности дипломирања

Овакав резултат можемо тумачити и да кадети су кадети који нису имали поштеде са наставе преданији циљу, односно успешном дипломирању на Војној академији. Рангирање свих чињеница које утичу на дипломирање кадета је следеће:

1. Број поштеда,
2. Завршена средња школа,
3. Ранг-резултат приликом уписа,
4. Чланство у секцијама,
5. Број коришћених књига,
6. Постојање дисциплинских, односно стимулативних мера.

Када се ради о утицају појединих димензија-чињеница на просек кадета ситуација је приказана на наредној слици. Иначе, и у овом као и у претходном случају у вези успешности дипломирања као улазне димензије-чињенице коришћено је следеће:

- Успех-ранг приликом уписа Војне академије
- Завршена средња школа
- Пол
- Број поштеда
- Чланство у секцијама
- Број коришћених књига у библиотеци
- Постојање дисциплинских, односно стимулативних мера.

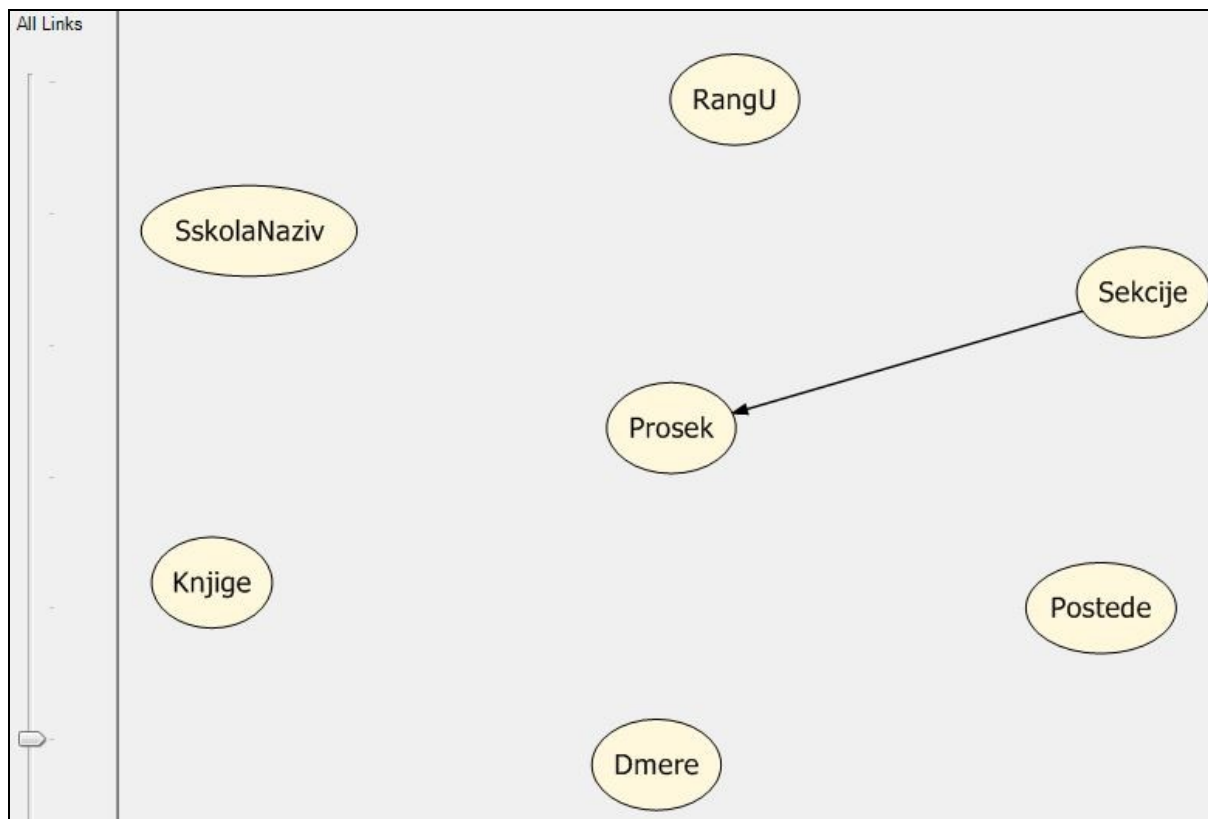


Слика 6.9 – Зависност просека студирања

Рангирање свих чињеница које утичу на просек кадета је следеће:

1. Чланство у секцијама,
2. Завршена средња школа,
3. Број коришћених књига,

4. Број поштеда,
5. Ранг-резултат приликом уписа,
6. Постојање дисциплинских, односно стимулативних мера.



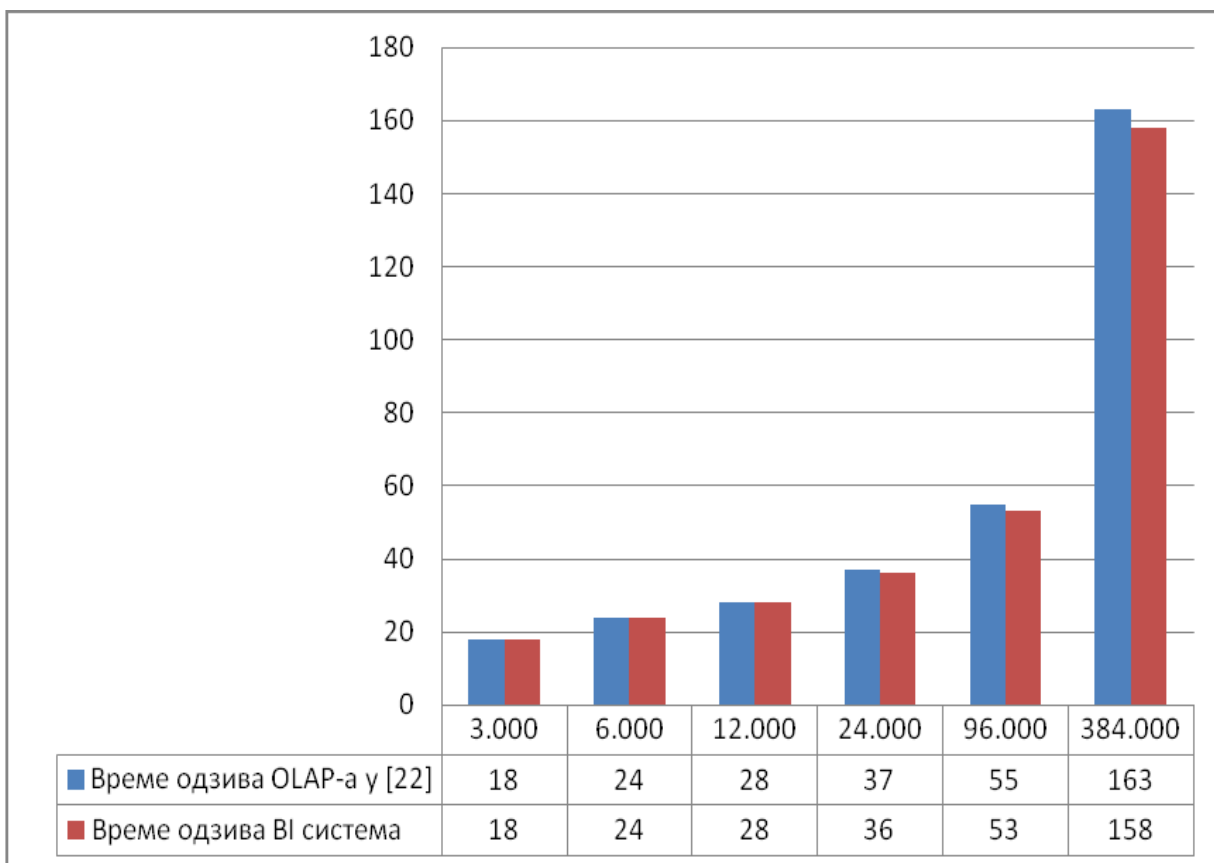
Слика 6.10 – Зависност просека студирања

Наведене резултате је интересно поредити са онима који су добијени на Факултету организационих наука у Београду [14]. Наиме на ФОН-у је један од параметара који утиче на просек пол студента, што на Војној академији није случај.

Добијени резултати, анализом проширеног скупа података из више различитих извора, говори и о чињеници који све параметри морају бити узети у обзир приликом рангирања кадета. Наведеним примерима је показано да подаци из студентске службе нису једини меродавни за успех-ранг кадета. Ово упућује на потребу факторисања утицаја свих расположивих чињеница о кадету, на основу чега би рангирање било далеко објективније него узимањем само просека како параметра.

У циљу додатне провере скалабилности са аспекта количине података вршено је и тестирање одзива упита у реализованом складишту података применом алата за

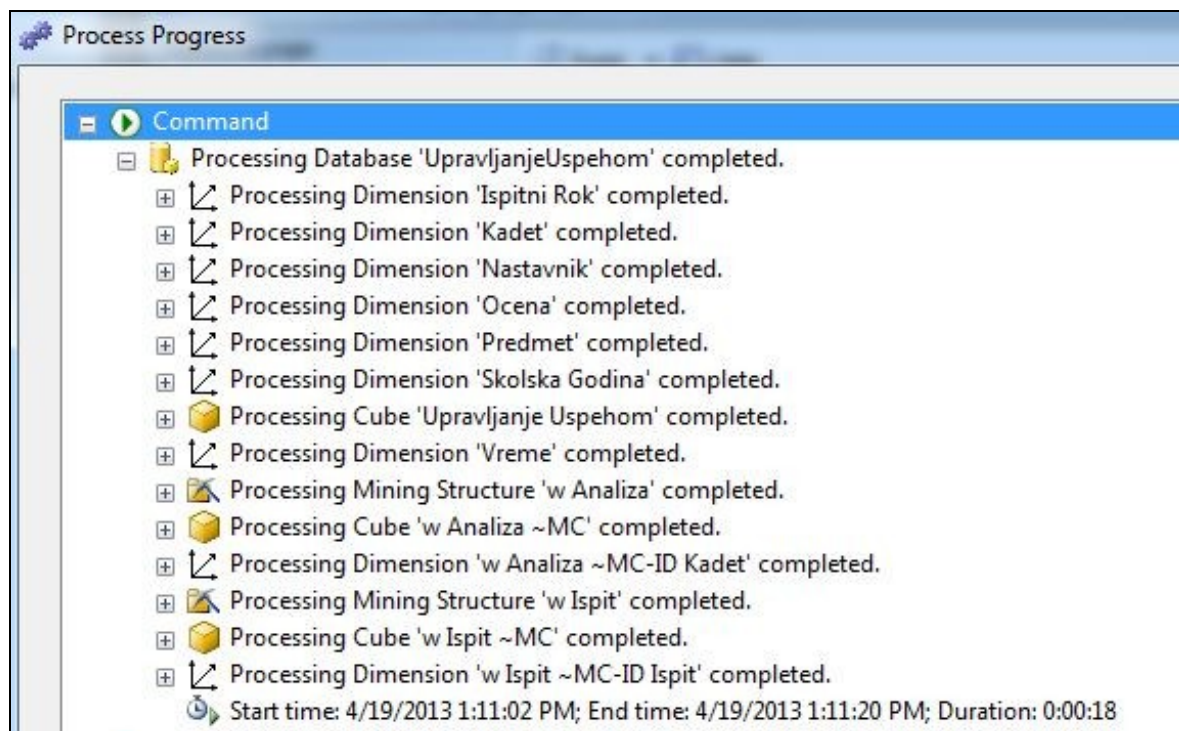
статистику који постоји у SQL серверу. Како би проверили ефикасност предложеног решења за анализу је коришћена иста количина података као у складишту података које је реализовано у раду “Развој *on-line analytical processing* (OLAP) система за подршку одлучивању“ [22].



Слика 6.11 – Поређење одзива упита у OLAP и BI систему

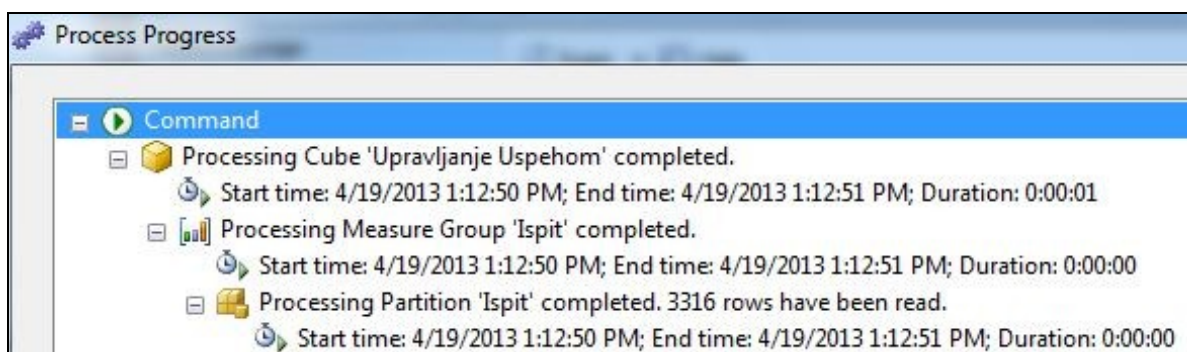
Добијени резултати показују да је предложено решење такође боље од система са трансакционом базом података. Мерењем су постигнути и бољи резултати код веће количине података, што је добар показатељ имајући у виду свакодневни тренд раста података у постојећим базама.

Тестирање је вршено и мерењем времена процесирања реализоване аналитичке базе у зависности од броја слогова у бази. Добијене вредности су се кретале од 15 до 20 секунди без обзира на количину података.



Слика 6.12 – Процесирање аналитичке базе података

Такође, тестирање је вршено мерењем времена процесирања реализоване OLAP коцке базе у зависности од броја слогова у бази. Добијене вредности су се кретале од 1 до 3 секунде без обзира на количину података која се налазила у бази.



Слика 6.13 – Процесирање OLAP коцке

Резултати у наведеним границама су добијени и експериментисањем мењањем броја димензија и чињеница у постојећој аналитичкој бази.

Имајући у виду наведена проширења, тестирања, као и карактеристике SQL Server-а, детаљно описане у оквиру избора технологије реализације (страница 121.), везане за

склабилност, као и чињеницу да коришћење CASE алата обезбеђује једноставну надоградњу система, **доказана је и посебна хипотеза везана за скалабилност.**

Као кључно за развој ВІ система за управљање успехом кадета Војне академије битно је напоменути да су коришћени кораци методологије предложене у четвртом поглављу. Ова методологија у себе интегрише класичне алате и стандарде, димензионо моделовање и објектно-оријентисани приступ у развоју ВІ система. ВІ систем је развијен кроз низ корака у којима је извршено дефинисање захтева, њихова анализа, и дизајн и реализација самог система. У делу реализације ВІ система коришћене су све предности интеграционих и аналитичких сервиса SQL Server-а.

Реализацијом истраживања постојећих методологија развоја информационих система (треће поглавље), истраживањем постојећих методологија развоја ВІ система и предлогом и оценом корака развоја ВІ система (четврто поглавље, посебно део 4.3), што је уједно и циљ истраживања, дат је конкретан научно-истраживачки допринос у области информационих система, односно система за подршку у одлучивању.

На крају, на основу потврђених посебних хипотеза везаних за *Bussines Intelligence* систем за управљање успехом кадета Војне академије, може се закључити да предложена методологија развоја *Bussines Intelligence* система омогућава дефинисање и анализу корисничких захтева, дизајн и имплементацију решења за анализу података ради доношења одлука потребних управљању, што доказује основну хипотезу.

7. Закључак

Убрзани развој информационах и комуникационих технологија диктирао је захтеве који су се постављали пред развојем информационах система. У почетку њиховог развоја и имплементације то су били искључиво системи базирани на трансакционим базама података, чији је основни задатак био да подрже свакодневне трансакције и обраду података. Данас је за опстанак на тржишту и остваривање стратешке предности неопходан такав концепт информационог система који ће, поред трансакционих, укључивати и аналитичке базе података, односно који ће обезбедити извлачење информација из трансакционих података и екстракцију „знања“ из тих података за потребе одлучивања.

Модерно управљање великим системима захтева правовремено доношење одлука. Време за доношење одлуке је оптерећено великом количином података које је потребно обрадити и пренети. У предности су они који доносе одлуке засноване на правовременим, поузданим и тачним подацима и информацијама.

Научни допринос овог рада лежи у предложеним корацима развоја ВІ система који интегришу класични и објектно-оријентисани приступ развоја информационах система. Иако се интеграција класичних метода развоја информационах система са објектно оријентисаним приступом већ примењује дужи низ година показано је да постоји потреба даљег истраживања у овој области када је у питању развој ВІ система, поготову када се ради о пројектима мањег обима. Такође, предложена методологија омогућава да паралелном применом класичних метода и стандарда са објектно-оријентисаним извучемо најбоље из сваке од њих, односно да обезбедимо пре свега најбоље могуће дефинисање корисничких захтева и њихову анализу. Такође, дефинисана методологија, својом флексибилношћу, омогућава ефикасан развој ВІ система у оптималном року.

Имајући у виду поређење предложене методологије са постојећим методологијама за развој ВІ система и практичну реализацију ВІ система за управљање успехом кадета Војне академије можемо закључити и да је предложена методологија ваљана за развој ВІ система и да представља и најбољи избор за реализацију пројеката мањег обима када већ постоји искуство везано за предмет ВІ система.

Практичан допринос лежи у чињеници да је резултат дисертације VI систем који омогућава управљање успехом кадета Војне академије. Увођењем VI система у високошколске институције омогућава се једноставно и брзо добијање аналитичких извештаја о успеху студената, који су од великог значаја за праћење квалитета наставног процеса, поготову у периоду када високошколске институције пролазе кроз период великих структуралних и организационих промена. VI систем ставља информационе ресурсе на располагање доносиоцима одлука који могу у сваком тренутку, интерактивно да добијају жељене извештаје без подршке ИТ сектора. Тиме се значајно смањује временско кашњење између идентификације евентуалних проблема и предузимања неопходних акција како би се ти проблеми решили. Сам систем је пре свега намењен менаџменту Војне академије, али га уједно могу користити студентска служба, командири, кадети, односно сви заинтересовани за успех кадета Војне академије.

Такође, допринос дисертације је и у чињеници да предложено решење управљање успехом кадета Војне академије у анализи користи знатно већи проценат расположивих димензија-чињеница о кадету. Наиме, тренутно се за анализу успеха кадета Војне академије користи свега 46% расположивих чињеница. Када се у обзир узме и проширење настало интеграцијом података ван студентске службе тај проценат је још мањи и износи 35%. Све ово, а поготову добијени резултати о утицају појединих димензија-чињеница на успех кадета, говори у прилог потреби да се рангирању кадета мора приступити свеобухватније применом свих расположивих информација, а не само на основу просека оцена. Наравно постоји потреба јасног дефинисања утицаја појединих информација на ранг кадета, кроз њихово факторисање, анализом утицаја на успех што нам и пружа предложено решење за управљање успехом кадета Војне академије.

Поред наведених, резултати рада на дисертацији су:

- анализа кључних методологија и стандарда за развој информационих система;
- анализа постојећих методологија развоја *Business Intelligence* система;
- функционални и информациони модел информационог система Војне академије, реализован применом IDEF0 и IDEF1X стандарда и UML-а као језика за објектно оријентисано моделовање;

- складиште података (савремено конципирано на аналитичком моделу података), *Business Intelligence* система Војне академије;
- свестрана анализа расположивих података о успеху кадета и њихово праћење у реалном времену;
- скалабилан и поуздан *Business Intelligence* систем.

На самом крају треба нагласити да рад нуди оригинално решење ВІ система за управљање успехом кадета Војне академије. Наведено решење показује да је, иако су ВІ системи намењени пре свега за анализу великих количина података, могућа и њихова примена за анализу релативно мале количине података. Ово решење се може самостално или у комбинацији са радом “Развој *on-line analytical processing* (OLAP) система за подршку одлучивању“ применити на Војној академији. Наравно, научно истраживачки рад у ово области треба наставити у правцу даљег усавршавања методологије развоја ВІ система и примене нових техника за интеграцију и анализу података, пре свега даље истраживање примене *Data Mining* алгоритама и техника. Како наведено решење није затворено, могућа је његова надоградња модулима за интеграцију података са новим информационим системом ВА, новим моделима анализе података и прилагођавање, у складу са евентуалним захтевима корисника.

8. Литература

Литература непосредно навођена у раду:

1. Балабан, Н., Ристић, Ж.: “Пословна интелигенција“, ЕКФ Суботица, 2006.
2. Ballard C.: “Dimensional Modeling In a Business Intelligence Environment“, IBM, 2006.
3. Бечејски-Вујаклија Д.: “Пословни информациони системи“, материјали са предавања
4. Booch, G., Rumbaugh, J., Jacobson, I.: “The Unified Modeling Language User Guide“, Addison-Wesley, 1999.
5. Вељовић А.: “Развој информационих система и базе података“, ВИЗ, Београд, 2000.
6. Вељовић А.: “Моделовање информационих система“ друго издање, ТФ Чачак, 2005.
7. Горник Д.: “IBM Rational Unified Process“, Rational Software Corporation, 2003.
8. Димитријевић Н.: “Примена елемената складишта података у анализи успеха студената“-магистарска теза, Чачак, 2007.
9. Димитријевић Н.: “Примена елемената Data Warehouse–а и визуелног програмирања у ВТА ВЈ“-дипломски рад, Београд, 2001.
10. Димитријевић Н., Станојевић Љ., Вељовић А.: “Contribution to methodology of Business Intelligence systems object oriented developing“, Valcor, Златибор, 2007.
11. Димитријевић Н., Тот И.: “Дизајн и развој корисничког интерфејса код система за подршку у одлучивању“, SYM-OP-IS, 2002.
12. Димитријевић Н., Петровић Н., Станојевић Љ., Вељовић А.: “Пословна интелигенција на примеру анализе успеха студената“, YuInfo, 2008.
13. Димитријевић Н., Вељовић А.: “Business Intelligence системи у управљању успехом студената“, YuInfo, 2010.
14. Димитријевић Н., Вељовић А.: “Објектно оријентисана анализа у развоју Business Intelligence система за управљање успехом кадета ВА“, YuInfo, 2012.
15. Круљ Д. Сукновић М., Мартић М.: Примена алгоритама Data Mininig-а у пословном одлучивању, ФОН, 2002.
16. Милићев Д.: “Model-Driven Development with Executable UML“, Wiley Publishing Inc., 2009.
17. Moss L., Atre S.: “Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications“, Addison Wesley, 2003.
18. Његуш А.: “Пословни информациони системи“, Универзитет сингидунум, 2009.

19. Сакан М.: “Методологија војних наука“, ВИЗ, 2006.
20. Стандарди IDEF0 и IDEF1X (www.idef.com).
21. Станојевић Љ.: “Информациони систем институције као подршка одлучивању о њеном развоју-докторска дисертација“, Мегатренд, 2007.
22. Тот И: “Развој on-line analytical processing (OLAP) система за подршку одлучивању“ - докторска дисертација, Војна Академија, 2010.
23. www.microsoft.com/sqlserver, разни материјали за SQL Server 2008.
24. www.va.mod.gov.rs
25. www.docs.oracle.com/html/B13970_01examples.htm
26. <http://rc.etf.bg.ac.rs/ostali/fis>
27. www.microsoft.com/learning/en/us/mcse-sql-business-intelligence.aspx#fbid=7HkRr3qMcTv, разни материјали
28. www.thescipub.com/abstract/10.3844/jcssp.2009.1075.1081, “Evaluating Defense Architecture Frameworks for C4I System Using Analytic Hierarchy Process“ Journal of Computer Science, 2009.

Шира литература коришћена у раду:

29. Ballard C., Herreman D.: “Data Modeling Techniques for Data Warehousing”, 1998.
30. Bruckner R.: “Developing requirements for data warehouse systems with use cases“, Seventh Americas Conference on Information System, 2001.
31. Вељовић А.: “Менаџмент информациони систем“ друго издање, Компјутер библиотека Чачак, 2005.
32. Veerman E.: “MCTS Self-Paced Training Kit: Microsoft SQL Server 2005 Business Intelligence - Implementation and Maintenance“, Microsoft Press, 2007.
33. Горник Д.: “Data Modeling for Data Warehouses“, Rational Software, 2002.
34. Golfarelli M., Rizzi S.: “Designing the Data Warehouse: Key Steps and crucial issues“, Journal of Computer Science and Information management, vol 2, 1999.
35. Димитријевић Н., Тот И.: “Дизајн и развој корисничког интерфејса код система за подршку у одлучивању“, SYM-OP-IS, 2002.
36. Димитријевић Н.: “ОО анализа као корак у процесу развоја апликације “Анализа испита“, применом елемената складишта података“, ETRAN, 2003.
37. Димитријевић Н.: “Складиште података и димензионо моделовање“, YuInfo, 2004.
38. Inmon W.: “Building the Data warehouse“, QED Publishing Group, Wellesley, 1992.
39. Jacobson R., Misner S.: “MS SQL Server 2005 Analysis Services Step By Step“, MS

- Press, 2006.
40. Jamie M.: "Data mining with Microsoft SQL server 2008", Wiley Publishing Inc., 2009.
 41. Johnson E.: "A developer's guide to data modeling for SQL server", AW, 2008.
 42. Kimball, R.: "The data warehouse toolkit", John Wiley, New York, 1996.
 43. Larson B.: "Delivering Business Intelligence with Microsoft SQL Server 2005", McGraw-Hill/Osborne, 2006.
 44. Месарић Ј.: "Моделирање и имплементација информацијских сустава", материјали са предавања.
 45. Ничић Б.: "ЕТЛ процес у развоју система пословне интелигенције", дипломски рад, Сингидунум, 2009.
 46. Harts D.: "Microsoft Office 2007 Business Intelligence", The McGraw-Hill, 2008.
 47. Петровић Д.: "Менаџмент", ФОН, материјали са предавања
 48. Thomsen E.: "OLAP Solutions: Building Multidimensional Information Systems", John Wiley&Sons, New York, 2002.
 49. Cameron S.: "MS SQL Server 2008 Analysis Services Step By Step", MS Press, 2009.
 50. <http://office.microsoft.com/sr-latn-cs/excel-help/pegled-tehnologije-olap-online-analytical-processing-HP010177437.aspx>
 51. http://infosys3.elfak.ni.ac.rs/nastava/attach/ISNP_vezbe/UML0snove.doc
 52. <http://www.businessintelligence.com/>
 53. www.gartner.com
 54. www.redbooks.ibm.com
 55. <http://www.skladistenje.com/>

9. Скраћенице

BPR	Business Process Reengineering
BPwin	Business Process for Windows
CASE	Computer Aided System Engineering
DBMS	DataBase Management System
DDL	Data Definition Language
DM	Data mining
DSS	Decision Support Systems
DTS	Data Transformation Services
DW	Data Warehouse
ERwin	Entity Relationships for Windows
ESS	Executive Support Systems
IDEF0	Integration Definition Functional Modeling
ISO	International Organization for Standardization
MOLAP	Multidimensional OLAP
ODBC	Open DataBase Connectivity
OLAP	OnLine Analytical Processing
OLE (DB)	Object Linking and Embedding (DataBase)
OLTP	Online Transaction Processing
PK	Primary Key
RDBMS	Relational DataBase Management System
RI	Referential Integrity
ROLAP	Relational OLAP
SQL	Structured Query Language
UML	Unified Modeling Language

10. Преглед слика

Слика 2.1 – Раскорак у знању и одлучивању	14
Слика 2.2 – Нивои одлучивања	15
Слика 2.3 – Логичка хијерархија података, информација и знања	22
Слика 2.4 – OLAP коцка	30
Слика 3.1 – Модел водопада	38
Слика 3.2 – Спирални модел	39
Слика 3.3 – Синтакса правоугаоника	42
Слика 3.4 – Синтакса стрелице	43
Слика 3.5 – Позиције и улога стрелица	44
Слика 3.6 – Дијаграм највишег нивоа	45
Слика 3.7 – Декомпозициона структура	46
Слика 3.8 – Референцирање детаља	47
Слика 3.9 – Идентификујућа веза	51
Слика 3.10 – Неидентификујућа веза	52
Слика 3.11 – Хијерархијска подела UML 2.0 дијаграма	58
Слика 3.12 – Дијаграм случај употребе	59
Слика 3.13 – Пример дијаграма активности	60
Слика 3.14 – Основни елементи дијаграма сарадње	61
Слика 3.15 – Пример класе	62
Слика 3.16 – Пример асоцијативности	62
Слика 3.17 – Пример генерализације	62
Слика 3.18 – Фазе и дисциплине RUP-а	64
Слика 3.19 – Пример Use Case модела	66
Слика 3.20 – MSF управљење процесима	72
Слика 3.21 – Шема звезде	78
Слика 3.22 – Хијерархија димензија Студент, Предмет и Време	80
Слика 3.23 – Врсте димензионих модела	81
Слика 4.1 – Шематски приказ развоја складишта података	83
Слика 4.2 – Фазе развоја ВІ система	84
Слика 4.3 – Кључни кораци развоја ВІ система према	84
Слика 4.4 – Шематски приказ развоја ВІ система	87

Слика 4.5 – Шематски приказ дефинисања захтева.....	88
Слика 4.6 – Шематски приказ Анализе ВІ система.....	92
Слика 4.7 – Шематски приказ Дизајна ВІ система.....	95
Слика 4.8 – Шематски приказ Реализације ВІ система	98
Слика 4.9 – Процес Data mining-а	101
Слика 4.10 – Приступ OLAP коцкама	102
Слика 5.1 – Изглед табеле Преглед полагања испита.....	109
Слика 5.2 – Изглед табеле Селекција	111
Слика 5.3 – Изглед форме Студент	113
Слика 5.4 – Подаци са претходног школовања	113
Слика 5.5 – Изглед форме Резултати положених испита	114
Слика 5.6 – Шема организације Војне академије	115
Слика 5.7 – Декомпозициони дијаграм пословних процеса на ВА	118
Слика 5.8 – Стабло активности на ВА.....	118
Слика 5.9 – Декомпозициони дијаграм послова за питања кадета.....	119
Слика 5.10 – IDEFX1 модел изворних података	120
Слика 5.11 – Модел пословних случајева употребе на ВА	120
Слика 5.12 – Дијаграма пословних активности Послови испита и успеха.....	121
Слика 5.13– Модел случајева употребе за анализу успеха	123
Слика 5.14 – Модел концепта решења	124
Слика 5.15 – Логички модел складишта података	127
Слика 5.16 – Димензиони модел складишта података.....	128
Слика 5.17 – Физички модел складишта података.....	129
Слика 5.18 – Припрема података	130
Слика 5.19 – Мапирање података и ETL пакет за учитавање	131
Слика 5.20 – Дизајн реализације система	131
Слика 5.21 – Microsoft SQL Server 2008 Business Intelligence.....	133
Слика 5.22 – Дијаграм хардверске архитектуре	134
Слика 5.23 – Физичка реализација складишта података у SQL Server-у 2008.....	135
Слика 5.24 – OLAP коцка	137
Слика 5.25 – Преглед мера и димензија OLAP коцке.....	137
Слика 5.26 – Мера “Просек“.....	138
Слика 5.27 – Мера “Пали“	138
Слика 5.28 – Дефинисање КРІ “Просек“.....	139

Слика 5.29 – Дефинисање КРИ “Пали“	140
Слика 5.30 –Приказ дефинисаних КРИ	140
Слика 5.31 – Data mining модел	141
Слика 5.32 – Дефинисање параметара за Data mining	142
Слика 5.33 – Графикони настали изградом Data mining модела	142
Слика 5.34 – Приступ OLAP коцки	144
Слика 5.35 – Pivot табела Просек по школским годинама	144
Слика 5.36 – Pivot табела и дијаграм Просек по предмету	145
Слика 5.37 – Pivot табела Просек по класама и модулима	146
Слика 5.38 – Pivot табела Оцене по наставнику	147
Слика 5.39 – Pivot табела Пало по предмету	148
Слика 5.40 – Pivot табела за анализу успеха	148
Слика 5.41 – Pivot табела за анализу испита	149
Слика 5.42 –Data mining “Стабло одлучивања“	150
Слика 5.43 – Зависност статуса кадета	151
Слика 5.44 – Зависност оцене на испиту	152
Слика 5.45 – Зависност оцене на испиту	152
Слика 5.46 – Утицај појединих параметара на упис кадета на ВА	153
Слика 6.1 – Аналитичка база података	154
Слика 6.2 – Проширени модел пословних случајева употребе	156
Слика 6.3 – Модел изворних података за задужење у Библиотеци	157
Слика 6.4 – Проширени модел случајева употребе	158
Слика 6.5 – Редизајнирана табела „Кадет“	159
Слика 6.6 – Проширени пакет за припрему података	159
Слика 6.7 – Зависност успешности дипломирања	160
Слика 6.8 – Зависност успешности дипломирања	161
Слика 6.9 – Зависност просека студирања	162
Слика 6.10 – Зависност просека студирања	163
Слика 6.11 – Поређење одзива упита у OLAP и BI систему	164
Слика 6.12 – Процесирање аналитичке базе података	165
Слика 6.13 – Процесирање OLAP коцке	165