

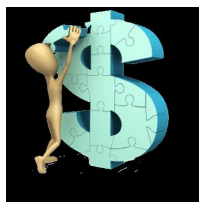
Reinženjering poslovnih procesa Business Process Reengineering	Knjiga Volume	3
---	------------------	----------

Urednik serije/ Editor of series: **Alempije V.Veljović**

**PRIMENA ANALITIČKIH BAZA PODATAKA U
FINANSIJSKOM MENAŽMENTU**

**Application of analytical databases in financial
Management**

Miroljub M. Zahorjanski



Fakultet tehničkih nauka u Čačku Univerzitet u Kragujevcu
Faculty of Technical Sciences Cacak University of Kragujevac

PRIMENA ANALITIČKIH BAZA PODATAKA U FINANSIJSKOM MENADŽMENTU

dr Miroљjub M. Zahorjanski docent

Recenzent:

Prof. dr Ljiljana Stanojević
Prof. dr Mirko Đapić

Izdavač:

Fakultet tehničkih nauka u Čačku

Za izdavača:

Prof. dr Jeroslav Živanić, dekan

Štampanje odobreno odlukom Naučno-nastavnog veća
Fakulteta tehničkih nauka u Čačku broj 14-1366/9 od 11. 09. 2013. god.

Tiraž: 100 primeraka

Štampa:

Štamparija SaTCIP, Vrnjačka Banja

ISBN: 978-86-7776-155-4

SADRŽAJ:

PREDGOVOR	7
TEORIJSKI PRISTUP FINANSIJSKOM POSLOVANJU	19
ANALITIČKE BAZE U POSLOVNOJ STRATEGIJI PREDUZEĆA ..	25
1. DEFINISANJE ZAHTEVA	31
1.1 Prikupljanje zahteva.....	34
1.1.1 Prikupljanje korisničkih zahteva	35
1.1.2 Prikupljanje izvornih zahteva.....	39
1.1.3 Analiza OLTP podataka	41
1.2 Izrada modela poslovnih slučajeva upotrebe	42
1.2.1 Dijagram poslovnih slučajeva upotrebe	44
1.2.2 Dijagram poslovnih aktivnosti	46
2. OBJEKTNO ORIJENTISANA ANALIZA.....	51
2.1 Izrada modela sistemskih slučajeva upotrebe	51
2.1.1 Sistemski slučajevi upotrebe	53
2.2 Izrada konceptualnog modela	58
2.3 Izrada dijagrama interakcije	63
3. OBJEKTNO ORIJENTISAN DIZAJN	69

3.1 Priprema podataka za analitičko procesiranje informacija	69
3.1.1 Ekstrakcija podataka	70
3.1.2 Čišćenje podataka	71
3.1.3 Transformacija podataka	72
3.2 Izrada dimenzionog modela	73
3.2.1 Definisane hijerarhije elemenata i atributa.....	76
3.2.2 Denormalizacija	76
3.2.3 Kreiranje agregacija	77
3.3 Izrada dijagrama klasa.....	79
4. IMPLEMENTACIJA.....	85
4.1 Izrada skladišta podataka	86
4.1.1 Kreiranje fizičkog modela BP	90
4.1.2 Generisanje baze podataka	91
4.1.3 Učitavanje podataka	94
4.2 Priprema analize podataka.....	94
4.2.1 Analiza podataka upitima (SQL).....	96
4.2.2 Višedimenziona analiza podataka (OLAP)	96
4.3 Analiza podataka - MS EXCEL dinamičke tabele	107
4.4 Formiranje Pivot tabele.....	109
4.5 Formiranje Pivot grafikona	117
4.6 Izrada korisničkog interfejsa	120
4.6.1 Korisnički interfejs transakcione baze podataka	121
4.6.2 Korisnički interfejs OLAP	125

4.7 Implementacija u MS SQL server 2005	127
ZAKLJUČAK.....	135
POJMOVI.....	141
SKRAĆENICE	149
LITERATURA.....	151



PREDGOVOR

Obzirom da živimo u vremenu sve bržih, dramatičnijih, kompleksnijih i nepredvidivih promena, tehnološki, tržišno-ekonomski, politički, društveni i globalni faktori, utiču na ubrzanje promena, kao i na oblikovanje teorije i prakse ekonomije i menadžmenta. Koncept „*Knowledge management*“ (Menadžment znanja) sve više je jedan od ključnih koncepata kreiranja konkurentne prednosti u novoj ekonomiji i menadžmentu. Po rečima *Michael Porter*-a “Jedina konkurentna prednost je sposobnost da se uči i menja.”

Pre svega, trebalo bi da definišemo pojam menadžmenta znanjem. “Menadžment znanjem je kontinuirani upravljački proces koji pomaže organizacijama da prepoznaju, izmere vrednost intelektualnog kapitala kojim raspolažu, planiraju, organizuju, upravljaju i naravno kontrolišu (definicija menadžmenta) važne informacije i iskustvo dopunjeno obrazovanjem koje predstavlja memorijski kod svake organizacije, ali nije deo postojeće strukture pravilnika i procedura.”^[23]

Prvo poglavlje ove monografije će se baviti definisanjem mogućih zahteva u preduzećima i institucijama u oblasti koje pokrivaju finansijsko poslovanje, kao i strategije koju ta oblast zahteva. U procesu prikupljanja korisničkih zahteva koja daje širinu u pristupu i bazira se na definisanju zahteva koje korisnik izvršava i izvodi, biće korišćeni *IDEFO* standardi. Ovim se definišu ciljevi, procesi i resursi. Metodologijom odozgo na dole, serijom intervjua, postići će se identifikacija granica posmatranog sistema i dinamička integrativnost poslova koji se odvijaju u posmatranom okruženju.

Drugo poglavlje, baviće se objektno orjentisanom analizom i izradom modela sistemskih slučajeva upotrebe, izradom konceptualnog modela i dijagrama interakcije. Sistemski slučajevi upotrebe, vezani su za odgovarajuće organizaciono tehnološko okruženje i podrazumevaju detaljan opis poslovnih procesa kao sekvence aktivnosti.

Treće poglavlje će obuhvatiti objektno orjentisani dizajn koji treba da omogućiti: pripremu podataka za analitičko procesiranje informacija, izradu dimenzionog modela, i izradu dijagrama klasa.

U okviru pripreme podataka za analitičko procesiranje finansijskih informacija biće obuhvaćene: Ekstrakcija podataka iz

OLTP baze, potom čišćenje podataka proverom logičkih grešaka i eliminacijom ostalih grešaka i podataka koji nemaju značenje i na kraju transformacijom podataka pre učitavanja.

Četvrto poglavlje biće posvećeno implementaciji koja bi se sastojala od sledećih faza: izrade skladišta podataka, pripreme analize podataka i izrade korisničkog interfejsa

Izrada skladišta podataka preudicira da se prethodno izabere sistem za upravljanje bazama podataka. S obzirom da je cilj rada, da se među različitim modelima informacionih sistema pronađe onaj koji će obezbediti najpouzdanije čuvanje podataka adekvatno izraženim zahtevima finansijskog poslovanja, kao i da primereno potrebama izveštavanja i analize, bude predložen onaj sistem koji će moći da pruži odgovore na svako razumno pitanje postavljeno od strane taktičkog i strateškog menadžmenta ove oblasti, očekuje se da se odluka donese tokom istraživanja. Od budućeg alata za izradu skladišta podataka očekuje se da ima mogućnosti za ekstrakciju i transformaciju podataka (DTS), a za *OLAP (OnLine Analytical Processing)* analizu takozvani *OLAP* server.

U zaključku će biti dat kritički osvrt na ostvarenost predmeta i cilja ovog istraživačkog rada, potom ocena ispunjenosti očekivanih rezultata, ograničenja metodologije objektno orjentisanog razvoja, a takođe će biti komentarisani pravac daljih istraživanja.

Autor

INTRODUCTION

Since we live in a time of faster, more dramatic, complex and unpredictable change, technology, market-economic, political, social and global factors affecting the speed of changes, as well as the design theory and practice of economics and management. The concept of "Knowledge Management" (Knowledge Management) is increasingly one of the key concepts in creating competitive advantage in the new economy and management. According to Michael Porter, "The only competitive advantage is the ability to learn and change."

First of all, we should define the concept of knowledge management. "Knowledge management is a continuous management process that helps organizations identify, measure the value of intellectual capital at their disposal, plan, organize, manage and control the course (the definition of management) important information and experience complemented by an education that is in the memory of every organization, but is not part of the existing structure rules and procedures".

The first chapter of this monograph will address the definition of possible requests in companies and institutions in the area covered by the financial management and strategies that the area is required. In the process of gathering user requirements, which gives the width of the approach and is based on the definition requires that a user executes and runs, will be used IDEF0 standard. This defines the objectives, processes and resources. Top-down methodology, a series of interviews, will be achieved by identifying the limits of the observed system and the dynamic integrativity activities that take place in the monitored environment.

The second section will deal with the object-oriented analysis and development of models of system use cases, and creating a conceptual model of interaction diagram. System use cases are related to the appropriate organizational and technological environment and include a detailed description of the business process as a sequence of activities.

The third chapter will include object-oriented design that should enable the preparation of data for analytical processing of information,

development of dimensional models, and create class diagrams. In preparing the data for analytical processing of financial information will include: extraction of data from the OLTP database, then data cleaning verification logic errors and other errors and eliminating data that have no meaning, and finally transform the data before uploading.

The fourth chapter will be devoted to the implementation of which would consist of the following stages: preparation of data warehouse, data analysis and preparation of making the user interface.

Creating a data warehouse without prejudice to pre-select a system database management. Considering that the aim of the paper, and among the different models of information systems find one that will provide the most reliable data storage requirements are adequately expressed financial operations, as well as to the needs of reporting and analysis, is proposed the system which will be able to provide answers to all reasonable question posed by tactical and strategic management of the area, is expected to take a decision by the survey. From the prospective of tools for data warehouse is expected to have features extraction and transformation of data (DTS) and OLAP (OnLine Analytical Processing) analysis of the so-called OLAP server.

The conclusion will be given a critical review of the achievement object and purpose of this research, then mark the fulfillment of the expected results, the limitations of the methodology of object-oriented development, and will also be commented direction for further research.

Author

UVOD

Postojanje adekvatnog i dobro organizovanog informacionog sistema predstavlja danas jedan od neophodnih uslova za efikasno i kvalitetno upravljanje u bilo kojoj od oblasti ljudskog delovanja. Skoro da se ne mogu ni zamisliti funkcije preduzeća ili institucija bez podrške informacionih tehnologija. ^[102]

Kasne '70 i rane '80 prošlog veka bile su obeležene ubrzanim razvojem informacionih tehnologija u oblasti izrade integriranih kola i mikročipova, koje su omogućile povećanje brzine i kapaciteta, što je prouzrokovalo i veću mogućnost u praktičnim primenama u odlučivanju i upravljanju preduzećem. Ako se na trenutak podsetimo činjenice da je organizacija sinergetska sinteza nauke i veštine, a po rečima francuskog teoretičara klasične organizacije Anri Fajol-a (*Henri Fauol*) računovodstvene aktivnosti ili finansijske informacije su "organ vida preduzeća", onda nam odmah bivaju jasne mogućnosti koje ove aktivnosti mogu imati od sve prisutnijih informacionih tehnologija današnjice.

Prvi softveri u oblasti računovodstva, razvijani u drugoj polovini prošlog veka imali su za cilj prikupljanje podataka, skladištenje na za to predviđene repozitorijume i pretvaranje podataka u informacije neophodne za rešavanje dnevnih knjigovodstvenih zadataka, periodičnih platnih obaveza i godišnjih završnih izveštaja. Uporedo sa rastom kapaciteta nosača informacija sa nekadašnjih kilobajta i megabajta na sadašnje gigabajte i terabajte, korišćenje informacionih tehnologija proširilo se sa područja operativnog menadžmenta na funkcije koje u preduzećima i institucijama obavljaju taktički i strateški menadžeri.

Da bi smo preciznije definisali predmet interesovanja ove disertacije navešćemo jednu od definicija sistema za podršku odlučivanju.

DSS je na modelima zasnovan skup procedura za obradu i tumačenje informacija koji pruža podršku menadžeru u procesu donošenja odluka. *DSS* pruža korisniku informacije potrebne za donošenje odluka i omogućava mu da kombinacijom ličnog rasuđivanja sa računarskim izlazom (u aktivnom korisničkom interfejsu sa mašinom) donosi valjane odluke. Takvi sistemi su sposobni da rešavaju sve vrste problema (strukturirane, polu-strukturirane i ne-strukturirane) i koriste upite (*query*) za dobijanje odgovora na *ad-hoc* zahteve. Na odgovarajući zahtev oni koriste baze podataka i kvantitativne metode da bi se našlo adekvatno rešenje za problem odlučivanja. U perspektivi *DSS* će biti integralni deo pristupa otkrivanju i rešavanju

problema odlučivanja i upravljačke doktrine utemeljene na principima *Management by Perception*.”^[6]

Nekadašnje odluke koje su trebale biti donešene na taktičkom ili strateškom nivou zasnivale su se na papirnatim izveštajima nastalim kao produkt periodičnih preseka stanja i godišnjih bilansa. Njihovo upoređivanje iziskivalo je pamćenje velikog broja cifara, a bilo kakva dublja analiza zahtevala je uvođenje novih obrazaca, naknadnu matematičku obradu podataka, uz prethodno ručno ekstrahovanje i agregiranje podataka i prikazivanje u vidu informacija na novim izveštajima.^[6, 7,49.]

Brzi rast proizvodnje u razvijenim zemljama, kao i zemljama koje hvataju korak sa takvim trendom, proizveo je nove zahteve za informacijama u pogledu strukture i kvaliteta ali i u pogledu brzine odziva. Nekadašnji softveri, zasnivani na direktnoj transakcionoj obradi (*OLTP - OnLine Transaction Processing*) postaju nedovoljno fleksibilni u pogledu karakteristika neophodnih za direktnu analitičku obradu podataka.

Razmatranje koncepta preduzeća

Svako preduzeće ima svoj opšti cilj funkcionisanja koji predstavlja njegovo željeno stanje (izlaz). Za ostvarivanje opšteg cilja, preduzeće mora da zadovolji određene pretpostavke: mogućnost funkcionisanja u vremenu i potrebu što efikasnijeg funkcionisanja u odnosu prema postavljenom cilju. Ako preduzeće posmatramo kao ekonomski sistem u malom onda svakako treba istaći njegove ekonomske ciljeve kao što su:

- Kontinuitet funkcionisanja,
- Povećanje efikasnosti funkcionisanja,
- Kontinuitet povećanja efikasnosti.

Preduzeće je ekonomski subjekt i ekonomski sistem. Ono je bitan element i predstavlja podsistem globalno privrednog sistema jednog društva, sa kojim je u međuzavisnosti. Vlasništvo kapitala je takođe važan element tržišnog sistema i preduzeća koje obuhvata. Pored kapitala, obuhvata učinke i dobit tog kapitala. Takođe se za preduzeće može reći da je osnovni privredni subjekt, u kome se korišćenjem sredstava za proizvodnju proizvode i razmenjuju dobra. Na njemu se zasniva privredni i celokupni društveni razvoj. Osnovna obeležja preduzeća su da preduzeće ima svoju ekonomiju, da tu ekonomiju ostvaruje u okruženju sa kojim je u stalnoj komunikaciji, da je u principu složen dinamički sistem koji se može tretirati kao

finansijski entitet i da procesima i pojavama u preduzeću upravlja kompletan menadžment.

Očigledno je da je važan odnos preduzeća i okruženja. Okruženje sistema je skup objekata izvan sistema, za koje važe sledeće pretpostavke: da promene odlika okruženja utiču na ponašanje sistema i da nasuprot tome promene ponašanja sistema deluju na promenu odlika okruženja. Složenost preduzeća kao ekonomskog sistema određuje veliki broj njegovih komponenata najrazličitije prirode, neprekidnih povratnih sprega i kvalitativnih svojstava ekonomskih (poslovnih) pojava i procesa. Po pravilu, svi procesi i pojave u preduzeću su dinamičke prirode.

Prvi od principa na kojima je jedan ovakav neizolovani (u interakciji je sa globalnom ekonomijom) sistem funkcioniše je strategija. Ona je neprikosnovena i na nju deluju faktori makroekonomije. Drugi princip bila bi struktura i ona je izvedena iz strategije, obzirom da na nju deluju činjenice okruženja, a često i činjenice nepoznate prirode. Strategiju možemo razumeti kao opšti plan akcija koje se preduzimaju i čija je svrha ostvarivanje određenih jasno definisanih ciljeva. Akcije su uglavnom delovi procesa strategije koja se sprovodi ili je u toku. Ovakvi procesi u preduzeću označavaju promene ulaza, izlaza, stanja i upravljanja sistemom, tokom vremena. Te promene su posledica različitih razmena ili u samom sistemu, ili između sistema i okruženja (materijalnih, energetskih, informacionih). Procese koji se odvijaju u preduzeću delimo u dve grupe: osnovne koji obezbeđuju razvitak i funkcionisanje preduzeća i pomoćne koji obezbeđuju poslovanje. Prve bismo mogli okarakterisati kao dugotrajne, a druge kao kratkotrajne i vezane za kraći period izvršenja jednog posla. Kada su u pitanju i jedni i drugi procesi ne možemo izbeći činjenicu upravljačkog uticaja na stanje u preduzeću.

Upravljanjem sistemom^[47] ostvaruju se njegovi ciljevi. Optimalno funkcionisanje preduzeća može se postići usklađivanjem funkcionisanja svih njegovih delova, uz težnju za ostvarivanjem cilja sistema u celini, ali ne ciljeva njegovih pojedinih delova. Suma pojedinih optimuma ne rezultira uvek zajedničkim optimumom. Bilo bi isuviše prosto rukovoditi preduzećem ulaganjem maksimuma u sve njegove delove ili delove njegovog cilja. Osim toga to bi na duži rok izazvalo prenapregnutost i neželjene efekte. Iz tog razloga pažnju u upravljanju treba usmeriti na odlike funkcionisanja sistema i njegovih pojedinih delova koje su najčešće različite. Funkcionisanje sistema pokazuje različite odlike u vremenu, zavisno od karakteristika njegovih elemenata. Pokazuje i njegovu strukturu i određene uslove u kojima se proces događa. Pored odlika moramo razumeti i oblike funkcionisanja i to najbolje kroz horizontalnu i vertikalnu podelu. Dok bi u horizontalnoj podeli oblici bili: proizvodnja, nabavka, prodaja,

finansijska funkcija, kadrovska funkcija; dotle bi se oblici vertikalne podele odnosili na upravljanje, rukovođenje i izvršenje.

Formiranje strategije u cilju poboljšanja stanja

Kad se govori o strategiji, čak i u opštem smislu, mora se pomenuti planiranje i modeliranje. Planiranje kao glavna funkcija upravljanja putem za postizanje cilja je nezaobilazni element svake strategije. Ono bi trebalo da nam da odgovor na pitanje „Šta treba uraditi u budućnosti?“. U segmentu taktike i pitanja „Kako to treba uraditi?“ poslužili bismo se modeliranjem kao najvažnijom metodom razvijanja procesa donošenja odluke. Ponašanje preduzeća kao sistema može se izraziti kao njegova koncepcija za ostvarivanje ciljeva. Ono je pod uticajem raznih delovanja, koji mogu narušiti njegovu stabilnost i ravnotežu (balans). Otuda organizacija uspešnih i efikasnih preduzeća mora biti fleksibilna, kreativna, inovativna (zasnovana na informacijama), intenzivno okrenuta novim znanjima, usmerena prema kupcima i dobavljačima i na kraju prilagodljiva sve turbulentnijem i izbirljivijem okruženju.

Sa druge strane upravljanje procesima je ciljno usmereno delovanje na sistem upravljanja. Upravljanje treba da ograniči broj stepeni slobode, da smanji neodređenost kako bi rezultat funkcionisanja potpunije odgovarao postavljenom cilju. Upravljanje se prati vezom između ulaza i izlaza određenog sistema. Ta veza je povratna sprega. Promene u sistemu stvaraju protivrečnosti, koje dovode do prilagođavanja sistema. Upravo to prilagođavanje sistema našeg preduzeća treba da bude okosnica svake strategije.

Dok klasični pristup raspravi o preduzeću u prvi plan stavlja razmatranje strukture, dotle sistemski pristup započinje razmatranjem ciljeva poslovanja i načina njihovog ostvarenja. Otuda i potreba u sistemskom pristupu za kreiranjem strategije jer se jedino na taj način kroz praksu (3-5 godina) može proveriti ispravnost poslovne politike.

Govoriti o formiranju strategije finansijskog poslovanja preduzeća znači ne zaobići činjenicu o važnosti sistema informacija. Sistem informacija u preduzeću je bitan element uspešnosti poslovanja preduzeća. Dobro izabrane i pravovremene informacije omogućavaju donošenje bolje odluke. Predviđanje korišćenjem bilo koje od metoda (regresija, ekonometrijski model, *input-output* model, simulacioni model i drugi) zahtevaju pravovremene i kvalitetne informacije kao preduslov. Eksterne i interne informacije su glavni izlazi iz sistema kada su u pitanju EKONOMSKO FINANSIJSKI POSLOVI što je vidljivo i na dijagramu dekompozicije prvog nivoa (slika 1.6)

Odnosi preduzeća sa okruženjem glavni su predmet finansijskog računovodstva, premda ono obuhvata i interne procese (utrošak materijala, obračun amortizacije, i dr.). Finansijsko računovodstvo je i jedini informacioni izvor o finansijskom položaju i uspehu preduzeća za korisnike informacija. Ono je regulisano računovodstvenim principima koji su opšte prihvaćeni, računovodstvenim standardima i zakonskim propisima, koji određuju formu i sadržinu finansijskih izveštaja preduzeća. I upravljačko računovodstvo se značajno oslanja na operativna knjigovodstva i druge analitičke informacione izvore. Ono se bavi interpretacijom izveštaja za potrebe menadžmenta. Upravljačko računovodstvo nije, kao finansijsko računovodstvo, usmereno na informacije eksternih korisnika, već se bavi internom alokacijom sredstava.

Analiza poslovanja kao preduslov planiranju i formiranju strategije

Sistemska analiza označava pristup upravljanju sistema koji karakterišu organizovanost i kreativnost. Cilj systemske analize je u određivanju pojedinih karakteristika elemenata i procesa u nekom realnom sistemu, kao i analizi međuzavisnosti koje se između njih dešavaju.

Sistemska analiza omogućava sagledavanje i rešavanje, na najracionalniji način, najsloženijih problema. Njihovim rešavanjem mogu se dobiti najadekvatniji odgovori o strukturi i procesu analiziranog sistema.

Sistemskom analizom se prikupljaju, obrađuju i upotrebljavaju informacije o relevantnim parametrima datog sistema. Ona omogućava sagledavanje međusobnih odnosa sistema kao celine i njegovih delova, kao i odnosa sistema ili pojedinih njihovih delova sa okruženjem. Postoje dva osnovna tipa analize, a to su:

- Deskriptivna ili opisna analiza
- Kvantitativna ili količinska analiza

Oba ova oblika analize mogu se koristiti za poboljšanje isporuke, kvaliteta i fleksibilnosti proizvodnog procesa u preduzeću, kao i snižavanje troškova.

Kada je reč o planu i prognozi, treba istaći da je prognoza nepristrasna projekcija onoga što se očekuje da će se dogoditi. Prognoza se ne sme mešati sa planom, ciljem, ili merilom performansi, koji pokazuju ono što mislimo da bi se trebalo dogoditi. Prognoza nije cilj, već preko prognoziranja i predviđanja dolazimo do cilja. Mogućnost prognoziranja u preduzeću bazira se na osnovu

verovatnoće: pesimističke, najverovatnije i optimističke vrednosti tražnje. Drugi aspekt je pitanje ko bi mogao da radi prognozu. Postoje mnoge mogućnosti, uključujući marketing, proizvodnju, finansije, središnje korporativno telo za predviđanje i različite korporativne i divizijske nivoe, zavisno od veličine i vrste preduzeća. Različite odluke zahtevaju različite metode, a one mogu biti: Kvalitativne (kada podaci iz prošlosti nisu raspoloživi ili su nepotpuni), potom Delhi metoda, istraživanje tržišta, analogija sa životnim ciklusima, prosuđivanje zasnovano na informacijama, ili pak metode dinamičke analize vremenskih serija, i druge.

Radi sagledavanja realne i objektivne finansijske situacije preduzeća, potrebno je izvršiti finansijsku analizu, koja prikazuje prošlost i sadašnjost preduzeća, a predstavlja i osnovu (bazu) za predviđanje budućeg finansijskog stanja. Ova analiza se bavi utvrđivanjem funkcionalnih odnosa koji postoje između pozicija bilansa stanja i bilansa uspeha, sa ciljem dobijanja verodostojne ocene o finansijskom položaju i aktivnosti preduzeća. Dobijene informacije iz finansijske analize predstavljaju polaznu osnovu za preduzimanje akcija usmerenih na popravljavanje boniteta i trenda rasta i razvoja poslovanja preduzeća.

Razlog za vršenje finansijske analize je uviđanje slabosti koje mogu dovesti do finansijskih problema u preduzeću u budućem periodu i preduzimanje adekvatnih mera za njihovo otklanjanje. Finansijska analiza bi trebala da ponudi odgovore o tome kako menadžment finansira investicije, da li preduzeće ostvaruje dovoljan iznos profita i svakako kakva je likvidnost preduzeća.^[32] Početnu osnovu za finansijske analize predstavljaju finansijski obračuni na kojima se i temelji finansijska analiza, a to su bilans stanja, bilans uspeha i izveštaj o finansijskim tokovima.

“Finansijska analiza može da se realizuje primenom različitih metoda, od kojih su najznačajniji: vizuelna analiza, analiza pomoću računa pokrića, analiza pomoću neto obrtnog fonda, *cash flow* analiza, *funds flow* analiza i *racio* analiza. *Racio* analiza predstavlja najkompleksniji pristup utvrđivanja kreditne sposobnosti preduzeća, jer najdirektnije ukazuje na sposobnost otplaćivanja ugovorenih kreditnih obaveza, stepen efikasnog poslovanja i korišćenja resursa, nivo operativnog korišćenja raspoloživih sredstava, sposobnost participacije i samo-finansiranja, odnosno ukupne poslovne performanse preduzeća od kojih zavisi otplatni kapacitet, efikasno korišćenje kreditnih resursa i nivo potencijalnog kreditnog rizika.“ Finansijska analiza predstavlja, u osnovi, *racio* analizu.^[24]

Radi obavljanja pouzdane analize poslovanja preduzeća, potrebno je da se obezbedi ispravnost podataka, odnosno potrebno je da knjigovodstveni i drugi podaci o poslovanju preduzeća budu

sastavljeni i prezentirani u skladu sa važećim finansijskim i privrednim propisima, te budu ispravni i objektivni. Isto tako je važna jednoobraznost podataka, odnosno da postupci ili metode koje se koriste za dobijanje podataka o poslovanju preduzeća budu unapred determinisani i da se ne mogu menjati prema trenutnim potrebama korisnika tih podataka. Ovaj zahtev je naročito značajan za potrebe komparativne analize, kada se porede preduzeća iz iste grane, za poređenja između grupacija, ili kao što je u ovoj tezi slučaj, između obračunskih perioda za isto preduzeće.

Racio analiza se zasniva na podacima iz bilansa uspeha i bilansa stanja. Baš taj odnos između bilansnih pozicija naziva se *racio* i izražen je u matematičkoj formuli. U osnovne *raci-e*, odnosno pokazatelje, spadaju :

- Pokazatelji likvidnosti
- Pokazatelji aktivnosti
- Pokazatelji finansijske strukture i
- Pokazatelji rentabilnosti.

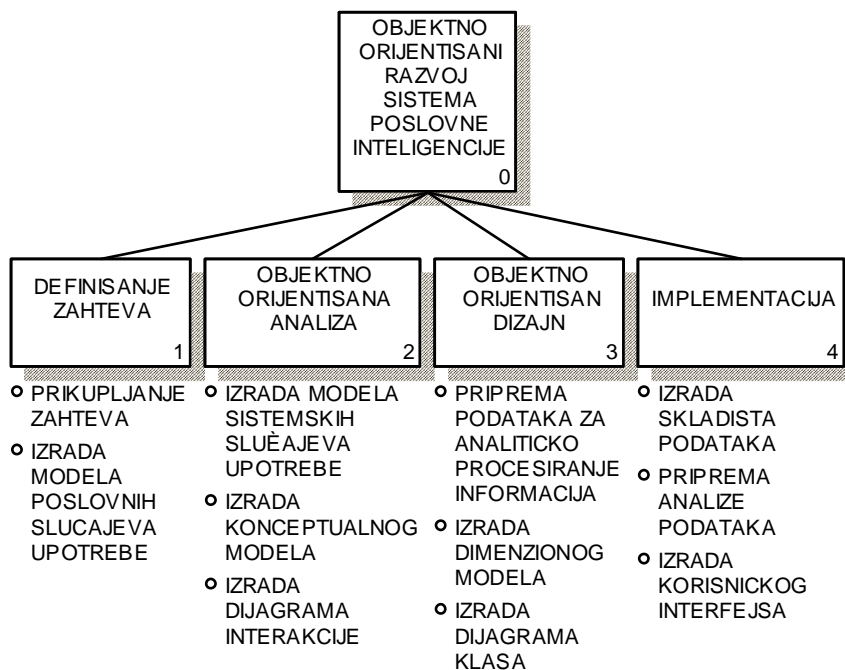
Pokazatelji likvidnosti ukazuju na sposobnost preduzeća da plaća dospele obaveze uz održanje potrebnog obima i strukture obrtnih sredstava i očuvanje dobrog kreditnog boniteta.

Pokazatelji aktivnosti ili efikasnost obuhvataju relativne odnose između prihoda i rashoda sa jedne strane i pojedinačnih i ukupnih ulaganja u poslovna sredstva sa druge strane. Ovi pokazatelji su poznati kao koeficijenti obrta. Pokazatelji finansijske strukture su od značaja za potencijalne poverioce jer ukazuju na stepen zaduženosti preduzeća i njegovu sposobnost da izmiruje obaveze po osnovu kamate i otplate glavnice.

Pokazatelj rentabilnosti kao merilo parcijalnog poslovnog uspeha pokazuje koliko je ostvarena dobit na 100 dinara angažovanih sredstava

Očigledno je da se pomenuta analiza korišćenjem pokazatelja likvidnosti, kao i kompletna finansijska analiza zasniva na bilansima stanja i uspeha proteklih obračunskih perioda. Činjenica je da je prošlost interesantna u onoj meri u kojoj daje indikacije za razvoj u budućnosti. Iako prošlost nije uvek dobra baza za procenu budućnosti, mora se priznati da je ona često jedina, uvek objektivna i po pravilu najpristupačnija osnova te procene.

Slika 1.0 Šematski prikaz metodologije objektno orijentisanog razvoja sistema poslovne inteligencije



Izvor: Stanojević Lj., Veljović A., Razvoj metodologije projektovanja poslovne inteligencije, Monografija, Megatrend univerzitet primenjenih nauka, Beograd, 2008.

TEORIJSKI PRISTUP FINANSIJSKOM POSLOVANJU

Pošto je za kvalitetno finansijsko poslovanje^[10] oduvek najvažnija bila prava i blagovremena informacija i mogućnost predviđanja budućih događaja i njihovog finansijskog uticaja, stoga se i teorijski pristup finansijskom poslovanju uglavnom bavi finansijskim izveštavanjem i maksimalnim iskorištavanjem dobijenih informacija. Sva pažnja teoretičara je usmerena upravo na otkrivanje boljeg, bržeg, efikasnijeg modela koji bi obezbedio pristup informacijama od važnosti.

Osnovne karakteristike finansijskog izveštavanja jesu obaveznost i redovnost. Obaveza vođenja poslovnih knjiga, kao i obaveza obelodanjivanja finansijskih izveštaja zakonom je sankcionisana u svim uređenim državama.

Finansijsko izveštavanje regulisano je na tri nivoa: zakonima i propisima, profesionalnim načelima i standardima i internom računovodstvenom regulativom.

Pitanja koja se tiču zakonske regulative mogu se podvesti pod: identifikovanje obveznika, propisivanje obaveze i načina vođenja poslovnih knjiga, identifikovanje svrhe izveštavanja, održavanje forme i strukture izveštaja i propisivanje obaveze javnog objavljivanja finansijskih izveštaja.

Kada je u pitanju nivo profesionalne regulative, njena se pažnja usmerava na: pravila u vidu načela i standarda koje propisuju profesionalne računovodstvene organizacije. Ovde se težina poklanja profesionalnoj etici i moralnoj obavezi poštovanja profesionalnih uputstava. Načela su opštija pravila, a standardi čine jedan vid njihove razrade, odnosno konkretizaciju radi lakše primene u praksi.

Treći nivo regulisanja finansijskih izveštaja odnosi se na internu računovodstvenu regulativu i predstavlja pravila preduzeća u toj oblasti. Ova pravila su manje ili više specifična za svako preduzeće jer se odnose na pojedinačnu poslovnu politiku i konkretne okolnosti u kojima se ono nalazi. Konkretizacija tih pravila formalizuje se kroz pravilnike o računovodstvu.

U nekim državama računovodstveno izveštavanje je strožije regulisano zakonima i propisima kao što je slučaj kontinentalnog dela zapadne evrope, dok se u Velikoj britaniji i SAD-eu, veća pažnja poklanja profesionalnoj regulativi. U savkom slučaju kada govorimo globalno, neophodno je sintetizovanje sva tri nivoa radi dobijanja kvalitetnih informacija za društvo u celini.

Konkretizacija cilja finansijskog izveštavanja, detaljnije posmatrana, predstavlja obezbeđenje kvalitetnih, istinitih, poštenih i nadasve korisnih informacija, na osnovu kojih će se donositi racionalne poslovne odluke. Ovo bi trebalo da bude u harmoniji sa ciljevima preduzeća u savremenim uslovima poslovanja od kojih treba istaći dva najvažnija: održati preduzeće i ostvariti profit.

Da bi se navedeni ciljevi uspešno realizovali nophodno je iscrpeti računovodstvene informacije iz osnovnih finansijskih izveštaja, internih finansijskih izveštaja i analitičkih evidencija. Neke od pomenutih informacija dostupne su samo određenim vrstama korisnika. U najopštijem smislu, korisnike bismo podelili na interne i eksterne.

Kao interni korisnici najčešće se pominju menadžeri. Strateški menadžeri sa aspekta kreiranja politike preduzeća predviđaju pravce kretanja u rasponima od 3 do 5 godina vodeći računa o okruženju i tržištu. Taktički menadžeri imaju zadatak da iz poslovno-finansijskih informacija odrede taktike napredovanja na strateški određenim pravcima.

U eksterne korisnike informacija ubrojićemo više grupa: akcionare, kreditore, investitore, zaposlene radnike, poslovne partnere, analitičare i vladine institucije. Svi oni imaju različite interese ali je u interesu svih da preduzeće dobro posluje.

Kada bismo trebali da damo definiciju šta su to finansijski izveštaji, mogli bismo da ih označimo kao skup računovodstvenih informacija o imovini, kapitalu, obavezama i rezultatu poslovanja preduzeća. Taj skup bi činili: bilans stanja, bilans uspeha, izveštaj o tokovima gotovine, izveštaj o promenama na kapitalu, napomene uz finansijske izveštaje i statistički aneks.

Bilans uspeha, kao krajnji rezultat, daje razliku između potrošenog i zarađenog novca, dakle profit ili gubitak. Tu se u opštim crtama daju troškovi proizvodnje (direktan rad, materijal i ostalo), operativni troškovi (troškovi prodaje, marketinga, administracije i ostalo), kao i prihodi.

Bilans stanja je klasičan računovodstveni prikaz "leve i desne strane". Na levoj strani tabele obično se prikazuju sredstva, a na desnoj obaveze. Leva i desna strana kao krajnji rezultat, odnosno zbir svih tačaka, moraju da imaju istu cifru. Ukoliko nemaju, to je znak da nije obračunat ili neki trošak ili neki prihod. U sredstva spadaju gotovina i potraživanja, ali i nekretnine i osnovna sredstva, kao i zalihe i eventualni gubici. Obaveze obuhvataju sva tekuća i buduća plaćanja, zajmove, osnovni kapital i zadržane zarade.

Izveštaj o gotovinskom toku, kako se "*keš flou*" zove u našem računovodstvu, zbirno pokazuje sve ulaze i izlaze gotovine u toku poslovanja preduzeća. Dakle, tu su prikazani svi troškovi i svi prihodi,

razvrstani u nekoliko kategorija. Ove kategorije su u stanju da daju odgovor na pitanja kao što su:

- Gde se usmerava profit?
- Zašto dividende nisu veće?
- Zašto su korišćeni pozajmljeni izvori sredstava?
- Kako su finansirane investicije?

Statistički aneks sadrži određene statističke i statusne podatke, dok Izveštaj o promenama na kapitalu pruža informacije o promenama na kapitalu pravnih lica, tokom određenog obračunskog perioda.

Svi ovi nabrojani finansijski obračuni su periodičnog karaktera i izrađuju se najmanje jednom godišnje, sa stanjem na dan 31.12 za proteklu godinu.

Ono što se smatra veoma bitnim kada se razmatra teorijski pristup finansijskom poslovanju jeste insistiranje na kvalitativnim obeležjima finansijskih izveštja. Prvo od njih je relevantnost. Ono predpostavlja da na osnovu informacije koja proistekne iz iveštaja može da se predvidi budući događaj i/ili potvrdi prošlost. Informacija takođe treba da bude blagovremena, odnosno da ne izgubi sposobnost da utiče na odluke. Potom informacija mora da poseduje odliku pouzdanosti i da verno prikazuje finansijsko stanje i uspešnost preduzeća. Ne sme da sadrži greške. Reprezentativnost informacije proističe iz pravilnog prezentovanja reči i brojeva uz uslov pravilnog evidentiranja poslovnih transakcija. Odlika uporedivosti predpostavlja jednoobraznost prikazivanja poslovnih događaja i stanja, jer se jedino tako mogu utvrditi trendovi u finansijskoj poziciji i rezultatima jednog preduzeća u različitim vremenskim periodima ili različitim preduzeća u istom vremenskom periodu. Odlika neutralnosti krase nedvosmislene informacije, one koje ne izazivaju nedoumice. Informacija takođe mora da bude lako razumljiva i iskoristiva. I konačno odlika konzistentnosti (čvrstine) koja informaciji iz perioda u period obezbeđuje sadržinsku i formalnu nepromenljivost.

Oni finansijski izveštaji koji daju tačnu i ispravnu sliku materijalno-finansijske pozicije preduzeća i njegovog poslovnog uspeha mogu se smatrati ispravnim. S druge strane, zamagljen finansijski izveštaj je onaj koji nameće željene informacije korisniku, a analitičare navodi na pogrešne zaključke.

“Pri sastavljanju finansijskih izveštaja moraju se poštovati računovodstveni principi:

- Princip nastanka poslovnog događaja kojim se efekti transakcija i ostalih događaja priznaju tek kad nastanu.

Evidentiranje u poslovnim knjigama i uključivanje u finansijske izveštaje perioda na koji se odnose moguć je samo po nastanku.

- Princip uzročnosti ili sučeljavanja odnosi se na vezivanje prihoda za rashode nastale zbog njihovog ostvarenja.
- Princip stalnosti koji pretpostavlja da će preduzeće koje ima obavezu sastavljanja bilansa nastaviti redovno poslovanje i u doglednoj budućnosti.
- Princip doslednosti treba da obezbedi uporedivost finansijskih izveštaja izborom istih računovodstvenih metoda koje će se koristiti u različitim obračunskim periodima.
- Princip identiteta obavezuje da bilans novo-otvorene poslovne godine bude identičan sa zaključnim bilansom prethodne godine.
- Princip opreznosti koji omogućava da se prihodi i imovina ne precene, a rashodi i obaveze ne potcene.
- Princip pojedinačnog procenjivanja obavezuje na pojedinačnu procenu imovine i obaveza ali i grupno iskazivanje vrednosti u bilansima. “¹

U narednom delu, na komparativan način izdvojene su neke različitosti pojedinih država u svetu kada je finansijsko izveštavanje u pitanju. Razlog tome je da u današnjoj globalizaciji poslovanja, postoji potreba postojanja uporedivih finansijskih izveštaja i harmonizacija računovodstvene prakse. Potreba za umanjivanjem postojećih razlika, kada je neusklađenost postupaka sastavljanja, i računovodstvene revizije završnih računa i konsolidovanih godišnjih obračuna u pitanju, postala je najvažnija za finansijske menadžere koji nastupaju na međudržavnom prostoru. Ovde se svakako misli na preduzeća koja imaju svoje delove na različitim lokacijama u svetu.

Finansijsko izveštavanje u **SAD**-u karakteriše detaljno razrađena računovodstvena regulativa u odnosu na ostale države. Svaka od pedeset američkih država ima svoje sopstveno zakonodavstvo. Pojedinačna zakonodavna tela u okviru svojih granica kontrolišu privredne aktivnosti i ubiraju porez. Američki kongres jeste zakonodavna vlast ali ne utiče direktno na računovodstvenu praksu. O računovodstvenoj politici odlučuje komisija za hartije od vrednosti i berze (*SEC*), a standarde predlaže američki institut ovlašćenih javnih računovođa (*AICPA*). Dakle, uplitanje države je na niskom nivou i težište je na privatnom sektoru udruženom u strukovnu asocijaciju.

¹ Internet izvor – www.IFAC.org

Kada su forme finansijskih izveštaja u pitanju i tu je praksa da uglavnom nema (sa par iznimki) posebno propisanih formata za izveštavanje. Tri su glavna finansijska izveštaja: Bilans stanja, Bilans uspeha i Izveštaj o tokovima gotovine. Karakteristika im je da su manje više deskriptivni (opisni) i da sadrže podatke za tri godine. Tekuću i prethodne dve. Osim ovih glavnih postoje i neki dodatni izveštaji: Izveštaj direktora, Izveštaj revizora, Analiza rezultata poslovanja i finansijske situacije uz razamtranje direktora, Izveštaj o akciskom kapitalu, Prilozi uz finansijske izveštaje, Desetogodišnje poređenje izabranih finansijskih podataka i Kvartalni podaci. Svi oni imaju za cilj da daju potporu glavnim izveštajima i omoguće istinitost i javnost podataka.

Situacija sa finansijskim izveštavanjem u **Velikoj britaniji** je nešto drugačija. Državni organi kreiraju Zakon o kompanijama, a organizacije koje se bave pitanjima računovodstvene prakse obelodanjuju računovodstvene standarde. Postoji visok nivo saradnje i usklađenosti između državnih organa i oragnizacija koje se bave računovodstvenom profesijom. Suština je u poštovanju dva osnovna računovodstvena principa: princip istinitosti i princip pravičnosti. Zakonom je utvrđeno pet principa za sastavljanje finansijskih izveštaja: princip stalnosti, princip konzistentnosti, princip opreznosti, princip uzročnosti i princip pojedinačnog procenjivanja. Britanski standardi usklađeni su sa (MRS – Međunarodni računovodstveni standardi). Za razliku od SAD-ea, u Velikoj Britaniji se često revalorizuje zemljište i zgrade, a kompanije koje to vrše moraju da imaju revalorizacionu rezervu. Konsolidovanje finansijskih izveštaja je normirano kao i u SAD-eu. Zakonom su propisana pravila obelodanjivanja i pravila vrednovanja. Velika Britanija, međutim prepoznaje i kreativno računovostvo koje često za cilj ima da sakrije, a ne da razotkrije informacije interesantne za korisnika finansijskih izveštaja.

Forme Bilansa stanja i Bilansa uspeha su izvedene iz Četvrte direktive Evropske unije i britanski zakon ih propisuje fleksibilno. Kompanijama je omogućeno da biraju između dve forme Bilansa stanja i četiri forme Bilansa uspeha. Interesantno je takođe, da male kompanije ne moraju da publikuju ni Bilans uspeha ni Izveštaj direktora, dok Bilans stanja mogu da publikuju u skraćenom obliku.

U **Nemačkoj** se sistem računovodstva i finansijsko izveštavanje znatno razlikuje od drugih zemalja. Jedino zakon (Trgovački zakon) i sudske odluke imaju obavezujući status. Ne postoje opšteprihvaćeni računovodstveni principi i standardi kao u prethodna dva slučaja. Profesionalno računovodstvo nema jak uticaj kao zakonska regulativa, tako da izostaju aktivnosti oko donošenja računovodstvenih standarda. Institut revizora i Računovodstvena komora Nemačke su konsultativna

tela u procesu donošenja zakona i njihov se rad ispoljava preko preporuka i neobavezujućih izveštaja.

Sva preduzeća su dužna da vode poslovne knjige i izrađuju godišnje izveštaje koji se sastoje od Bilansa stanja i Bilansa uspeha. Samo kompanije moraju da sačine anekse uz bilanse, izveštaje direktora i konsolidovane izveštaje. Računovodstveni principi nisu kodifikovani već se izvode iz zakona.

Forme bilansa su strogo propisane. Bilans stanja mora imati formu dvostranog pregleda, a bilans uspeha može biti samo u vertikalnoj formi, bez obzira da li je sastavljen po metodi ukupnih troškova ili po metodi prodatih učinaka. Bilansi sadrže podatke iz prethodne i tekuće godine.

Što se **Japana** tiče valja istaći da se računovodstvena praksa oslanja na Trgovački zakon po uzoru na francusko-nemački model. Razvoj računovodstvenih standarda nezaisno od zakona, je relativno mlada kategorija. Na japansku računovodstvenu praksu osim Trgovčkog zakona uticaj imaju i Zakon o hartijama od vrednosti i berze, kao i poreski zahtevi. Pravila koja proističu iz zakona su više usmerena na vrednovanje sredstava nego na utvrđivanje rezultata i dobiti. Ovo s toga da bi pored kreditora bili zaštićeni i akcionari. Više se naglašava ekonomski uticaj od legalnog. Osnovna sredstva se u bilans stanja unose po nabavnim cenama umanjenim za amortizaciju. Ovakav način iskazivanja osnovnih sredstava dovodi često do stvaranja skrivenih rezervi, a revalorizacija nije dozvoljena.

Kada su forme finansijskih izveštaja u pitanju, zakon u Japanu nalaže Bilans stanja, Bilans uspeha, Izveštaj o predloženoj raspodeli profita ali i razne dopunske izveštaje. Bilans stanja ima tri dela: sredstva, obaveze i akcijaksi kapital, dok Bilans uspeha ima dve sekcije: redovan dobitak ili gubitak i poseban dobitak ili gubitak.

Kada je **Srbija** u pitanju, povodom računovodstvene prakse i formi finansijskog izveštavanja, treba istaći da se pomaci osećaju u zadnje dve decenije od promene ekonomskog i političkog sistema na ovamo. Tek u nekoliko poslednjih godina oseća se potreba za standardizacijom računovodstva i finansijskog izveštavanja. U ovome, svakako veliki uticaj ima *IFAC - International Federation of Accountants*, *IFAC*-ov etički kodeks za profesionalne računovođe i Međunarodni računovodstveni standardi.

Iz svega prethodno iznetog, jasno je da postoje značajne razlike između navedenih zemalja. Najveće razlike su između evropskih zemalja i SAD-ea, s tim što su britanska i američka praksa slične. Ovakve razlike kao krajnju posledicu imaju to da se finasijski izveštaji preduzeća razlikuju. Informacije iz finansijskog izveštaja jedne zemlje često mogu biti dezinformacija prilikom tumačenja istih u drugoj zemlji sa različitom računovodstvenom praksom. Koliko to zna da

bude ozbiljan problem u vreme globalizacije svetske privrede, samo se da naslutiti. Finansijski menadžeri, pogotovo oni koji se bave formiranjem strategije finansijskog poslovanja preduzeća, imaju veoma težak zadatak, da iz svih informacija koje znaju često zbog ovakvih različitosti da budu kontroverzne i zamagljujuće, donesu adekvatne i korisne odluke.

Analičke baze u poslovnoj strategiji preduzeća

Finansijska funkcija u preduzeću je jedna od veoma važnih i prati relacije između sredstava, dobavljača, kupaca i naplate. Menadžment treba da donese finansijske odluke koje se odnose na pitanja :

- Kako povećati sredstva?
- Šta treba učiniti da se vrši brže pretvaranje sredstava u novčani oblik?
- Koja slobodna sredstva udružiti u druge organizacije?
- Kako uspostaviti obaveze da bi se mogle u rokovima podmiriti?
- Koja novčana sredstva upotrebiti za podmirenje obaveza?
- Kako ostatak dobiti rasporediti i sl.?

„Računovodstvo je proces identifikovanja, registrovanja, merenja i komuniciranja ekonomskih informacija. Ovi procesi omogućavaju kvalifikovanje, prosuđivanje, kontrolu i donošenje odluka od strane korisnika informacija u toku posmatranog perioda.“^[38]

Računovodstvo preduzeća je glavni oslonac menadžmentu preduzeća :

- U planiranju budućih aktivnosti
- U merenju i vrednovanju efikasnosti izvršenja poslovnih aktivnosti
- U rešavanju upravljačkih problema

Računovodstvene odluke su one koje se odnose na pitanja kao što su :

- Kako organizovati praćenje aktivnosti ostalih poslovnih funkcija?
- Kakva treba da bude dokumentacija koja prati sve poslovne tokove?

-
- Kakva treba da bude obrada podataka?
 - Kako treba prikazati rezultate te obrade?

Strategijska orijentacija moći će se ostvarivati samo ako uprava preduzeća pravovremeno dobije informacije o poslovnim događajima. Isto tako, na taktičkom nivou svaka pojedina funkcija, a naročito finansijska, zavise od toga kada i kakve informacije dobije vođa te funkcije i stigne li da reaguje na promene. Koncept utemeljen na organizaciji kao subjektu, ima višestruke prednosti. Preduzeće kao pravno lice odgovara prema trećim licima svom svojom imovinom, ali ne imovinom svojih osnivača, preko onog dela koji su oni uneli u preduzeće.

Preduzeće kao pravno lice izdvaja se od osnivača i deluje potpuno samostalno, preko fizičkih zastupnika, direktora ili advokata. Osnivačka prava lako se prenose na druga lica, a da preduzeće pri tome ne trpi nikakve promene i nesmetano obavlja svoju delatnost.^[38,39] Trajnost delovanja preduzeća pretpostavka je poslovnog uspeha, a ograničenje u tom pogledu izazvalo bi nepoverenje poslovnog partnera. Prikupljanje dopunskih sredstava je olakšano i nema sputavanja broja učesnika, niti visine njihovih uloga. Jedino ograničenje sastoji se u očekivanoj profitabilnosti organizacije.

Budući da se opisani koncept temelji na državnim propisima, naročito je važno da računovodstvo preduzeća prikaže celovito poslovanje u skladu sa zakonom i omogući državnim organima uvid u poslovne događaje. Ovo posebno ističe obveze preduzeća što se tiče plaćanja poreza, tako da je evidencija plaćanja poreza još jedna izuzetno važna uloga računovodstvene funkcije. Porez je naime naknada koju preduzeće daje državi za opšte uslove i sigurnost poslovanja, premda se s unutrašnjeg gledišta tretira kao trošak. Računovodstvene informacije moraju omogućiti upravi preduzeća laku i brzu analizu poslovnih događaja i proveru kako se ostvaruje zamišljena strategija.

Knjigovodstvene aktivnosti podrazumevaju :

- Obavljanje poslova evidentiranja promena novčanog poslovanja
- Vođenje glavne knjige i celokune sumarne evidencije
- Izradu finansijskih izveštaja

„Finansijsko računovodstvo ima za cilj periodično izveštavanje o stanju i uspehu preduzeća, tj. obračun i izveštaj o dobitku ili gubitku preduzeća u toku perioda, kao i korektno vrednovanje i iskazivanje sredstava, obaveza i vlasničkog kapitala preduzeća na poslednji dan

perioda. Ono je usmereno na celinu preduzeća i njegovo okruženje i predstavlja jedini računovodstveni izvor informacija za eksterne korisnike, kao što su državni organi i agencije, akcionari, trgovački kreditori i banke.

Upravljačko računovodstvo se odnosi na interno izveštavanje menadžmenta i menadžera, za potrebe donošenja poslovnih odluka i sistematske kontrole poslovanja. Predstavlja interno orijentisan sistem globalnog i analitičkog izveštavanja koji se bavi prošlošću, ali pre svega budućnošću preduzeća. Upravljačko računovodstvo koristi informacije finansijskog računovodstva i operativne informacije za potrebe menadžmenta. Ono se odnosi na obračun troškova kao i računovodstveno planiranje i kontrolu. ^[25]

Analiza profitabilnosti ^[25] sve više dobija na značaju. Čak i najprofitabilnije organizacije treba da znaju šta je izvor njihovog profita, a manje profitabilne gde su podbacile. Ova analiza je neophodna da bi se utvrdila ispravna politika cena (i popusta), definisale promotivne aktivnosti, izabrale oblasti investiranja i deinvestiranja, anticipirao pritisak konkurencije, itd. Pored toga važna je i analiza troškova, a popularan način da se utvrde i alociraju troškovi je tehnika određivanja troškova na bazi aktivnosti (*Activity Based Costing*). Na tržištu postoje brojni ABC alati koji imaju mnoge OLAP karakteristike, a moguće je za ovu primenu direktno koristiti OLAP tehnologiju.

Tehnologije skladišta podataka (*Data Warehouse*), stvorila je mogućnost za razvoj savremenih sistema za podršku u odlučivanju i još je jasnije povučena granica između transakcionih sistema (za podršku i automatizaciju poslovnih procesa) i informacionih sistema za menadžment preduzeća. ^[13] Ova dva podsistema nikako nisu u koliziji, već su naprotiv komplementarni. Transakcioni sistem, odnosno podaci nastali u njemu, su čak šta više osnova za izgradnju sistema za podršku u odlučivanju. Upotrebom tehnologije skladišta podataka u ISM-u (Informacioni sistem za menadžment), pogodno se transformišu podatci iz transakcionih sistema i omogućuju njihovu analizu. Pored toga, ISM obuhvata i druge podatke: sa tržišta, od institucija koje se bave praćenjem i procenama ekonomskih kretanja, itd. Često se u ISM sistemima govori o "tvrdim" i "mekim" podacima. "Tvrdi" podaci su oni koje možemo smatrati tačnima (obično podaci iz transakcionog sistema; na primer: vrednost fakturisane robe, otpremljena količina, itd.). "Meki" podaci su oni koje smatramo približno tačnim ("verovatno tačnim") i koji pre svega kvalitativno opisuju pojave od interesa. Obuhvat ovih podataka je veoma važan jer ih menadžeri moraju uzeti u obzir prilikom analize situacije i donošenja odluka.

Tehnologija skladišta podataka i analitičke obrade (*On Line Analytical Processing*) omogućava da se navedena ograničenja OLTP sistema uklone. Skladišta podataka omogućavaju da se, tipično heterogeni podaci, objedine u homogenom skladištu podataka. Korisnicima koji se bave analizom podataka ova tehnologija pruža sledeće koristi:

- podaci su organizovani na način koji olakšava analizu, a ne transakcionu obradu,
- razrešavaju se nehomogenosti u podacima koji se crpe iz heterogenih izvora,
- podaci se transformišu i konsoliduju, a često i agregiraju pre prepisivanja u skladište,
- zaštita podataka i performanse sistema se mogu poboljšati, a de se ne dira u postojeći transakcioni sistem.

Skladišta služe za čuvanje podataka, dok *OLAP* predstavlja tehnologiju koja korisnicima omogućava efikasan pristup podacima. *OLAP* korisnicima omogućava:

- preagregaciju često korišćenih podataka, a time i brz pristup podacima (kratko vreme izvršenja upita),
- intuitivan višedimenzioni podel podataka, koji omogućava laku selekciju, navigaciju i pretraživanje podataka,
- nov pogled na podatke korišćenjem niza analitičkih (statističkih) funkcija,
- upravljanje zaštitom podataka i optimizaciju performansi sistema.

Iako se pojmovi, skladište podataka i *OLAP*, često koriste kao sinonimi i u suštini predstavljaju dva lica istog sistema, potrebno je razumeti gore navedene razlike.

Kad se govori o konceptu analitičkih baza podataka u formiranju strategije finansijskog poslovanja preduzeća, onda je od naročitog značaja da pokažemo da "tvrdi" podaci, nastali u transakcionom sistemu, mogu korišćenjem tehnike skladišta podataka biti adekvatno analizirani, da menadžeri mogu samostalno i u realnom

vremenu (dakle: kad im je to potrebno) vršiti analizu podataka u skladištu i da sve to zajedno predstavlja zdravu podlogu za izgradnju savremenog ISM-a. U daljem korišćenju i razvijanju sistema za podršku u odlučivanju moguće je postupno uvoditi i druge napredne tehnike: statističke tehnike, tehnike crpljenja podataka iz skladišta, ekspertne sisteme, neuronske mreže i slično. ^[14]

Za potrebe tokvog sistema za podršku u odlučivanju razvijeni su dimenzionalni modeli, koji omogućuju da se stekne uvid ne samo u osnovne elemente poslovanja već i mogućnost da se analiziraju trendovi, osnovni indikatori (racio brojevi) koji kompleksno opisuju poslovanje.

Ovde će u nastavku biti prikazano jedno od mogućih modularnih rešenja, koje omogućuje da se podaci u skladištu podataka integrišu iz postojećeg transakcionog sistema.

Dva su osnovna načina izrade modela, konvencionalni i objektno orjentisani pristup ^[36,65,73, 85, 87,127] Nedostaci konvencionalnog pristupa bili bi:

- mala modularnost sistema
- statički koncept modela podataka
- nesaglasnost jezika baze podataka i programskih jezika
- razvoj sistema u jednom prolazu

Prednosti objektno orjentisanih pristupa bile bi:

- objekti se mogu koristiti više puta
- nezavisno testiranje objekata
- lakši timski rad
- razvoj objekata koji su nezavisni od mašine na kojoj rade
- velika prilagodljivost na promene



1. DEFINISANJE ZAHTEVA

Prilikom izrade zahteva koji se postavljaju pred budući informacioni sistem, a koji pokriva oblast finansijskog poslovanja nekog preduzeća ili institucije, mora se poći od trenutno važećih domaćih zakona i propisa ovičavajućih za ovu oblast s jedne strane i Međunarodnih računovodstvenih standarda (*IAS - International Accountant Standards*) strukturiranih po *ISO 9001/2000* s druge strane. Obzirom da je aktuelno postupno približavanje ovih dveju regulativa u daljem tekstu će biti vođena pažnja o konsultaciji obe.

U definisanje zahteva ka izradi softvera, koji će za cilj imati: koncept analitičkih baza podataka u formiranju strategije finasijskog poslovanja preduzeća, treba poći od premise da je menadžment skup funkcija usmerenih na efikasnu i efektivnu upotrebu resursa da bi se ostvarili ciljevi organizacije. Valjalo bi odmah definisati šta je to strateški menadžment. Odgovor na ovo pitanje proizilazi iz podele menadžmenta koju su mnogi značajni teoretičari organizacije isticali kao valjan primer. Podela je zamišljena kao piramida, čiju osnovu čini operativni menadžment (*first-line*), potom dolazi srednji menadžment (*middle*), da bi se na samom vrhu piramide nalazio strateški (*top*) menadžment. Već iz ovako slikovite postavke jasno je da manja grupa najviših rukovodilaca vodi celu organizaciju, definiše ciljeve, pretstavlja organizaciju i obezbeđuje resurse. Sledeći niži segment predstavlja relativno veliku grupu rukovodilaca odgovornih za implementaciju politike i planova strateškog menadžmenta. Na samoj osnovi piramide nalaze se menadžeri koji koordiniraju i kontrolišu aktivnosti izvršilaca zaduženih za proizvodnju i usluge.

Zahtevi strateškog menadžmenta su u domenu finansijskog poslovanja od apsolutno prioritnog značaja. Top menadžment mora da bude u stanju da na osnovu informacija dobijenih iz finansijske funkcije preduzeća ili institucije, može da donese najkvalitetniju odluku u domenu poslovanja.^[107]

Na slici 1.1 predstavljen je polazni OLTP model "Računovodstva proizvodnog preduzeća" iz kojeg treba izvući samo neophodne entitete koji će podržati glavnu aktivnost koju ova disertacija obrađuje, odnosno EKONOMSKO FINANSIJSKE POSLOVE. Predstavljeni informacioni model^[108] sadrži preko trideset tabela informacionog podsistema "Računovodstva proizvodnog preduzeća". Ovaj podsistem je samo deo globalnog sistema jednog proizvodnog preduzeća.

Arhitektura informacionih sistema obezbeđuje unificirajući okvir unutar koga je različitim ljudima sa različitim perspektivama moguće

organizovanje i prikaz građenja osnovnih blokova informacionih sistema.

Različiti su ključni korisnici (*stakeholders*)^[99,103,119] odnosno perspektive iz kojih oni posmatraju arhitekturu informacionog sistema:

- Vlasnici sistema plaćaju da sistem bude ugrađen i održavan.
- Korisnici sistema koriste sistem za izvršenje ili podršku poslovima koje treba završiti.
- Projektanti sistema projektuju sistem odgovarajući korisničkim zahtevima.
- Graditelji sistema konstruišu, ispituju, i isporučuju sistem u radu.
- Analitičari sistema omogućuju razvoj informacionih sistema i primenu kompjutera u premošćenju prekida komunikacija između netehničkih korisnika sistema i tehničkih sistem projekatara i sistem konstruktora.
- Prodavci i konsultanti IT prodaju hardver, softver, i usluge u poslovanju za uključivanje unutar njihovih informacionih sistema.

Pošto u fazi definisanja zahteva^[28] posmatramo podatke kao sirovinu za stvaranje budućih korisnih informacija, onda je potrebno da na ovim stranama osvetlimo činjenice kako pojedini ključni korisnici fokusiraju podatke:

- Iz perspektive vlasnika sistema:
 - Podaci se shvataju kao poslovno znanje koje je postignuto blagovremenim i relevantnim informacijama. (Podsetimo da su informacije proizvod sirovih podataka.)
- Iz perspektive korisnika:
 - Fokusiraju se zahtevi za podacima, jer su oni predstavljanje korisnikovih činjenica u kategorijama entiteta, atributa, veza i pravila. Zahtevi za podacima treba da budu eksplicitno formatirani to jest nezavisni od tehnologije koja može ili će biti korišćena za uskladištenje podataka:
- Iz perspektive projektanta sistema:
 - fokus je na šemi baze podataka
- Iz perspektive konstruktora sistema:
 - Sistem za upravljanje bazama podataka predstavlja krajnje ishodište shvatanja podataka.

Kako podaci nisu sami sebi cilj, već informacija, razjasnićemo i taj pojam. Informacija je bivši prečišćeni podatak organizovan i po

1.1 Prikupljanje zahteva

Prikupljanje zahteva bi se moglo u osnovi podeliti na prikupljanje korisničkih zahteva i prikupljanje izvornih zahteva. Cilj i jednih i drugih je definisanje sveukupnih zahteva koji dovode do izrade logičkog modela funkcija i fizičkog modela poslovnih procesa.^[46]

Kada se razmišlja u kontekstu formiranja strategije finansijskog poslovanja, jasno je da će prava adresa za prikupljanje zahteva biti finansijski direktor ili šef računovodstva. Ove osobe bi smo mogli svrstati u kategoriju sistem analitičara za našu rečenu oblast. No i pored toga, na samom početku, neophodno je da se projektni tim upozna sa vlasnikom sistema i od njega iscrpi osnovne informacije.

Deklarativno izneti zahtevi nakon intervju sa vlasnikom sistema izgledali su kao na slici 1.2, predloženi u 9 tačaka.

Slika 1.2 Prikaz zahteva vlasnika sistema

Захтеви власника система	
Захтев бр.	Опис захтева
	Да се изради софтвер на нивоу предузећа који ће омогућити књиговодствено-рачуноводственом делу: унос, обраду и складиштење података од интереса за овај посао са једне стране и аналитичку обраду ускладиштених података као основу за подршку
1	одлучивању финансијском менаџменту с друге стране.
2	Да се хардвер подигне на актуелни ниво Информационе технологије
3	Да се користи савременији оперативни и мрежни систем
4	Да се за СУБП користи такође неки из групе новије технологије
5	Да апликација буде компактна, са могућношћу оператора да у зависности од датих му привилегија бира модуле програма
6	Да софтвер буде урађен на принципу клијент-сервер архитектуре, што подразумева рационално коришћење ресурса свих рачунара.
7	Неопходно је да се уради сва техничка документација око израде софтвера у некој од графичко-текстуалних технологија данас признатих у свету
8	Да постоји прилаз формама и извештајима кроз GUI и преко менија командне табле
9	Да постоје одвојене групе модула софтвера намењене OLTP-у и OLAP-у

Izvor: Miroљub Zahorjanski

1.1.1 Prikupljanje korisničkih zahteva

Metoda intervjua (odozgo na dole), koja se koristi u prikupljanju korisničkih zahteva, daje širinu u pristupu i bazira se na definisanju zahteva koje korisnik izvršava. Ova metoda izvodi se sa stanovišta već pomenutog top menadžmenta. Za ovu priliku projektni tim je intervjuisao sistem analitičara u preduzeću, zaduženog za finansijsko-računovodstvene poslove. Rezultat ankete je dat na slici broj 1.3, kroz 18 tačaka.

Slika 1.3 Prikaz zahteva sistem analitičara

Захтеви ситем аналитичара	
Захтев бр.	Опис захтева
1	Да постоје припрема и одржавање шифарника (партнери, конта, тарифне групе.....). Шифарници ће имати табеларни изглед, а шифарнк Контног плана, ће бити направљен употребом "TreeView Control" због примене хијерархије
2	Да постоји књижење докумената насталих из књиговодствених исправа (извод са текућих рачуна, благајна, књижна писма, компензације, асигнације, записници, полиси и слично)
3	Да постоји финансијско рачуноводство, где ће се књижити подаци из осталих делова књиговодства (робно, погонско...) али и остале књиговодствене исправе.
4	Да постоји анализирање података робног и погонског књиговодства кроз извештаје за књиговодство, менаџмент и инспекцију.
5	Да извештаји за менаџмент буду одрађени уз помоћ ОЛАП система. Одлучено је да се помоћу MS QUERY-а омогући више различитих извештаја.
6	Да постоји анализирање података финансијског рачуноводства као што су (налози, дневници, картице, биланси...)
7	У току ревизије одлучено је да се догради могућност преноса података из биланса у Програмски пакет за израду финансијских извештаја Народне банке Србије.
8	Раздвојен унос погонских и финансијских докумената удовољава тежњи да сваки извршилац ради онај део књиговодства-рачунводства за који је оспособљен и задужен.
9	Добро класификована анализа података са доступношћу тачно оних података који су неопходни за конкретни извештај повећава задовољство наручиоца. Могућност да се сваки извештај претходно погледа на екрану повећава удобност и ефикасност, а смањује нервозу због евентуалног губљења времена и трошења папира.
10	Структурирани хелп по принципу стабла са доступношћу темама и брзи приступ теми из одговарајуће форме или извештаја помоћу дугмета Ф1 кориснику дају самопоуздање и аутономност у раду без превелике потребе да се ослања на техничку подршку.

11	Да се увек зна тачно стање количина у магацинима без обрачунавања промета.
12	Да количине не могу да иду у минус
13	Да сваки документ може да се одштампа у изворном облику (како је унешен). Омогућено је штампање документа у стању пре књижења.
14	Да се количине у свим магацинима воде по просечним ценама.
15	Да употребљаване рецептуре (саставница материјала) не могу да се бришу нити мењају
16	Да употребљаване шифре партнера не могу да се бришу нити мењају
17	Да се може водити погонско књиговодство за већи (неограничени) број клијента.
18	Упутство за коришћење програма да буде троспојно. 1. Генерално упутство дато у менију и командној табли. 2. Хелп подршка везана за сваку форму, извештај или упит појединачно. 3. Поруке које програм шаље кроз дијалог са корисником. Ово прво је урађено применом "TreeView Control"

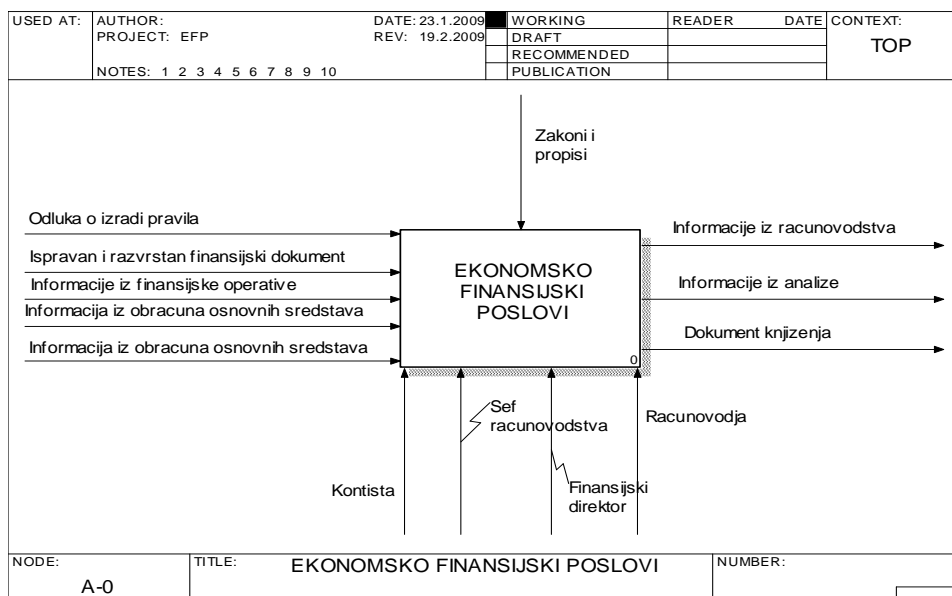
Izvor: Miroljub Zahorjanski

Ovakvi pobrojani zahtevi sistem analitičara predstavljaju zbirku zapisa na koju projektni tim obraća pažnju tokom celog procesa izrade projektnog zadatka.

No ono što je stvarni produkt intervjuja sa sistem analitičarem izgleda nešto drugačije. Upotrebom i korišćenjem *IDEFO* standarda, kao i alata *BPwin* u ovom konkretnom slučaju, izvedena je funkcionalna specifikacija budućeg informacionog sistema. Ovakvom aktivnošću identifikovane su granice posmatranog sistema, potom vertikalno povezivanje funkcija putem definisanja stabla logičkih funkcija, kao i horizontalno povezivanje kroz izradu dijagrama dekompozicije. Opisanom metodom je na kraju postignut prikaz dinamičke integrativnosti poslova koji se odvijaju u posmatranom okruženju projektnog zadatka.

Da bi smo sistem definisali moramo uvesti granice sistema koje grafički predstavljamo kontekstnim dijagramom. Dijagram konteksta vizuelizuje granicu modela koji se proučava dok u sistemu i van njega teku informacije preko strelica. Ovo je najviši nivo apstrakcije i putem dekompozicionih dijagrama prevodi se u njene niže nivoe.

Slika 1.4 Dijagram konteksta



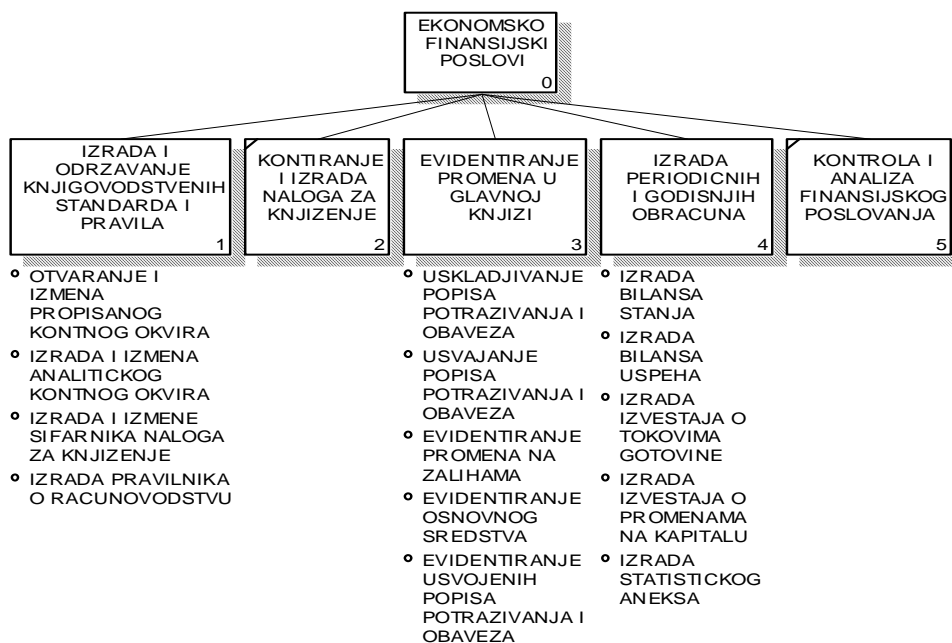
Izvor: Miroljub Zahorjanski

Na slici 1.4 prikazan je dijagram konteksta za ekonomsko finansijske poslove sa istoimenom glavnom aktivnošću. [79]

Strelice sa leve strane pravougaonika definišu se kao ulazi (*Input*). Strelice koje ulaze u pravougaonik odozgo definišu se kao kontrole (*Control*). Strelice koje izlaze iz pravougaonika na desnoj strani predstavljaju izlaze (*Output*). Izlazi su podaci ili objekti proizvedeni od strane aktivnosti. Strelice na donjoj strani pravougaonika predstavljaju mehanizme. Strelice okrenute prema gore identifikuju značenja koje podržavaju izvršenje aktivnosti. Strelice mehanizma koje su okrenute nadole definišu se kao strelice poziva (*Call arrows*).

Stablo procesa, (slika 1.5) omogućuje da se polazni složeni posao razvije u hijerarhiju podređenih poslova, čija je struktura tipa izvrnutog stabla. Koren stabla, kao najviši čvor stabla, sadrži polazni posao (u našem slučaju EKONOMSKO FINANSIJSKI POSLOVI), dok listovi, tj. čvorovi koji nemaju potomke, sadrže poslove čije je rešavanje relativno jednostavno. Rešavanjem svih podređenih poslova iz listova rešen je i polazni složeni posao.

Slika 1.5 Stablo aktivnosti za polazni posao



Izvor: Miroljub Zahorjanski

Dakle, stablo poslova predstavlja hijerarhiju definisanih poslova, očišćenu od strelica, i omogućuje funkcionalnu dekompoziciju i uvid u dubinu odvijanja veza između poslova.

Poslovi na vrhu (*root*) uvek su označeni sa 0. Brojevi se koriste da bi prikazali koliko detalja sadrži posao. Posao A0 je dekomponovan (razdvojen) na 1, 2, 3, itd. Posao 1 je dekomponovan u 11, 12, 13, itd. Nadređeni posao se zove roditelj (*parent*), a podređeni poslovi su deca (*childs*).

Razbijanje poslova „roditelja“ na svoju „decu“ treba da ima od 2 do 6 podređenih poslova. Ako je više od šest podređenih poslova, to znači pokušaj da se previše detalja smesti na jedan nivo.

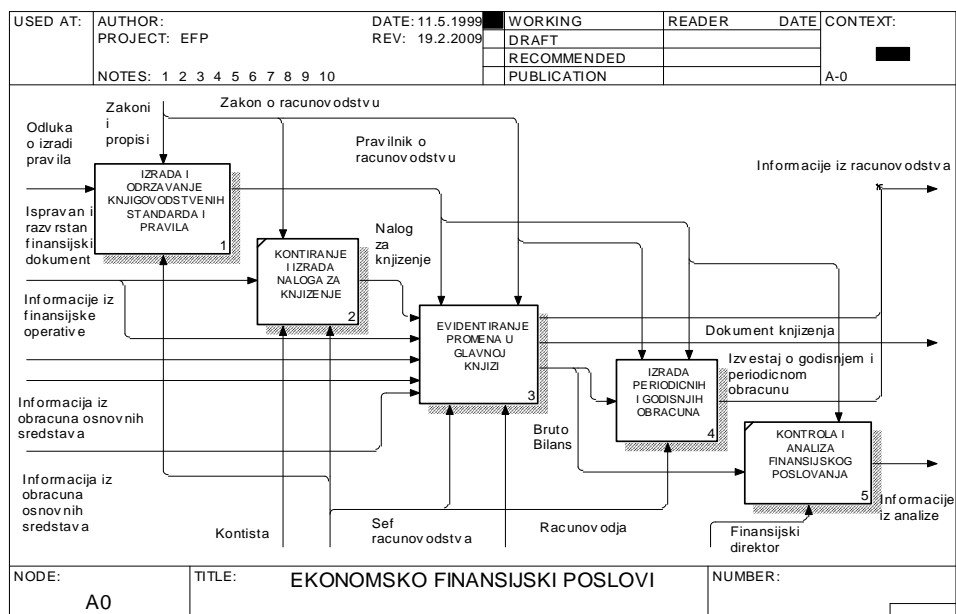
Na slici 1.5 vidimo kompletno stablo aktivnosti za polazni posao EKONOMSKO FINANSIJSKI POSLOVI. Na ovaj način su prikazane vertikalne veze između aktivnosti kompletnog modela.

Korišćenjem *IDEFO* metode izvodi se definisanje fundamentalnih funkcija. Pravougaonici predstavljaju poslove definisane kao funkcije i imaju nazive i brojeve u okviru svojih granica. Dekompozicionim dijagramima uspostavljaju se horizontalne veze između poslova istog nivoa. Eksplicitne ili interne strelice (*Internal Arrows*), uazuju na funkcionalni pristup dekompoziciji. U suprotnom bi

smo imali čisto organizacioni pristup. Na osnovu definisanog dijagrama konteksta za Ekonomsko finansijske poslove prikazanog na

slici 1.4 i stabla aktivnosti sa slike 1.5 definiše se dijagram dekompozicije prikazan na slici 1.6.

Slika 1.6 Dijagram dekompozicije prvog nivoa



Izvor: Miroljub Zahorjanski

1.1.2 Prikupljanje izvornih zahteva

Dok je za prikupljanje korisničkih zahteva primenjivana metoda odozgo na dole (*Top down*), dotle se u radovima i aktivnostima na prikupljanju izvornih zahteva (*source-driven*) koristi metoda odozdo na gore (*Bottom up*). Ova metoda omogućuje preciznost i bazirana je na definisanju zahteva analiziranjem podataka za transakcione baze korišćenjem *ER* modela podataka. ^[100,101]

Za primenu ove metode neophodno je bilo prikupljanje svih dokumenata koji su u vezi sa budućom modelovanom aktivnosti EKONOMSKO FINANSIJSKI POSLOVI. Članovi projektnog tima, u saradnji sa licima, koje je odredio vlasnik sistema i koji poznaju određene uže oblasti, prikupljaju dokumente. Ove dokumente u osnovi možemo podeliti na tri grupe sa aspekta namene:

- Održavanje podataka
- Održavanje šifarnika
- Izrada izveštaja

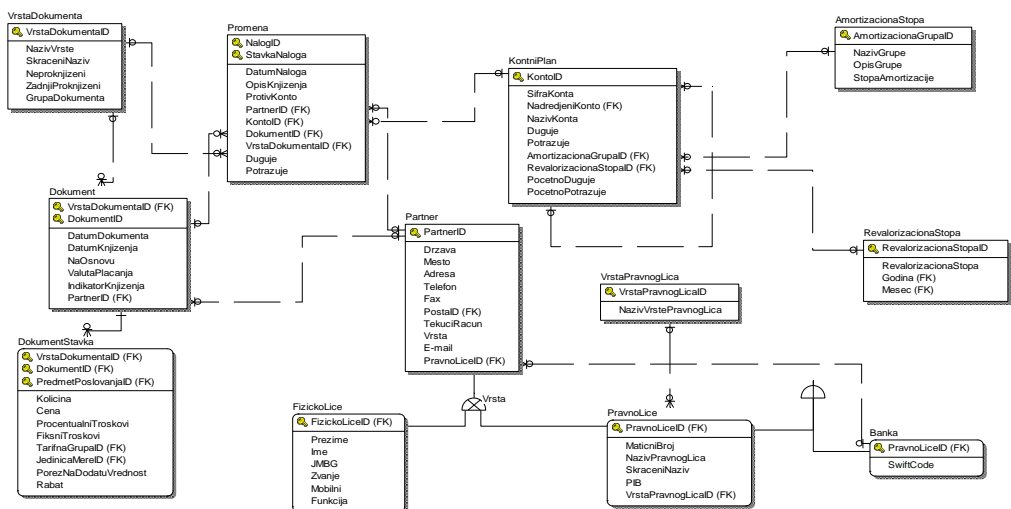
U primeru koji sledi, učinjen je napor da se svi dokumenti generalizuju (odrede im se zajednički atributi) i svrstaju pod jedan glavni entitet, u ovom slučaju entitet

“*Dokument*”. Iz egzistencijalnih razloga entitetu “*Dokument*” pridružen je i entitet „*DokumentStavka*”, dok je kao subkategorijalni entitet predviđen entitet “*NalogZaKnjizenje*” sa stavkama iskazanim u “*NalogZaKnjizenjeStavka*”.

Na ovaj način ER model prikazan na slici 1.7 može bez problema da postane deo nekog kompleksnijeg modela koji korespondira sa poslovima i aktivnostima na nivou nekog preduzeća ili institucije. Zavisni subkategorijalni entiteti nepotpune strukture daju mogućnost projektantu da definiše sve različitosti i specifičnosti dokumenata iz gore pomenute grupe sa aspekta namene.

Sa aspekta izrade izveštaja vođeno je računa da dokumenti budu usklađeni sa Međunarodnim standardima finansijskog izveštavanja (*IFRS – International Financial Reporting Standards*). Ovo se svakako odnosilo na izlazne izveštaje *OLTP* sistema.

Slika 1.7 ER model za Ekonomsko finansijske poslove



Izvor: Miroljub Zahorjanski

Pri kreiranju entiteta, odnosno njihovih atributa, velika pažnja se poklanjala verodostojnosti dokumenata, korišćenju najnovijih verzija i njihovoj usklađenosti sa gore pomenutim standardima.

Na slici 1.7 prikazana je *IDEF1X* notacija. Identifikujuće veze prikazane su punom linijom, dok su neidentifikujuće prikazane kao isprekidane. Nezavisni entiteti su u pravougaonicima koji imaju oštre uglove, dok su zavisni entiteti prikazani zaobljenim pravougaonicima.

Iako entitet „*Dokument*” čini okosnicu modela, valja napomenuti da njegovi generički podentiteti „*NalogZaKnjizenje*” i „*NalogZaKnjizenjeStavka*” predstavljaju izvor podataka za 90% budućeg skladišta (*Data Warehousing*).^[50,51,53,55,74,78]

1.1.3 Analiza OLTP podataka

Ova vrsta analize vezana je za upitni jezik SQL^[4,41] koji omogućava realizaciju *ad-hoc* upita i čijom analizom dolazimo do potrebnih informacija vezanih za kasniji razvoj višedimenzionih modela. SQL je jezik za interaktivno definisanje baze podataka (*Data Definition Language - DDL*), kao i jezik za manipulaciju podacima (*Data Manipulation Language - DML*). SQL jezik je neproceduralan jer specificira operacije u smislu ŠTA treba uraditi a ne KAKO. Na osnovu analize upita nad *OLTP* podacima, prikupljaju se zahtevi koji treba da ukažu na pravce kojima je potrebno ići prilikom definisanja višedimenzionog modela.

Poređenje *OLTP* i *OLAP* sistema:

OLTP

Obrada podataka
Mnogo kratkih transakcija
Mb-Gb podataka
Sirovi podaci
Mnoštvo različitih korisnika
Sadrži ažurne podatke itd.

OLAP

Čitanje podataka
Dugački i kompleksni upiti
Gb-Tb podataka
Sumarni i prerađeni podaci.
Korisnici su donosioci odluka
i analitičari
Sadrži istorijske podatke itd.

Na konkretnom primeru EKONOMSKO FINANSIJSKI POSLOVI, analiza *OLTP* podataka se satojala s jedne strane u kreiranju upita za mnoštvo različitih korisnika kojima su potrebni svakodnevni izveštaji iz transakcionih procesa, a s druge strane u kreiranju onih upita kojima bi trebalo prikupiti podatke u cilju kasnijeg definisanja višedimenzionog modela. Po jedan od upita iz ova dva domena prikazan je u nastavku teksta.

a) Primer upita za svakodnevne izveštaje

```
SELECT      Konto.KontoID, Konto.BrojKonta, Konto.NazivKonta,
            Konto.NadredjeniKonto, [BrojKonta] & " " & [NazivKonta]
            AS Naziv
FROM        Konto
ORDER BY   [BrojKonta] & " " & [NazivKonta];
```

b) Primer upita u cilju kasnijeg definisanja višedimenzionog modela:

```
SELECT      QDokumentCinjenica.DatumDokumenta,
            QDokumentCinjenica.Godina, QDokumentCinjenica.Mesec,
            Mesec.NazivMeseca, QDokumentCinjenica.DanUNedeljiID,
            DanUNedelji.DanUNedelji, QDokumentCinjenica.Dan
FROM        (QDokumentCinjenica
            INNER JOIN Mesec
            ON QDokumentCinjenica.Mesec = Mesec.Mesec)
            INNER JOIN DanUNedelji
            ON QDokumentCinjenica.DanUNedeljiID =
            DanUNedelji.DanUNedeljiID
GROUP BY   QDokumentCinjenica.DatumDokumenta,
            QDokumentCinjenica.Godina, QDokumentCinjenica.Mesec,
            Mesec.NazivMeseca, QDokumentCinjenica.DanUNedeljiID,
            DanUNedelji.DanUNedelji, QDokumentCinjenica.Dan;
```

1.2 Izrada modela poslovnih slučajeva upotrebe

Neosporno je da je razumevanje procesa poslovanja ključ uspeha. Način poslovanja se stalno menja, evoluirá i postaje sve kompleksniji. S informatičke tačke gledišta, primena objektnog modeliranja, odnosno vizuelnog modeliranja procesa poslovanja, predstavlja način da se stekne uvid u tok poslovanja onakav kakav on jeste, sa ciljem njegovog daljeg unapređenja i poboljšanja. Vizuelno modeliranje^[116] povećava efikasnost rada i omogućuje brže i lakše shvatanje velikih procesa i struktura.

„Za objektno modeliranje poslovnih procesa koristi se *UML (Unified Modeling Language)* standard, a cilj je napraviti dobar poslovni model koji će poslužiti kao osnova za razvoj softvera poslovne inteligencije. Ovakav pristup kao rezultat daje opšti model dizajna i nezavistan je od programskih jezika ili softverskih platformi na kojima će se informacioni sistem kasnije implementirati. Objektno orijentisani principi omogućuju da se implementacijom dodatnih

funkcionalnosti dopunjuju, a ne značajno menjaju arhitektura i model softvera”^[99].

UML pruža mogućnosti da se jednim jezikom modeluje poslovni sistem, aplikacija, baza podataka i arhitektura sistema.

U okviru modela poslovnih slučajeva upotrebe, definiše se dijagram poslovnih slučajeva upotrebe i dijagram poslovnih aktivnosti.

Kada su procesi u pitanju, odnosno aktivnosti (uključujući upravljanje) koji izvršavaju zadatak za posao, onda se takođe može reći da su različiti fokusi različitih grupa ključnih korisnika:

- Iz perspektive vlasnika sistema
 - Poslovne funkcije su aktivnosti koje se odvijaju dajući podršku poslovanju. Funkcije mogu biti rastavljene na druge podfunkcije i eventualne procese tako izvršavajući specifične zadatke.
- Iz perspektive korisnika sistema
 - Poslovni procesi su aktivnosti koje reaguju na poslovne događaje. Poslovni procesi su “rad” obavljen od strane sistema.
 - Zahtevi procesa su predstavljanje korisničkih poslovnih procesa u kategorijama aktivnosti, tokova podataka ili tokova rada.
 - Politika je skup propozicija koje vladaju poslovnim procesima.
 - Procedura je skup instrukcija tipa korak po korak i logike za izvršenje poslovnih procesa.
- Iz perspektive projektanta sistema
 - Aplikaciona šema je model koji saopštava kako su odabrani poslovni procesi ili će biti izvršena upotreba softvera i hardvera.
 - Specifikacija softvera predstavlja tehnički projekat poslovnih procesa koji će biti automatizovani ili podržani kompjuterskim programom napisanim od konstruktora sistema.
- Iz perspektive konstruktora sistema
 - Aplikacioni programi su jezički bazirane, mašinski čitljive reprezentacije za ono što je softverski proces zamišljen da radi, ili kako je softverski proces zamišljen da izvrši zadatak.
 - Prototip je tehnika za brzu izradu funkcionalnog ali nekompletnog modela informacionog sistema korišćenjem brzih aplikacionih alatki za razvoj.

1.2.1 Dijagram poslovnih slučajeva upotrebe

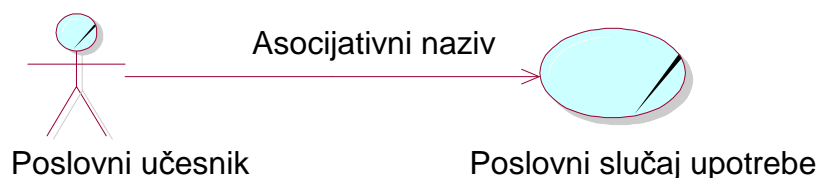
Ovaj dijagram prikazuje poslovne slučajeve upotrebe i učesnike, kao i njihove relacije. Dijagram poslovnih slučajeva se koristi da bi se predstavio statički pogled na funkcionalnost (ponašanje) sistema. Korišćenjem ovog dijagrama može da se modeluje kontekst sistema: granica sistema i učesnici koji su sa njima u interakciji.

Za svaki definisani poslovni slučaj upotrebe, razvija se poslovni dijagram aktivnosti. Za pojedine poslovne aktivnosti iz dijagrama poslovnih aktivnosti razvija se dijagram sistemskih slučajeva upotrebe, a sistemski slučaj upotrebe opisuju se sistemskim dijagramom aktivnosti. ^[98]

Dijagram korisničke funkcije je grafički prikaz pojedinih ili svih izvođača, korisničkih funkcija i njihovih interakcija identifikovanih u sistemu. Uobičajeno je da svaki sistem ima glavni dijagram korisničkih funkcija (*Main Use Case diagram*), koji je slika granice sistema (izvođači) i glavnih funkcionalnosti koje sistem obezbeđuje (korisničke funkcije). Po potrebi se mogu napraviti i drugi dijagrami korisničkih funkcija.

Poslovni učesnik (*Business Actor*) pokreće poslovne slučajeve upotrebe (*Business Use-Case*) preko veze koja se prikazuje asocijativnom (koja može da se dodatno opiše asocijativnim nazivom) tipa "u interakciji je sa" ili "naređuje".

Slika 1.8 Grafički prikaz poslovnih slučajeva upotrebe



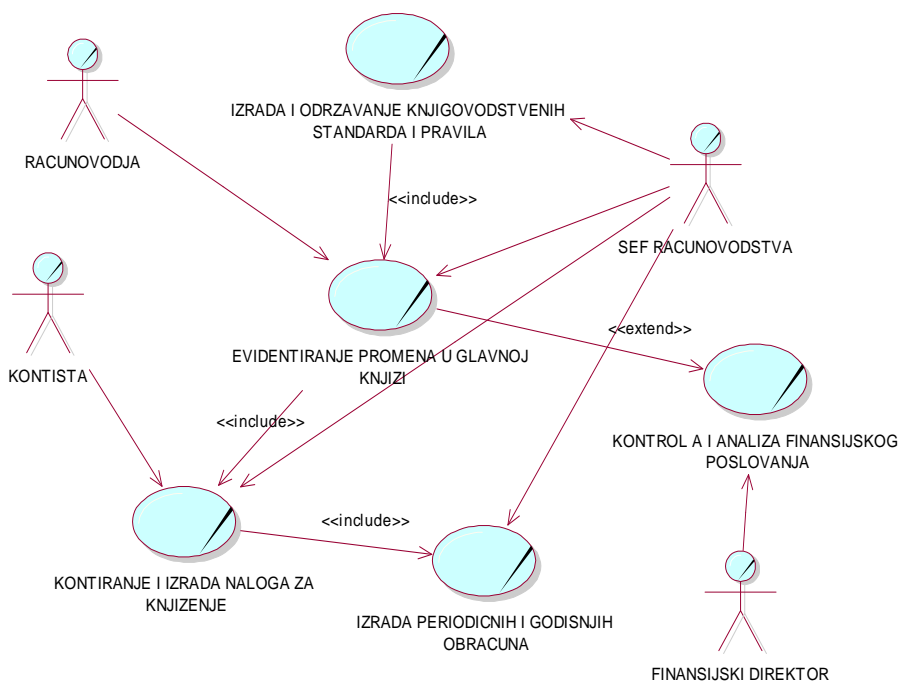
Izvor: Miroljub Zahorjanski

Kada je dijagram poslovnog slučaja upotrebe u pitanju može se sa sigurnošću reći da on predstavlja pandam dijagramu konteksta u

BPwin alatu. Upoređivanjem slike 1.9. i 1.4. to nedvosmisleno pokazuje.

Model poslovnih slučajeva upotrebe detaljno opisuje poslovne procese korišćenjem *UML* dijagrama poslovnih slučajeva upotrebe koji se obavljaju u konkretnom organizacionom i tehnološkom okruženju. Poslovni slučajevi upotrebe opisuju funkcionalnost sistema iz perspektive rukovodstva.

Slika 1.9. Dijagram poslovnog slučaja upotrebe za EKONOMSKO FINANSIJSKE POSLOVE



Izvor: Miroљjub Zahorjanski

U konkretnom slučaju, poslovni učesnici su: finansijski direktor, šef računovodstva, kontista i računovođa:

- Finansijski direktor vrši kontrolu i analizu celokupnog finansijskog poslovanja.
- Šef računovodstva vrši izradu i održavanje knjigovodstvenih standarda i pravila, a nadzire kontiranje, izradu naloga za knjiženje i evidentiranje promena u glavnoj knjizi. Najmanje jednom godišnje šef računovodstva vrši izradu periodičnih i godišnjih izveštaja.

-
- Kontista je zadužen isključivo za kontiranje i izradu naloga za knjiženje.
 - Računovođa evidentira promene u glavnoj knjizi.

Poslovni slučajevi upotrebe koji se razmatraju u ovom radu su:

- Izrada i održavanje knjigovodstvenih standarda i pravila, koja treba da omoguće usklađenost sa zakonskom regulativom iz oblasti računovodstva.
- Kontiranje i izrada naloga za knjiženje, koji omogućuju pravilno usmeravanje iznosa na odgovarajuće račune.
- Evidentiranje promena u glavnoj knjizi, treba da omogući sistematsko obuhvatanje stanja i promena na imovini obavezama, kapitalu prihodima i rashodima i to je podloga za sastavljanje finansijskih izveštaja.
- Izrada periodičnih godišnjih obračuna treba da omogući celokupni pregled poslovanja tokom perioda.
- Kontrola i analiza finansijskog poslovanja treba da oceni i prati adekvatnost i efikasnost računovodstvenog sistema u skladu sa zakonom.

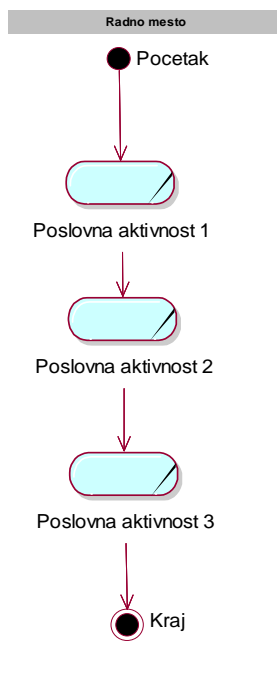
1.2.2 Dijagram poslovnih aktivnosti

Na osnovu prikupljenih zahteva i definisanih poslovnih slučajeva upotrebe, definiše se dijagram poslovnih aktivnosti koji treba da bude osnova za izradu modela sistemskih slučajeva upotrebe za izabrane poslovne aktivnosti iz dijagrama poslovnih aktivnosti.

U okviru dijagrama poslovnih aktivnosti izdvajaju se aktivnosti koje će u sledećoj fazi objektno orijentisane analize biti opisane kroz systemske slučajeve upotrebe i systemske dijagrame aktivnosti. ^[46]

Na slici 1.10 definisan je dijagram poslovnih aktivnosti koji se definiše za samo jednog poslovnog učesnika sa naredbodvnom ulogom. Međutim nije baš uvek tako. Poslove aktivnosti često "prelaze iz ruke u ruku" i menjaju poslovnog učesnika koji nad njima radi.

Slika 1.10 Grafički prikaz izrade dijagrama poslovnih aktivnosti poslovnog slučaja upotrebe



Izvor: Miroljub Zahorjanski

Inače dijagram poslovnih aktivnosti se koristi za objektno orijentisanu specifikaciju informacionih sistema, za opis aktivnosti vezanih za poslovni proces, tj. poslovni slučaj upotrebe. Dijagram poslovnih aktivnosti prikazuje sekvencijalni tok aktivnosti, a sastoji se od: stanja, akcija i prelaza. Dijagram poslovnih aktivnosti jednog posla predstavlja opis odvijanja aktivnosti u poslovnom slučaju upotrebe.

U krajnjem ishodištu, dijagram poslovnih aktivnosti predstavlja pandam dekompozicionom dijagramu u *BPwin* alatu. Kao što se tamo (Slika 1.6.) definiše tok aktivnosti kroz dekomponovanje na podaktivnosti, funkcionalnost putem internih strelica i odvijanje podaktivnosti pod uticajem različitih mehanizama, tako se u ovde za poslovnu aktivnost definiše dijagram sa plivačkim stazama (*Swimlanes*).

Na slici 1.11. je prikazan dijagram poslovne aktivnosti za glavnu aktivnost EKONOMSKO FINANSIJSKI POSLOVI. Da bi se ova aktivnost započela neophodno je na početku izraditi knjigovodstvene

standarde i pravila u skladu sa zakonskom regulativom i kasnije tokom upotrebe održavati iste. Ovu aktivnost u celosti obavlja šef računovodstva. Nakon ispunjenosti ovih pretpostavki kontista, kao sledeći u nizu poslovnih učesnika, u mogućnosti je da primenom pravila i standarda izvrši kontiranje dokumenta i izradi nalog za knjiženje.

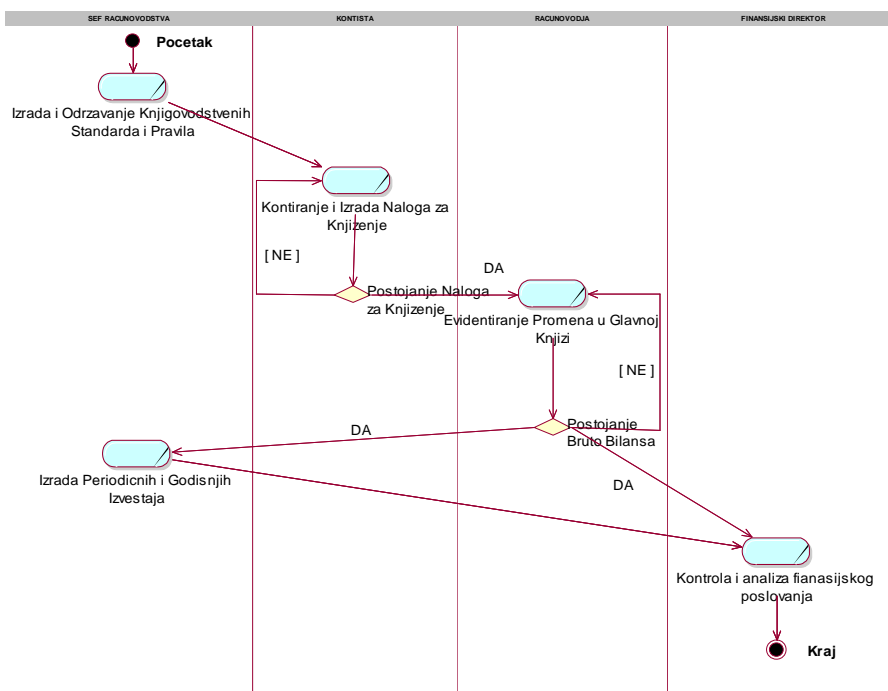
Između druge i treće podaktivnosti na ovom dijagramu postoji odluka (*Decision*) "Postojanje naloga za knjiženje". Ukoliko uz dokument postoji nalog za knjiženje, prelazi se na evidentiranje promena u glavnoj knjizi i tu podaktivnost obavlja računovođa, kao sledeći poslovni učesnik u nizu. Ukoliko je situacija suprotna, dokument se vraća na prethodnu aktivnost radi upotpunjavanja.

Nakon završene podaktivnosti "Evidentiranje promena u glavnoj knjizi" u dijagramu, a i u realnosti postoji još jedna odluka na putu ka drugim dvema podaktivnostima: "Izrada periodičnih i godišnjih izveštaja" i "Kontrola i analiza finasijskog poslovanja". Ova odluka se tiče postojanja bruto bilansa. Ukoliko je odgovor afirmativan, uporedo se mogu raditi obe zadnje pomenute podaktivnosti. Ukoliko je odgovor negativan, vraćamo se na izradu bruto bilansa. Kao što strelica na dijagramu pokazuje izrađen "Periodični i godišnji izveštaj" garantuje mogućnost "Kontrole i analize finasijskog poslovanja".

Ovim se završava tok u dijagramu glavne poslovne aktivnosti i omogućuje ciklično ponavljanje radnji.

Timovi za poslovnu analizu, razvoj softvera i razvoj baze podataka mogu na ovaj način, koristeći *UML* dijagram poslovnih aktivnosti da rade usaglašeno i u potpunosti razumeju jedni druge.^[70,71] A pošto je razvoj softvera iterativni proces u kome učestvuju više timova sa različitim nivoima iskustva i često u različitim fazama životnog ciklusa projekta, rad u ovakvom okruženju postaje olakšan.^[16]

Slika 1.11. Dijagram glavne poslovne aktivnosti EKONOMSKO FINANSIJSKI POSLOVI



Izvor: Miroljub Zahorjanski

Dijagrami aktivnosti su osnovni prikaz načina na koji se odvijaju poslovi. Međutim oni ne sadrže tehničke detalje tako da svi učesnici u izradi sistema, od poslovnih korisnika do članova tehničkog osoblja, mogu da razumeju predstavljene informacije. Dijagrami aktivnosti imaju višestruku ulogu:

- služe razumevanju poslovnog sistema onakvog kakav je trenutno
- identifikuju oblasti poslovnog sistema koje treba promeniti
- otkrivaju redundansu u poslovnim procesima
- identifikuju aktivnosti koje bi interno ili eksterno trebalo bolje izvesti
- utvrđuju informacione potrebe pojedinih aktivnosti ili poslovnih korisničkih funkcija.

Dok dijagrami poslovnih slučajeva upotrebe i dijagrami poslovnih aktivnosti, predstavljaju poslovni proces onako kako ga vidi poslovni učesnik sa naredbodavnom ulogom, dotle sistemski slučajevi upotrebe predstavljaju poslovni proces onako kako ih vide korisnici, odnosno učesnici koji nemaju naredbodavnu ulogu.

Ako bi smo onaj prvi pristup okarakterisali kao pristup odozgo na dole (*Top down*), onda bi ovaj drugi bio odozdo na gore ili (*Bottom up*).

2. OBJEKTNO ORIJENTISANA ANALIZA

Proces Objektno orijentisane analize (u daljem tekstu OO analiza) definiše objekte sadržane u realnom sistemu i njihovu međusobnu saradnju. Ono što OO analiza stavlja u prvi plan je istraživanje problema, odnosno nalaženje i opisivanje objekata (konceptata) u domenu problema, pri tome ne dajući odgovore na pitanje kako su definisana rešenja. OO analiza predstavlja najkritičniju fazu, jer je potrebno uočiti koji se objekti pojavljuju u realnom sistemu i specificirati attribute i interakciju između objekata.

Osnovu za OO analizu predstavlja aktivnost pod nazivom "KONTROLA I ANALIZA FINANSIJSKOG POSLOVANJA" definisana u okviru dijagrama aktivnosti koji opisuju poslovne slučajeve upotrebe (poglavlje definisanje zahteva) što predstavlja osnovu za:

- izradu modela sistemskih slučajeva upotrebe
- izradu konceptualnog modela i
- izradu dijagrama interakcije

Prilikom modeliranja realnog sistema za sistemske slučajeve upotrebe, objekti i njihove veze se predstavljaju određenim brojem konceptata koji služe za formiranje jasnog i potpunog modela realnog sistema kao uslov za implementaciju na računaru CASE alat (*RationalRose*).^[82] Pri tome se može zaključiti da je proizvod procesa OO analize konceptualni model sistema koji je kroz svoju nadgradnju vezanu za izradu dijagrama interakcije, osnova za objektno orijentisan dizajn.

2.1 Izrada modela sistemskih slučajeva upotrebe

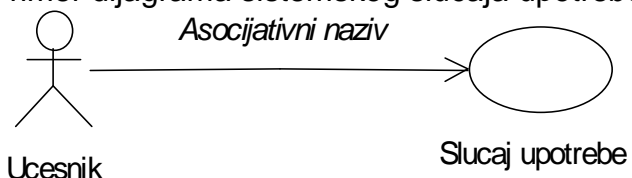
Model sistemskih slučajeva upotrebe predstavlja:

- atomsku transakciju, jer po njegovom završetku informacioni sistem ostaje u konzistentnom stacionarnom stanju, i
- logičku jedinicu posla u realnom vremenu, nešto što ima značenje za korisnika, bez obzira na njegovu složenost.

Dijagrami slučajeva upotrebe su dijagrami koji po *UML* standardu sadrže:

-
- objekte (učesnici i slučajevi upotrebe), i
 - veze.

Slika 2.1. Primer dijagrama sistemskog slučaja upotrebe



Izvor: Miroljub Zahorjanski

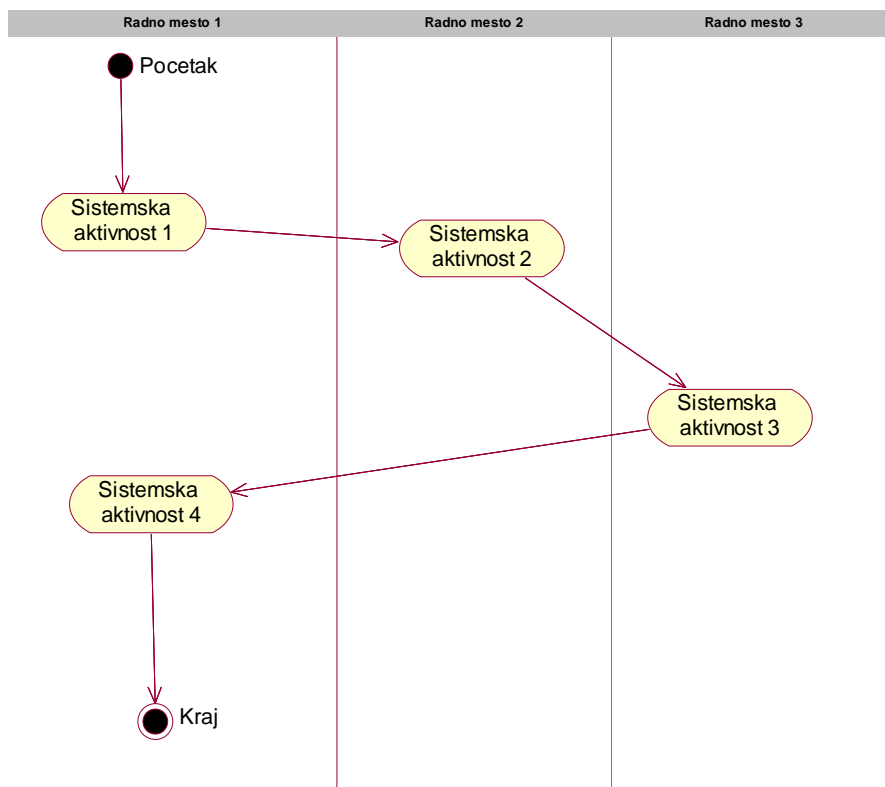
Između učesnika i slučaja upotrebe može postojati relacija asocijacije. Ovaj tip asocijacije često se naziva komunikacionom asocijacijom, jer predstavlja komunikaciju između učesnika i slučaja upotrebe, odnosno izvođača i korisničke funkcije. Asocijacija može biti prohodna u oba smeru: učesnik ka slučaju upotrebe i slučaj upotrebe ka učesniku ili može biti prohodna samo u jednom smeru. Smer u kojem je asocijacija prohodna ukazuje na to ko inicira komunikaciju.

Mogu postojati dva tipa relacija između slučajeva upotrebe: *include* (uključiti) i *extend* (proširi). Relacije *include* se formiraju kada novi slučaj upotrebe "koristi" funkcionalnost već postojećeg slučaja upotrebe. Relacija *extend* se koristi da prikaže neko opciono ponašanje ili različite tokove koji mogu biti pokrenuti na osnovu izbora učesnika.

Dijagram sistemskih aktivnosti sadrži:

- plivačke staze (*Swimlanes*) koje specificiraju odgovornosti za delove celokupne aktivnosti i nemaju neku duboku semantiku,
- stanja dijagrama aktivnosti predstavlja poslovnu aktivnost, i
- tranzicije koje predstavljaju prelazak iz faze u fazu koja je prouzrokovana nekim događajem.

Slika 2.2 Primer dijagrama sistemskih aktivnosti



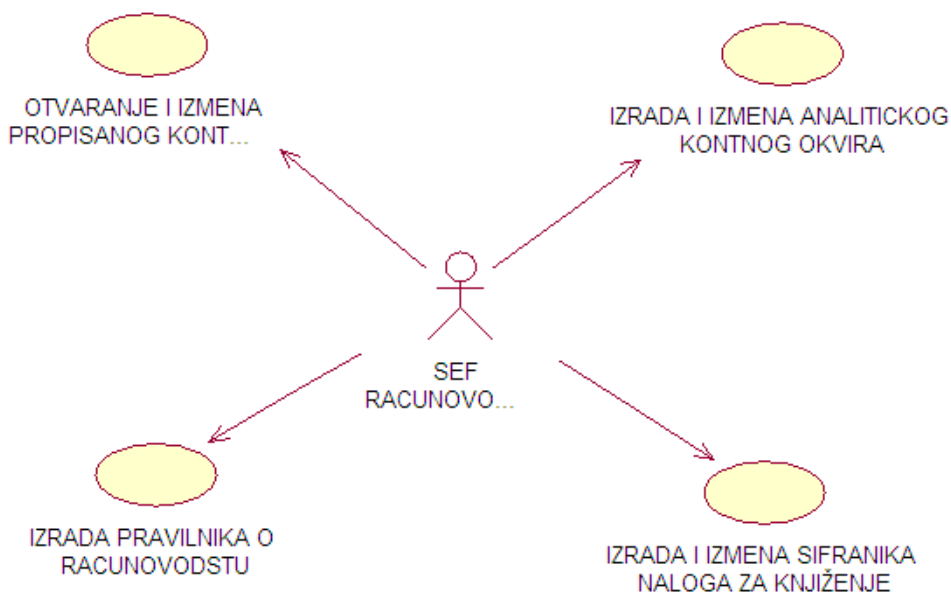
Izvor: Miroljub Zahorjanski

U dijagramu sistemskih aktivnosti se predstavljaju samo aktivna stanja (aktivnosti) i samo završeci aktivnosti kao događaji koji diktiraju redosled aktivnosti. ^[120] Sve druge vrste događaja posmatraju se kao izuzeci koji mogu da promene normalan redosled aktivnosti, u bilo kojem trenutku, odnosno stanju posla.

2.1.1 Sistemski slučajevi upotrebe

Nakon izrađenog modela poslovnog slučaja upotrebe u kome je specificiran i dijagram aktivnosti koji opisuje tok glavne poslovne aktivnosti, sačinjeni su modeli sistemskih slučajeva upotrebe. Sistemski slučajevi upotrebe omogućavaju grubi opis dinamike aktivnosti budućeg rada učesnika.

Slika 2.3. Dijagram sistemskog slučaja upotrebe za Izradu i održavanje knjigovodstvenih standarda i pravila.

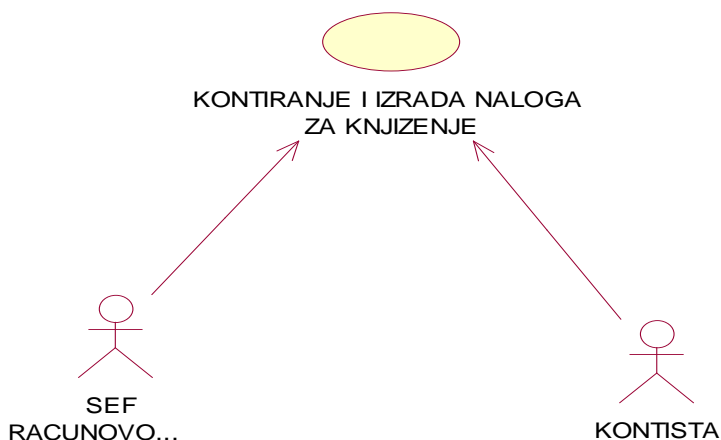


Izvor: Miroljub Zahorjanski

Na slici 2.3. prikazan je opis poslova šefa računovodstva iz perspektive samog korisnika, a vezano za podaktivnost “Izrada i održavanje knjigovodstvenih standarda i pravila”. Ovaj nivo bi se mogao uporediti sa dekompozicionim dijagramom drugog nivoa u *BPwin* alatu. Obzirom da taj nivo dekompozicije zbog skućenog prostora ovog rada nije prikazan, moguće je usporedbu načiniti koristeći sliku 1.5. Stablo aktivnosti za polazni posao.

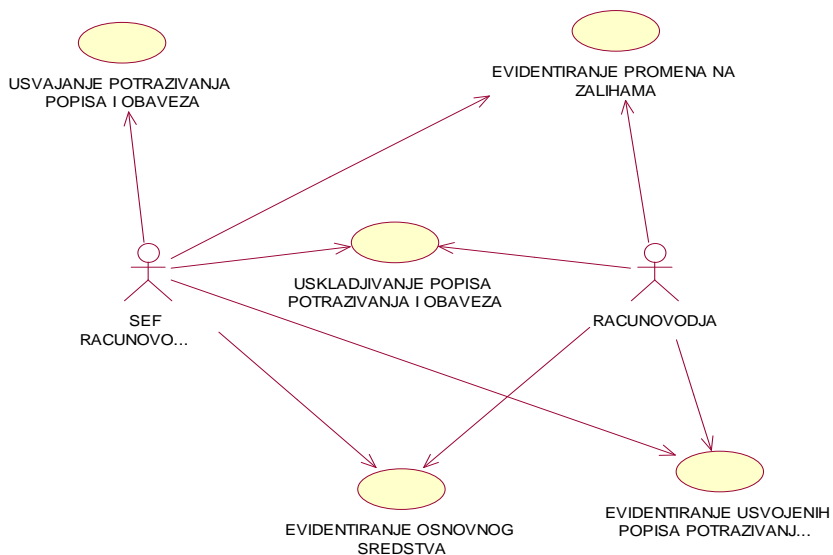
Na slici 2.4. prikazan je Dijagram sistemskog slučaja upotrebe za Kontiranje i izradu naloga za knjiženje. Ovde se vidi da na ovu aktivnost sa aspekta kako korisnik vidi poslovni proces postoje dva učesnika: kontista i šef računovodstva, gde ovaj drugi ima kontrolnu ulogu u obezbeđivanju pravilnog usmeravanja iznosa na odgovarajuće račune.

Slika 2.4. Dijagram sistemskog slučaja upotrebe za Kontiranje i izradu naloga za knjiženje.



Izvor: Miroљjub Zahorjanski

Slika 2.5. Dijagram sistemskog slučaja upotrebe za Evidentiranje promena u glavnoj knjizi.

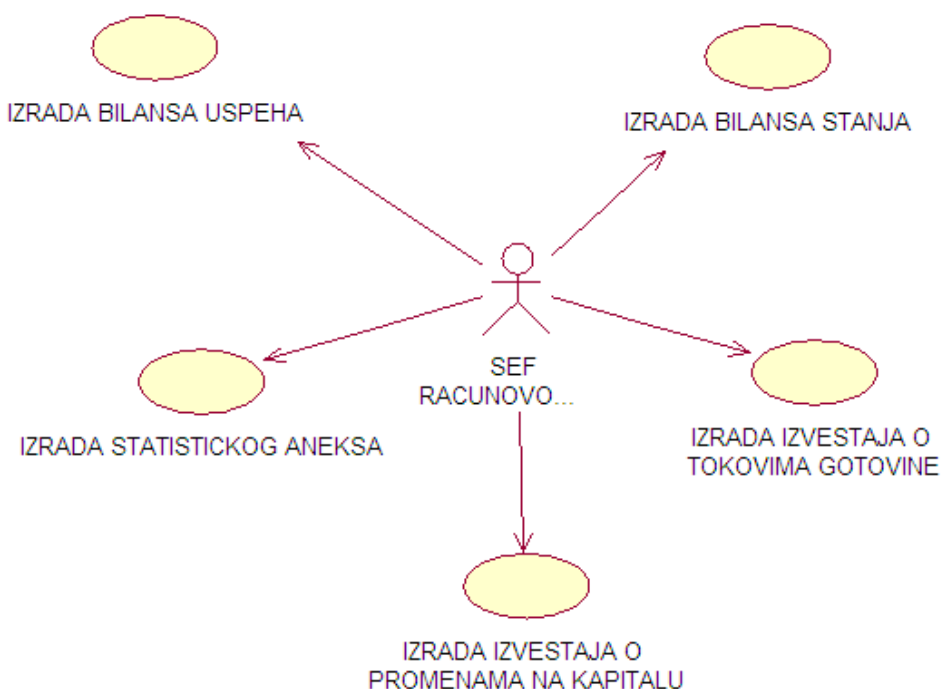


Izvor: Miroљjub Zahorjanski

Na slici 2.5. vidi se dijagram sistemskog slučaja upotrebe za "Evidentiranje promena u glavnoj knjizi". Ovde je tok sledeći: prvo oba

učesnika rade na “Usklađivanju popisa potraživanja i obaveza”, a potom šef računovodstva donosi odluku o “Usvajanju popisa potraživanja i obaveza”. Nakon toga se redom pod aktivnošću oba učesnika vrše evidentiranje promena na zalihama, osnovnih sredstava i konačno popisa, potraživanja i obaveza.

Slika 2.6. Dijagram sistemskog slučaja upotrebe za Izradu periodičnih i godišnjih obračuna.



Izvor: Miroljub Zahorjanski

Izrada Bilansa stanja treba da obezbedi informacije o stalnoj i obrtnoj imovini iz domena aktive, kao i o kapitalu, dugoročnim i kratkoročnim obavezama iz domena pasive.

Izrada Bilansa uspeha ima za cilj da obezbedi iz poslovnih prihoda i poslovnih rashoda pre svega poslovnu dobit ili poslovni gubitak. Potom na osnovu još nekih informacija među kojima su : vanredne stavke, dobit ili gubitak pre oporezivanja i porez na dobit, da obezbedi informaciju o neto dobitku ili neto gubitku.

Izrada Izveštaja o tokovima gotovine treba da pruži informacije o tokovima gotovine iz poslovnih aktivnosti, investiranja i finansiranja.

Izrada Izveštaja o promenama na kapitalu ima za cilj da obezbedi informacije o osnovnom kapitalu, kao i o akcijama, emisionim premijama, revalorizacionim rezervama i ostalom kapitalu.

Izrada Statističkog aneksa treba da obezbedi pre svega neke opšte podatke o pravnom licu odnosno preduzetniku, potom bruto promene nematerjalnih ulaganja, nekretnina, postrojenja i opreme. Nakon toga od izuzetnog su značaja informacije o strukturi zaliha, strukturi osnovnog i akcijskog kapitala, informacije o potraživanjima i dugovanjima, kao i drugim troškovima i rashodima.

Slika 2.7. Dijagram sistemskog slučaja upotrebe za Kontrolu i analizu finansijskog poslovanja.



Izvor: Miroљjub Zahorjanski

Zadnje prikazani dijagram sistemskog slučaja upotrebe na slici 2.7. sugeriše način na koji će se finansijski direktor nalaziti u interakciji sa budućim softverskim rešenjem, tj. moraju se predvideti događaji koje će sistemski učesnik u liku finansijskog direktora generisati.

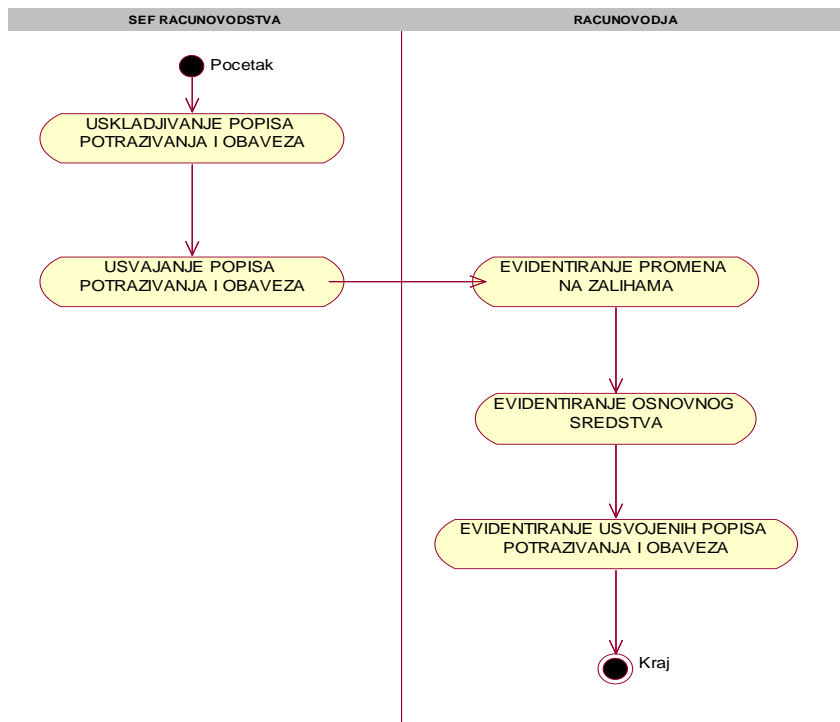
Sistemski učesnik koji izvodi analize u preduzeću je finansijski direktor. Identifikovan je kao korisnik funkcionalnosti sistema, koji će imati podršku sistema u obavljanju dnevnih zadataka i zainteresovan je za rezultate koje sistem proizvodi.

Sistemski učesnik, finansijski direktor komunicira sa sistemskim slučajem upotrebe kao što je pokazano na prethodnoj slici.

Od sad, pa nadalje u tekstu, bavićemo se isključivo funkcijom Kontrola i analiza finansijskog poslovanja i sve će bit podređeno tome.

Slika 2.7a predstavlja dijagram sistemskih aktivnosti za EVIDENTIRANJE PROMENA U GLAVNOJ KNJIZI i pokazuje sekvencijalni tok poslovnih aktivnosti, a sastoji se od stanja akcija i prelaza.

Slika 2.7a. Dijagram sistemskih aktivnosti za EVIDENTIRANJE PROMENA U GLAVNOJ KNJIZI



Izvor: Miroljub Zahorjanski

2.2 Izrada konceptualnog modela

Konceptom se opisuju stvari u realnom sistemu. Na osnovu tako opisanih stvari, kasnije se u fazi objektno orijentisanog dizajna, definišu odgovarajuće klase, a potom i objekti koji definišu odgovarajuća softverska rešenja.

Konceptima se prave apstrakcije, koje su deo problema koji se rešava. Svaka od ovih apstrakcija je ujedno deo rečnika sistema, tako da one skupa predstavljaju stvari koje su važne kako za korisnike tako i za projektante sistema.

Pošto se apstrakcijama označavaju stvari, korisnicima nije teško da identifikuju te iste apstrakcije koje koriste za opis svog sistema. Za one koji rešavaju problem, ove apstrakcije su samo stvari u tehnologiji koje su deo rešenja.

Kako je koncept opis stvari u realnom sistemu, iskaz se odnosi na atribute implementirane u kontekstu entiteta realnog sistema i odgovarajuće operacije.

U okviru aktivnosti definisanja koncepata treba, na osnovu prethodno definisanih zahteva i spoznaje sistema odrediti objekte (koncepte) koji se javljaju u sistemu i svaki od njih opisati određenim atributima.

Atribut je logička vrednost podataka za objekat. Atributi se mogu definisati i u okviru slučajeva upotrebe, sugerišući ili dajući nagoveštaj o potrebi za informacijom. Attribute na konceptualnom nivou treba predstaviti jednostavnim tipovima vrednosti (*Image*, *Date/Time*, *Number*, *Text*), a ne treba predstavljati prenesene ključeve.

Veze između koncepata predstavljaju asocijacije između prethodno definisanih koncepata. U okviru *UML*-a asocijacije se opisuju kao "strukturne relacije" između objekata za različite tipove.

Pomoću apstrakcija će se utvrditi da veoma mali broj koncepata obitava sam.^[15, 28,34] Veći broj njih saraduje sa drugima na različite načine. Zbog toga, kada se sačinjava konceptualni model, ne samo da se moraju identifikovati stvari koje formiraju rečnik modela, već se takođe mora definisati i kako te stvari stoje jedna u odnosu na drugu.

Koriste se sledeći tipovi veza: asocijacija, agregacija i generalizacija. Asocijacija je strukturalni odnos koji predstavlja grupu veza slične strukture i jedinstvene semantike. Ovaj tip veze odgovara neidentifikujućoj vezi u *IDEF1X* metodologiji.

Slika 2.8. Tip veze asocijacija



Izvor: Miroljub Zahorjanski

Agregacija je specijalni oblik asocijacije i jači je oblik veze, gde se uspostavlja veza između celine i dela (celina je sastavljena od delova). Agregacija se prikazuje kao linija koja spaja klase sa malim rombom uz klasu koja se odnosi na celinu (*Parent*) i pandan je identifikujućoj vezi u *IDEF1X* metodologiji:

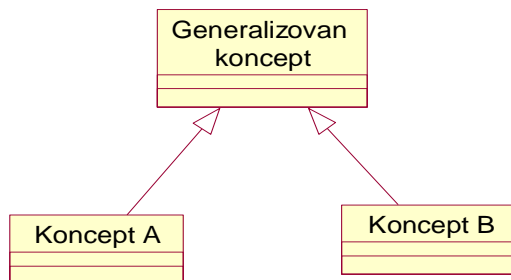
Slika 2.9. Tip veze agregacija



Izvor: Miroljub Zahorjanski

Generalizacija predstavlja hijerarhijsku vezu između koncepata:

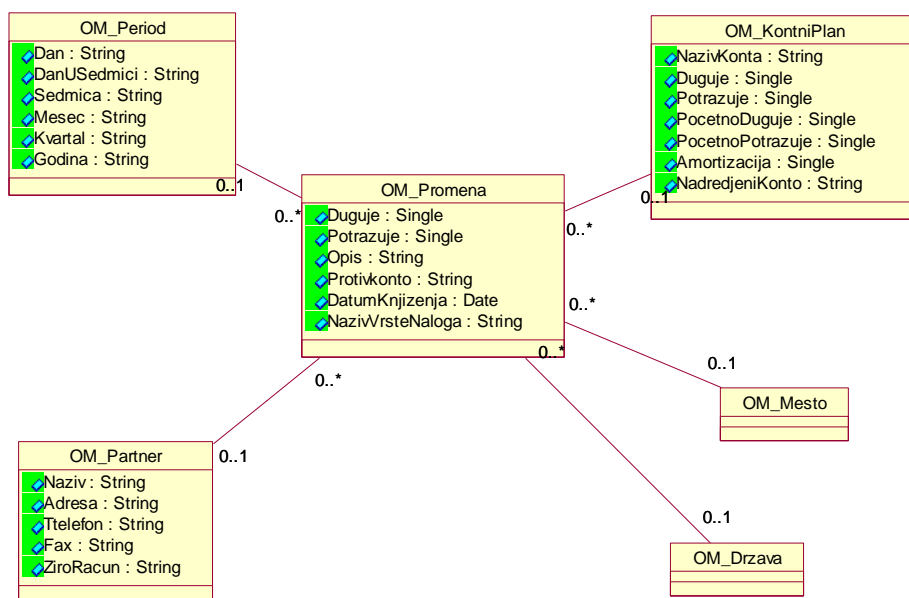
Slika 2.10 Tip veze generalizacija



Izvor: Miroljub Zahorjanski

Konceptualni model definiše se za sistemske slučajeve upotrebe i predstavlja osnovu za izradu dijagrama interakcije, pripremu podataka za analitičko procesiranje informacija i dimenziono modeliranje gde će pretrpeti korekcije i dopune.

Slika 2.11. Konceptualni model Kontrole i analize finansijskog poslovanja



Izvor: Miroljub Zahorjanski

Na slici 2.11. prikazan je konceptualni model Kontrole i analize finansijskog poslovanja. Ovaj konceptualni model postavljen je za jedan jedini sistemski slučaj upotrebe i iz njega proističu mogućnosti za bavljenje raznim vrstama analiza.

Od svih mogućih analiza ovde ćemo spomenuti samo neke:

- Analiza prihoda od prodaje robe
- Analiza prihoda od prodaje usluga
- Analiza rashoda po pitanju nabavke robe
- Analiza rashoda za plate radnika
- Analiza rashoda za održavanje opreme
- Analiza potreživanja od kupaca
- Analiza dugovanja poveriocima

Za gore pomenute i još mnoge druge analize koristiće se jedan jedini konceptualni model podataka.

Budća tabela činjenica^[105,112,114] biće centralna tabela u dimenzio-nalnom modelu. Ova tabela po pravilu sadrži detaljne ili

sumirane poslovne transakcije: novčani iznos prodane robe, priliv-odliv sredstava na žiro-računi i slično. Ovo je obično tabela sa najvećim brojem redova i okružena je drugim tabelama (dimenzijama) koje dodatno opisuju poslovne transakcije o kojima je reč. Tabela činjenica, zapravo, na sažet način čuva sve podatke o segmentu poslovanja preduzeća koje višedimenzioni model opisuje.

Vrednosti su osnovne veličine koje su od interesa za analizu određenog segmenta poslovanja. Na primer: suma duguje po određenom računu za određenog kupca, suma potražuje po određenom računu za određenog dobavljača i slično. Ovo su po pravilu aditivne veličine (veličine koje se slobodno mogu sabirati ili kojima se na drugi način može manipulirati na različitim nivoima agregacije po dimenzijama modela) ili poluaditivne veličine (veličine koje se agregiraju na specifičan način, poput stanja na žiro-računu, itd.) Vrednosti mogu biti izvorne vrednosti preuzete iz poslovnih transakcija ili pogodno sumirane vrednosti (na primer: saldo duguje za određenog kupca u toku dana pod pretpostavkom da je bilo više transakcija).

Koncepti kojima se definišu činjenice su:

- Duguje, (koncept koji u sebi sadrži meru o vrednosti iznosa na dugovnoj strani stavke)
- Potražuje, (koncept koji u sebi sadrži meru o vrednosti iznosa na potražnoj strani stavke)
- Opis, (koncept koji nosi informaciju o broju dokumenta na koji se odnosi radi zatvaranja stavki)
- Protivkonto, (koncept koji nosi informaciju o tome sa kojim kontom je knjižena stavka povezana)
- DatumKnjiženja, (Koncept koji nosi informaciju o datumu izvršenog knjiženja)
- NazivVrsteNaloga, (koncept koji u sebi sadrži informaciju o vrsti naloga)

U logičkom smislu dimenzije predstavljaju ose višedimenzionog modela. Na primer: datum novčane transakcije, konto ili račun, partner i slično. Tabele dimenzija okružuju tabelu činjenica (zato se često model naziva i zvezdasta šema (*star schema*)). Ove tabele detaljno opisuju poslovne transakcije koje sadrži tabela činjenica. Po pravilu dimenzije su hijerarhijski strukturisane. Na primer: datum fakturisanja može da ima sledeću strukturu: godina, kvartal, mesec, dan, itd. Pored podataka koji detaljno opisuju hijerarhijsku organizaciju dimenzije ^[18], tabela dimenzija može da sadrži i dodatne attribute, kao što je dan u sedmici.

Koncepti kojima se definišu dimenzije su:

- NazivKonta, (koji sadrži puni naziv računa)
- NadređeniKonto (koji sadrži hijerarhijski gledano roditeljski račun)
- Naziv, (koji u sebi sadrži naziv partnera)
- Adresa, (koji u sebi sadrži adresu partnera)
- Telefon, (koji u sebi sadrži tefon partnera)
- ZiroRacun, (koji u sebi sadrži žiro račun partnera)
- Dan, (koji u sebi sadrži tekstualnu informaciju o danu u nedelji)
- DanUSedmici,
- Sedmica, (koji u sebi sadrži red i broj sedmice u godini)
- Mesec, (koji u sebi nosi informaciju o rednom broju mesecu u godini)
- Kvartal, (koji sarži informaciju o rednom broju kvartala u godini)
- Godina, (koji nosi informaciju o kalendarskoj godini)

2.3 Izrada dijagrama interakcije

Dijagrami interakcije vrše nadgradnju koncepata za sistemske slučajeve upotrebe. Izvode se izradom dijagrama sekvenci i dijagrama saradnje.

Nakon definisanja sistemskih dijagrama slučajeva upotrebe i konceptualnog modela, potrebno je sačiniti dijagram sekvenci, koji predstavlja realizaciju sistemskih slučajeva upotrebe i pokazuje redosled događaja koje generišu spoljni učesnici za svaki slučaj upotrebe. Ono što sistemski slučaj upotrebe treba da prdstavi, jeste način na koji se učesnik nalazi u interakciji sa softverskim sistemom. Odnosno, učesnik generiše buduće događaje. Osnovna podela dinamike u dijagramu sekvenci definisana je sledećim elementima:

- Događaj, tj. spoljašnji ulazni događaj koji generiše učesnik u sistemu.
- Operacija, koja predstavlja odziv na događaj u sistemu.

Dijagram sekvenci je prikaz dinamičke saradnje između objekata u vremenu. To je na izvestan način grafička ilustracija dinamičke interakcije gde objekti komuniciraju preko sekvenci poruka. Uz to, dijagram sekvenci se koristi za prikaz korisničkog interfejsa, jer

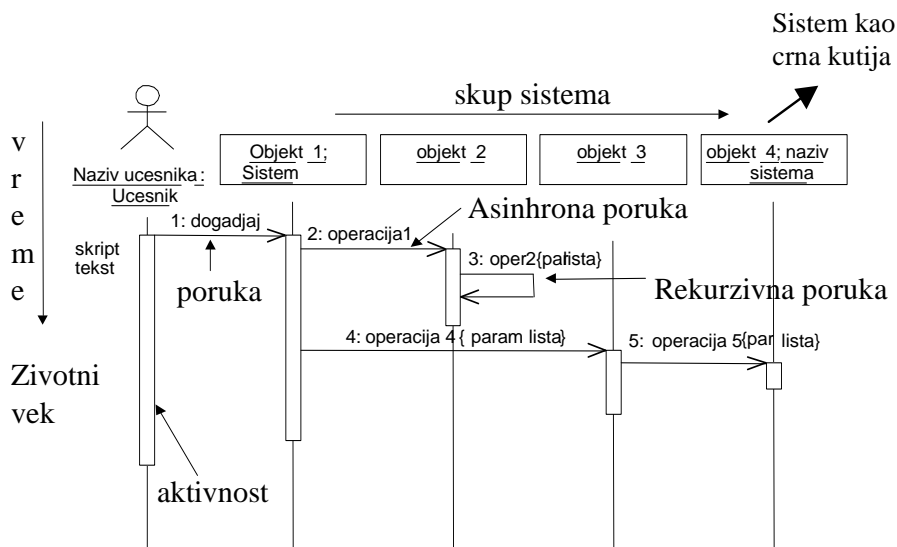
se definiše sekvenca događaja, koje korisnik (interfejs ili okolina) prosleđuje sistemu u jednom slučaju upotrebe.

Dijagram sekvenci, kao što se vidi, pokazuje interakciju između objekata u vremenskoj sekvenci. Ovi dijagrami poseduju dve dimenzije: vreme i kolekciju objekata. Obično se vreme prikazuje po vertikalnoj, a kolekcija objekata po horizontalnoj dimenziji (slika 2.12). Na vertikalnoj liniji se može, na pogodan način, predstaviti i vremenska skala. Isto tako je uobičajeno da se na dijagramu sekvenci prikazuju i granice sistema. Ovi dijagrami predstavljaju, obično za jedan slučaj upotrebe, objekte koji su u interakciji i sekvencu događaja (poruka) koje objekti razmenjuju. Uz levu ivicu dijagrama sekvenci moguće je opisati sekvencu operacija nad prikazanim objektima, koristeći strukturni prirodni jezik ili neki formalniji način.

Na dijagramu sekvenci objekti su predstavljeni vertikalnim linijama. Na vrhu linije je naziv objekta i simbol objekta. Aktiviranje objekta se prikazuje uskim pravougaonikom na liniji objekta, a vizuelizuje operaciju (akciju) koju objekat obavlja u periodu predstavljenom dužinom aktivacije. Na vrhu aktivacionog objekta se prikazuje događaj koji je aktivirao objekt, a na dnu povratna poruka objektu koji je aktivirao posmatrani objekt. Povratna poruka se često ne prikazuje.

Na slici 2.12 pokazan je dijagram sekvenci koji se formira tako što se prvo na vrh dijagrama, duž ose X, postavljaju objekti koji učestvuju u interakciji. Objekti se postavljaju s leva na desno, dok se poruke koje ovi objekti šalju i primaju smeštaju duž Y ose, odozgo na dole. Ovo pruža jasnu asocijaciju na tok kontrole u proteku vremena.

Slika 2.12. Primer dijagrama sekvenci



Izvor: Stanojević Lj., Veljović A., Razvoj metodologije rojektovanja poslovne inteligencije, Monografija, Megatrend univerzitet primenjenih nauka, Beograd, 2008.

Dijagram sekvenci preuzima učesnike iz dijagrama slučajeva upotrebe, a iz dijagrama koncepta preuzima ose i objekte. U okviru dijagrama sekvenci učesnici mogu biti korisnik ili neki veštački entitet (softver ili sistem). Objekt je element koncepta koja otprema i prima poruke. Takođe je moguće da objekt otpremi poruku samom sebi. Objekti se definišu nazivom objekta i nazivom koncepta, koji su podvučeni i to na sledeći način: Naziv objekta: Naziv koncepta.

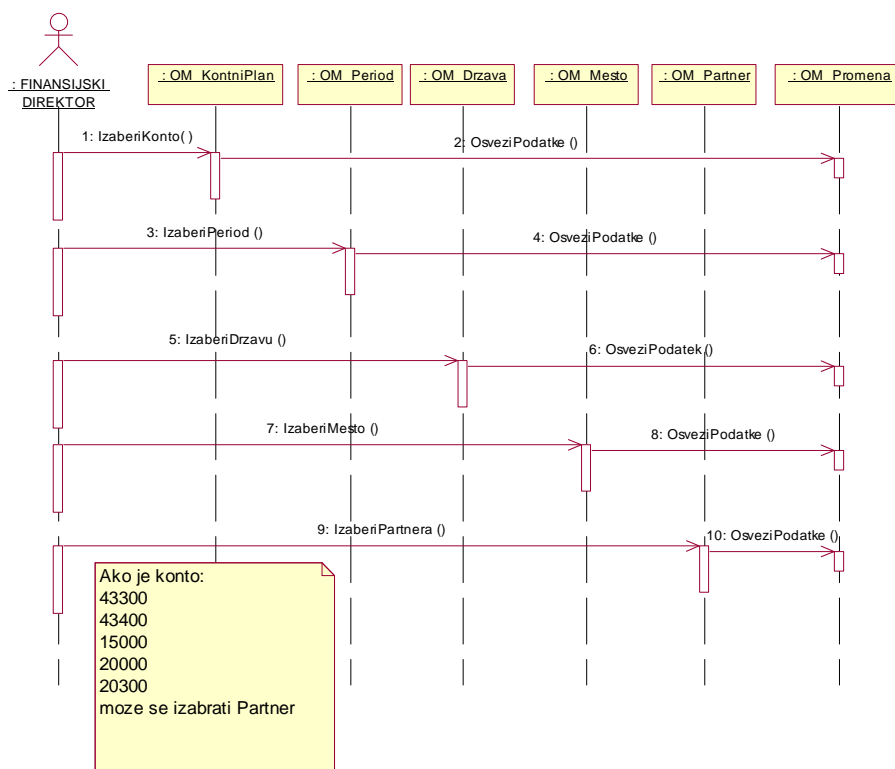
Dijagram sekvenci za funkciju “Kontrola i analiza finansijskog poslovanja” prikazan na slici 2.14. opisuje način na koji objekti : (OM KontniPlan, OM Period, OM Drzava, OM Mesto, OM Partner, OM Promena) u sistemu međusobno komuniciraju, ostvarujući na taj način očekivano ponašanje.

Ovi dijagrami se koriste za specifikaciju vremenskih zahteva i opis složenih scenarija. Poruke između objekata kojima se realizuje odgovarajuća akcija u sistemu koji se opisuje, jesu:

- tipa: *IzaberKonto()*, *IzaberiPeriod()*, *IzaberiDrzavu()*, *IzaberiMesto()*, *IzaberiPartnera()*, gde se bira objekat posmatranja.

- tipa *OsveziPodatke()*, što znači suziti izbor prema prethodno izabranom objektu, koji je prethodio osvežavanju podataka.

Slika 2.13. Dijagram sekvenci za funkciju Kontrola i analiza finansijskog poslovanja



Izvor: Miroljub Zahorjanski

Iz svakog objekta polazi nadole isprekidana linija koja predstavlja njegov životni vek (*lifeline*). Životni vek predstavlja postojanje nekog objekta u periodu vremena. Većina objekata koji se pojavljuju u dijagramu sekvenci postojeće dok traje interakcija, pa su svi ti objekti poređani na vrhu dijagrama, sa svojim životnim vekom povučenim od vrha ka dnu dijagrama. Objekti se mogu praviti u toku interakcije.

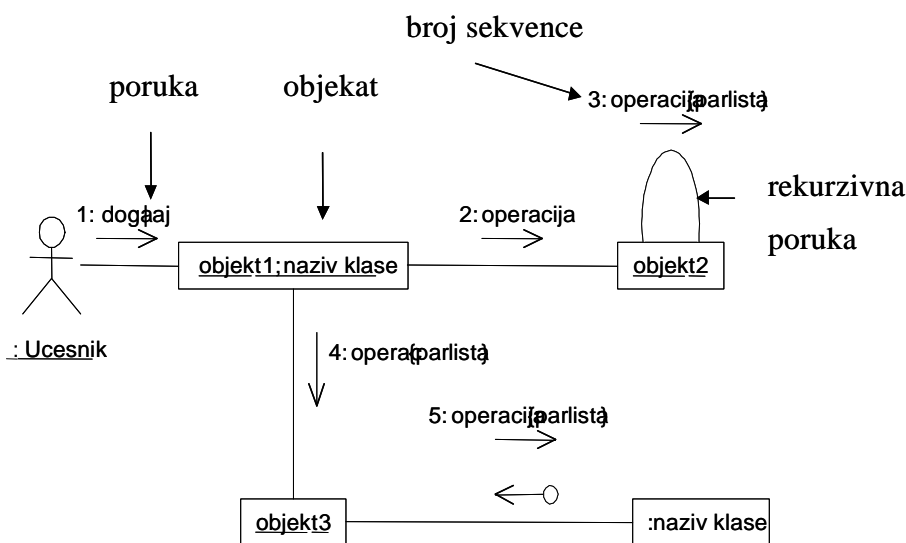
Dijagram sekvenci treba da pokaže redosled događaja koji generiše finansijski direktor za svaki sistemski slučaj upotrebe.

Dijagram saradnje ima sličnu strukturu kao dijagram sekvenci, samo što, pored objekata i poruka, prikazuje i veze između objekata.

Dijagram saradnje prikazuje interakciju između skupa objekata koji se predstavljaju kao čvorovi nekog grafa. Kako razvoj sistema napreduje i kako struktura koncepta prelazi u strukturu klasa, tako i značaj dijagrama saradnje raste, a dijagrama sekvenci opada, jer je semantika dijagrama saradnje bogatija.

Drugim rečima, pošto su izvedeni iz istih informacija u UML-ovom metamodelu, dijagram sekvenci i dijagram saradnje su međusobno ekvivalentni. Posledica toga je da se jedan može prevesti u drugi, bez bilo kakvog gubitka informacija. Na slici 2.14 prikazan je opšti slučaj dijagrama saradnje sa pripadajućom notacijom.

Slika 2.14. Primer dijagrama saradnje



Izvor: Miroljub Zahorjanski

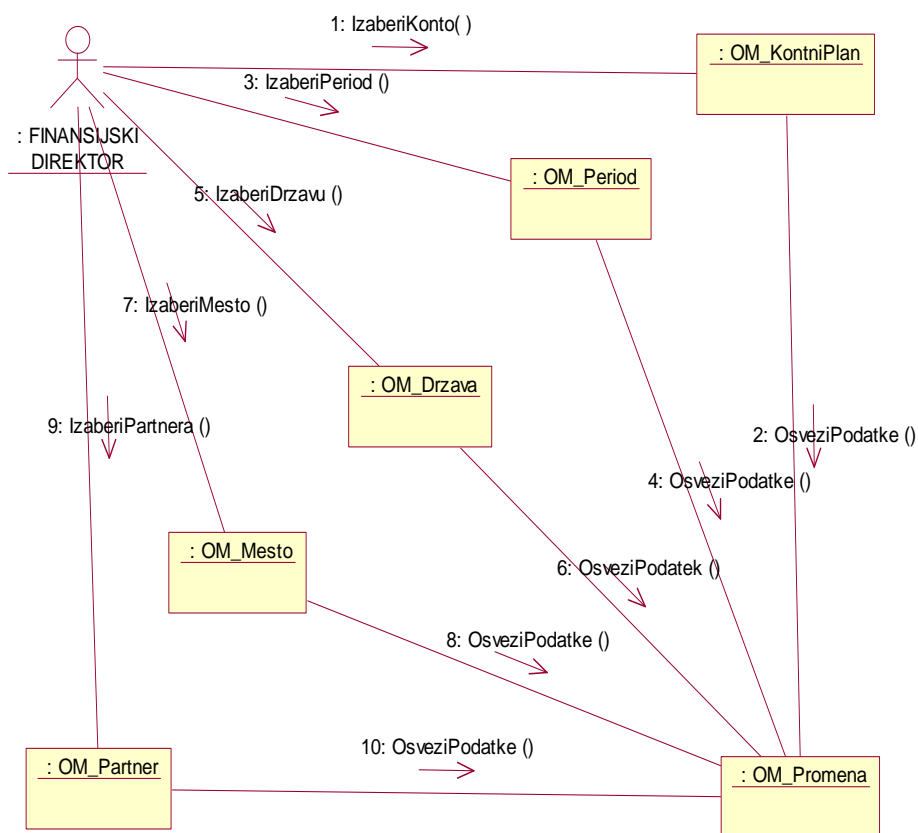
Na slici 2.15. prikazan je dijagram saradnje za funkciju Kontrola i analiza finansijskog poslovanja. Iz nje se zapaža da je radi izvršenja bilo koje analize spominjane u prethodnom tekstu, neophodno izabrati:

- konto (poruka 1: *IzaberiKonto()* za objekat :OM KontniPlan),
- period (poruka 3: *IzaberiPeriod()* za objekat :OM Period),

- državu (poruka 5: *IzaberiDrzavu()* za objekat *:OM Drzava*),
- mesto (poruka 7: *IzaberiMesto()* za objekat *:OM Mesto*),
- partner (poruka 9: *IzaberiPartnera()* za objekat *:OM Partner*)

koji će na osnovu redosleda koraka prikazanih na slici koja sledi izvršiti izbor operacije (2:,4:,6:,8:,10: *OsveziPodatke()* za objekat *:OM Promena*) užeg skupa podataka Promena.

Slika 2.15. Dijagram saradnje za funkciju Kontrola i analiza finansijskog poslovanja



Izvor: Miroљjub Zahorjanski

3. OBJEKTNO ORIJENTISAN DIZAJN

U fazi Objektno orijentisana analiza (OOA) koja je opisana na prethodnim stranicama bavili smo se istraživanjem, a ne toliko rešavanjem problema. Faza koja sledi, Objektno orijentisan dizajn (OOD), daće nam odgovor na pitanje kako da se izvrši proces fizičkog dekomponovanja sistema na manje softverske celine i blokove, a da se pri tom izvrši specifikacija statičkih i dinamičkih programskih celina. [72.]

Pre svega treba pažljivo i precizno da se definišu sva svojstva koja objekti poseduju i sve interakcije između tih objekata. Tokom objektno orijentisanog dizajna pojedini objekti će dobiti još neka svojstva, koja nisu uočena tokom Objektno orijentisane analize.

Na ovo proširivanje utiču koraci definisani u okviru objektno orijentisanog dizajna:

- priprema podataka za analitičko procesiranje informacija gde je potrebno izvesti ekstrakciju, čišćenje, i transformaciju podataka iz transakcione baze podataka,
- izrada dimenzionog modela, gde je potrebno izvesti definisanje hijerahije elemenata i atributa, denormalizaciju i kreiranje agregacija i
- definisanje konačnog dijagram klasa kao osnove za izradu odgovarajućih softverskih rešenja.

3.1 Priprema podataka za analitičko procesiranje informacija

Jedna od najbitnijih aktivnosti y procesu razvoja skladišta podataka je svakako priprema podataka. Razvoj skladišta podataka će nadalje zavisiti upravo od uspešnog okončanja ove aktivnosti. Priprema podataka se vrši na osnovu ranije određenog izvora podataka, potom pravila za preuzimanje tih podataka, same procedure pripreme i zahteva korisnika. Priprema se vrši određenim ekstrakciono-transformacionim alatima kroz^[130] sledeće korake:

- ekstrakcija podataka,
- čišćenje podataka, i
- transformacija podataka.

Rezultat ovih triju aktivnosti treba da budu podaci koji će nam omogućiti generisanje meta-podataka (podatci o podacima), na osnovu kojih se može pristupiti dizajniranju skladišta podataka.

Proces ekstrakcije, transformacije i učitavanja podataka (u daljem tekstu *ETL* kao akronima za *Extraction, Transaction, Loading*) je ključni korak u implimentaciji skladišta podataka, odnosno koncepta baze podataka za vršenje analize finansijskog poslovanja.^[79]

Svi podaci koji će se učitati u skladište podataka^[56,57,61,63,66,67,128,129] moraju se transformisati u odgovarajući oblik. Pri tom je izuzetno važno da svi podaci budu transformisani i integrisani u skladište podataka na postojan i održiv način. To znači da se kroz celo skladište podataka moraju zadržati permanentne konvencije imenovanja, iste takve merne jedinice promenljivih i strukture šifriranja.

3.1.1 Ekstrakcija podataka

Ekstrakcija podataka se izvodi iz *OLTP* baze podataka koja je pominjana u okviru prikupljanja zahteva. Za konkretni slučaj je vršeno inicijalno ekstrahovanje podataka u procesu kreiranja skladišta podataka. Kasnije se na osnovu određenih procedura vrši dodavanje novih podataka u skladište podataka. Cilj procesa ekstrakcije podataka je da sve potrebne podatke, u pogodnom i konzistentnom formatu, pripremi za učitavanje u skladište podataka. Ekstrakcija podataka je vrlo jednostavna operacija ako se potrebni podaci nalaze u jednoj relacionoj bazi^[21], ali može da bude i veoma kompleksna operacija ako su podaci smešteni u višestrukim heterogenim operacionim sistemima.

Pre procesa ekstrakcije trebalo bi proveriti da li u bazi podataka iz koje vršimo ekstrakciju nema logičkih grešaka (konkretno u tabeli *Partner* nema partnera koji se nalazi u tabeli *Promena*). Ovakve greške bi pre ekstrakcije trebalo ukloniti korišćenjem procedura za proveru grešaka.

Postoji mogućnost da se ne može utvrditi eventualno postojanje logičkih grešaka. To se dešava u situacijama kada se ekstrakcija vrši iz više izvora podataka. Prilikom ekstrakcije iz više izvora podataka može se javiti i problem nekonzistentnosti podataka usled različitog označavanja istih pojmova (nazivi država se mogu skraćeno označavati sa tri ili sa dva simbola).

3.1.2 Čišćenje podataka

Proces čišćenja podataka ima tri osnovne komponente: 1) proveru podataka, 2) poboljšanje podataka i 3) upravljanje greškama.

1) Provera podataka se sastoji od brojnih provera, kao na primer:

- ispravne vrednosti za attribute (provera domena),
- provera atributa u kontekstu vrednosti ostatka reda,
- provera atributa u kontekstu vrednosti povezanih redova u toj ili drugim tabelama, i
- provera veza između redova u toj i drugim tabelama (provera stranih ključeva).

2) Poboljšanje podataka je proces čišćenja proverenih podataka kako bi oni postali smisleniji. Najuoobičajeniji primer na konkretnom modelu je informacija o imenu i adresi poslovnih partnera. Često se ove dve informacije smeštaju na različita mesta. Tokom vremena one mogu postati nesinhronizovane. Spajanje podataka na zahtev korisnika je veoma teško, jer se podaci koji su se nekad poklapali više ne poklapaju. Proces poboljšanja podataka prepoznaje ovakve podatke.

3) Upravljanje greškama je proces koji utvrđuje šta da se radi sa podacima koji nisu baš savršeni. Oni mogu biti odbačeni, smešteni u prihvatnu oblast radi popravke ili propušteni u skladište podataka zajedno sa svojim nedostacima. Iz perspektive modela podataka, treba brinuti samo o podacima koji prođu u skladište podataka. Meta-podaci za loše podatke treba da sadrže stavke o kvalitetu podataka (tipu greške) i o tačnosti podataka (učestalosti grešaka). ^[104]

U konkretnom primeru nad tabelom *Partner* primenjen je upit (*Find duplicates for Partneri*) za pronalaženje i eliminisanje duplikata u toj tabeli:

```
SELECT Partner.SifraPartnera, Partner.Naziv, Partner.Mesto
FROM Partner
WHERE ((([Partneri].[Sifra]) In (SELECT [Sifra] FROM [Partneri] As Tmp
GROUP BY [Sifra]
HAVING Count(*)>1 ))) ORDER BY Partner.SifraPartnera;
```

Ova tabela koje je predviđena za dimenzionu tabelu buduće šeme zvezde sadrži preko 2000 zapisa.

Proverom tabele *Promena*, takođe je utvrđeno da postoje zapisi koji ne sadrže podatak o partneru. Takvi zapisi su izdvojeni posebno kreiranim upitom (*Promene Without Matching Partneri*) i dodat je nedostajući podatak.

```
SELECT Promena.SifraPartnera  
FROM Promena LEFT JOIN Partner ON Promena.SifraPartnera =  
Partner.SifraPartnera WHERE (((Partner.SifraPartnera) Is Null));
```

Upiti su pisani u dijalektu *ANSI SQL-92*.

Tabela *Promena*, koja je u budućoj šemi zvezde viđena kao tabela činjenica sadrži preko 30.000 zapisa.

3.1.3 Transformacija podataka

Transformacija podataka je kritičan korak u razvoju bilo kojeg skladišta podataka. To je proces koji ima za cilj da osigura razumevanje učitanih podataka i njihovu konzistentnost u skladištu podataka. U transakcionoj bazi podataka često se za pojmove i imena atributa koriste teško razumljiva imena (npr. *DatDok*) koja bi krajnjem korisniku bila teška za upotrebu. Takođe, različiti izvori mogu imati i različita imena za isti pojam (npr. *DatDokum* i *DatDok*). Tokom transformacije podataka, ti pojmovi i imena se transformišu u jednoznačne, standardne poslovne izraze koji su razumljivi krajnjem korisniku (*DatDok* je *DatumDokumenta*).

Različiti sistemi mogu imati za iste attribute različite tipove podataka i različite dužine (npr. jedan sistem za šifru artikla ima 12, a drugi 14 numeričkih znakova). Transformacijom se svi različiti tipovi podataka i dužine prevode u jednoznačan, zahtevani oblik.

Tipična transformacija podataka uključuje:

- prevođenje polja sa više imena u jedno polje,
- “razbijanje” polja sa datumom u posebna polja za godinu, mesec i dan,
- prevođenje polja sa jednom reprezentacijom u drugu (npr. sa 1 i 0 u DA i NE),
- kreiranje i dodavanje ključeva za tabele dimenzija.

Po završenoj transformaciji postoje svi uslovi da se pristupi generisanju meta baze podataka, tj. izradi konačnog dimenzionog modela na osnovu konceptualnog modela definisanog u okviru objektno orijentisane analize.^[60]

U nastavku je priložena procedura koja je poslužila za punjenje tabele *Period* kako bi se dobila nova, trenutno nepostojeća dimenzija.

```
Function Datumi(OdD As Date, DoD As Date) As String
    Dim Brojac As Integer
    Dim I As Integer
    Dim Baza As Database
    Dim rs As Recordset
    Set Baza = CurrentDb
    Set rs = Baza.OpenRecordset("tblDatumi")
    OdD = OdD - 1
    Brojac = DoD - OdD
    For I = 1 To Brojac
        Datumi = Format(OdD + I, "dd.mm.yyyy")
        rs.AddNew
        rs![Datum] = Datumi
        rs![Dan] = Format(Datumi, "dddd")
        rs![DanUSedmicij] = Format(Datumi, "w", vbMonday)
        rs![Sedmica] = Format(Datumi, "ww", vbMonday)
        rs![Mesec] = Format(Datumi, "mm")
        rs![Kvartal] = Format(Datumi, "q")
        rs![Godina] = Format(Datumi, "yyyy")
        rs.Update
    Next
    MsgBox "Tabela je napunjena", vbInformation, "Paznja"
    rs.Close
    DoCmd.Close acForm, "frmDatumi"
    Kraj:
    Set Baza = Nothing
End Function
```

Tabela *Period* je transformisanjem i generisanjem zapisa dostigla kapacitet od preko 2000 zapisa odnosno sadrži podatke za pet kalendarskih godina koje pokrivaju podatke tabele činjenica.

3.2 Izrada dimenzionog modela

Izrada dimenzionog modela izvodi se u nekoliko faza:

- definisanjem hijerarhije elemenata i atributa,
- denormalizacijom
- kreiranjem agregacija.

Definisanje hijerarhije elemenata i atributa vezano je za uspostavljanje hijerarhija, gde se jedan hijerarhijski nivo nastavlja sa drugim hijerarhijskim nivoom i to se definiše kao dimenzija.

Druga faza, denormalizacija je vezana za dimenzije i činjenice (ili mere) gde se definiše mali broj tabela i definisanih veza između njih i pritom je model dimenzija organizovan u šemu zvezde.

Samo kreiranje agregacija je definisanje sumarnih podataka u okviru table činjenica i time se poboljšavaju performanse upita.

Znači, izrada dimenzionog modela vezana je za definisanje:

- tabela činjenica i
- dimenzionih tabela.

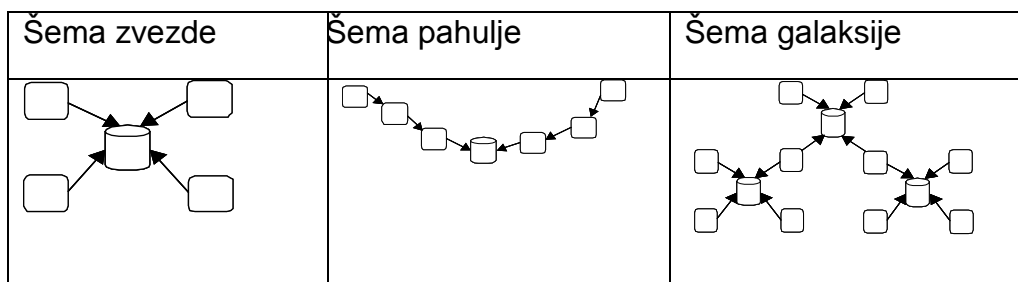
“Ove prve, table činjenica sadrže kvantitativne podatke o poslovima, tj. podatke koje korisnici analiziraju. Ovi podaci su najčešće numeričkog tipa i mogu se sastojati i od nekoliko miliona redova i kolona.

Tabela činjenica modeluje podatke koji se koriste za izradu izveštaja. Ta tabela sadrži dve vrste atributa: ključne attribute ili spoljne ključeve pomoću kojih je tabela činjenica povezana sa tabelama dimenzija i činjenice, odnosno attribute koji sadrže samo podatke koji se obrađuju.”^[84]

Druge pak, dimenzione table, znatno su manje i sadrže podatke koji opisuju dati posao, tj. one podatke po kojima se vrši analiziranje. Ti podaci se nazivaju atributi.

Prednost koju šeme zvezde^[84] ima je što omogućava definisanje složenih višedimenzionih podataka u vidu jednostavnog modela, potom smanjuje broj fizičkih veza koje se moraju procesirati pri zadavanju upita, čime se postiže poboljšanje performansi sistema i na kraju omogućava proširenje skladišta podataka uz relativno jednostavno održavanje. Najveća mana šeme zvezde je što se povećava redundantnost podataka.

Slika 3.0 Šema zvezde, pahulje i galaksije

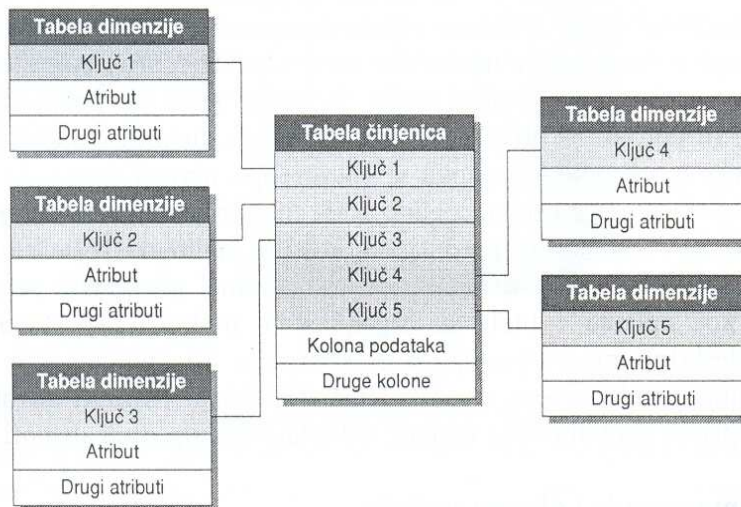


Izvor : Veljović A. Njeguš A., Osnove relacionih i analitičkih baza podataka, Megatrend univerzitet primenjenih nauka, Beograd, 2004.

Bitna karakteristika šeme zvezde je da su dimenzione tabele denormalizovane. Postupak denormalizacije je pristup gde se podaci u bazi podataka ponavljaju zbog pojednostavljenja dizajna i karakteristika. Sama denormalizacija je proces kombinovanja tabele da bi se poboljšale performanse sistema. Ovim postupkom se smanjuje broj potrebnih veza koje se moraju procesirati zadavanjem upita. Time se direktno utiče na poboljšanje performansi sistema. Rezultat je manji broj veza, što uzrokuje da sistem brže nalazi tražene podatke.

Iz ovog proizilazi da dimenzioni atributi mogu biti smešteni više puta u dimenzione tabele, u zavisnosti od toga koji nivo dimenzione hijerarhije se opisuje atributom. Svaka tabela mora sadržati primarni ključ koji predstavlja kolonu ili grupu kolona u tabeli čiji sadržaj jedinstveno identifikuje zapise. Na slici 3.1 dat je izgled jednostavne šeme zvezde.

Slika 3.1 Primer realizacije jednostavne šeme zvezde



Izvor: Miroljub Zahorjanski

3.2.1 Definisavanje hijerarhija elemenata i atributa

Tabela dimenzija može biti organizovana u hijerarhiji gde se hijerarhijski nivo nastavlja sa nekim drugim hijerarhijskim nivoom. Jedan dobar primer bi bila vremenska dimenzija. Unutar vremenske dimenzije, dani se nastavljaju na nedelje, nedelje na mesece, a meseci na kvartale.

Ovakvi dimenzioni elementi su specijalna kategorija podataka, koja predstavlja određeni nivo u dimenzionoj hijerarhiji. Za svaki hijerarhijski nivo postoji po jedan dimenzioni element.

Sa aspekta zahteva korisnika imamo dve vrste pogleda ili perspektive sa različitom detaljnošću. Postupak prelaska sa nivoa, sa manjim brojem detalja, na nivo sa većim brojem detalja naziva se spuštanje u dubinu (*drill down*) i predstavlja zahtev korisnika da mu se prikaže više detalja.

Druga perspektiva bi bila postupak prelaska sa nivoa sa većim brojem detalja na nivo sa manjim brojem detalja, na tzv. sumarne podatke i ona se naziva dizanje naviše (*drill up*). Dizanje naviše nasuprot spuštanju nadole, zahteva zbirni pogled na podatke.^[16]

Tako posmatrajući, geografski podaci mogli bi se organizovati u sledeću hijerarhiju:

OKEAN → MORE → SLIV → REKA → POTOK → IZVOR

Osim ovih pomenutih operacija *drill down* i *drill up*, postoji i operacija *drill across*, koja se koristi za povezivanje dve ili više činjeničnih tabela na istom nivou hijerarhije.

3.2.2 Denormalizacija

Najjednostavnije objašnjenje pojma denormalizacije kod dimenzionih modela vezano je za pojam normalizovanja šeme baze podataka kod transakcionog modela. Transakcioni model je suštinski relacoioni, dok dimenzioni nije uvek takav. Sva pravila za normalizovanje primenjena kod transakcinog modela ne važe kod dimenzionog. Gotovo da ni jedna od normalnih formi ne važi u ovom slučaju.

Pojam denormalizacije je neposredno vezan za definisanje dimenzija i mera. Kada je denormalizovan model u pitanju, dimenzije su organizovane u šemu zvezde.

Šema zvezde se sastoji od relativno malog broja tabela sa dobro definisanim vezama. Ona polako postaje standard za izradu skladišta podataka zbog svojih prednosti u odnosu na ostale relacione strukture:

- smanjenje broja fizičkih veza između tabela, obezbeđuje kraće vreme odziva na upit,
- jednostavnost modela obezbeđuje lakoću modifikacije,
- pojednostavljuje razumevanje i navigaciju meta podataka,
- jednostavno je održavanje,
- proširuje skup alata koji se mogu koristiti za rad sa podacima.

Arhitektura dimenzionog modela opisana je pomoću šeme zvezde definisane sa dve vrste tabela – dimenzione tabelle (*dimension table*) i tabelle činjenica (*fact table*) kao što se vidi na slici 3.1.

3.2.3 Kreiranje agregacija

Proces agregacije predstavlja sakupljanje činjeničnih podataka po unapred definisanim atributima. Pomoću agregacija se sumiraju detalji podataka i potom smeštaju u posebne tabelle. Tabele nastale na taj način se koriste od strane aplikacija da bi se eliminisala potreba da se ponovo vrše neki proračuni koji bi se inače morali sprovesti ukoliko ove tabelle ne bi postojale.

Agregacije se kreiraju uglavnom da bi se poboljšale performanse upita. Kao najbitniju performansa upita pomenućemo vreme odziva koje bi trebalo da bude kraće nakon agregacija. Takođe će se smanjiti i broj resursa potrebnih za izvršenje upita. Prilikom kreiranja agregacije mora se voditi računa da se kreiraju samo one zaista potrebne, jer se u suprotnom umanjuju performanse skladišta podataka, a u krajnjoj liniji i upita. Svakako ne treba kreirati agregacije koje obrađuju podatke po nekoliko sati, a potom se koriste jednom godišnje. Nasuprot tome, dobro je kreirati agregaciju koju će upotrebljavati svi ili većina korisnika i pri tome vrlo učestalo.

Skladište podataka sadrži podatke atomskog nivoa. Mere se smeštaju u tabelle činjenica tako da se kasnije mogu koristiti za potrebe analiziranja. Preuzimanje podataka atomskog nivoa iz skladišta podataka ne obezbeđuje optimalne performanse. Tabele činjenica mogu biti vrlo velike, te izvođenje operacija nad podacima atomskog nivoa, smeštenih u njima, može vremenski da traje dugo. Najveći broj upita zadatih nad skladištem podataka odnosi se na

sumiranje (agregaciju) podataka. Poželjno je izvršiti sumiranje za podatke kojima se češće pristupa. Sumiranjem se omogućava da se već postojeći sumarni podaci mogu odmah koristiti, čime se znatno smanjuje vreme odziva na upit koji treba da procesira te sumarne podatke.

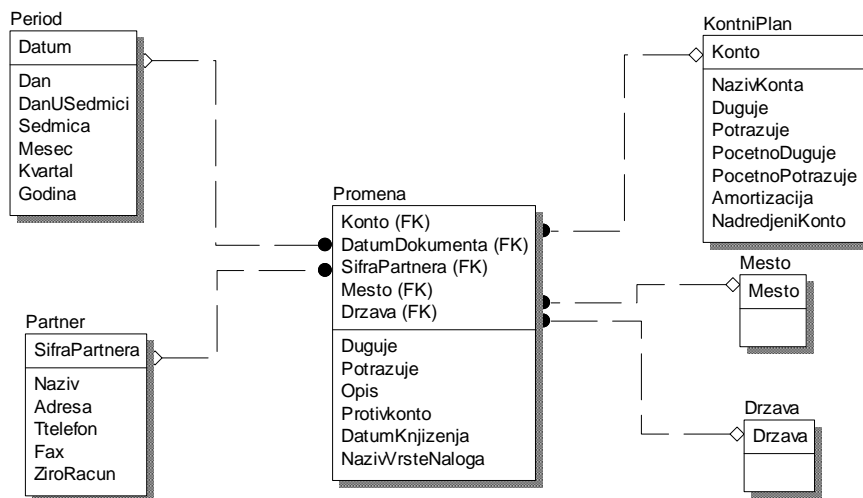
Mnogi upiti koje postavlja korisnik mogu zahtevati agregaciju stotina hiljada redova, te vršenje agregacija unapred može značajno da smanji vreme odziva na upit. Vreme odziva na upit smanjuje se upotrebom agregacija, ali se istovremeno i povećava sama baza podataka.

Zbog toga se može zaključiti da je kreiranje unapred definisanih agregacija neophodno da bi se omogućio rad sa velikim brojem podataka. Agregacije koje vrši korisnik za vreme rada sa skladištem podataka, koje zovemo dinamičkim agregacijama su najčešće dugotrajne, te su za potrebe odlučivanja neprihvatljive.

Jedan od načina na koji se mogu kreirati agregacije su one zasnovane na *SQL* naredbama. Ovaj način je najjednostavniji iako nije najbolji po pitanju performansi sistema.

One druge agregacije, koje se ne zasnivaju na *SQL* naredbama, vezane su za razvijanje specijalizovanih programa, što usložnjava procese sačinjavanja i održavanja skladišta podataka. Prednosti ovog načina kreiranja agregacija leži u mogućnosti izvršavanja agregacija jednim prolazom po podacima. Sama priroda procesa agregacije je takva da se može dekomponovati na više paralelnih procesa.

Slika 3.2 Šema zvezde- Dimenzioni model za Kontrolu i analizu finansijskog poslovanja.



Izvor: Miroљjub Zahorjanski

U ovom slućaju metapodaci (podaci o podacima) predstavljeni su u Prilogu, deo; “Rećnik podataka završnog OLAP modela”. Saznanja o strukturnim podacima tabela završnog sistema, su od suštinskog znaćaja za dalji rad. Ukoliko njima ne bismo posvetili dovoljno paźnje, doveli bismo u opasnost dalji razvoj budućeg skladišta podataka i OLAP sistema koji na njega treba da se osloni.

3.3 Izrada dijagrama klasa

U odeljku 2. u okviru faze Objektno orijentisane analize (OOA) definisan je konceptualni model “Kontrole i analize finansijskog poslovanja”, koji je opis realnog sistema, dok je sada zadatak da se kroz izradu dijagrama klasa za istu funkciju opiše buduće softversko rešenje koje za osnovu ima definisan dijagram koncepta postavljen u OOA, a razrađen i proširen u okviru pripreme podataka za analitićko procesiranje informacija i izradu dimenzionog modela.

U objektno orijentisanom dizajnu (OOD) postavlja se konaćan izgled klasa, atributa operacija, veza i kardinalnosti kojima se definiše konaćno softversko rešenje.

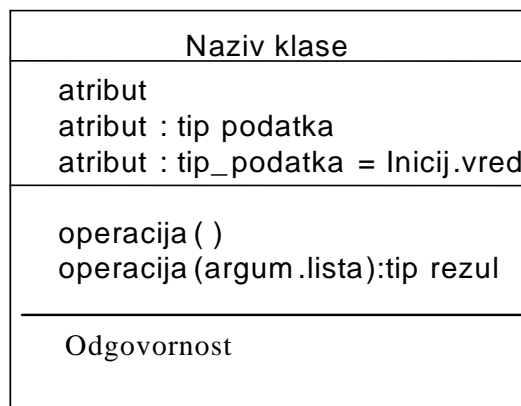
Metodologija *UML*-a opisuje klasu kao skup objekata sa zajedničkim atributima, operacijama, metodama, vezama i semantikom, a kao termin za implementaciju softvera koristi se implementaciona klasa.

Ova metodologija ponašanje klase definiše se operacijom, a metodu tretira kao implementiranu operaciju specificiranu u okviru operacionog algoritma ili procedure.

Izradom konceptualnog modela definisali smo strukturu objekata, a dijagramima interakcije (sekvenci i saradnje), ponašanje objekata odnosno operacije nad njima. S toga je potrebno da te objekte u okviru aktivnosti definisanja klasa prevedemo u pojam klasa i nadogradimo ih.

Struktura objekata definisana je delu "Izrada konceptualnog modela", dok je ponašanje (tj. struktura operacije) definisano u dijagramima interakcije (aktivnost "Izrada dijagrama sekvenci" i aktivnost "Izrada dijagrama saradnje"), pa ih je, u okviru aktivnosti "Definisanje klasa", potrebno prevesti u pojam klasa i nadgraditi ih. Objasnjavajući pojam klase rekli bi smo da je to apstrakcija stvari, koja može biti direktno upisana u programski jezik i predstavljati deo rečnika. Klasa nije individualni predmet (objekt), već skup objekata koji imaju zajedničku strukturu i ponašanje. Klasa može da sadrži neku klasu i da bude sadržana u njoj. S druge strane objekt predstavlja konkretizaciju nečeg što postoji u vremenu i prostoru, tj. objekt je primerak (instanca) klase i ima svoje attribute kao što su osobine, sadržaj i struktura, potom operacije poput akcija i ponašanja i na kraju svoj identifikacioni broj. Objekti izvršavaju i neke radnje definisane kao ponašanje (*behaviors, operations, methods*).

Slika 3.3 Primer klase



Izvor: Miroljub Zahorjanski

Svaka klasa ima svoje ime koje je odvaja od drugih klasa. Naziv klase je tekstualni niz (*string*) definisan imenicom ili frazom iz rečnika. Prvo slovo naziva klase piše se velikim slovom.

Naziv klase treba da bude u skladu sa rečnikom u određenom domenu. Opciono se može navesti stereotip i/ili svojstvo (*property string*), kojima se dodatno opisuje klasa. Odeljak za naziv sadrži naziv klase sa opcionim nazivom paketa kojem pripada (Ime Paketa:ImeKlase).

Atribut je svojstvo ili karakteristika koja je uobičajena za neki ili sve objekte entiteta. Jedan atribut prikazuje korišćenje domena u kontekstu entiteta. U objektnom dizajnu posmatrano, atribut je apstrakcija vrste podatka ili je stanje objekta klase koja se posmatra. Znači da će jednog trenutka objekat klase imati specifične vrednosti za svaki atribut. Naziv atributa je kratka imenica ili fraza koja se može označiti prvim velikim slovom.

Operacija je implementacija servisa koji se može zahtevati od bilo kojeg objekta klase radi toga da utiče na ponašanje, odnosno to je apstrakcija nečega što može da bude urađeno objektu a zajedničko je za sve objekte klase. Klasa može, a ne mora, da ima operacije. Naziv operacije je glagol ili glagolska fraza. Operacija može da se specificira dajući signaturi ime, tip i default vrednost, kao i povratnu vrednost.

Operacije i atributi se zajedno grupišu u klase. Za razliku od struktura tradicionalnih programa, klasa je struktura podataka sa tačno definisanim pridruženim operacijama.

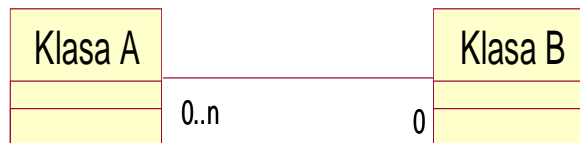
Pojedina klasa može da sadrži prazne odeljke, a može da bude i bez odeljaka. Prazan odeljak atributa/operacija ne znači da ih klasa nema, već da nisu relevantni za dati pogled (dijagram).

Veze ili linkovi obezbeđuju komunikaciju između objekata, odnosno fizičku ili konceptualnu vezu pojavljivanja objekata. Veza se formalno definiše kao n-torka, odnosno uređena lista pojavljivanja. Dijagramima sekvence i saradnje se određuje, kakve veze moraju da postoje između objekata da bi se obezbedilo željeno ponašanje. Ako dva objekta žele da komuniciraju, treba da postoji veza između njih.

Koriste se sledeći tipovi veza:

- Asocijacija je strukturalni odnos koji specificira vezu objekta jedne stvari sa objektima druge stvari, tj. asocijacija predstavlja grupu veza slične strukture i jedinstvene semantike. Ovaj tip veze odgovara neidentifikujućoj vezi u *IDEF1X* metodologiji.

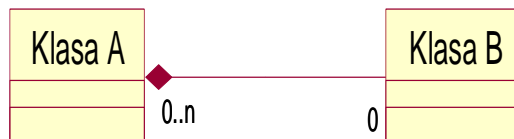
Slika 3.4 Primer veze tipa asocijacija



Izvor: Miroljub Zahorjanski

- Agregacija je specijalni oblik asocijacije i jači je oblik veze, gde se uspostavlja veza između celine i dela (celina je sastavljena od delova). Agregacija se prikazuje kao linija koja spaja klase sa malim romбом uz klasu koja se odnosi na celinu (*Parent*) i pandan je identifikujućoj vezi u *IDEF1X* metodologiji.

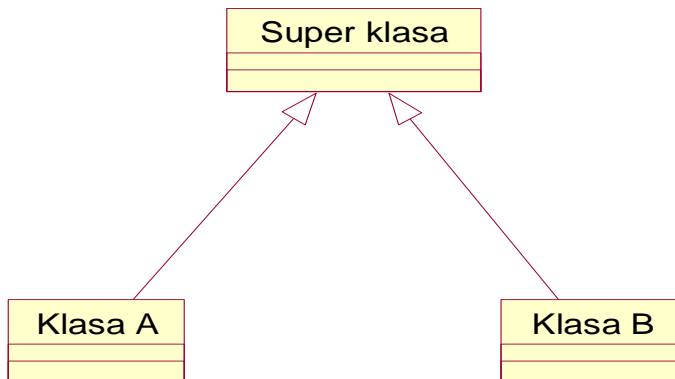
Slika 3.5 Primer veze tipa agregacija



Izvor: Miroljub Zahorjanski

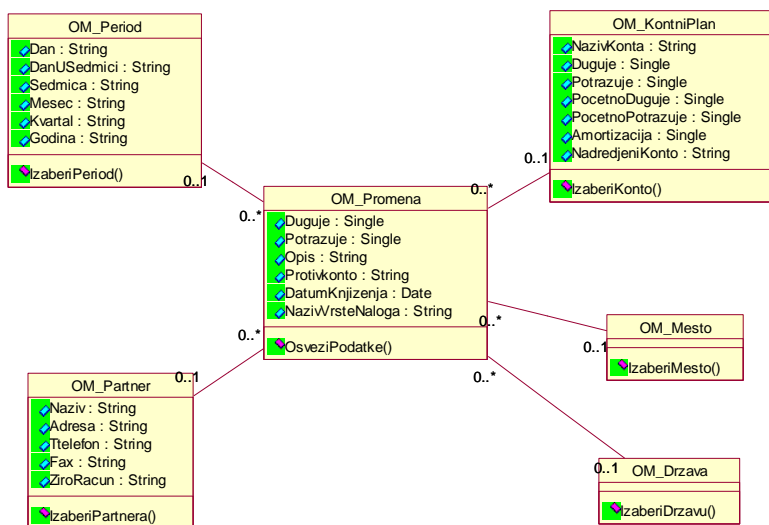
- Generalizacija (osobina nasleđivanje) predstavlja hijerarhijsku vezu između klase (*superclass* – *subclass*).

Slika 3.6 Primer veze tipa generalizacija



Izvor: Miroljub Zahorjanski

Slika 3.7 Dijagram klasa za Kontrolu i analizu finansijskog poslovanja



Izvor: Miroljub Zahorjanski



4. IMPLEMENTACIJA

Nakon procesa Definisane zahteva u kojem smo sagledavali sistem da uočimo potrebe korisnika, potom procesa Objektno orjentisane analize (OOA), gde smo za prethodno definisane poslovne aktivnosti u okviru poslovnih slučajevau potrebe izradili dijagram konceptualnog modela i dijagrame interakcije (sekvenci i saradnje), i na kraju procesa Objektno orjentisanog dizajna (OOD), gde smo radili na pripremi podataka za analitičko procesiranje, izradi dimenzionog modela i dijagrama klasa, ostalo je da se upustimo u proces implementacije.

Ako proces implementacije posmatramo sa aspekta upravljanja onda bismo mogli da ga podelimo na faze: Uvođenje, Testiranje i Održavanje. No kako u ovom radu sagledavamo koncept analitičkih baza podataka u formiranju strategije finansijskog poslovanja preduzeća, sam proces implementacije posmatraćemo iz ugla potrebe za izradom skladišta podataka, pripreme analize podataka i izrade korisničkog interfejsa. Dakle procese koji prethode samom uvođenju softvera u preduzeće i njegovom puštanju u rad.

Implementacija u ovom slučaju predstavlja završne faze izrade softvera, a ne njegovo uključivanje u IS preduzeća.

S toga možemo reći, da ćemo implemetacijom objektno orjentisanog razvoja sistema za finansijsko poslovanje^[11] preduzeća, prikazati rezultate ovog istraživanja i dokazati hipoteze s početka ovog rada.

Prikaz rezultata biće dat u obliku SQL upita i višedimenzione analize podataka (*OLAP*), a na osnovu izgrađenog skladišta podataka, što podrazumeva kreiranje fizičkog modela skladišta podataka, generisanje baze podataka i učitavanje podataka.

Prilikom implementacione faze, *UML* model se preslikava u *OLAP* bazu podataka. To podrazumeva da se, strukturni model prevodi u statički, tj. *OLAP* bazu podataka (tabele i polja) u okviru izrade skladišta podataka i pripreme analize podataka. Model ponašanja se prevodi u definicije tela metoda (funkcija) koje implementiraju definisane interakcije u okviru izrade korisničkog interfejsa.

Na dalje će u ovom radu, biti na svojevrsan način prikazana implementacija baze podataka do 2 GB korišćenjem alata (*MS Access 2003*, *MSquery32.exe* i *MS Excel 2003*) i baze preko 2 GB, pomoću *MS SQL Server 2005* i njegovih alata za ekstrakciju i transformaciju podataka (*DTS*), kao i za *OLAP* analizu (*OLAP server*), koji uključuje i alate za vizuelni dizajn skladišta podataka.

4.1 Izrada skladišta podataka

Pristupanje procesu izrade skladišta podataka ima za cilj da obezbede informacije za proces odlučivanja, a da se pri tom ne ugrozi svakodnevno poslovanje. Da bi smo nastavili dalje sa obrazlaganjem procesa, neophodno je skladište podataka, objasniti sa teorijskog aspekta.

Skladište podataka ^[9,110] ili (*Data Warehouse - DW*) je proces integracije podataka u jedan repozitorijum iz kojeg krajnji korisnici mogu sprovesti *ad-hock* analize podataka i praviti izveštaje.

Warehousing koncept je skladištenje agregiranih, ekstrahovanih i filtriranih podataka u meta baze, koje omogućavaju slojevit, multidimenzionalni pristup podacima, kakav je potreban za donošenje odluka najvišg strateškog nivoa.

Koncept je postavljen veoma fleksibilno i omogućuje naknadno korišćenje različitih alata i modela i posebno se mora naglasiti da transakciona baza nije uslov za primenu koncepta. Koncept se primenjuje nad operativnim podacima pa ukoliko su oni transakcionog tipa, onda i nad njima.

Osnovni cilj skladištenja podataka je prikupljanje i distribucija informacija kroz preduzeće tj. korišćenje bilo koje informacije, sa bilo kog mesta, u bilo koje vreme. Tačnije, ostvarenje principa "Biti uvek na usluzi korisniku informacija".

Menadžeri i analitičari moraju imati pristup do podataka koji su bitni za tu organizaciju. Taj pristup treba biti neposredan, brz, na zahtev korisnika i mora omogućiti visoke performanse. Sve ovo korisnici moraju biti u mogućnosti da postignu koristeći svoj personalni računar.

Podaci u skladištu podataka takođe moraju biti konzistentni. Konzistentnost znači da ako dva korisnika traže isti podatak moraju dobiti isti odgovor iako su oni to tražili u različito vreme. Takođe znači da ako podaci od juče nisu do kraja učitani korisnik mora biti upozoren. Podaci se u skladištu podataka mogu kombinovati na sve moguće načine, što omogućava dimenzionalni model. Skladište podataka nisu samo podaci, već ono mora sadržati i skup alata za postavljanje upita (*query tools*), alata za analizu i predstavljanje informacije.

Karakteristike skladišta podataka je, da je to baza podataka koja sadrži istorijske, nepromenljive podatke koji su logički i fizički izvučeni iz raznih izvora. Ti podaci se u skladu s definisanim modelom učitavaju u skladište i integrišu s postojećim podacima, a sve to u svrhu podrške poslovnom odlučivanju. "Skladištenje podataka je

proces integracije podataka o poslovanju neke organizacije u jednu bazu podataka iz koje krajnji korisnici mogu raditi izvještaje, postavljati upite i analizirati podatke.”^[84] Skladištenje podataka je takođe i proces koji se ne završava inicijalnim učitavanjem podataka, već se skladište podataka osvežava novim podacima u nekim, vremenskim intervalima (svaki dan, nedeljno, mesečno). Iz toga sledi da je skladištenje podataka kontinuiran i dugotrajan proces.

Iz prethodno navedenog, jasna je potreba aktivnog sudelovanja korisnika naručioca u projektnom timu i na nivou operativnih poslova, čak je i nemoguće zamisliti izgradnju kvalitetnog sistema skladišta podataka bez zajedničkog rada. Neke od ključnih uloga u projektnom timu za koje je potreban ovakav angažman su: stručnjak za zahvatanje izvornih podataka, administrator podataka, analitičar i naravno krajnji korisnici. Stručnjak za zahvatanje izvornih podataka osigurava dostupnost izvornih podataka potrebnih u bazi skladišta podataka i poznaje vrlo dobro produkcione sisteme. Administrator podataka u fazi izgradnje ključna je osoba za proveru ispravnosti podataka u bazi skladišta podataka koje upoređuje s podacima iz produkcionih baza postavljajući iste kriterijume upita na obe strane. Na osnovu takvih testiranja potvrđuje se ispravnost algoritma zahvatanja podataka. U fazi rada skladišta podataka vodi se računa o radu procesa prenosa podataka (da li je prenos prošao u predviđenom vremenskom periodu, da li su sve potrebne radnje pripreme podataka za krajnje korisnike izvršene) te o tome obaveštava korisnike. Krajnji korisnici kojima su namenjene aplikacije razlikuju se od korisničke populacije produkcionih sistema. Prvenstveno su to analitičari i određeni nivoi upravljačke strukture, tu se još javljaju i "Data Mining" specijalisti^[3,43,86,88], istraživači tržišta, stručnjaci za promociju, marketing, kontroling i u većini slučajeva prodaju i finansije.

Spremnost korisnika na upotrebu sistema jedan je od ključnih faktora uspeha projekta u celini. Naravno da spremnost korisnika zavisi od upotrebljivosti sistema, odnosno osećaja krajnjih korisnika o rešavanju konkretnih problema upotrebom sistema. Kako bi se pridobili korisnici za upotrebu sistema potreban je njihov angažman za sve vreme trajanja projekta. Oni su stalni korektor aktivnosti, a u

slučaju velikog broja krajnjih korisnika, može se oformiti grupa ključnih korisnika koju čine predstavnici pojedinih grupa, kako bi se olakšala komunikacija. Uvođenje sistema skladišta podataka predstavlja uvođenje nove tehnologije s kojima se većina krajnjih korisnika nije susrela niti je imala priliku da radi. U većini slučajeva to je profil korisnika koji je naučen da radi s programima za tabelarno izračunavanje, alatima za obradu teksta i slično, koji dozvoljavaju vrlo

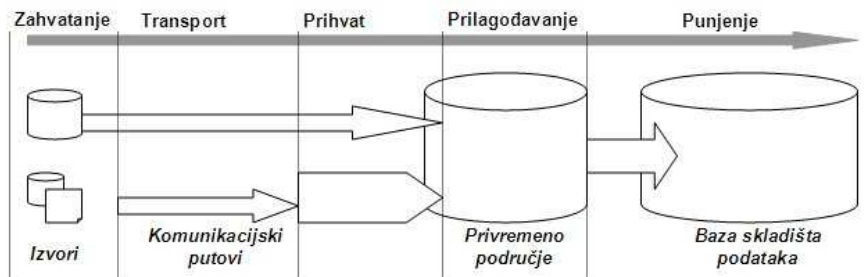
veliku slobodu u kreiranju "output-a", te ih treba sistemski obučavati za prelazak na nove korisničke alate i aplikacije i njihovo prihvatanje u svakodnevnom obavljanju posla.

Izrada sistema prenosa podataka predstavlja najteži zadatak u celom procesu izgradnje sistema skladišta podataka i na njega otpada približno 60% ukupnog vremena. Proces prenosa podataka definiše se kao proces koji obavlja prenos podataka iz izvornih sistema u skladište podataka u predviđenim vremenskim razmacima. Sam proces može se podeliti u nekoliko faza i to:

- Zahvat – zahvatanje izvornih podataka po utvrđenom algoritmu iz predviđenih izvora kako bi se osigurala konzistentnost zahvaćenih podataka (ekstrakcija u užem smislu).
- Transport – transportovanje zahvaćenog seta podataka komunikacijskim kanalima do prihvatnog područja.
- Prihvat – prihvatanje podataka i njihovo smeštanje u prihvatne tabele privremenog područja skladišta podataka.
- Prilagođavanje – prilagođavanje strukture podataka, unutar pripremnog područja u skladišnom modelu podataka kako bi se moglo obaviti punjenje.
- Punjenje – krajnja faza koja rezultira punjenjem tabela skladišta podataka i pripremu za korisničku upotrebu.

Pre samog projektovanja procesa prenosa, potrebno je osigurati početnu definiciju korisničkih zahteva odnosno modela skladišta podataka, dostupnost i spremnost na saradnju stručnjaka za zahvat izvornih podataka, kao i stabilnost logičke strukture izvornih sistema. Ako bilo koji od navedenih uslova nije zadovoljen, potrebno je odgoditi fazu izgradnje prenosa podataka jer se u protivnom može računati s mesecima uzaludno obavljenog posla.

Slika 4.1 Struktura procesa prenosa podataka



Izvor: Miroljub Zahorjanski

U okviru procesa izrade skladišta podataka možemo identifikovati bar četiri osnovna podprocesa.

Pre svega to je analiza izvora podataka, koja se obavlja na osnovu informacija o operativnim (transakcionim) skladištima podataka. U našem slučaju *OLTP (On-Line Transaction Processing)* podaci nastali su kao istorijat poslovanja. Ovaj podproces podpomognut je mehanizmom *RDBMS (Relation Data Base Management Sistem)* i odvija se pod kontrolom pravila za preuzimanje podataka.

Drugi podproces izrade skladišta podataka bio bi postupak pripreme podataka koji se odvija delom pod već rečenim pravilima za preuzimanje, a delom pod kontrolom procedura u kojima su ta pravila inkorporirana. U postupku pripreme podataka ulazni element su zahtevi korisnika. Dakle moramo unapred znati šta od skladišta podataka korisnik očekuje (koje buduće analize) kako bi smo znali koje podatke da pripremimo.

Treći podproces bio bi konkretno uskladištenje podataka koje se odvija pod kontrolisanim postupkom za uskladištenje i ovde deluju podjednako dva mehanizma: *RDBMS* i *OLAP* alati. Krajnji produkt (izlaz) ovog podprocesa bili bi zbirni podaci.

Ovakvi zbirni podaci tretiraju se kao ulaz u zadnji podproces Analiza podataka. Kontrole su definisani postupci za analize podataka, mehanizmi su *OLAP*, *Data Marts*^[88] i *Data Mining* alati, dok je izlaz iz ovog zadnjeg podprocesa *DW* analiza.

4.1.1 Kreiranje fizičkog modela BP

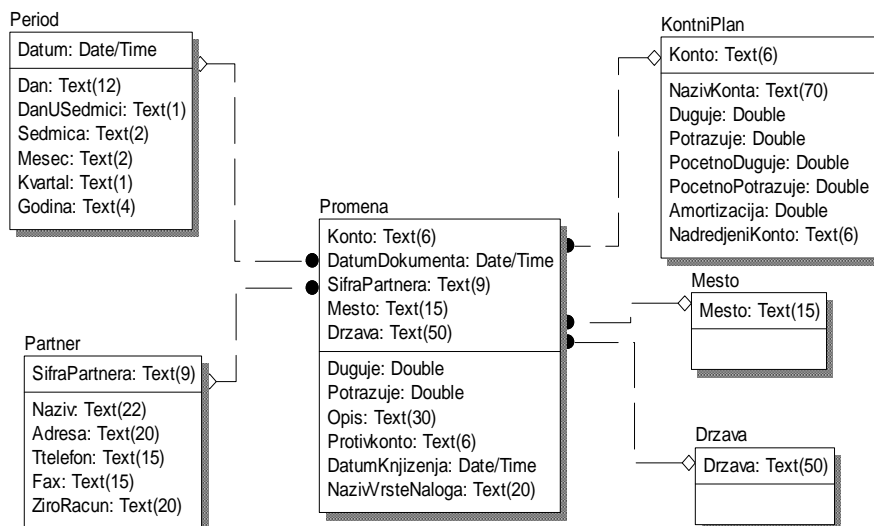
Kreiranje fizičkog modela skladišta podataka je postupak prevođenja logičkog modela podataka u fizički model skladišta podataka. Fizički model skladišta podataka prevodi dimenzioni (logički) model skladišta podataka i tom prilikom dolazi do sledećih konvertovanja:

- entiteta iz modela podataka u tabele fizičke baze podataka,
- atributa u kolone, u odgovarajućim tabelama, i
- kandidata za ključeve entiteta u primarne ključeve u tabelama.

Domeni i tipovi podataka u skladištu podataka definisani su na osnovu domena i tipova podataka iz transakcione baze podataka.

Na slici 4.2 je prikazan fizički model šeme skladišta podatka i to *IDEF1X* notacijom.

Slika 4.2 Fizički model skladišta podataka za analizu finansijskog poslovanja



Izvor: Miroljub Zahorjanski

Na osnovu *OLTP* šeme baze podataka za sistemske slučajeve upotrebe (u okviru poglavlja izrada modela sistemskih slučajeva upotrebe) *OLAP* šema zvezde baze podataka ima sledeće elemente:

- činjenice su finansijske promene
- dimenzije su period, partner, kontni plan, mesto i država

4.1.2 Generisanje baze podataka

“Na osnovu fizičkog modela podataka, vrši se generisanje šeme baze podataka koju čine fizičke tabele, kolone i relacije, koje se u *CASE* alatu automatski generišu iz fizičkog modela.

Proces generisanja šeme baze podataka naziva se direktni inženjering. Kada se generiše šema baze podataka, entiteti prelaze u tabele, atributi u kolone, a veze u relacije i definiše se referencijalni integritet, trigeri, procedure, indeksi i druge osobine koje podržava izabrani *SUBP*. „^[101]

Za samo generisanje baze podataka neophodno je da model podataka u *ERwin*-u bude prebačen na fizički nivo, potom izaberi odgovarajuću ciljnu platformu u ovom slučaju *Access 2003*. U *Access*-u napraviti jednu praznu *.mdb* datoteku na putanji gde je i fajl *.er1*. Ta datoteka treba da bude zavorena (da se ne koristi) u trenutku generisanja, a prethodno je neophodno da se kao prazna snimi (*Tools, Options..., Advanced, Default file format*) u onom formatu koji je izabran u *ErWin*-u 4.1.2522 (*Database, Chose Database..., Target Desktop DBMS, Access, Access Version, ... Version*). Sledeći korak je da se iz *ErWin*-a treba logovati-konektovati na ciljnu platformu u prozoru (*Database, Database Connection*). Tom prilikom *ERwin* kreira aktivnu bidirekcionu vezu sa sistemskim katalogom izabranog servera (*Target Desktop DBMS*) koja omogućava uz praznu *.mdb* datoteku u *Access*-u direktno generisanje šeme baze podataka. Naredni korak je da se u *ERwin*-u (*Tolls, Forward Enginerr/Schema Generation... , Generation*) izvrši generisanje. Nakon generisanja u *ERwin*-u treba izvršiti diskonekciju i otvoriti *.mdb* datoteku u *Access*-u. Tabele bi trebalo da su prisutne u *Dztabase* prozoru *Access*-a.

Tipovi relacija definisani u *MS Access*-u su siromašniji po mogućnostima od onih koje sadrži *ERwin* model podataka, što ne bi trebalo da čudi obzirom da *ERwin* podržava oko petnaestak *Target SQL DBMS* i oko pet do šest *Target Desktop DBMS* u zavisnosti od verzije.

No i pored toga neke mogućnosti kojih nema u *Access*-ovom *Relationships* nadomeštene su opcijama *Field Properties* u delu

kreiranja tabela, kao što su : *Field Size, Format, Input Mask, Default Value, Validation Rule, Required, Indexed* i slično.

Stoga se u ovom konkretnom slučaju naknadno pristupilo dograđivanju dobijenog modela podataka, uporedo vodeći računa o kardinalnosti relacija na nivou *MS Access*-a i tipovima relacija kao što su *Cascade Update Related Fields* koji omogućuje da se u slučaju izmene sadržaja polja u prvoj tabeli promene i vrednosti polja koji su u relaciji sa njima u drugoj tabeli i *Cascade Delete Releted Records* koji omogućava da se u slučaju brisanja redova u jednoj tabeli izaziva brisanje redova iz druge tabele.

Što se tiče aktivnosti izrade aplikacije kao što su : Definisane menija, Definisane izgleda formi, Definisane upita i Definisane izeštaja, o tome će biti reči na stranama vezanim za korisničke interfejsne transakcione i *OLAP* baze podataka.

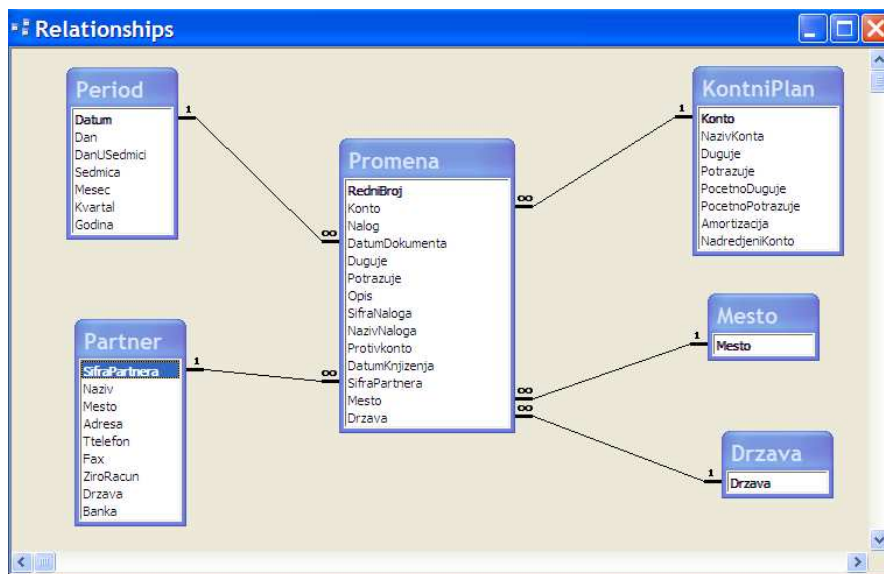
Access je multifunkcionalni SUBP; sastoji se od mnoštva povezanih alata za generisanje, organizovanje, izdvajanje, prikazivanje, štampanje i objavljivanje podataka.

Obzirom da je *ERwin* alat sa višestrukim izborom pri mogućnosti generisanja baze podataka, ova činjenica je iskorišćena da se budućem *OLAP* sistemu pruži šansa delovanja i u uslovima veličine skladišta podataka do 2 GB (*Access 2003*) i preko 2GB (*MS SQL Server 2005*).

Za prvi slučaj kada je baza podataka, u ovom slučaju skladište podataka ispod kriterijuma od 2 GB izvršeno je generisanje u *Access Target Desktop DBMS*. Rezultat je prikazan na Slici 4.3 Fizička realizacija skladišta podataka u *MS Access*.

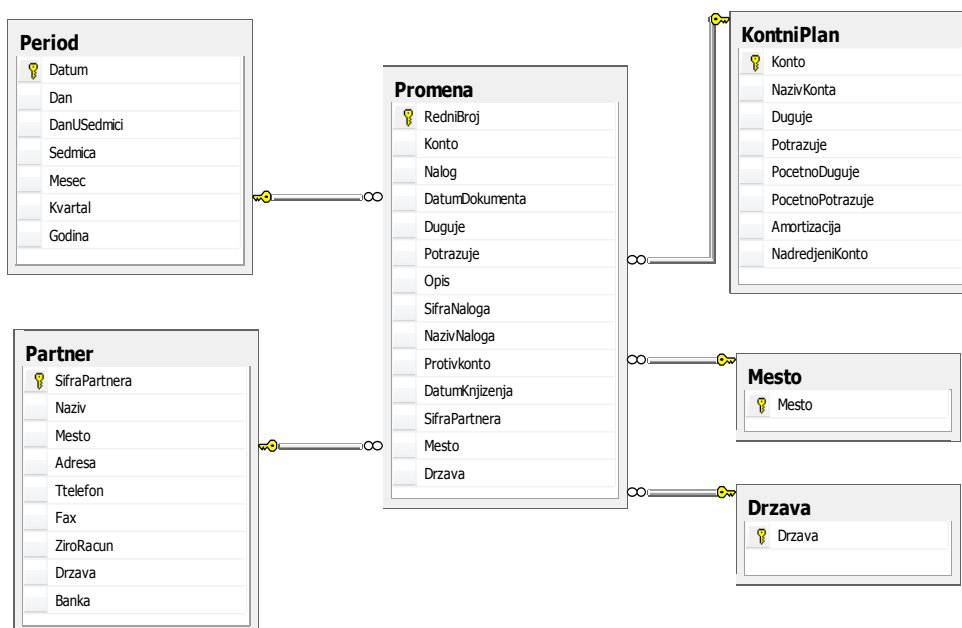
Za varijantu kada se od skladišta podataka očekuje da će preći veličinu od 2 GB izgenerisana je šema baze podataka (prazne tabele koje su povezane) u *MS SQL Server 2005*. Ovaj drugi alat je iz grupe *Target Desktop DBMS*-a. Rezultat ovog drugog generisanja prikazan je na slici 4.4 Fizička realizacija skladišta podataka u *MS SQL Server 2005*.

Slika 4.3 Fizička realizacija skladišta podataka u MS Access.



Izvor: Miroljub Zahorjanski

Slika 4.4 Fizička realizacija skladišta podataka u MS SQL Server 2005



Izvor: Miroljub Zahorjanski

4.1.3 Učitavanje podataka

Nakon ekstrakcije i transformacije podataka, sledi faza učitavanja podataka, odnosno kreiranja paketa za učitavanje podataka. U tu svrhu, za varijantu kada se od skladišta podataka očekuje da će preći veličinu od 2 GB korišćen je *Export Wizard MS SQL Server-a 2005*. Prvi korak bio je izbor izvora podataka, odnosno tabele iz koje se preuzimaju podaci i odredišne tabele, odnosno tabele u koju se učitavaju podaci.

Kod varijante u kojoj je za očekivati skladište podataka ispod kriterijuma od 2 GB korišćena je postojeća *OLTP* baza podataka, čiji je model prikazan na početku ovog rada u odeljku Definisane zahteva slika 1.1. Najveći deo podataka u ovom slučaju učitani su korišćenjem *SQL* upita kreiranih *Access*-ovim alatom *Append query* i njegovom *SQL* naredbom *INSERT INTO (ANSI SQL-92)*.

4.2 Priprema analize podataka

Razlog za izgradnju skladišta podataka bio je da se obezbedi lako pristupačan izvor podataka visokog kvaliteta, dok ćemo razlog ili svrhu analize finansijskih podataka objasniti u nekoliko rečenica. Pre svega za investitore je bitno ne samo da znaju da čitaju finansijske izveštaje, već da znaju kako da analiziraju informacije date u osnovnim finansijskim izveštajima – bilansu stanja i uspeha. Finansijska analiza omogućava brz i relativno jednostavan uvid u finansijsko stanje preduzeća upoređivanjem različitih finansijskih pozicija (racio analiza).

Finansijska ili racio analiza koristi se u cilju komparacije performansi jedne firme sa drugim firmama iz iste grane (međusektorska analiza), kao i za komparaciju sa poslovanjem iste firme u proteklom periodu ili sa planiranim performansama (analiza vremenskih serija).

Ako se pogleda struktura potencijalnih korisnika rezultata finansijskih analiza, a to su:

- Akcionari (postojeći, potencijalni)
- Poverioci (postojeći, potencijalni)
- Menadžeri
- Zaposleni
- Kupci, dobavljači
- Država

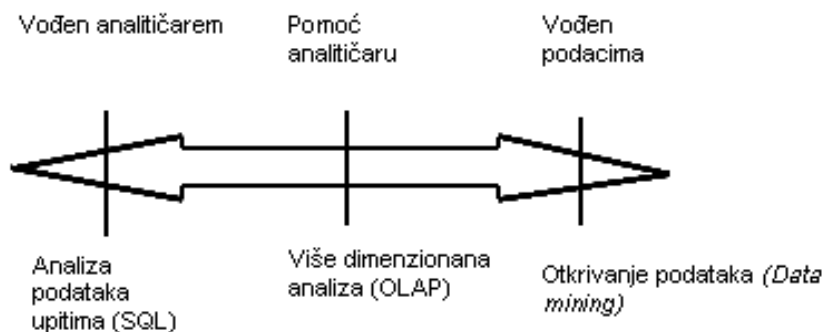
onda se da uočiti njihova raznolikost kako po lokaciji u odnosu na centralno skladište podataka (spoljni i unutrašnji tražioci informacija) tako i po analitičkim znanjima i sposobnostima. Na finansijske izveštaje koji su zakonom propisani ne može se mnogo uticati ali na softversku analizu *OLAP* alata i njihovu prilagodljivost svim mogućim razumnim pitanjima svakako da.

“U zavisnosti od uključenosti analitičara, postoji tri pristupa analizi podataka i to:

- ako analitičar vodi posao onda se koriste *SQL* upiti i izveštaji,
- ako je orijentacija na višedimenzione analize onda dobijeni rezultati služe kao pomoć analitičarima, i
- ako se koristi metodologija otkrivanja znanja u podacima (*Data mining*) onda je analitičar vođen podacima.

Izbor pristupa analizi podataka može uticati na tip odabranog modela podataka i njegov sadržaj. Jasno je i da se, u većini implementacija, može koristiti više tipova modela podataka da bi se najbolje zadovoljili različiti zahtevi skladišta podataka.”^[99]

Slika 4.5 Izbor načina analiziranja podataka



Izvor: Veljović A. Njeguš A., Osnove relacionih i analitičkih baza podataka, Megatrend univerzitet primenjenih nauka, Beograd, 2004.

Na narednim stranama biće pojašnjena analiza podataka upitima (*SQL*), ali će glavno težište ovog rada, odnosno Koncepta analitičkih baza podataka u formiranju strategije finansijskog poslovanja preduzeća biti na višedimenzionoj analizi (*OLAP*).

4.2.1 Analiza podataka upitima (SQL)

Procesom analize podataka upitima upravlja analitičar. On mora da postavi pitanje da bi dobio odgovor, što znači da je analiza upitima i izveštajima proces postavljanja pitanja na koje se traži odgovor. Procs započinje izdvajanjem podataka od značaja, njihovom transformacijom u odgovarajući kontekst i prikazivanjem u čitljivom formatu.

Objašnjenje pojma upit bilo bi, da je to proces uzimanja poslovnih pitanja ili pretpostavki i njihovo prevođenje u format upita koji može koristiti određeni alat za podršku pri odlučivanju. Nakon izvršenja upita, alat na kojem se upit izvršava, generiše odgovarajuće komande za dobijanje traženih podataka, koji se smeštaju u skup odgovora. Osoba koja vrši analizu podataka potom obavlja potrebne manipulacije i kalkulacije na skupu odgovora da bi dobila očekivane rezultate. Potom se ovi rezultati formatiraju da bi odgovarali obrascu prikaza ili izveštaja koji je odabran da olakša razumevanje krajnjem korisniku. Ovaj obrazac se može sastojati od kombinacije grafike, teksta, vido i audio prokaza. Kao krajnji rezultat, izveštaj se dostavlja korisniku na željenom izlaznom medijumu koji može biti monitor, papir, ali zvuk.

Menadžeri kao krajnji korisnici su pre svega zainteresovani za obradu numeričkih podataka koje imaju na raspolaganju da bi analizirali ponašanja poslovnih procesa. Osim toga oni mogu da računaju ili istražuju kvalitativne mere, kao što su stepen zadovoljstva korisnika, vremenske zaostatke u poslovnim procesima ili pogrešne događaje. Krajnji korisnici takođe mogu analizirati efekte poslovnih transakcija ili događaja, analizirati trendove ili vršiti ekstrapolaciju (izbacivanje detalja) njihovih predviđanja za budućnost. Nije redak sličaj da će prikazani podaci inicirati korisnika da formuliše drugi upit nebi li razjasnio skup odgovora ili prikupio detaljnije informacije. Ovakav proces se nastavlja do dobijanja željenog rezultata.

4.2.2 Višedimenziona analiza podataka (OLAP)

OLAP ili višedimenziona analiza predstavlja način da se prošire mogućnosti upita i izveštaja. U suštini to znači da se umesto izvršavanja višestrukih upita (upit nad upitom) podaci strukturiraju tako da se omogući brz i lak pristup odgovorima na sva razumna pitanja koja korisnici najčešće postavljaju.

Dok su produkcionni sistemi^[27] namenjeni ažuriranju baza podataka i obradi transakcija (*On Line Transaction Processing* –

OLTP), dotle je interaktivno analitičko procesiranje (*On Line Analytical Processing – OLAP*) namenjeno kako sam naziv govori *on line* analizama i izveštajima.

Ono što krajnji korisnik očekuje od višedimenzionalne analize, odnosno ono što njemu treba je:

- da može da postavi bilo koje poslovno pitanje,
- da može da koristi za analizu bilo koji podatak iz transakcione baze podataka,
- da ima mogućnost neograničenog izveštavanja.

U današnjim uslovima koji vladaju na tržištu, finansijskim menadžerima su potrebni odgovori na pitanja koji će im povećati kompetencije i obezbediti veću sigurnost u donošenju odluka. Dakle potrebni su im jasni odgovori i na najteža pitanja i to u što kraćem vremenskom periodu. Upravo iz tih razloga i u tu svrhu se koriste analitički *OLAP* sistemi koji obezbeđuju korisne informacije za analizu poslovnih problema ili finansijskih situacija. Ono što je karakteristično za analitičko procesiranje je da se ono vrši prevashodno korišćenjem raznih poređenja i analiziranjem gotovih obrazaca (šablona) i osnovnih pravaca kretanja (trendova).

Razlog što analitičke baze podataka ne sadrže ažurne podatke, već čuvaju informacije iz određenog trenutka vremena je što analiziranje šablona podataka i trendova zahteva postojanje velikog broja istorijskih podataka.

Dva su pojma vezana za *OLAP* sisteme, a u službi su nadogradnje skladišta podataka ^[28,29]. To su višedimenzionalnost i denormalizacija.

Višedimenzioni pogled je pojam vezan za bazu podataka koja omogućava korisnicima analize velikih količina podataka. Predstavlja podatke kao nizove koji su organizovani u višestruke dimenzije. Promenljive su objekti koji se čuvaju u višedimenzionim bazama. To su jednostavni nizovi vrednosti (numeričkih najčešće) koji su dimenzionisani po dimenzijama u bazi podataka. Može da ima višestruke promenljive, sa različitim ili jedinstvenim skupom dimenzija. Ovaj višedimenzioni pogled na podatke naročito je važan za *OLAP* aplikacije.

Pojam denormalizacije objašnjen je već u odeljku 3.2 Izrada dimenzionog modela i nećemo ga naknadno elaborirati.

Osnovni elementi *OLAP* sistema su:

- baza podataka, koja služi kao osnova za analizu;
- *OLAP* server, za upravljanje i manipulaciju podacima;

-
- interfejs sistem, prema korisniku i prema drugim aplikacijama, i
 - alati za administriranje.

“Pokušaj korišćenja *OLAP* pristupa nad bazama podataka koje su nastale na osnovu modela podataka projektovanog da podrži transakcioni nivo informacionih sistema i obezbedi zahtevani nivo integracije podataka, ne može se izvesti dovoljno efikasno za praktičnu upotrebu, a takođe ugrožava nivo performansi transakcionog nivoa. Za korišćenje *OLAP* složene procedure potrebno je transakcione podatke prebaciti u posebnu bazu podataka.”^[99]

OLAP server je obavezni deo hardvera da bi se ostvario takozvani *OLAP* pristup. Na njega se povezuju relacione baze podataka, eksterni izvori podataka i ostali interni podaci, koji su podržani grafičkim interfejsima, radnim tabelama i ostalim alatima. Ovaj deo hardvera koristi višedimenzione strukture za čuvanje podataka i veza između njih. Same višedimenzione strukture se najbolje vizuelizuju kao kocke podataka i kao kocke u kockama podataka. Pojedinačna strana kocke se naziva dimenzijom. Dimenzije će u našem slučaju predstavljati kategoriju podataka, kao što su vreme, konta ili računi, partneri, lokacije i slično. Svaka ćelija kocke sadrži agregirane podatke koji su u vezi sa dimenzijama. Na primer, jedna ćelija može sadržati podatke o ukupnom rashodu za održavanje opreme jednog pružaoca usluga u toku jednog meseca.

Opracije koje podržavaju *OLAP* serveri su:

- Konsolidacija – kojom se vrši agregacija podataka po zadatom kriterijumu;
- *Drill down / Drill up* – koje omogućavaju prikazivanje podataka više ili manje detaljno;
- Isecanje (*slice & dice*) – kojom se obezbeđuju prikazivanje podataka iz različitih perspektiva. Isecanje se najčešće vrši po vremenskoj dimenziji radi analiziranja trendova. (na primer, jedan isečak kocke može prikazivati sve podatke o plaćanjima vezano za sve kupce po regionima, a drugi isečak može prikazivati sve podatke o plaćanjima pojedinačnog kupca po periodima).

OLAP serveri takođe imaju karakteristiku da smeštaju podatke u sabijenom, zgusnutom obliku. Ovo se postiže učestalim monitoringom blokova na disku i tehnikama za kompresiju podataka da bi se što bolje iskoristili prostori za čuvanje podataka. Ređe popunjene matrice se čuvaju dislocirano od zgusnuto popunjenih

matrica. *OLAP* serveri na ovaj način umanjuju zahteve za čuvanje podataka.

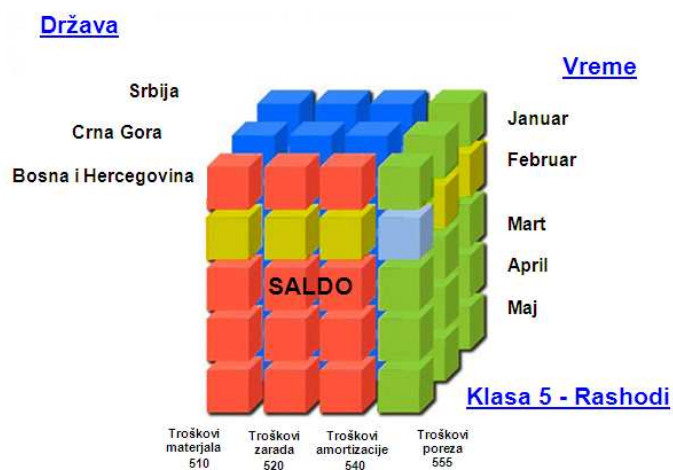
U konkretnom slučaju, prilikom istraživanja vezanog za ovaj rad korišćen je za skladište podataka do 2 GB, alat *MS Query32.exe* za kreiranje glavnog upita podesnog za analizu.

Obzirom da je aplikacija kao krajnji produkt ovog projekta namenjena podršci analitičkih baza u formiranju strategije finansijskog poslovanja preduzeća, prikazaćemo i koncept *OLAP* kocke namenjen za jedan od mogućih izveštaja predviđen još na samom početku prilikom funkcionalnog modeliranja.

Zbog sličnosti sa geometrijskom reprezentacijom na slici 4.6 osnovni gradivni element skladišta podataka se naziva "kockom". Zapravo to je slikovit prikaz dimenzionalnog modela, koji se sastoji od jedne centralne tabele činjenica i više zvezdasto (*star schema*) ili u obliku pahuljice (*snowflake schema*) oko te tabele postavljenih tabela dimenzija

Skladište podataka se često sastoji od više kocaka. Kocku treba zamisliti kao višedimenzioni prostor podataka u kome se nalaze podaci o poslovanju preduzeća. Dimenzije tog prostora mogu biti različite: vreme, prostor, poslovni partneri, konta-računi, itd. Podaci koje kocka sadrži su veličine relevantne za analizu poslovanja: u ovom slučaju saldo dugovne i potražne strane. Obično su to aditivne ili poluaditivne veličine.

Slika 4.6 Primer *OLAP* kocke za budući dinamički izveštaj troškova u vremenu i prostoru.



Izvor: Miroljub Zahorjanski

Na Slici 4.6 se jasno uočavaju tri dimenzije budućeg izveštaja (prostor, vreme i računski okvir), čijim će parametrima menadžment tim moći da operiše i na brz i efikasan način dobija informacije putem upotrebe takozvanih Pivot-tabela i Pivot-grafikona.

MS QUERY je alat koji se dobija samo prilikom kompletne instalacije paketa *MS OFFICE*. Valja napomenuti da se ova alatka čiji je pun naziv i ekstenzija (*Msqry32.exe*) nalazi na istoj putanji ispod foldera *Program files* gde su i ostale izvršne aplikacije ovog paketa u zavisnosti od verzije *MS OFFICE*-a. Jednostavnim kreiranjem prečice treba dovesti ovu alatku na radnu površinu kako bi bila pristupačna pri budućem kreiranju upita i kreiranju *OLAP* kočke.

Pre formiranja bilo kojeg upita potrebno je da znamo kakve podatke (informacije) želimo da dobijemo na bazi posmatranog poslovnog procesa u ovom slučaju (Kontrola i analiza finansijskog poslovanja). Imajući u vidu raspoloživu strukturu *Access* baze podataka, a za potrebe menadžmenta preduzeća, mogu se zahtevati, između ostalih, sledeći sintetički podaci:

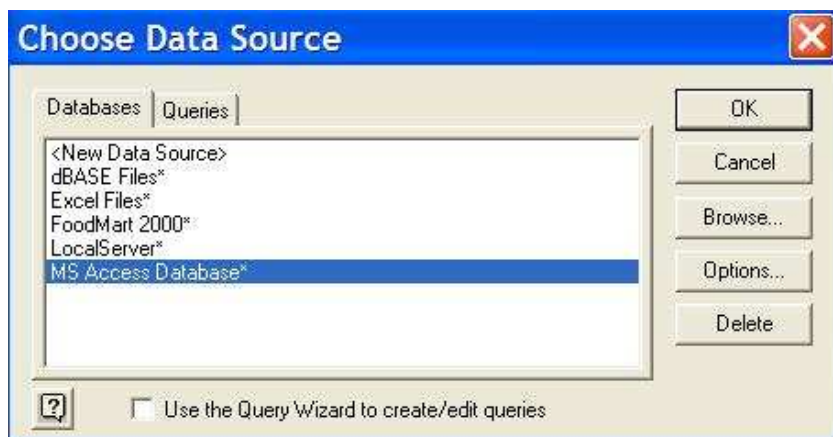
- dugovanje po kupcu u vremenu (naša potraživanja)
- potraživanje po dobavljaču u vremenu (naša dugovanja)
- utrošak repromaterijala po proizvodu u vremenu (utrošci)
- prodaja proizvoda po kupcu u vremenu (prihod)
- nabavka repromaterijala po dobavljaču u vremenu (troškovi)

Navedeni primeri zahtevanih sintetičkih podataka mogu da se kombinuju čime se dobija višestruko veći broj mogućih informacija koje mogu biti od interesa za posmatrani poslovni proces. Takođe, mogu se generisati i drugi izveštaji po želji, a naročito na predlog menadžmenta koji svakodnevno koriste ove podatke.

Analiza slučaja “Veliki kupac”

Za ovaj slučaj ćemo se usredsrediti na analizu kupaca po veličini. Pojam “Veliki kupac” su oni partneri kod kojih ($\sum Duguje \rightarrow +\infty$) suma duguje teži plus beskonačno. Znači što je veće zaduženje to je kupac veći, ne uzimajući trenutno u razmatranje izvršenja njegovih obaveza plaćanja. Pokrenućemo *Ms Query*, izabrali opciju *File, New* i dobiti prozor *Chose Data Source*.

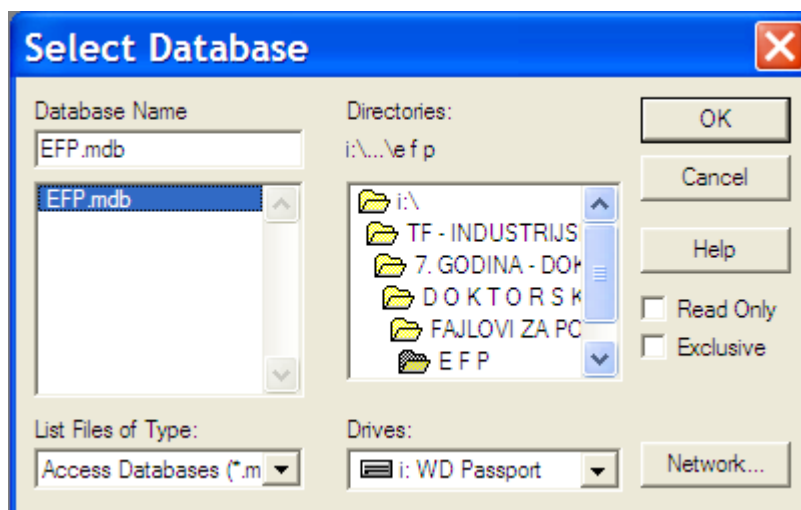
Slika 4.7 Izgled prozora za izbor izvora podataka



Izvor: Miroljub Zahorjanski

Pošto nam se podaci nalaze u Access-ovoj datoteci *EFP.mdb*, obraćemo ponuđenu opciju *MS Access Database* i pritisnuti *OK*. Na ekranu ćemo dobiti novi prozor *Select Database*.

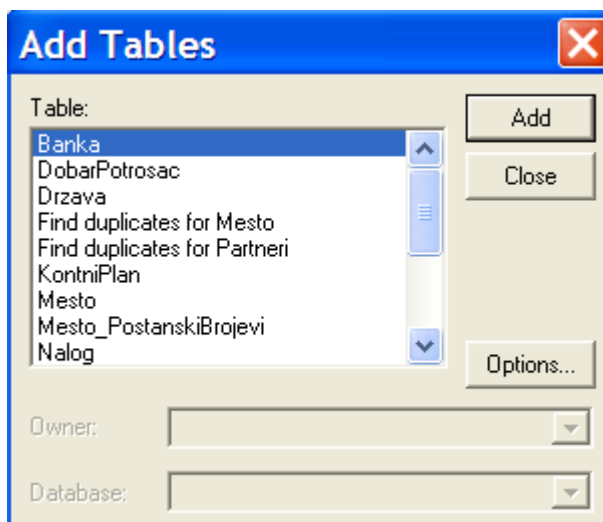
Slika 4.8 Izgled prozora za izbor konkretnog izvora podataka



Izvor: Miroljub Zahorjanski

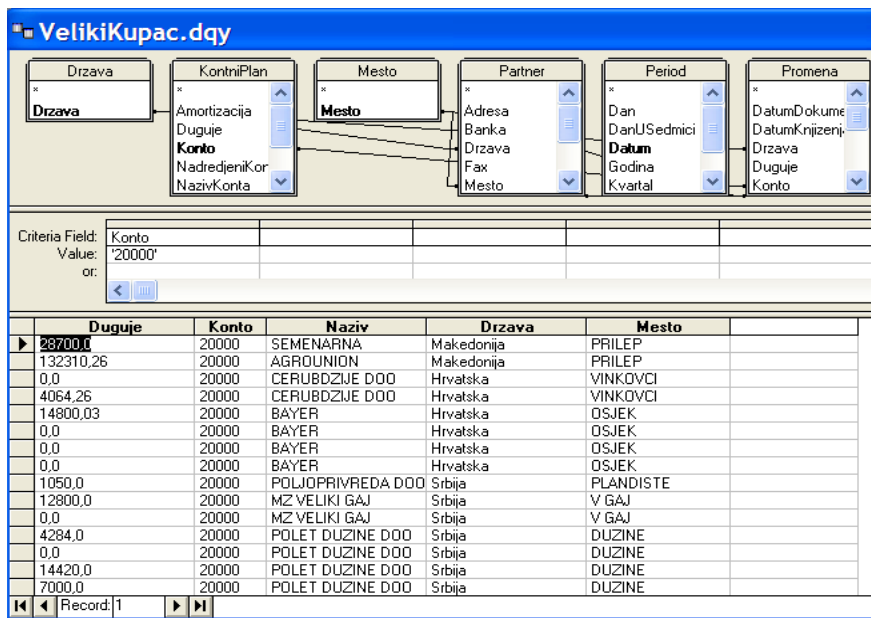
Pronaći ćemo na stablu direktorijuma željenu .mdb datoteku i kliknuti na *OK*. Program nas odmah prebacuje u naredni prozor *Query from MS Access Database* sa već otvorenim podprozorom za dodavanje tabela *Add Tables*. Iz prozora *Add Tables* treba odabrati tabele koje će služiti za formiranje upita. Ukoliko je neophodno da se neke tabele dodatno povežu relacijama to se može naknadno učiniti po dodavanju tabela.

Slika 4.9 Odabir tabela koje će služiti za formiranje glavnog upita




Izvor: Miroljub Zahorjanski

Slika 4.10 Izgled povezanih tabela



Izvor: Miroljub Zahorjanski

Potom treba kliknuti na šestu sličicu počevši od leve strane menija koja se zove *Show/Hide Criteria* . Na taj način otvaramo kao i kod Access-a, *Design* mod za kreiranje upita.

Sasvim je svejedno da li ćemo prvo kreirati uslove za upit u redovima *Criteria Field* i *Value* ili ćemo prvo prevući željena polja iz tabela u donji deo prozora za upit. Najvažnije je da smo zadovoljni kreiranim upitom i da smo sigurni u ishod koji će on proizvesti. Ukoliko se ne ispune naša očekivanja uvek se upit može popraviti ili načiniti nov.

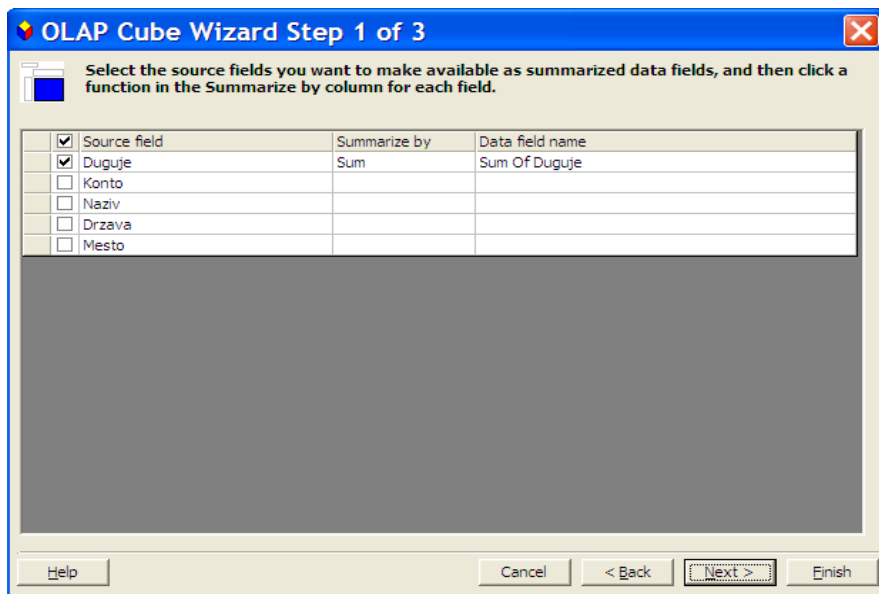
Upit koji je kreiran iz povezanih tabela u SQL sintaksi izgleda ovako:

```
SELECT Promena.Duguje, Promena.Konto, Partner.Naziv,
       Drzava.Drzava, Mesto.Mesto
FROM   Drzava Drzava, KontniPlan.KontniPlan, Mesto. Mesto,
       Partner.Partner, Period. Period, Promena.Promena
WHERE  Promena.Drzava = Drzava.Drzava AND Promena.Konto =
       KontniPlan.Konto AND Promena.Mesto = Mesto.Mesto
       AND Promena.SifraPartnera = Partner.SifraPartnera AND
       Period.Datum = Promena.DatumDokumenta AND
       ((Promena.Konto='20000'))
```

Upit ćemo izvršiti sa *File, Execute SQL....*

Ovako kreirani upit potrebno je snimiti u ovom slučaju pod nazivom *VelikiKupac* i na lokaciji koja je odabrana klasično iz menija *File, Save As*. Na taj način dobili smo fajl *VelikiKupac.dqy*. Nakon ovog neophodno je da upit *VelikiKupac.dqy* ostane otvoren, jer je jedino tada iz menija *File* dostupna opcija *Create OLAP cube...* čijim startovanjem se prvo pokreće uvodni ekran koji ovde nećemo prikazivati, već se odabirom *Next* prelazi u sledeći prozor koji najavljuje kreiranje *OLAP* kočke u tri koraka.

Slika 4.11 Izbor sumarnih polja



Izvor: Miroljub Zahorjanski

1. Korak - Izbor sumarnih polja

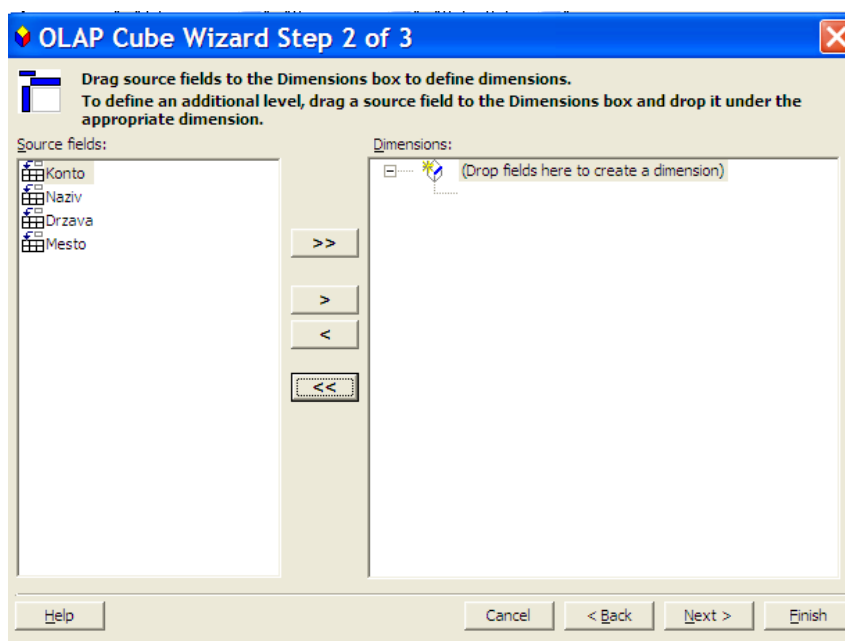
Prvi korak je izbor sumarnih polja i eventualnih polja za prebrojavanje. Dostupne funkcije su Sum, Count, Min i Max. Polja koja se ne koriste za sumarne podatke predstavljaju kandidate za dimenzije kočke. Kad ste zadovoljni izborom funkcija za polja koja će se zbrajati ili sumirati, treba pritisnuti *Next* za nastavak.

2. Korak - Definisiranje dimenzija OLAP kočke

Drugim korakom Slika 4.12 se praktično iz levog okna prebacuju u desno, ona polja koja predstavljaju dimenzije OLAP kočke i preferiraju da budu klizeći parametri po kojima će se upit dinamički izvršavati.

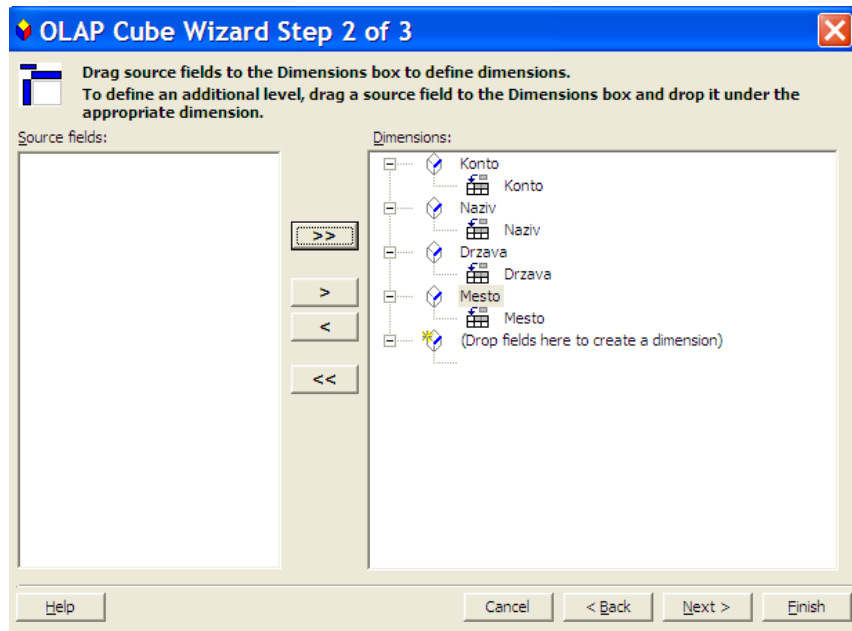
Nakon prebacivanja polja prizor bi trebalo da izgleda kao na Slici 4.13. Na njoj se vide slektovane dimenzije.

Slika 4.12 Izgled pre definisanja OLAP kočke



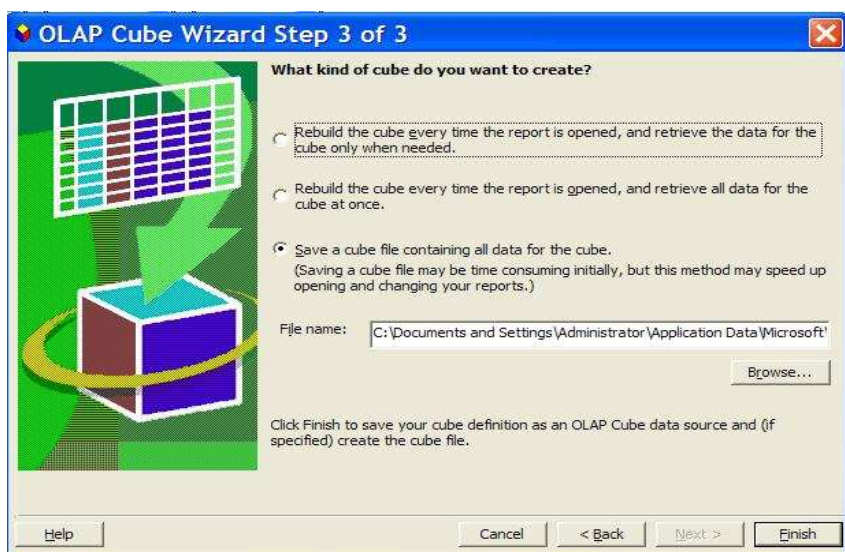
Izvor: Miroljub Zahorjanski

Slika 4.13 Izgled nakon definisanja OLAP kocke



Izvor: Miroljub Zahorjanski

Slika 4.14 Izbor vrste OLAP kocke



Izvor: Miroljub Zahorjanski

Prva opcija, odnosno prva vrsta *OLAP* kocke koja je ponuđena u izboru, nudi da ponovo sagradi *OLAP* kocku, svaki put kad se izveštaj otvori i da ponovo uzme podatke za *OLAP* kocku samo kada je potrebno.

Druga opcija nudi da ponovo sagradi *OLAP* kocku, svaki put kad se izveštaj otvori i da obavezno istovremeno uzme podatke za *OLAP* kocku.

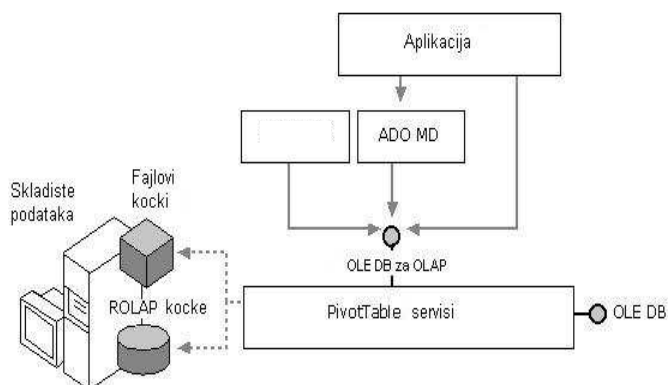
Treća opcija nudi i lokaciju za snimanje fajla sa ekstenzijom *.cub* i u prevodu znači "Snimate fajl *OLAP* kocke koja sadrži sve podatke za *OLAP* kocku. (Štedljivi *OLAP* fajl će možda trošiti početno vreme, ali ipak ova metoda može ubrzati otvaranje i izmenu vaših izveštaja)".

Za ovu priliku odabrali smo treću opciju i putem dugmeta *Browse...* odabrali putanju na kojoj će biti snimljen fajl *VelikiKupac.cub*. Nakon pritiska na dugme *Finish* program će od nas zatražiti da definišemo i putanju gde će biti snimljen fajl tipa *.oqy*, u ovom slučaju *VelikiKupac.oqy*. Nakon kratke animacije na kojoj je prikazano formiranje *OLAP* kocke na destinaciji koju smo odabrali trebalo bi da se nađu sva tri već pomenuta fajla: *VelikiKupac.dqy*, *VelikiKupac.cub* i *VelikiKupac.oqy*. Time su se stekli svi uslovi da se u narednim koracima posredstvom *MS Excel-a* vrši kreiranje pivot tabele za pristup podacima koje ta kocka sadrži.

4.3 Analiza podataka - MS EXCEL dinamičke tabele

U prethodnom delu dat je prikaz formiranja *OLAP* kocki koje sadrže pripremljene podatke za analizu. Na slici 4.15 su prikazana dva načina pristupa podacima u *OLAP* kockama: korišćenjem *Microsoft Excel-a* ili izradom posebne aplikacije, primenom takozvanih *ADO* mehanizama.

Slika 4.15 Pristup OLAP kockama

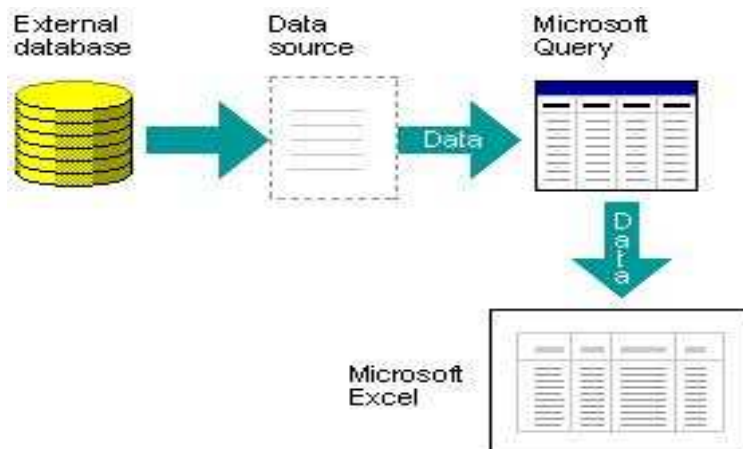


Izvor: Veljović A. Njeguš A., Osnove relacionih i analitičkih baza podataka, Megatrend univerzitet primenjenih nauka, Beograd, 2004.

U ovom radu, analiza podataka je izvršena primenom programa *Microsoft Excel*, tj. korišćenjem Pivot-tabela (*Pivot Table*). Pivot tabela predstavlja dinamičku tabelu sa objedinjenim podacima iz neke baze podataka. Ona služi za tabelarno prikazivanje više vrsta (dimenzija) podataka. U okviru nje se sumarni podaci mogu prikazivati na bilo kom nivou detaljnosti.

Korisnik ima mogućnost da direktno iz *Excel*-a vrši štampanje izveštaja za određeni pogled na podatke (izabrani nivo detaljnosti i raspored dimenzija).

Slika 4.16 Šematski prikaz puta od eksterne baze do *Excel* izveštaja



Izvor: *Microsoft Query Help*

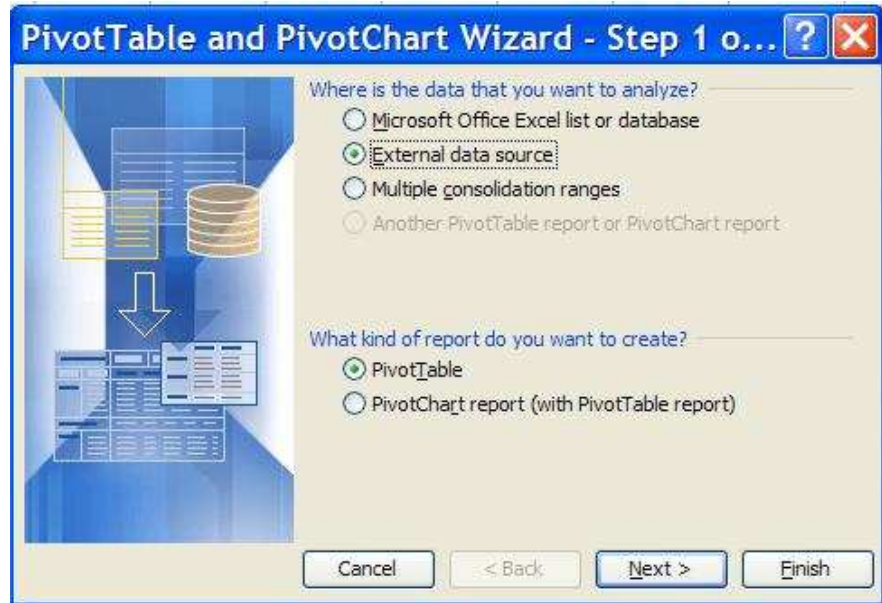
4.4 Formiranje Pivot tabele

Kreiranje Pivot-tabele se vrši preko menija *Data/Pivot Table and PivotChart Report*. Aktiviranjem ove opcije startuje se čarobnjak (*Wizard*) koji od nas, u tri koraka, traži da izvršimo izbor opcija za formiranje Pivot-tabele:

1. Izbor izvora podataka – u našem slučaju je to spoljni izvor podataka (*External data source*);
2. Izbor vrste i fajla sa podacima – u našem slučaju izbor *OLAP* kočke (*OLAP Cubes*);
3. Izbor pozicije pivot-tabele na *Excel*-ovom radnom listu.

1. Korak - Izbor izvora podataka

Slika 4.17 Izbor izvora podataka



Izvor: Miroљub Zahorjanski

2. Korak - Izbor vrste i fajla sa podacima

Nakon toga i pritiska na taster *Next* dolazimo do drugog koraka gde se od nas traži da izvršimo izbor vrste fajla sa podacima, u našem slučaju to je *VelikiKupac.cub*.

Slika 4.18 Međuprozor u drugom koraku



Izvor: Miroљub Zahorjanski

Pritiskom na taster *Get Data...* pojavljuje se prozor iz programa *MS Query Choose Data Source*, biramao treću karticu po redu *OLAP cubs*, a potom dugme *Browse*.

Slika 4.19 Prozir sa tri kartice za izbor vrste izvora podataka



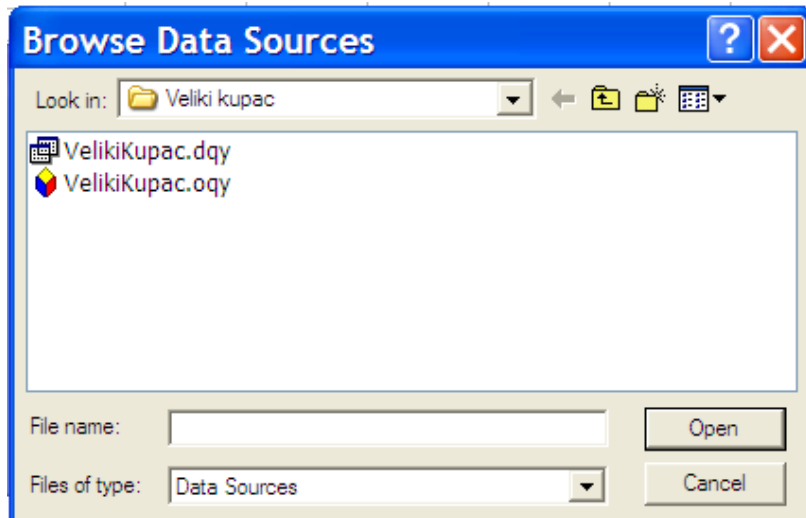
Izvor: Miroljub Zahorjanski

Sa slike 4.19 možemo videti da postoji izbor između tri različita tipa izvora podataka za analizu putem Pivot-tabela:

- *Databases* – izbor neke od ponuđenih relacionih baza podataka (npr. *MS Access*) sa skupom primitivnih podataka razvrstanih u tabele (normalizovani podaci);
- *Queries* – izbor upita formiranog u programu *MS Query* (denormalizovana tabela);
- *OLAP Cubes* – izbor fajla kocke sa definisanim sumirajućim podacima i kandidatima za dimenzije.

Pritiskom na dugme *Browse*, Prelazimo u prozor *Browse Data Sources*, gde nalazimo putanju do našeg fajla *VelikiKupac.cub*

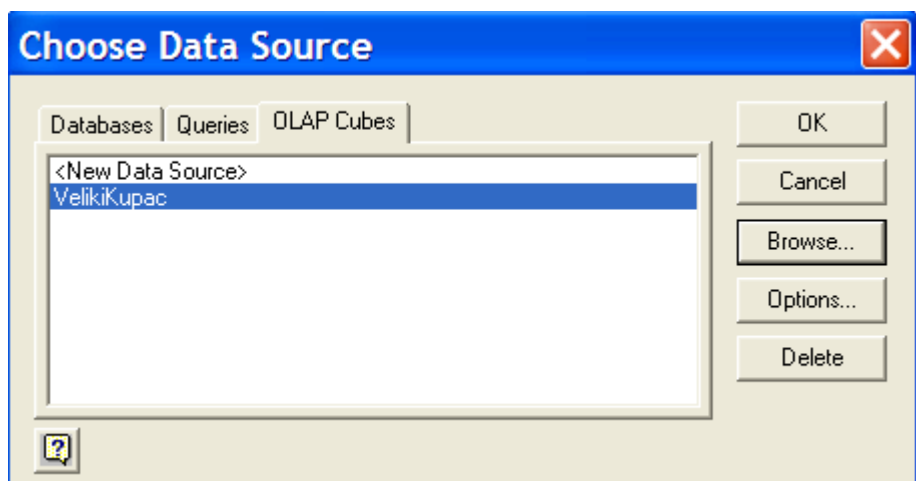
Slika 4.20 Prikaz prozora za izbor konkretnog fajla



Izvor: Miroljub Zahorjanski

Odabiramo fajl *VelikiKupac.cub* i pritiskamo *Open*. Tako se vraćamo u prethodni prozor *Choose Data Source* sa odabranim fajlom ekstenzije *.cub*.

Slika 4.21 Povratak na prethodni nivo



Izvor: Miroljub Zahorjanski

Pritiskom na *OK* vraćamo se u prozor *Pivot Table and PivotChart Wizard* odnosno da dovršimo korak 2. Izbor vrste i fajla sa podacima.

Slika 4.22 Povratak na međuprozor sa slike 4.18



Izvor: Miroljub Zahorjanski

Nakon pritiska na *Next* prelazimo na 3. korak .

3. Korak - Izbor pozicije pivot-tabele na *Excel*-ovom radnom listu

Slika 4.23 Prvi prozor trećeg koraka

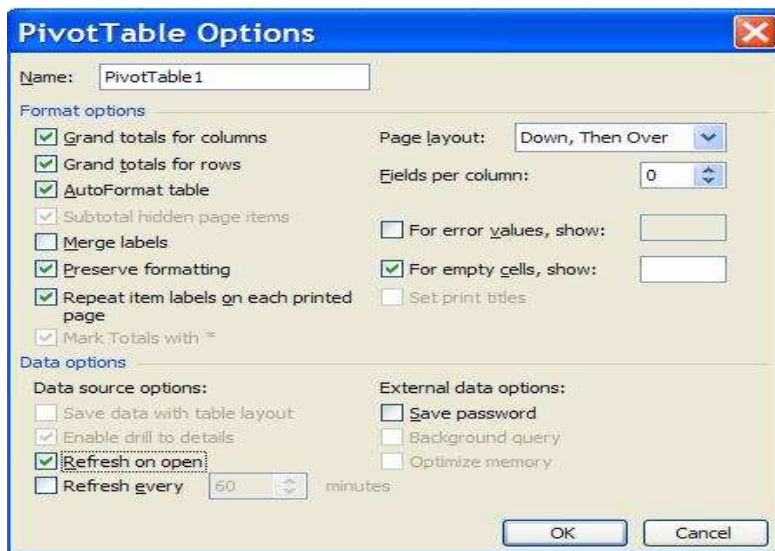


Izvor: Miroljub Zahorjanski

Da bismo se osigurali da će se kasnijim povezivanjem sa *Access*-om svakim ponovnim pokretanjem izveštaj redovno ažurirati shodno novim podacima iz tabele *.mdb* u prozoru *Pivot TableOptions*

do koje se dolazi pritiskom dugmeta *Options...* treba u području *Data source options*: čekirati *Refresh on open* i pritisnuti *OK*.

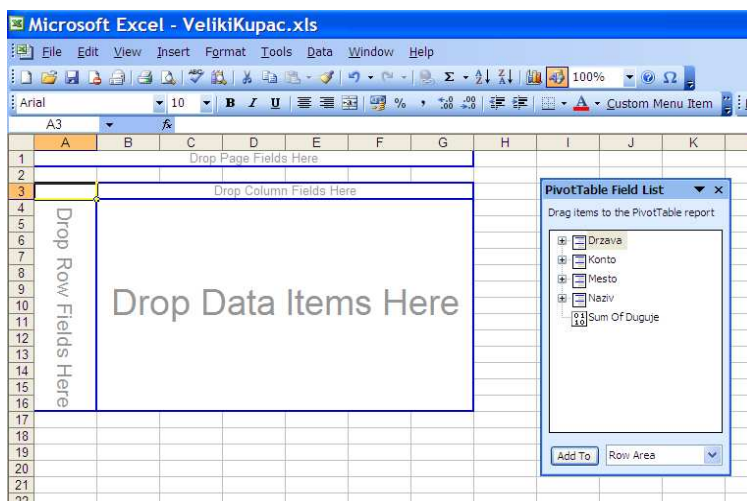
Slika 4.24 Podešavanje opcija za Pivot tabele



Izvor: Miroljub Zahorjanski

Povratkom na prethodni prozor, izborom nove ili postojeće *Excel*-ove strane i pritiskom na dugme *Finish* završavamo 3. korak, a time zatvaramo *PivotTable and PivotChart Wizard*.

Slika 4.25 Šablon za formiranje Pivot tabele sa osnovnim elementima



Izvor: Miroljub Zahorjanski

Po završetku rada, na ekranu dobijamo šablon za formiranje pivot tabele sa osnovnim elementima:

- *Page Fields* – dimenzija kocke (polja strane);
- *Column Fields* – dimenzija kocke (polja kolone);
- *Row Fields* – dimenzija kocke (polja reda);
- *Data Items* – sumirajući podaci.

Prevlačenjem iz oblasti *PivotTable Field List* recimo, kolone *Drzava* u polje strane (*Drop Page Fields Here*), kolone *Mesto* u polje kolone (*Drop Column Fields Here*), potom kolone *Naziv* u polje reda (*Drop Row Fields Here*) i na kraju sumarnog polja *OLAP* kocke *Sum of Duguje* na poziciju (*Drop Data Items Here*) dobićemo sledeću pivot tabelu kao na slici 4.26.

Dobijenu Pivot tabelu možemo podvrći različitim sintetičkim pogledima koristeći padajuće liste  sa podacima pored svake od dimenzija *OLAP* kocke.

Svako polje u pivot-tabeli ima na kraju kontrolno dugme-strelicu za brzi izbor konkretne vrednosti iz tog polja, slika 4.27. Time se, zajedno sa kombinacijom pozicije polja u tabeli, dobija mogućnost za formiranjem bilo koje pivot-tabele tj. moguće je dobiti svaku sintetičku informaciju na bazi raspoloživih kolona i instanci podataka.

Nakon formiranja željene pivot-tabele možemo je brzo formatizovati, tj. estetski i funkcionalno preurediti, opcijom *AutoFormat*, iz trake sa alatima, izborom nekog od ponuđenih formata.

Slika 4.26 Izgled Pivot table

Drzava	Mesto	B JARAK	B PALANKA	B CRKVA	B KARLOVAC
AGRO SAVIC	B JARAK	3233100.88	0		
ALTEX DOO	B JARAK				
BAMBI-BANAT A.D.	B JARAK				
BANATSKA SUBOTICA MZ	B JARAK				
DEMETRA STR.	B JARAK				706544357.8
DOSITEJ OBRADOVIC ETS	B JARAK	50471080			
ECOAGRI SERBIA	B JARAK			835742571.7	
FISHCORP 2000	B JARAK				
GRANICAR	B JARAK		7028000		
KRUSCICA ZZ	B JARAK				
MEDJUN.OEG ZA MIGRACIU	B JARAK				
MILENIJUM ZZ	B JARAK				
MOMA STR.	B JARAK				
NEDIM DOO	B JARAK	11626320			
NERA TRANS	B JARAK				
PIG FARM ZZ	B JARAK				
SWISSLION DOO	B JARAK				
VETA-BEL	B JARAK				
Grand Total		65330500.88	0	7028000	835742571.7

Izvor: Miroljub Zahorjanski

Ukoliko se želi dobiti neki analitički pogled na tabelu ili upit, potrebno je iz *MS Excel*-a pokrenuti *Data/Import; External Data/Import; Data....*, iz menija odrediti izvor podataka, kreirati analitički pogled, a potom pomoću opcije, takođe iz menija *Data/List/Create List....* svakoj koloni u analitičkom pogledu dodeliti padajuću listu.

Slika 4.27 Izgled analitičkog pogleda na tabelu ili upit

RedniBroj	Konto	Nalog	DatumDokumenta	Duguje	Potrazuje	Opis	SifraNaloga	NazivNaloga
3	Sort Ascending	2	3.1.2008	-230575.45		0	2	MK
9	Sort Descending	2	4.1.2008	-29155		0	3	MK
15	(All)	2	8.1.2008	-188099.5		0	4	MK
21	(Top 10...)	2	8.1.2008	-11603.8		0	5	MK
27	(Custom...)	2	9.1.2008	-91446.3		0	6	MK
33	02300	2	11.1.2008	-12564		0	7	MK
39	02301	2	11.1.2008	-43987.45		0	8	MK
45	02302	2	11.1.2008	-48804.82		0	9	MK
51	02305	2	12.1.2008	-51253.35		0	12	MK
57	02307	3	3.1.2008	-2237		0	2	MK
63	02390	3	4.1.2008	-19515		0	3	MK
69	02391	3	8.1.2008	-13042.5		0	4	MK
75	13100	4	4.1.2008	-29225		0	2	MK
81	13190	4	8.1.2008	-12698		0	3	MK
87	13401	4	8.1.2008	-6621.5		0	4	MK
94	98 13100	2	4.1.2008	0	31147.5	1	IF	IZLAZNA FAKTURA
99	101 13100	2	8.1.2008	0	1772	2	IF	IZLAZNA FAKTURA
102	101 13100	2	4.1.2008	97501	0	2	UF	ULAZNA FAKTURA
106	105 13100	2	8.1.2008	3080	0	6	UF	ULAZNA FAKTURA
109	108 13100	2	10.1.2008	35767.2	0	8	UF	ULAZNA FAKTURA
113	112 13100	2	10.1.2008	19305.6	0	9	UF	ULAZNA FAKTURA

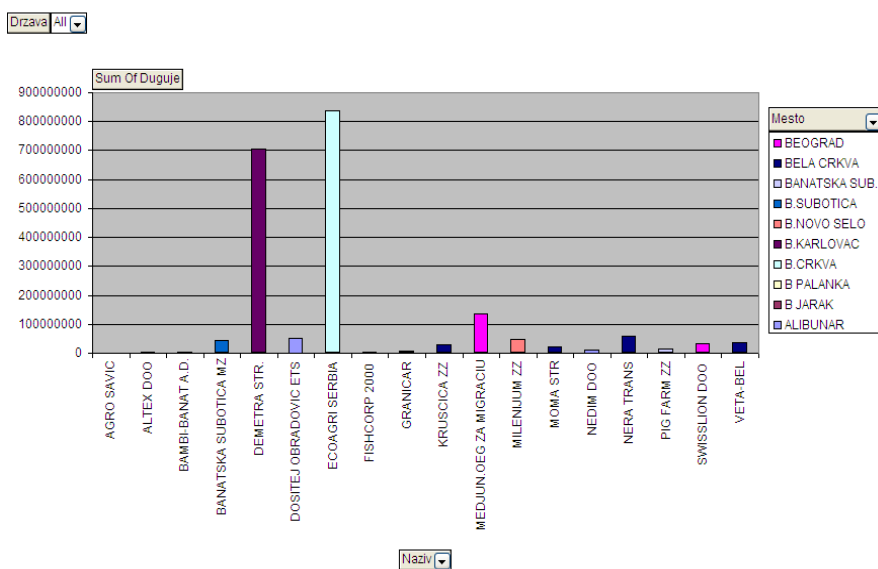
Izvor: Miroljub Zahorjanski

4.5 Formiranje Pivot grafikona

Grafički prikaz podataka iz tabele veoma često može na bolji način da pruži uvid u podatke i pomogne pri analizi podataka i donošenju zaključaka. U toku rada sa Pivot-tabelama preko *Pivot Table* trake sa alatima pritiskom na opciju *Chart Wizard* automatski dobijamo gotov grafikon, predefinisani na tip *Column*. Potom možemo lako da menjamo tip i podtip grafikona, da intervenišemo na samom grafikonu ispravljajući neke od elemenata grafikona, sve dok ne dobijemo konačni željeni izgled grafikona.

Najjednostavniji način da napravimo Pivot grafikon je da u otvorenom *Excel* fajlu selektujemo Pivot tabelu od koje želimo da kreiramo grafikon, a potom da sa Standard tulbara izaberemo ikonicu *Chart Wizard* ili iz menija *Insert* opciju *Chart...*

Slika 4.28 Izgled Pivot grafikona za slučaj Veliki kupac



Izvor: Miroljub Zahorjanski

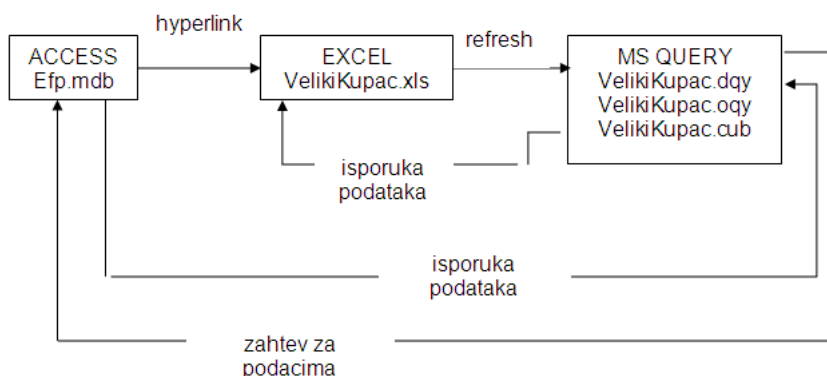
Na taj način dobijamo *Column* grafički prikaz podataka (slika 4.28) prethodno prikazanih u Pivot tabeli sa mogućnošću da po svakoj od dimenzija kocke ponovo kreiramo sintetički pogled bilo kojeg slučaja. To se čini jednostavnim čekiranjem iz ponuđenih padajućih

lista kao u slučaju *Naziv, Mesto* ili izborom iz stabla (konzole) kao u slučaju *Drzava*..

Svaki stubić na grafikonu prikazuje prodatu količinu određenog proizvoda sa učešćem kupca u ukupnoj prodaji u određenom geografskom području.

Mogućnosti, da se grafikon predefiniše tipovima i podtipovima kao što su: pite, linijski dijagrami ili trodimenzionalna geometrijska tela su velike. Na ovom mestu se nećemo detaljnije baviti tim opcijama, jer sama izrada mnogo zavisi i od ličnih afiniteta onog ko grafikon sačinjava ali ćemo opisati komunikaciju ovde pomenutih alata za izradu *OLAP* izveštaja.

Slika 4.29. Šema ostvarene komunikacije između prikazanih alata



Izvor: Miroljub Zahorjanski

Na slici 4.29 prikazano je kako *Access* kao izabrani SUBP poziva *Excel*-ov fajl putem hiperlinka, jer je predviđena mogućnost pozivanja iz *Access*-ovih padajućih menija. *Edit Hiperlink, Open*. Ukoliko *Excel* već poseduje podatke, generisaće izveštaj. U slučaju da se korisnik odluči za opciju *Refrsch*, šalje se zahtev *MS Query*-ju za obnavljanje podataka. *MS Query* šalje zahtev *Access*-u za novim podacima. *Access* vraća i isporučuje sveže podatke. Potom *MS Query* od poslanih podataka obnavlja *.dqy*, *.oqy* i *.cub* fajlove i šalje *Excel*-u novi *.cub* fajl od kojeg ovaj generiše obnovljeni izveštaj. Ciklus se ponavlja kad god korisnik odluči da osveži podatke za budući izveštaj, dok se u suprotnom dinamički izveštaj generiše od postojećih podataka.

Analiza slučaja “Veliki platiša“

Analogno ovom primeru veoma je lako pokazati analizu za situaciju koju bi smo mogli nazavati Veliki platiša. Filtriranje po kontu kupaca bi ostalo na snazi, kao i sve dimenzije. Jedino bi sumarni podaci bili iz kolone Potražuje, shodno objašnjenju da su velike platiše oni kod kojih ($\sum Potrazuje \rightarrow + \infty$) suma potražuje teži plus beskonačno.

Analiza slučaja “Dobar potrošač”

Ako želimo da uvedemo pojam Dobar potrošač i njega analiziramo, potrebno je prethodno definišemo taj pojam. U našem slučaju dobri potrošači bi bili oni partneri koji osim što vrednosno ostvaruju veliku kupovinu, redovno i blagovremeno izmiruju svoje obaveze. Tako da su Dobri potrošači oni kod kojih $Saldo = (\sum Duguje - \sum Potrazuje) \rightarrow 0$ ili saldo suma duguje manje suma potražuje teži nuli. Znači da su izmirili svoje obaveze. Ostali sa vrdnostima u (*Data Items Sum Of Saldo*) su dužnici.

Za izvršavanje ove analize, prethodno je bilo neophodno da u Access-u iz tabele činjenica (*Promene*) ekstrahujemo sve kupce sa uslovom (konto = 20000). To je urađeno sledećim upitom:

```
SELECT Promena.RedniBroj, Promena.Konto, Promena.Nalog,
       Promena.DatumDokumenta, Promena.Duguje,
       Promena.Potrazuje, Promena.Opis, Promena.SifraNaloga,
       Promena.NazivNaloga, Promena.Protivkonto,
       Promena.DatumKnjizenja, Promena.SifraPartnera,
       Promena.Mesto, Promena.Drzava, Left([konto],3) AS Kupci
FROM   Promena
WHERE  (((Left([konto],3))=200));
```

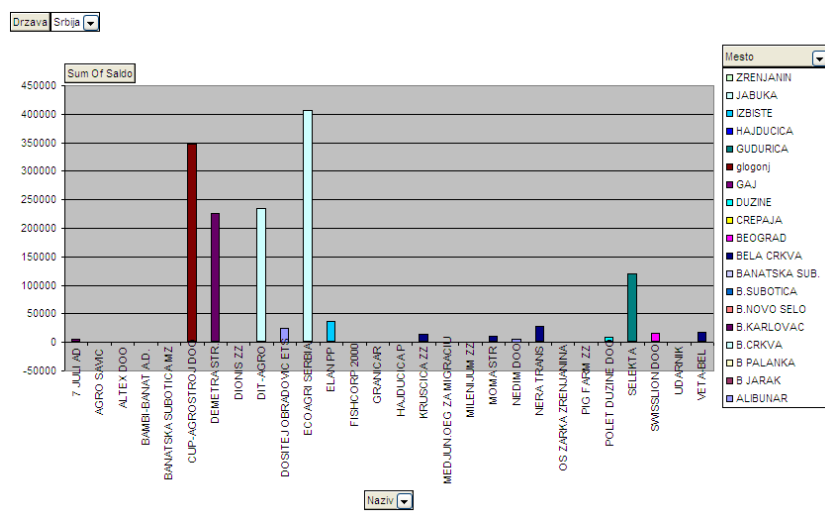
Potom je korišćen taj upit i tabela *Partner* da bi napravili saldo, odnosno (suma duguje - suma potrazuje) za odabranog kupca, pribegavajući u SQL jeziku agregacionim opcijama sumiranja u okviru grupisanja.

```
SELECT   Partner.Naziv, Sum(SviKupci.Duguje) AS SumOfDuguje,
         Sum(SviKupci.Potrazuje) AS SumOfPotrazuje, Sum([Duguje]-
         [Potrazuje]) AS Saldo
FROM     SviKupci INNER JOIN Partner ON SviKupci.SifraPartnera =
         Partner.SifraPartnera
GROUP BY Partner.Naziv;
```

Na kraju je taj upit napravljen u Access-u poslužio za pravljenje krajnjeg upita u alatu MSQuery u fajlu *DobarPotrosac.dqy* u relaciji sa tabelom *Partner*, koja je poslužila da bi se iz nje dobili dimenzioni podaci za naziv partnera, državu i mesto.

Postupak oko izgradnje Pivot table i Pivot grafikona je tehnički gledano potpuno identičan i nema ga svrhe ovde eksplicitno ponavljati. Rezultat jedne takve analitičke “mašine” prikazan je na sledećoj slici 4.30.

Slika 4.30 Izgled Pivot grafikona za slučaj Dobar potrošač



Izvor: Miroljub Zahorjanski

4.6 Izrada korisničkog interfejsa

Prezentacioni sloj je deo poslovne aplikacije, koji obezbeđuje komunikacioni mehanizam između korisnika i sloja servisa poslova u aplikaciji. Prezentacioni sloj sadrži komponente potrebne za omogućavanje interakcije korisnika i aplikacije. Mnogi primeri prezentacionih slojeva sadrže korisničke interfejs komponente, slične *Windows* formama ili *ASP.NET Web* formama.

Korisnički interfejs je veoma važna komponenta bilo kog tipa poslovnog softvera pogotovu u oblasti pružanja podrške u odlučivanju, a u sklopu toga i podsistemima za vođenje finansijskog poslovanja preduzeća. Od interfejsa se zahteva da bude lak za upotrebu i poseduje osobinu *user friendly* (prijateljski nastrojen prema

korisniku). Ovo u principu znači “Lako se uči i lako se koristi“. Ukoliko su ove osobine zastupljenije, veće su šanse da će korisnik koristiti predpostavljeni softver.

“Efikasan interfejs minimizuje vreme koje je korisnicima potrebno da sistem savladaju i počnu da ga koriste. Nakon uvođenja sistema u redovnu upotrebu, produktivnost korisnika biće veća ako ne moraju da se bore da bi sistemu nametnuli svoju volju.”^[84]

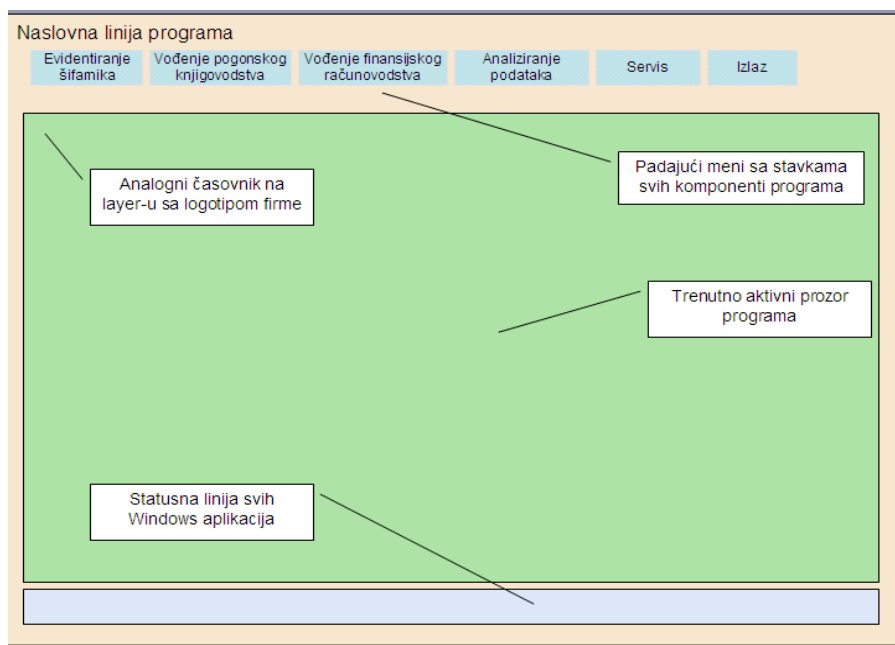
Ono što korisnik misli da se događa je u stvari korisnikov mentalni model i on se razlikuje od onog što se stvarno događa i što predstavlja stvarni model. Negde na sredini, odnosno između ova dva modela nalazi se korisnički interfejs ili prividni model. Za korisnički interfejs bi se moglo reći da je to model procesa kako ga sistem prikazuje korisniku (odnosno kako ga korisnik vidi). Idealan korisnički interfejs treba da se što više poklapa sa korisnikovim mentalnim modelom i da ima osobinu intuitivnosti. Značenje ove reči u psihologiji je predosećaj, domišljaj, instiktivno shvatanje. Kada interfejs dosegne ovu crtu intuitivnosti onda se sa sigurnošću može tvrditi da je ako ne idealan, onda bar veoma blizu toga.

4.6.1 Korisnički interfejs transakcione baze podataka

Za transakcionu bazu podataka sačinjen je korisnički interfejs koji u suštini odslikava grupisane procese rada finansijskog knjigovođe. Najčešći tip objekata u većini SUBP^[19] koji su namenjeni oblasti vođenja finansijskog poslovanja preduzeća jesu ekranske forme i izveštaji.

Niža preciznost dizajna ilustruje glavne osobine, strukturu i navigaciju, dok visoka preciznost dizajna ilustruje detalje izgleda ekrana i interfejs elemente. Manje precizan dizajn dopušta brzo i lako ispitivanje alternativnih dizajna, dok precizniji dizajn, mada brz za obezbeđivanje promena zahteva računar i softver. Manje precizno dizajniranje je odlično za inenadnu ideju i brz povratak na staro, dok je preciznije dizajniranje tipično za razvoj iz manje preciznog dizajna.

Slika 4.31. Izgled ekrana nakon startovanja programa (precizniji dizajn)



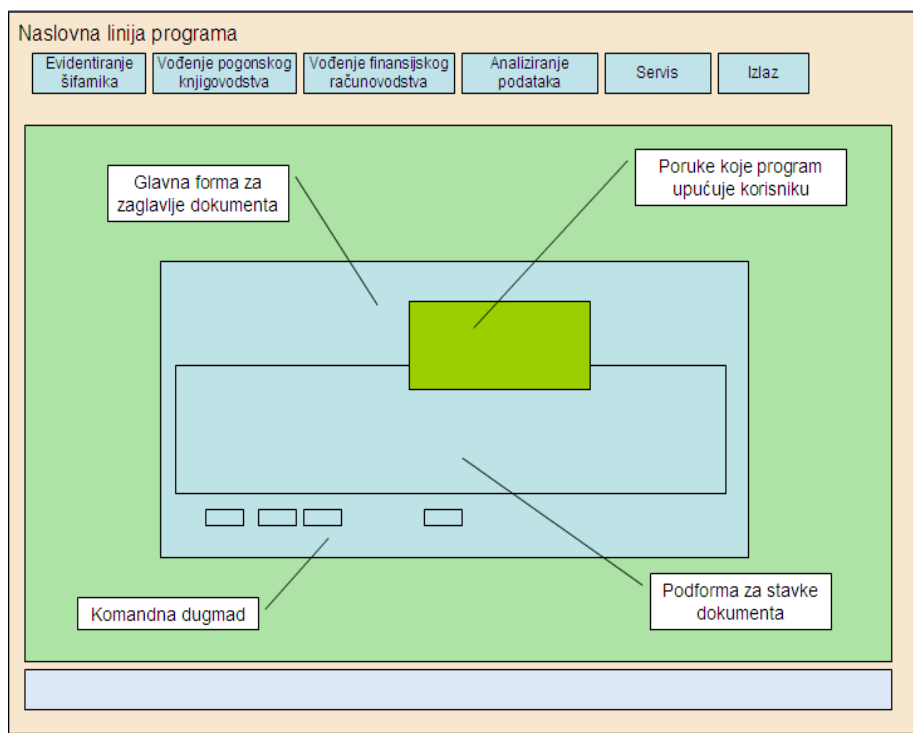
Izvor: Miroljub Zahorjanski

U postupku izrade prezentacionog sloja manje preciznosti u ovom projektu korišćeni su papir i olovka, da bi se potom iskristalisane ideje realizovale pomoću računara i nekog programa za crtanje i prezentovanje, u ovom slučaju *MS Power Point-a*.

Na slici se jasno vidi da su padajući meniji na vrhu strane odmah ispod naslovne linije, dok najveći deo ekrana između menija i statusne linije na dnu zauzima deo namenjen za postavljanje podloge sa logotipom firme preko koje će se smenjivati svi aktivni prozori formi i izveštaja koje će program sadržati.

Obzirom da knjigovodstveni podaci firme sadrže manje više i neke poslovne tajne za širu javnost, kao što su to kupci i dobavljači, pa i količine i cene roba, odlučeno je da aplikacija bude isključivo orjentisana prema intranetu ili još bolje rečeno ograničena na lokalnu mrežu. Iz tog razloga se odustalo od izrade *Online helpa*, koji je toliko popularan u zadnje vreme pri izradi aplikacija. S druge strane omogućiti će se korisniku nekoliko drugih načina za upoznavanje sa programom i podrška prilikom eksploatacije.

Slika 4.32 Prikaz najčešćih komunikacija programa i korisnika

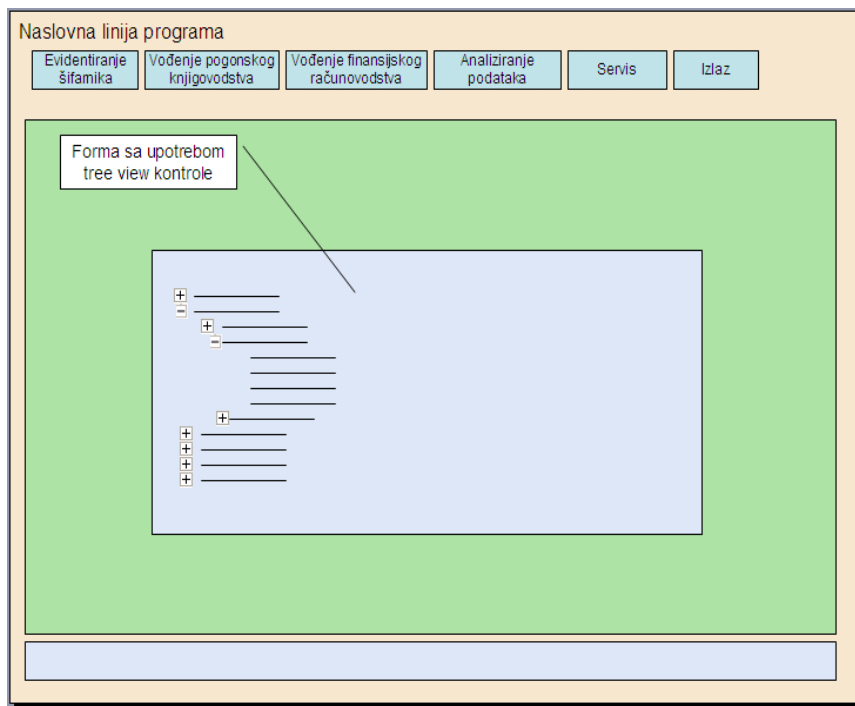


Izvor: Miroљjub Zahorjanski

Jedna od tih podrški jesu takozvani *MsgBox*-ovi ili kutije za poruke. Program je u konceptualnoj fazi zamišljen sa potpunom proverom podataka koji se unose i odgovori na validaciju podataka upravo predstavljaju ove poruke programa. Te iskačuće poruke su smeštene u centralni deo ekrana i prekrivaju zadnji aktivni prozor samo dok ih korisnik jednim klikom ne ukloni. Ovo povećava lagodnost u upravljanju aplikacijom i doprinosi opuštenijem odnosu korisnika kada je siguran da su uneti podaci validni.

Za najveći broj šifarnika predviđen je tabelarni način unosa i ažuriranja, jer su šifarnici sami po sebi tako organizovani podaci da ih je najjednostavnije u određenom redosledu posmatrati unutar tabele. Kod šifarnika *KontniPlan*, uvedena je novina i on će biti odrađen putem *Tree view* kontrole, koja podrazumeva podatke sa nadređenim i podređenim nivoima do zahtevane dubine.

Slika 4.33 Izgled buduće forme za kontni plan



Izvor: Miroљjub Zahorjanski

Korisnički interfejs je *Windows* baziran u globalu gledajući, što i nije toliko nelogično, obzirom da je pisan za taj operativni sistem. Komponente korisničkog interfejsa čine da korisnik aplikacije bude više interaktivan. Korisničke procesne komponente sažimaju specifične procese koje bi korisnik bio sposoban da izvede u interakciji sa aplikacijom.

Osim kod kontnog plana *Tree view* kontrola je upotrebljena i kod izrade forme "Help sadržaj", gde je na istovetan način vodeći računa o nadređenosti i podređenosti određenih stavki padajućeg menija, postignuta laka pretraga koja u osnovi ima kontekst ideju.

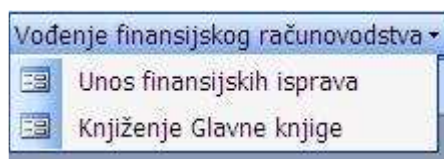
Slika 4.34 Izgled dela menija koji sadrži forme za evidentiranje šifarnika



Izvor: Miroljub Zahorjanski

Kada je u pitanju vođenje finansijskog poslovanja preduzeća i pogled na interfejs sa aspekta transakcione baze podataka, onda je održavanje šifarnika jedna od važnih aktivnosti koje korisnik obavlja. Druga i savkako najvažnia aktivnost je unošnje finansijskih isprava i njihovo knjiženje.

Slika 4.35 Izgled dela menija koji sadrži forme za vođenje finansijskog knjigovodstva



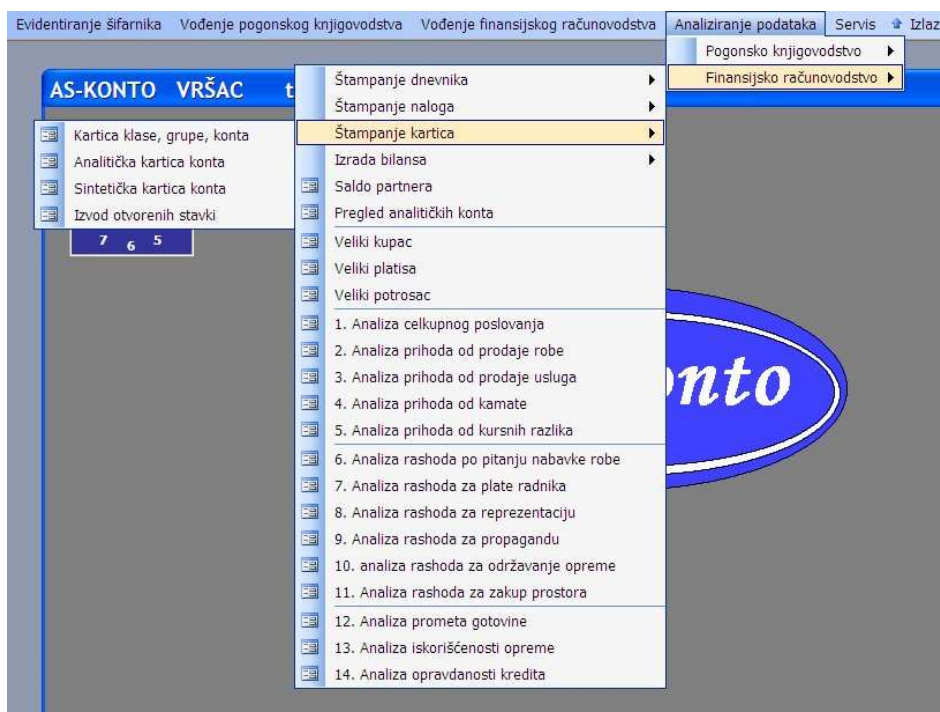
Izvor: Miroljub Zahorjanski

4.6.2 Korisnički interfejs OLAP

Osnovna funkcija *OLAP* korisničkog interfejsa je da omogući da se za unapred formiranu kocku koja u sprezi sa pivot tabelama sačinjenim u *Excel*-u omogući konfor za dobijanje grafičkih izlaza na osnovu kojih će rukovodstvo donositi poslovne odluke.

Šema ostvarene komunikacije između *Access*-a i *Excel*-a prikazana je ranije na slici 4.29. Konfor ili udobnost u radu postignuta je tako što svaka stavka menija za analiziranje podataka vodi pomoću *Hyperlinka* do adekvatnog *Excel* fajla tipa *.xls*.

Slika 4.36 Izgled dela menija namenjen OLAP-u



Izvor: Miroљjub Zahorjanski

Stavke u meniju su organizovane tematski i adekvatno grupisane, tako da je menadžeru-korisniku nakon prvog susreta sa delom interfejsa namenjenog OLAP-u, jasno koje radnje treba da preduzme kako bi došao do odgovora na željena pitanja.

Korisnik ima mogućnost da u određenoj analizi koristi svu svoju kreativnost i da od ponuđenih dimenzionih i činjeničnih elemenata, postavi kriterijum za filtriranje podataka, što konačno rezultuje tabelarnim prikazom "prosejanih" podataka, odnosno njihovim grafičkim prikazom.

Ovde postoji mogućnost prikazivanja više vrsta-dimenzija podataka i u okviru toga se sumarni podaci mogu prikazivati na bilo kom nivou detaljnosti.

4.7 Implementacija u MS SQL server 2005

Da bismo razumeli tok implemetacije u *MS SQL Server 2005* moramo prethodno pojasniti sve elemente takve implemetacije koja se u mnogo čemu razlikuje od implementacije u već prikazanim alatima. Težište je ovde na umreženoj pozadinskoj bazi podataka koja za osnovu ima skladište podataka.^[4] Ovo skladište podataka smo preuzeli na kraju procesa objektno orjentisanog dizajna opisanog u odeljku 3.3 pod nazivom Izrada dijagrama klasa.

Dizajniranje biznis intelegentne aplikacije u *MS SQL Server 2005* započinje kreiranjem projekta u alatu *Business Intelligence Development Studio*. Unutar tog projekta definišemo sve elemente, započevši sa kreiranjem izvora podataka za upite. Pošto smo definisali izvore podataka, definišemo i inforamacije vezane za konekciju ka izvoru podadaka.

Nakon što smo definisali izvore podataka koje ćemo koristiti unutar *Microsoft SQL Server 2005 Analysis Services* projekta, sledeći korak je da generalno definišemo pogled na izvor podataka za projekat. Pogled na izvor podataka je jedinstven unificiran pogled na metapodatke za specificirane tabele i poglede koje izvor podataka definiše u projektu. Snimajući metapodtake u pogledu na izvor podataka (*data source view*), omogućuje nam da radimo sa metapodacima za vreme razvoja bez otvaranja konekcije sa bilo kojim izvorom podataka.

Business Intelligence Development Studio koristi metapodatke za objekte u pogledu na izvore podataka kako bi definisao dimenzije, attribute, i odvojene grupe. Međutim, ovaj alat koristi takozvani *FriendlyName property* za objekte umesto *Name property*. Ovde možemo da promenimo vrednost osobine za objekte unutar pogleda na izvore podataka, kako bi povećali *user-friendliness* (prijateljsku raspoloženost prema korisniku) za imena, dimenzije, i *cube* objekte koji su kreirani iz pogleda na izvore podataka.

Definisanje i razvijanje kocke

Sledeći korak je definisanje i razvijanje kocke. Nakon definisanja upita izvora podataka u projektu, prelazimo na definisanje inicijalne kocke. Može se početi sa definisanjem dimenzija bilo koje kocke, a onda definisati jednu ili više kocki, koje koriste te dimenzije. U našem slučaju kao i pređašnjim imple-mentacionim primerima definisaćemo samo jednu kocku. Ako bismo planirali neku složeniju

soluciju višestrukih kocki koje dele zajedničku bazu dimenzija, generalno bismo počinjali definisanjem dimenzija na nivou baze podataka. Ovo se naziva *database dimensions*. Potom definišemo svaku kocku u soluciji, onako kako će svaka baza dimenzija biti korišćena za svaku pojedinačnu kocku. Ovo se naziva dimenzijama kocke (*cube dimensions*). Jedna dimenzija iz baze može se koristiti kao osnova dimenzija višestrukih kocki.

Kada definišemo kocku koja je zasnovana na već postojećem izvoru podataka, *Wizard* vrši konekciju sa bazom podataka, koja je definisana kao izvor podataka, i čita podatke u specifičnim tabelama i pomaže nam da definišemo veličine i dimenzije. Veličine i dimenzije koje smo definisali su smeštene u tabelama, koje su identifikovane kao stvarne tabele, kao tabele dimenzija, ili i jedno i drugo. Kada koristimo ovaj metod možemo dozvoliti opcije *Auto bild* i *Cube Wizard* (*Wizard* za pravljenje kocke), koje će tada automatski definisati attribute iz kolona u tabele dimenzija. Opciono, možemo imati da *Wizard* automatski pokuša da kreira više nivovsku hijerarhiju. Ako onemogućimo *Auto bild*, možemo kreirati attribute i hijerarhiju ručno u *Cube Wizard*, ili ih možete kreirati kasnije u *Cube Designer*.

Primena *Analysis Services* projekta

Da bismo videli podatke koji su u projektu, moramo dodeliti projekat posebnoj instanci *Analysis Services* i tek onda obraditi kocku i njene dimenzije. Implementiranje *Analysis Services* projekta kreira definisani objekat u instanci *Analysis Services*. Procesiranje objekata u instanci *Analysis Services* kopira podatke iz osnovnih izvora podataka u objekte kocke.

U ovom trenutku u procesu razvoja, razvijamo kocku u instancu *Analysis Services* na razvojnom serveru. Kada završimo razvoj našeg projekta, generalno ćemo koristiti *Analysis Services Deployment Wizard* da biste ga postavili na server.

Pretraživanje razvijene kocke

Pretraživanje razvijene kocke nam pomaže da razumemo promene koje smo trebali napraviti da bismo povećali funkcionalnost kocke. Na primer, možda bismo trebali da definišemo redove sortiranja elemenata dimenzija, obrišemo nepotrebne atribite dimenzija, definišemo nove hijerarhije korisnika, modfikujemo postojeće hijerarhije korisnika, ili da konfiguriramo attribute mernih jedinica. Pošto razvijamo kocku, njeni podaci se mogu videti na *Browser tabu* u *Cube Designer*, i podaci o dimenzijama se mogu videti na *Browser tabu* u *Dimension Designer*.

Modifikovanje mera, atributa i hijerarhija

Posle definisanja, razvoja i procesiranja neše inicijalne kocke, zatim ponovnog pregledanja dimenzija i podataka kocke u *Business Intelligence Development Studio*, spremni smo da uvedemo takozvanu korisnost i poboljšanje kocke. Možemo takođe povećati karakteristiku *user-friendly* mera kocke korišćenjem *FormatString* osobine za svaku meru kako bi definisali podešavanja formatiranja koja kontrolišu kako su prikazane mere u klijent aplikacijama.

Postoji mnogo načina da se povećaju *user-friendly* osobine i funkcionalnost dimenzija kocke. U konkretnom primeru ovog istraživačkog rada modifikovana je dimenzija *“Period”* promenom atributa i imena hijerarhije, promenom korisničkih osobina hijerarhije definisanjem atributa dimenzije baziranih na novijim imenovanim proračunima u *data source* pregledu, brisanjem nepotrebnih atributa i grupisanjem atributa u logične korisničke hijerarhije. Zatim možemo razviti ove promene, procesirati modifikovanje objekata i učitati dimenziju da bi smo pogledali promene. Takođe možemo imenovati proračun, koji je u suštini *SQL* izraz i koji je predstavljen kao proračunata kolona, u tabeli u *data source* pogledu. Izraz se pojavljuje kao kolona u tabeli. Kada kreiramo ime proračuna, opisujemo ime i *SQL* izraza. Imenovanjem proračuna proširujemo šemu veza postojećih tabela u *data source* pogledu, bez modifikovanja tabele u osnovnom izvoru podataka.

Pošto smo kreirali imenovani proračun u *data source* pogledu, možemo koristiti imenovani proračun kao osobinu atributa, tako što osobina imena kolone povećava korisnost atributa pravljnjenjem njegovog imena više prijateljskim. Korišćenjem *display foldera* za grupne korisnike i atribut hijerarhiju unutar strukture foldera takođe se povećava prijateljska raspoloženost prema korisniku, pri učitavanju dimenzija kocke.

Modifikacija vremenske dimenzije u tabeli *“Period”* takođe je moguća definisanjem složenih ključeva za attribute da bi kontrolisali članove koji su vraćeni od *Select Distinct* procesirajućeg upita. Kada koristimo složene ključeve, poredak sortiranja članova atributa je određen redom članova složenog ključa.

Sakrivanje i isključivanje hijerarhije atributa

Standardno, u *Microsoft SQL Server 2005 Analysis Services*, hijerarhija atributa se kreira za svaki svaki atribut u dimenziji, i svaka hijerarhija je dostupna za dimenzionisanje činjeničnih podataka. Kao što je već napomenuto, možemo organizovati attribute u korisničke

hijerarhije da bi obezbedili navigacionu putanju u kocki. Pod određenim okolnostima, korisnik možete poželeti da sakrije ili isključi pojedine attribute i njihove hijerarhije. Na primer, odeređeni atributi tabele “*Partner*” kao što su adresa, telefon, faks, ili žiro-račun nisu atributi na osnovu kojih će korisnici dimenzionisati informacionu kocku. Umesto toga, oni se generalno posmatraju kao detalji određenog člana atributa. Ove hijerarhije atributa možemo da sakrijemo, ostavljajući attribute vidljivim samo kao delove svojstava određenog atributa. Takođe možemo poželeti da napravimo članove drugih atributa^[117,122,124] kao što su ime osobe za kontakt pojedinog partnera ili poštanski kod, vidljivim samo kada su oni posmatrani kroz hijerariju korisnika umesto kroz hijerarhiju atributa. Jedan od razloga da se to uradi može biti mali broj različitih članova u hijerarhiji atributa. Ovo, dakako ima veze sa blagovremenim logičkim modelovanjem i na vreme pripremljenim mogućim podacima iz početnog *OLTP* sistema.

Sotiranje članova atributa na osnovu sekundarnih atributa

Ponekad je potrebno da članove atributa uredimo na osnovu nekog sekundarnog atributa da bi smo postigli željeno uređenje, na primer ako ni ime ni ključ atributa ne obezbeđuju uređenje koje želimo. Da bi smo uredili atribut na osnovu imena ili ključa sekundarnog atributa, moramo upotrebiti sekundarni atribut s kojim je on u vezi.

Veze atributa određuju veze ili zavisnosti između atributa. U dimenziji koja je zasnovana na samo jednoj tabeli, svi atributi su povezani međusobno preko atributa ključa. Zbog toga svi atributi obezbeđuju dimenziji informacije o članovima povezanim preko atributa ključa dimenzije. U dimenziji koja je zasnovana na više tabela, atributi su obično povezani na osnovu združenog ključa tabela.

Ipak, korisnike *OLAP* aplikacije EKONOMSKO FINANSIJSKIH POSLOVA takođe mogu interesovati dodatne informacije o članovima na određenom nivou u hijerarhiji. Dizajner dimenzija nam dozvoljava da definišemo dodatne relacije između atributa, ili da promenimo podrazumevane relacije s ciljem poboljšanja performansi. Glavno ograničenje kada kreiramo vezu između atributa je da taj atribut ima samo jednu vrednost za bilo kog člana u atributu sa kojim je povezan. Kada kreiramo vezu između dva atributa, možemo je definisati kao rigidnu ili kao fleksibilnu, u zavisnosti od toga da li će se veza menjati u budućnosti. Na primer, partner se može preseliti u drugi grad, ali se grad neće preseliti u drugu državu. Ako je veza definisana kao rigidna, onda se agregacije atributa ne računaju svaki put kad se dimenzija

inkrementalno procesira. Ali, ako se veza između atributa menja, dimenzija se mora svaki put potpuno procesirati.

Korisnička hijerarhija može predstavljati prirodnu hijerarhiju, kao grad, država, zemlja, ili može samo predstavljati navigacijski deo, kao naziv partera i naziv odeljenja.^[30] Korisnička navigacija i hijerarhija, ta dva tipa korisničkih hijerarhija su skoro iste. Sa prirodnom hijerarhijom, ako definišemo attribute veza između atributa koji podižu nivo, *MSSQL Analysis Services* može koristiti agregaciju od jednog atributa da dobije rezultate od povezanih atributa. Ako postoje nedefinisane veze između atributa, ovaj alat će generisati sve neključne attribute od ključnih atributa.

Definisanje stvarnih veza

Korisnici ponekad žele da budu u mogućnosti da mere dimenzije podacima koji su u stvarnoj tabeli ili da ispitaju stvarnu tabelu za određene dodatno povezane informacije, kao što je spisak partnera ili datumi dokumenta povezanih sa određenim faktorima prodaje. Kada definišmo dimenzije bazirane na takvim pojedinačnim stvarnim tabelama, dimenzija se naziva *fact dimension*. *Fact dimensions* su korisne za grupisanje međusobno povezanih redova stvarnih tabela, kao i svi redovi koji se tiču detaljnog izveštaja partnera. Iako ovu informaciju možemo staviti u odvojenu dimenzionu tabelu, u relacionu bazu podataka, stvarajući odvojenu dimenzionu tabelu za informaciju, ona ne omogućava korist zbog toga što će ta tabela rasti istom brzinom kao i stvarna tabela, i samo će stvoriti duplikat podatka i nepotrebne komplikacije.

Sa *Microsoft SQL Server 2005 Analysis Services (SSAS)*, možemo odrediti bilo duplikat stvarne dimenzije podataka u *MOLAP* dimenzionoj strukturi za povećano ispitivanje performansi, ili da definišemo stvarnu dimenziju kao *ROLAP* dimenziju da bi sačuvali prostor o trošku za ispitivanje performansi. Kada smestimo dimenziju sa *MOLAP* u modu za skladištenje, svi članovi dimenzije su smešteni u primeru *Analysis Services* u visoko kompresovanoj *MOLAP* strukturi. Kada smestimo dimenziju u *ROLAP* modu za skladištenje, samo je definicija dimenzije smeštena u *MOLAP* strukturi, a sami članovi dimenzije su ispitani u osnovnoj dimenzionoj tabeli u upitnom vremenu. Definisanje dimenzije kao što *ROLAP* radi ne podrazumeva da će sve kocke koje su korišćene za dimenziju takođe biti smeštene pomoću *ROLAP* moda za skladištenje. Ovo je drugačije od *SQL Server 2000 Analysis Services*.

Kada definišemo stvarnu dimenziju, možemo definisati vezu između stvarne dimenzije i merne grupe kao stvarnu vezu. Sledeća ograničenja odnose se na stvarne veze:

-
- Zrnasti atributi moraju biti ključni stub za dimenziju, a to stvara *one-to-one* vezu između dimenzije i činjenica u stvarnoj tabeli.
 - Dimenzija može imati stvarnu vezu samo sa jednom mernom grupom.
 - Merna grupa može imati stvarnu vezu samo sa jednom dimenzijom.

Definisanje veze više prema više (*Many-to-Many*)

Kada definišemo dimenziju, tipično svaka činjenica se pridružuje jednom i samo jednom dimenzionom broju, s obzirom na to da jednodimenzioni član može biti udružen sa velikim brojem različitih činjenica. Na primer svaki partner može imati mnogo promena. U terminologiji relacionih baza podataka, ovo je predstavljeno kao veza više-više.

U *Microsoft SQL Server 2005 Analysis Services (SSAS)*, definišemo vezu više-više između veličina i grupe mera specificiranjem među-činjenične tabele koja je pridružena dimenzionoj tabeli. Među-činjenična tabela je pridružena, redom, među-dimenzionoj tabeli kojoj je u stvari činjenična tabela pridružena.

Definisanje dimenzione granularnosti unutar merne grupe

Korisnici će želeći dimenzionisanje različitih činjenica podataka različitih granularnosti i specifikacija za različite svrhe. Na primer, u *OLTP* sistemu, prodaja može biti zapisivana svakog dana, s obzirom na to da informacija o prodaji može postojati kratko vreme. Po već nekoj utvrđenoj vrednosti u *Microsoft SQL Server 2005 Analysis Services (SSAS)*, kada je dimenzija korišćena unutar merne grupe, parčence podataka bez dimenzije je bazirano na ključevima atributa dimenzije. Da bi dali detaljan opis zrna za kub dimenziju drugačijim od već utvrđene vrednosti zrna, moramo modifikovati zrnasti atribut za kub dimenziju koji je korišćen unutar merne grupe na *Dimension Usage tab-u Cub Dizajnera*. Kada promenimo zrno dimenzije bez specifične merne grupe jednog atributa drugačijeg od ključ atributa te dimenzije, moramo garantovati da su svi ostali atributi u mernoj grupi direktno ili indirektno povezani sa novim granularnim atributom. To će biti urađeno specificiranjem atributske veze između svih ostalih atributa i atributa koji je specificiran kao granularni atribut u mernoj grupi. Atribut koji je specificiran kao granularni atribut efektivno postaje ključ atribut unutar merne grupe. Ako ne specificiramo atributske

veze prikladno, *Analysis Services* neće biti u mogućnosti da korektno sakupi vrednosti.

Definisanje kalkulisanih članova

Kalkulisani članovi su individualno određene mere ili dimenzije članovi, koji su definisani na osnovu kombinacije podataka iz kocke, aritmetičkih operacija, brojeva, i funkcija. Na primer, možemo napraviti kalkulisani član koji izračunava sumu dve fizičke mere iz kocke. Definicije kalkulisanih članova se smeštaju u kocke, ali njihove vrednosti se računaju u vreme izvršavanja. Međutim, za razliku od *Microsoft SQL Server 2000 Analysis Services*, u *SQL Server 2005 Analysis Services (SSAS)* izračunate vrednosti se keširaju i dostupne su ostalim korisnicima tek pošto su izračunate.

Primena *Analysis Services* projekta

Da bismo videli podatke koji su u kocki u *Analysis Services Tutorial* projektu, moramo dodeliti projekat posebnoj instanci *Analysis Services* i tek onda obraditi kocku i njene dimenzije. Implementiranje *Analysis Services* projekta kreira definisani objekat u instanci *Analysis Services*. Procesiranjem objekata u instanci, *Analysis Services* kopira podatke iz osnovnih izvora podataka u objekte kocke.

U ovom trenutku u procesu razvoja, razvijamo kocku u instancu *Analysis Services* na razvojnom serveru. Kada smo završili razvoj našeg *business intelligence* projekta, generalno ćemo koristiti *Analysis Services Deployment Wizard* da bismo ga postavili na server.

Pretraživanje razvijene kocke

Pretraživanje razvijene kocke nam pomaže da razumemo promene koje smo trebali napraviti da bismo povećali funkcionalnost kocke. Na primer, možda bismo trebali da definišemo redove sortiranja elemenata dimenzija, obrišemo nepotrebne atribite dimenzija, definišemo nove hijerarhije korisnika, modifikujemo postojeće hijerarhije korisnika, ili da konfigurišemo attribute mernih jedinica. Pošto razvijemo kocku, podaci kocke se mogu videti na *Browser tabu* u *Cube Designer*-u, i podaci o dimenzijama se mogu videti na *Browser tabu* u *Dimension Designer*-u.

Definisanje i listanje perspektiva

Ono što je za krajnje korisnike analitičare interesantno, je svakako mogućnost da predvide perspektivu i trendove na osnovu

kavliteno osmišljenih izveštaja. Perspektive mogu da pojednostave poglede specijalne namene. Standardno, korisnici mogu da vide sve elemente za koje imaju ovlašćenja. Ovo standardno svojstvo može biti namena standardne perspektive. Izgled može biti složen za korisnike u smislu upravljanja njime, posebno za korisnike kojima je jedino potrebno da utiču na mali deo da bi zadovoljili svoju poslovnu inteligenciju i tražene uslove. Da bismo smanjili složenost, možemo da kreiramo vidljive podskupove nazvane perspektive, koje pokazuju korisnicima samo definisane delove mernih grupa, granica, dimenzija, atributa, hijerarhije, akcije i slično.

Proces dodele prava pristupa bazama podataka

Na kraju ne možemo zaobići proces dodele prava pristupa bazama podataka, pogotovo ako znamo činjenicu, da *OLAP* aplikacijom, naslonjenom na skladište podataka treba da upotrebljavaju administratori, finasijski analitičari, ali i strateški menadžment koji želi samo da ima nekoliko raznorodnih izveštaja iz oblasti poslovanja i mogućnost dnevnog praćenja trendova. Ovako različiti skup korisnika zahteva i raznorodna prava pristupa bazi podataka.

Pošto smo instalirali *Microsoft SQL Server 2005 Analysis Services (SSAS)*, svi članovi *Administrators* lokalne grupe su članovi *Analysis Services Server* uloge u toj instanci i imaju sva prava pristupa serveru za izvršavanje istih zadataka vezanih za *Analysis Services*. Po defaultu, ni jedan drugi korisnik nema pravo da administrira ili pregleda objekte u instanci *Analysis Services*.

Član uloge *Analysis Services Server* može da dozvoli korisnicima sve administrativne pristupe na bazi servera imenujući ih članovima Serverske uloge, koja im dopušta neograničeni pristup svim *Analysis Services* objektima i njihovim podacima. Član uloge *Analysis Services Server* takođe može dozvoliti korisnicima pristup preciziranim ograničenjima ili kompletna administrativna prava ili prava pristupa na nivou baze podataka. Ograničena administrativna prava uključuju procese ili definicije prava čitanja iz baze podataka, kocke, ili dimenzionog nivoa. U zavisnosti od potreba može biti kreirano više grupa sa različitim nivoima pristupa bazi podataka.

Osnovni cilj ove disertacije bio je da se među različitim modelima informacionih sistema pronađe i saobrazno predloženoj metodologiji prikaže adekvatnost primene koncepta analitičkih baza podataka u finansijskom poslovanju preduzeća. Primenom metodologije objektno orijentisanog razvoja poslovne inteligencije, koja u ovoj oblasti predstavlja novost, potvrđuje očekivane rezultate po pitanju *OLTP* sistema, izgrađenog skladišta podataka i *OLAP* aplikacije. Ovo je u potpunosti iskazano i u odeljku 4.8 Testiranje hipoteza.

Ovaj rad predstavlja napor da se ohrabri primena koncepta analitičkih baza podataka u formiranju strategije finansijskog poslovanja preduzeća, na naučno zasnovanom načinu i primenom dokazane metodologije, kroz četiri istraživačka poglavlja teme, u sadržaju označene kao: XII ANALITIČKE BAZE U POSLOVNOJ STRATEGIJI PREDUZEĆA.

U **prvom istraživačkom poglavlju** koje se odnosi na definisanje zahteva upotrebljena je kombinacija metoda za prikupljanje korisničkih i izvornih zahteva oslonjenih na intervju (*Top down*) i prikupljanje svih dokumenata koji su u vezi sa modelovanom aktivnosti EKONOMSKO FINANSIJSKI POSLOVI (*Bottom up*). Obe ove metode sprovedene su korišćenjem *IDEF0* i *IDEF1X* standarda kroz alate *BPwin* i *ERwin*. Nakon izvršene *OLTP* analize korišćenjem *SQL* jezika za kreiranje *ad-hoc* upita izrađen je model poslovnog slučaja upotrebe koji se sastojao od dijagrama poslovnog slučaja i dijagrama poslovne aktivnosti (slike 1.9 i 1.11). Za obe ove potonje aktivnosti korišćen je *UML (Unified Modeling Language)* standard. Korišćenjem *CASE* alate tipa *BPwin* i *ERwin* postignuto je praćenje čitavog životnog ciklusa projekta u delu koji se tiče prikupljanja korisničkih i izvornih zahteva. Ovo je veoma važno zbog jasnoće projektnog zadatka. Relativna nefleksibilnost alata za čitav životni ciklus mogla je da oteža obavljanje celokupnog posla. Pošto je razvoj softvera interaktivni proces u kome učestvuju tim sa više različitih stručnjaka sa različitim nivoima iskustava i u različitim fazama životnog ciklusa projekta, bilo je celishodno u delu izrade dijagrama poslovnog slučaja i dijagrama poslovne aktivnosti, primeniti fleksibilniju metodu *UML (Unified Modeling Language)*, odnosno standard. Najjači razlog za to je i što se od tog momenta u projekat uključuju stručnjaci različitih domena.

Drugo istraživačko poglavlje donelo je objektno orijentisanu analizu sa izradom konceptualnog modela i dijagrama interakcije koji zajedno čine model sistemskog slučaja upotrebe. U ovom segmentu, opredeljenjem za metodologiju objektno orijentisanog razvoja sistema poslovne inteligencije, dobijeno je kroz korišćenje *UML* standarda i *RationalRose* alata, na jasnoći sistemskih aktivnosti kod svih članova tima. Na ovaj način svi članovi tima počev od vlasnika projekta, preko sistem analitičara, administratora mreže, inženjera baze podataka pa do programera i prezentatora dobili su jasnu sliku o sistemskim aktivnostima, specificirajućim odgovornostima pojedinih učesnika u procesu EKONOMSKO FINANSIJSKI POSLOVA. Kod izrade konceptualnog modela primenom ovakve metodologije svi članovi tima mogli su podjednako kvalitetno da učestvuju u donošenju odluka o mogućim atributima i vezama između entiteta. Kvalitet projekta je dobio na težini samim tim što je prihvaćen od svih članova tima i što su svi ravnopravno učestvovali u izradi koncepta šeme zvezde za skladište podataka. Ovu ravnopravnost je dakako obezbedila pomenuta metodologija kroz mogućnosti alata *RationalRose*. Izradom dijagrama interakcije, odnosno dijagramom sekvenci i dijagramom saradnje, ni jedan od događaja ili operacija nije ostao nedorečen ili pak nejasan pojedincu iz tima. Obzirom da ovakva metodologija i primena alata lakih za upotrebu čak i krajnjim korisnicima omogućuje rano uključivanje tih istih krajnjih korisnika, jasno je da je njihovo zadovoljstvo projektom veće u onoj meri u kojoj su bili učesnici istog.

U **trećem istraživačkom poglavlju** koji se tiče objektno orijentisanog dizajna i dela pripreme, dato je težište na pojašnjenje svih operacija koje izvodi stručni tim, a koje pripadaju delu pripreme podataka za analitičko procesiranje informacija. Detaljno je objašnjeno korišćenje ekstrakciono-transformacionih alata kroz korake ekstrakcija, čišćenje i transformacija. Prezentovani su i neki od upita direktno napisanih u tu svrhu. Kada je izrada dimenzionog modela u pitanju, definisanjem hijerarhije elemenata i atributa, potom, denormalizacijom i kreiranjem agregacija, pripremljen je teren za izradu dijagrama klasa. Konačno je iz početnog *OLTP* modela prikazanog u prvom poglavlju (slika 1.1), dobijen model skladišta podataka u izvedbenom obliku Dijagramom klasa (slika 3.7) i otvorena su vrata za implementaciju.

Četvrto istraživačko poglavlje, kao završna, imlementaci-ona faza, dovela je do kreiranja fizičkog modela baze podataka. Nakon toga izvršeno je generisanje baze sa učitavanjem podataka. U pripremi za analizu korišćen je strukturni upitni jezik *SQL*. Da bi pripremljeni podaci bili iskoristivi u oba predviđena slučaja

implemetacije: lokalni i mrežni, ostalo se na *Microsoft* platformi. Za slučaj prikaza lokalne implementacije, podesne za manja preduzeća koja u svom *IT* okruženju imaju server sa nekoliko (do desetak radnih stanica) i ne poseduju potrebe za korišćenjem korporativnih sistema, implementacija je izvedena na *MSQuery32.exe* platformi potpomognutoj *MS Excel* poslovnom aplikacijom iz paketa *Office 2003*. Pivot tabele i pivot grafikoni ovog alata dali su efikasan uvid u mogućnosti izrađenog skladišta podataka. Paralelno sa ovim izvršeno je kreiranje *OLAP* aplikacije na *MS Access*-u kao najpodesnijem za izradu korisničkog interfejsa kada je u pitanju baza podataka do 10.000 dokumenata, odnosno par stitina hiljada zapisa. *OLAP* korisnički interfejs pridružen je već postojećem interfejsu transakcione baze podataka i time omogućeno ugodno korišćenje u *user friendly* okruženju.

Za potrebe srednjih i većih preduzeća, implementacija je izvršena u *MS SQL Server 2005*. Baze podataka u ovakvom okruženju mogu ići i do nekoliko miliona zapisa, a broj radnih stanica (što za analitički pristup i nije bitno) nema važnih ograničenja. U ovom slučaju, preko definisanja i razvijanja kočke primenom *Analysis Services* projekta, prikazano je pretraživanje kočke, modifikovanje mera, atributa i hijerarhija, sakrivanje i isključivanje hijerarhije atributa, sotriranje članova atributa na osnovu sekundarnih atributa, definisanje stvarnih veza, definisanje veze više prema više (*Many-to-Many*), definisanje dimenzione granularnosti unutar merne grupe, definisanje kalkuliranih članova, definisanje i listanje perspektiva i na koncu proces dodele prava pristupa bazama podataka.

Ograničenja metodologije objektno orijentisanog razvoja

“Iako se intuitivno čini da su objektno orijentisane notacije poput *UML*-a pogodne za opisivanje šablona dizajna, postoji nekoliko nedostataka koji takve notacije čine nepodesnim za specifikaciju arhitekturnih šablona. Osnovni problem pri pokušaju korišćenja objektnih notacija za definisanje šablona dizajna predstavlja to što su objektno notacije stvorene da bi se opisao proces dizajna sistema a ne šablona dizajna. Proces dizajna sistema se uobičajeno odvija u toku faze konstrukcije softverskog sistema i kao rezultat daje konkretan softverski sistem čiji se opis sastoji od definicija klasa i odnosa između njih. Arhitekturni šabloni dizajna, nasuprot tome, govore o generičkim aspektima određene klase softverskih sistema umesto o instancama tih klasa tj. o konkretnim softverskim sistemima. *UML* se može iskoristiti za opis meta-klasa od kojih se sastoji

određeni šablon, kao i veza između njih. Moguće je na primer definisati meta-klase „Klasa“, „Metoda“, „Promenljiva“ a zatim definisati relacije između njih i na taj način generički opisati šablon putem njegovih sastavnih delova. Ipak, *UML* kao i druge objektne notacije nije pogodan za definisanje šablona dizajna iz nekoliko razloga.

Prvi nedostatak ogleda se u tome što objektne notacije mogu da izraze relacije obično na nivou klasa tj. nije jednostavno predstaviti odnose između ostalih koncepata objektno orijentisanog sveta npr. između metoda ili između metoda i same klase. Ovaj tip relacija koji nije generički podržan u objektnim notacijama je neophodan da bi se arhitekturni šablon mogao opisati.

Još jedan nedostatak predstavlja činjenica da objektne notacije ne mogu predstaviti skupove višeg reda (npr. skupove skupova klasa), koji predstavljaju osnovu specifikacije šablona dizajna. Objektne notacije konkretno ne podržavaju korelacije između skupova (kao što su npr. odnosi 1:1 između skupova bilo kog reda). Zbog svega navedenog, asocijacije i kardinaliteti koji postoje u *UML*-u i ostalim objektnim notacijama nisu dovoljno izražajne čak ni kada se koriste kao jezik za definisanje meta entiteta.”^[131]

Pravac daljih istraživanja

U naprednijim ekonomijama se pored proizvoda traže i usluge koje dodaju vrednost. Ovo je u tesnoj vezi sa upravljanjem odnosima sa kupcima (*Customer Relationship Management, CRM*). Oni proizvođači softvera koji su zaglavljani u starom sistemu, gde se samo proizvodi, uvideće da se takmiče na tržištu u kojem imaju sve manje šanse da prežive.

Kada je reč o skladištima podataka, arhitektura podataka u osnovi možda i neće uopšte morati da se menja jer je malo verovatno da će se menjati osnovna struktura podataka. Ono što je sigurno viđeno za promenu jesu interfejsi i fizička implementacija. Pravac u kojem će najverovatnije ići baze podataka jeste pravac takozvanih vrmenskih proširenja. Tip podatka (*Date/Time*) u sadašnjim SUBP-ovima je samo neka vrsta korisnički definisane vremenske podrške. Za očekivati je dalji razvoj u tom pravcu obzirom da su skladišta podataka po prirodi vremenski orjentisana. Vreme jeste fundamentalni atribut i još jedna dimenzija. Uporedo sa time i najviše korišćeni strukturni jezik za upite *SQL* će verovatno dobiti novu klauzulu *WHEN*. Čini se da je za sada najveći problem dvodimenzionalnost tabela.

Što se tiče *OLAP* proširenja za *SQL* očekuje se omogućavanje proširenja nekih funkcija vezanih za pokretne proseke, agregatne

grupe i funkcije rangiranja, obzirom da su one najviše vezane za informatičku analizu.

Generalno gledano, do sada razvijena skladišta podataka pripadaju pasivnoj vrsti. Znači čekaju postavljanje pitanja od strane čoveka ili aplikacije u najboljem slučaju. Od budućih aplikacija iz domena OLAP-a se očekuje aktivnija podrška odlučivanju u smislu da sam sistem neprestano traži nove informacije o promenama situacija i obaveštava korisnike kad pronađe nešto što bi moglo da ih zanima. Ovo svakako ime korene u ideji da se softver nauči da "uči". To bi podrazumevalo i proširivanje opsega izvora podataka. Spoljni podaci za sistem nebi bili samo oni strukturirani, koji se uvoze u skladište podataka tokom procesa izdvajanja, transformacije i punjenja, već i nestruktuirani podaci, kakvi se nalaze u dokumentima *Web* stranicama, časopisima, dnevnoj štampi i slično. Prednost nestruktuiranih podataka je u tome što oni postaju dostupni mnogo brže od strukturiranih za čije je kreiranje potrebno vreme.

Ako još malo više zaronimo u maštanje, a ne pobegnemo od realnosti, mogli bismo da zamislimo aktivne softverske agente za traženje stvari koje bismo voleli da saznamo. Ti napredni agenti za traženje bili bi u stanju da koriste strukturane informacije skladišta podataka i nestruktuirane spoljne informacije. Naravno da bi oni u budućnosti bili sposobni da tumače finansijske rezultate kompanija i donose finansijske odluke.

Inače za prethodno postavljene poslove kojih se dotiče ova disrtacija, pravac daljih istraživanja vidi se u razmatranju *Data mining*-a tj. postupka otkrivanja znanja u podacima. Za kvalitetnu primenu ove metodologije potrebno je osim gore zamišljenih alata imati što više podataka za upoređivanje. Što znači, što se više bude unosilo podataka u informacioni sistem EKONOMSKO FINANSIJSKOG POSLOVANJA, to će se dobijati kvalitetnije informacije za potrebe odlučivanja top menadžmenta.



Ad-hok upit (Ad-hoc query): Svako spontano i neplanirano pitanje, ili upit. To je upit koji se sastoji od dinamički generisanog SQL-a, koji je obično generisan preko neke desktop alatke.

Agregacija (Aggregation): Specijalna forma asocijacija koja specificira odnos između agregacije (celine) i komponentnog dela.

Alati poslovne inteligencije (BI Tools - BI Software): Softver koji omogućava poslovnim korisnicima pogled na, i korišćenje velikih količina kompleksnih podataka. Aplikacije su dizajnirane tako da omogućuju lak pristup poslovnim informacijama svima u organizaciji, u cilju donošenja boljih poslovnih odluka, unapređenja poslovanja i jačanja odnosa s kupcima i dobavljačima.

Asocijacija (Association): Semantička veza između dva ili više klasifikatora koji obuhvataju veze između njihovih instanci.

Atribut (Attribute): Svojstvo ili karakteristika koja je uobičajena za neke ili sve objekte entiteta. Jedan atribut prikazuje korišćenje domena u kontekstu entiteta.

Baza podataka (Database): Kolekcija podataka koji su u međusobnoj relaciji, često sa kontrolisanom redundansom podataka, organizovanom po modelu (šemi) da koristi jednoj ili više aplikacija.

Dijagram klasa (Class diagram): Diagram koji pokazuje kolekciju deklarativnih (statičkih) elemenata modela, kao što su klase, tipovi i njihove sadržaje i veze.

Dimenziona tabela (Dimension Table): Tabela u šemi zvezde koja sadrži podatke za jednu od dimenzija višedimenzione kocke.

Domen (Domain): Imenovani skup vrednosti podataka istih tipova podataka, preko kojih se formiraju stvarne vrednosti atributa objekata. Svaki atribut može biti definisan pod samo jednim domenom.

Drill Down/Up: Tehnika analize koja dopušta korisnicima navigaciju po nivoima podataka rangiranim od najsumarnijih (*up*) do najdetaljnijih (*down*).

Druga normalna forma (Second Normal Form-2NF): Entitet je u drugoj normalnoj formi, ako je prvo u prvoj normalnoj formi i ako je svaki atribut koji nije ključ u direktoj zavisnosti sa primarnim ključem.

Egzistencijalna zavisnost (Existence Dependency): Uslov između dva entiteta u relaciji, koji pokazuje da ne može postojati objekat jednog entiteta koji nije u relaciji sa objektima drugog entiteta.

Eksplicitno znanje (Explicit Knowledge): Znanje artikulirano formalnim jezikom, prenosi se u obliku informacija putem različitih medija i relativno lako se uklapa u strukture inteligentnih informacionih sistema jer ne postoje njegove apstraktne kategorije.

Element podatka (Data Element): Najelementarnija jedinica podataka koja može da bude prepoznata i opisana u rečniku ili skladištu i koja ne može dalje da bude dekomponovana.

Entitet roditelj (Entity Parent): Entitet čiji objekti mogu da budu u vezi sa više objekata drugog entiteta (entiteta dete).

Entitet (Entity): Presentacija realnih i apstraktnih stvari (ljudi, objekata, slučaja...) koji se prepoznaju pod istim tipom podataka, jer dele iste karakteristike i mogu učestvovati u istim relacijama.

Entitet dete (Entity Child): Entitet u specifičnoj povezujućoj relaciji, čiji objekti mogu biti u vezi sa nulom ili jednim objektom drugog entiteta (roditelja).

ETL procesi (Extract, Transform and Load): ETL procesi podrazumevaju postupke dobijanja podataka iz nekog skladišta podataka (ekstrakcija), modifikiranje tih podataka (Transform) i umetanje u različita skladišta podataka (Load).

Funkcionalna zavisnost (Functional Dependency): Veza entiteta kojom se opisuje uslov "bar jedan".

Grafički korisnički interfejs (Graphical User Interface – GUI): Programski interfejs koji koristi grafičke mogućnosti računara u cilju olakšavanja rada pri upotrebi računara. Grafički interfejs koristi pokazivačke uređaje za selektovanje objekata, uključivanje ikona, menija, tekst boksova itd.

Identifikator zavisnosti (Identifier Dependency): Iskaz između dva entiteta u vezi, koji zahteva da primarni ključ u jednom (entitetu detetu) sadrži primarni ključ drugog (entiteta roditelja).

Ime uloge (Role Name): Ime dodeljeno prenesenom ključu i predstavlja upotrebu prenesenog ključa u entitetu.

Informacija (Information): Podatak koji se obrađuje radi dobijanja nekog značenja i znanja za osobu koja je prima. Ona je izlaz iz informacionog sistema.

Intelektualni kapital (Intellectual Capital): Ukupna intelektualna imovina i intelektualni potencijal kojim se kompanija koristi za stvaranje nove vrednosti. Sadrži akumulirano znanje koje neka organizacija poseduje u svojim ljudima, metodama, patentima, dizajnima i vezama. Deli se na tri segmenta: ljudski kapital, strukturalni ili organizacijski kapital i kapital klijenata.

Kardinalnost (Multiplicity): Specifikacija opsega dozvoljene kardinalnosti koja se može prihvatiti. Specifikacija kardinalnosti može biti data za uloge unutar asocijacija, delova unutar kompozicija, ponavljanja i drugih svrha. Obično je kardinalnost podskup nenegativnih celih brojeva.

Kategorija entiteta (Entity Category): Entitet čiji se objekti prikazuju podtipom i potklasifikacijom drugog entiteta (podtip, potklasa).

Klasa (Class): Opis skupa objekata koji dele iste atribute (osobine), operacije, metode, veze i semantiku. Klasa može da koristi skup interfejsa da specifikira kolekciju operacija, koji snabdevaju svoju okolinu.

Klijent/server arhitektura (Client/server architecture): Mrežna arhitektura u kojoj računari na mreži učestvuju kao serveri u upravljaju podacima i servisima mreže, ili kao klijenti, gde korisnici pokreću aplikacije i pristupaju serveru.

Ključ, Kandidat (Key, Candidate): Atribut, ili kombinacija atributa entiteta čije vrednosti jednoznačno određuju sve objekte entiteta.

Ključ, Opcioni (Key, Alternate): Svaki ključ kandidat koji nije primarni ključ.

Ključ, Preneseni (Key, Foreign): Atribut ili kombinacija atributa deteta, ili nekog drugog entiteta čija se vrednost primarnog ključa poklapa sa vrednošću primarnog ključa entiteta roditelja.

Ključ, Primarni (Key, Primary): Kandidatski ključ koji jednoznačno identifikuje entitet.

Ključ, Složeni (Key, Composite): Ključ sastavljen od dva ili više atributa.

Korisnički interfejs (User Interface): Komponenta računarskog sistema za podršku u odlučivanju, koja omogućava bidirekcionu (dvosmernu) komunikaciju između sistema i korisnika.

Korišćenje (Uses) - Relacije od jednog slučaja upotrebe do drugog u kojima ponašanje definisano za prethodne slučajeve upotrebe uračunava ponašanje za kasnije.

Lanac znanja (Knowledge chain) - Sadrži četiri elementa koja čine bit korisnosti

Metapodaci (Metadata or Meta Data): Podaci o podacima u skladištu podataka. Pomažu u definisanju sadržaja skladišta podataka. To su semantičke informacije odgovarajućih promenljivih. Moraju da uključuju poslovne definicije podataka, tačne opise tipova podataka, potencijalne vrednosti, originalni izvorni sistem, formate podataka i druge karakteristike. Definišu i opisuju poslovne podatke. Sadrže stvari poput imena, dužine, validne vrednosti i opisa podataka nekog podatka elementa. Čuvaju se u rečniku podataka. Izoluju skladište podataka od promena usled rada pod nekim operativnim sistemom.

Metoda (Method) - Implementacija operacije. Specifikira algoritam, ili proceduru koja daje rezultate operacije.

Model podataka (Data Model): Grafička i tekstualna prezentacija analize koja identifikuje podatke koji su potrebni organizaciji koja učestvuje u poslu. Prezentuje entitete, domen

(atribute) i relacije sa drugim podacima i konstruiše konceptualni pogled podataka i relacija između podataka.

Multidimenzijaska kocka (Multidimensional cube): Osnovna jedinica skladišta podataka, koja u sebi sadrži dimenzije interesantne korisnicima za analizu. Niz multidimenzijaskih kocki iz srodne organizacijske jedinice ili jednog segmenta poslovanja čini data mart.

N-arna asocijacija (n-ary association) - Asocijacija preko tri ili više klasa. Svaka instanca asocijacije je n-ta vrednost odgovarajuće klase.

Normalizacija (Normalization): Proces redefinisavanja i regrupisanja atributa u entitetima, u skladu sa normalnom formom.

Normalna forma (Normal Form): Stanje entiteta koje relativno zadovoljava skup normalizacija njegovih atributa. Specifična normalna forma je izvedena sukcesivnom redukcijom entiteta iz njegovog izvornog stanja u neki željeni oblik forme.

Nul (Null): Stanje gde vrednost nekog atributa nije poznata za neki objekat entiteta.

Ograničenje (Constraint): Pravilo koje pokazuje validnost stanja podataka.

Ograničenje egzistencije (Constraint existence): Uslov gde objekti jednog entiteta ne mogu da postoje ukoliko ne postoje objekti entiteta sa kojim je ovaj u relaciji.

Ograničenje kardinalnosti (Constraint cardinality): Ograničenje broja objekata entiteta, koje može da bude asociirano u relaciji.

OLAP alati (On-Line Analytical Processing): Kategorija aplikacija i tehnologije namenjena za skupljanje, upravljanje, obradu i prezentaciju multidimenzijaskih podataka namenjenih analizama za potrebe upravljanja. Omogućuje brz, konzistentan i interaktivan način pristupa i manipulacije multidimenzijaskim podacima memorisanim u skladištu podataka. OLAP postoji u dva temeljna oblika s obzirom na formu u kojoj su podaci memorisani: relacioni (ROLAP) i multidimenzionalni (MOLAP), te u hibridnom obliku (HOLAP) koji za više nivoe sumarizacije koristi multidimenzionalni oblik, ali omogućuje i drill-down do nižih nivoa koji su smešteni u relacionoj tabeli.

On-line Analytical Processing (OLAP): Softver koji se koristi za rad sa višedimenzionim podacima iz različitih izvora koji se smeštaju u skladište podataka. Formira različite poglede na podatke. Omogućuje brži, sadržajniji i interaktivniji pristup višedimenzionim podacima.

Organizacija koja uči (Learning Organization): Organizacija koja je prilagođena kupcima, karakteristika su joj kreativnost, intenzivno znanje, visoko obrazovan kadar, spremnost i sposobnost menadžmenta i zaposlenih na stalno učenje.

Osnovni entitet, generički (Entity Generic): Entitet čiji su objekti klasifikovani u jedan ili više podtipova ili potklasa (supertip, superklasa).

Otkrivanje „znanja“ u podacima (Data mining): Postupak traženja skrivenih modela, odnosa i zakonitosti unutar baze podataka. Današnji moćni softver za Data mining omogućuje pronalaženje uzoraka i trendova koji su značajni za pomoć pri definisanju poslovne strategije, pronalaženje povezanosti između raznih varijabli i pronalaženje interesantnih segmenata i preseka informacija. Kombinacijom tehnika iz područja statističke analize, neuronskih mreža, tehnika modeliranja, pronalaze se strukture i odnosi među njima, te izvode pravila i modeli koji omogućuju predviđanje i odlučivanje u novonastalim situacijama.

Podaci (Data): Binarna (digitalna) prezentacija atomskih činjenica, teksta, grafika, bit mapa, zvuka, analognih ili digitalnih video segmenata. Podatak je sirovina sistema koju ovaj dobija preko procedura i koja se koristi radi kreiranja informacija.

Poslovna inteligencija (Business Intelligence): Proces prikupljanja raspoloživih internih i značajnih eksternih podataka i njihovo pretvaranje u korisne informacije koje pomažu menadžmentu pri donošenju odluka.^[77,93] Označava i široku kategoriju softverskih solucija za prikupljanje, konsolidovanje, analiziranje i osiguranje pristupa podacima korisnicima u organizaciji u cilju donošenja boljih poslovnih odluka. Izraz i koncept poslovne inteligencije osmislili su stručnjaci Gartner Grupe 1993. godine.

Poslovne transakcije (Business Transaction): To je jedinica posla nad strukturama podataka u cilju kreiranja, modifikacije ili brisanja poslovnih podataka. Svaka transakcija predstavlja jednu vrednovanu činjenicu, koja opisuje jedan poslovni slučaj.

Poslovni model (Business Model): Dizajnerski pogled na to kako posao funkcioniše. Pogled može biti sa aspekta posla, podataka, slučaja ili resursa i može da bude o prošlom, sadašnjem ili budućem stanju posla.

Poslovni podaci (Business Data): Podaci o ljudima, mestima, stvarima, poslovnim pravilima i slučajevima koji se upotrebljavaju pri vođenju posla. Nisu metapodaci.

Pravilo (Rule): Formalni pristup specifičnim preporukama, direktivama ili strategiji, iskazanim kroz IF-THEN konstrukcije.

Prva normalna forma (First Normal Form 1NF): Entitet je u prvoj normalnoj formi ako su njegov sadržaj samo atomske vrednosti.

Rapidni razvoj aplikacija (Rapid Application Development-RAD): Deo metodologije koji navodi na inkrementalni razvoj uz podršku naručioca. Cilj je da razvoj projekta ostane usredsređen na

stalno ostvarivanje komunikacije. Jedino ograničenje pri ovakvom radu jeste različitost govora lica koja su u komunikaciji.

Rečnik podataka (Data Dictionary): Baze podataka o podacima i strukturama.

Reinženjering poslovnih procesa (Business Proces Reengineering): Fundamentalna reorganizacija institucije, koja se bazira na procesima koji donose vrednost kupcu. Uključuje reorijentaciju poslovanja od usmerenosti proizvodu ka usmerenost kupcu. ^[26, 42,64]

Sistem (System): Kolekcija povezanih jedinica koje su organizovane da izvršavaju određenu svrhu. Sistem može biti opisan jednim modelom ili sa više njih, najverovatnije sa različitih aspekata posmatranja.

Sistem za podršku odlučivanju (Decision Suport System): Kompjuterski sistem dizajniran za svrhu asistencije organizaciji u postupku donošenja poslovnih odluka.

Skalabilnost (Scalability): Sposobnost raslojavanja radi održavanja većih ili manjih količina podataka i više ili manje korisnika.

Skladište podataka (Data Warehouse): Skup integrisanih podataka izolovanih i prikupljenih iz operacijskih sistema iz svih područja poslovanja za sprovođenje analiza kao pomoć pri donošenju poslovnih odluka. Radi se o velikoj bazi koja omogućuje relativno brzo i jednostavno izvođenje složenih upita nad velikim količinama podataka.

Slučaj upotrebe (Use case) - Dijagram koji pokazuje relacije između aktera i slučajeva upotrebe unutar sistema.

Spremište podataka (Data Mart): Baza podataka koja ima jednake karakteristike kao i skladište podataka, ali u pravilu manja i fokusirana na podatke jednog segmenta poslovanja ili jedne organizacijske celine unutar organizacije.

Šema (Schema): Definicija strukture podataka.

Šema zvezde (Star Schema): Dizajn baze podataka koji se sastoji od tablice činjenica i jedne ili više dimenzijskih tablica.

Tip podataka (Data Type): Kategorizacija apstraktnog skupa vrednosti, karakteristika i skupa operacija koji se odnose na attribute. Celi brojevi, realni brojevi, znakovni tipovi podataka itd.

Transkacione baze podataka (Operational or Transaction Database): Baze podataka za transakciju (razmenu) podataka. One su izvor za skladišta podataka. ^[45]

Treća normalna forma (Third Normal Form-3NFT): Jedan entitet je u trećoj normalnoj formi najpre ako je u drugoj normalnoj formi, i ako svaki atribut koji nije ključ nije u direktnoj zavisnosti od primarnog ključa.

Upit (Query): Postavljanje pitanja (kriterijuma). Obično koristi složene SQL konstrukcije.

Upravljanje znanjem (Knowledge Management): Predstavlja konstantni proces obnove znanja organizacije. Opšti cilj upravljanja znanjem je povezati one kojima treba znanje s izvorima znanja u organizaciji, kao i usklađivanje transfera tog znanja.

Višedimenziona baza podataka (Multy dimensional Database – MDBS): Baza podataka koja omogućava korisnicima analize velikih količina podataka. Predstavlja podatke kao nizove koji su organizovani u višestruke dimenzije. Promenljive su objekti koji se čuvaju u višedimenzionim bazama. To su jednostavni nizovi vrednosti (numeričkih najčešće) koji su dimenzionisani po dimenzijama u bazi podataka. Može da ima višestruke promenljive, sa različitim ili jedinstvenim skupom dimenzija. Ovaj višedimenzioni pogled na podatke naročito je važan za OLAP aplikacije.

Znanje (Knowledge): Nematerijalni resurs, slika stvarnosti iskazana zamislama čoveka. Sastoji se od intuicije, skupa ideja, iskustva, veština i učenja. Ima potencijal stvaranja nove vrednosti.

Znanje zasnovano na podacima (Data-Based Knoledg): Znanje izvedeno iz podataka korišćenjem alata poslovne inteligencije iz skladišta podataka.

Životni ciklus razvoja sistema (Systems Development Life Cucle-SDLC): Proces sistemske analize, softverskog inženjeringa, programiranja i korisničke izgradnje sistema.



SKRAĆENICE

<i>BPR</i>	<i>Business Process Reengineering</i>
<i>BPWin</i>	<i>Business Process for Windows</i>
<i>CASE</i>	<i>Computer Aided System Engineering</i>
<i>CGI</i>	<i>Common Gateway Interface</i>
<i>CRM</i>	<i>Customer Relationship Management</i>
<i>DBMS</i>	<i>DataBase Management System</i>
<i>DDL</i>	<i>Data Definition Language</i>
<i>DM</i>	<i>Data mining</i>
<i>DML</i>	<i>Data Manipulation Language</i>
<i>DMO</i>	<i>Distributed Management Objects</i>
<i>DPC</i>	<i>Deferred Process Call</i>
<i>DRI</i>	<i>Declarative Referential Integrity</i>
<i>DSN</i>	<i>Data Source Name</i>
<i>DSS</i>	<i>Decision Support Systems</i>
<i>DTS</i>	<i>Data Transformation Services</i>
<i>DW</i>	<i>Data Warehouse</i>
<i>EDIFACT</i>	<i>Electronic Data Interchange For Administration, Commerce and Transport</i>
<i>EIS</i>	<i>Executive Information Systems</i>
<i>ERwin</i>	<i>Entity Relationships for Windows</i>
<i>ESS</i>	<i>Executive Support Systems</i>
<i>IAS</i>	<i>International Accounting Standards</i>
<i>IFRS</i>	<i>International Financial Reporting Standards</i>
<i>IDEF0</i>	<i>Integration Definition Functional Modeling</i>
<i>IE</i>	<i>Information Engineering</i>
<i>IFRS</i>	<i>International Financial Reporting Satnadards</i>
<i>ISM</i>	<i>Informacioni Sistem za Menadžment</i>
<i>ISO</i>	<i>International Organization for Standardization</i>
<i>ISQL</i>	<i>Interactive Structured Query Language</i>
<i>KE</i>	<i>Knowledge Engineering</i>
<i>KM</i>	<i>Knowledge Management</i>
<i>MOLAP</i>	<i>Multidimensional OLAP</i>
<i>ODBC</i>	<i>Open DataBase Connectivity</i>
<i>OLAP</i>	<i>OnLine Analytical Processing</i>
<i>OLE (DB)</i>	<i>Object Linking and Embedding (DataBase)</i>
<i>OLTP</i>	<i>Online Transaction Processing</i>
<i>PK</i>	<i>Primary Key</i>
<i>RAD</i>	<i>Rapid Application Development</i>
<i>RDBMS</i>	<i>Relational DataBase Management System</i>

<i>RI</i>	<i>Referential Integrity</i>
<i>ROLAP</i>	<i>Relational OLAP</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>TQM</i>	<i>Total Quality Management</i>
<i>UML</i>	<i>Unified Modeling Language</i>
<i>UNC</i>	<i>Universal Naming Convention</i>
<i>XML</i>	<i>Extensible Markup Language</i>

LITERATURA

1. Adamson, Christopher and Michael Venerable, *Data Warehouse Design Solutions*, New York: John Wiley and Sons, 1998.
2. Adelman, S. & Moss, L. , *Data Warehouse Project Management*, Boston: Addison-Wesley, 2000.
3. Adriaans, P., and D. Zaantinge, *Data Mining*, Addison-Wesley, 1996.
4. Bain T., Benkovich M., Dewson R., Ferguson S., Graves C., Joubert T., Lee D., Scott M., Skoglund R., Turley P., Youness S., *Professional SQL Server 2000 Data Warehousing with Analysis Services*, Wrox Press Ltd., Chicago, 2001.
5. Balard C., Herman D., Bell R., *Data Modeling Techniques for data warehousing*, IBM – International Technical Support Organization, San Jose, 1998.
6. Balaban N. Ristić Ž., *Sistemi za podršku odlučivanju*, Ekonomski fakultet, Subotica, 1998
7. Balaban, N., Ristić, Ž., *Poslovna inteligencija*, Ekonomski fakultet Subotica, 2006.
8. Ballard C., Beaton A., Chiou D., Chodagam J., Lowry M., Perkins A., Phillips R., Rollins J., *Leveraging, DB2 Data Warehouse Edition for Business Intelligence*, IBM Redbooks, November 2006
9. Barquin, R., and H. Edelstein (eds.), *Building, Using, and Managing the Warehouse*, Prentice Hall, 1997.
10. Benković S., *Operativno finansijsko poslovanje*, Priručnik, Junior, Beograd, 2006
11. Benković S., *Budžetiranje kapitala*, FON, Beograd, 2007.
12. Bečejski D., Veljović A., *Data Warehouse System For Design And Review Of Analytical Reports Of The Government Bureau Of Development*, Fakultet organizacionih nauka Univerziteta u Beogradu, Tehnički fakultet Čačak , Univerzitet u Kragujevcu, 6th BALKAN CONFERENCE ON OPERATIONAL RESEARCH, Solun, 2002. str. 84
13. Bečejski-Vujaklija D., *Metodologija izgradnje sistema za podršku odlučivanju tipa upravljačke igre*, Doktorska disertacija, FON, Beograd, 1992. str. 78-82
14. Bečejski-Vujaklija, D., *Metodološke osnove ekspertskeg ocenjivanja u funkciji podrške odlučivanju*, SIMORG 91, zbornik radova, str. 158-165, Kopaonik, 1991.
15. Bennett, S., McRobb, S., Farmer, R., *Object-oriented systems analysis and design using UML*, McGraww-Hill, 1999.
16. Bischoff, J., and T. Alexander (eds.), *Data Warehouse: Practical Advice from the Experts*. Prentice Hall, 1997.
17. Blok, C., *UML 2 Composition Model*, Journal of Object Technology, p. 3, 47-73, 2004.

-
18. Booch, G., Rumbaugh, J., Jacobson, I., *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
 19. Collins K., *Analytical CRM: Driving Profitable Customer Relationships*, Gartner, Inc., 2001.
 20. Codd E.F., (1990), *The Relational Model for Database Mangement Version 2.*, AWP Company, USA
 21. Chen P.P., (1976), *The Entity-Relationship Model - Toward a Unified View of Data*, ACM TODS, Vol.1, No.1.
 22. Chris Todman, *Projektovanje skladišta podataka*, Cet, 2001.
 23. Clark, T., *Implementing a Knowledge Strategy for Your Firm*, Research Technology Management, March-April, 2000. p. 23-24
 24. Ćirović M., *Bankarstvo, Bridge Company*, Beograd, internet izvor, 2001.
http://www.poslovnapolitika.com/index.php?option=com_content&task=view&id=352&Itemid=41 (16. 04.2009)
 25. Dendić, D., *Osnovi finansijskog računovodstva*, Beograd, 2000.
 26. Davenport, T. H., *Process Innovation: Reengineering Work through Information Technology*, Boston: Harvard Business School Press, 1993.
 27. Davenport, T.H., Prusak L. *Working knowledge*, Harvard Business School Press, Boston Massachusetts, 2000.
 28. Gojgic Natasa Veljovic Alempije Zahorjanski Miroljub Stanojevic Ljiljana Cvijovic
Natasa , *Implementation of Data Warehouse in Analysis of Qms Inconsistencies*, METALURGIJA INTRNATIONAL, (2012), vol. 17 br3, str. 5-10
 29. Devlin, B., *Data Warehousing: From Architecture to Implementation*, Addison-Wesley, 1996.
 30. Dulanović N., Veljović A., *Otkrivanje obrazaca u korišćenju Web-a*, Web Usage Mining, YU INFO 2002, Kopaonik, www.cit.org.yu, 2002.
 31. Đerković Z. i Petrović Z., *Računovodstvo*, Beograd, 1999
 32. English, L. P., *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*, New York: Wiley, 1999.
 33. Finnegan, P. & Sammon, D., *Foundations of an Organisational Prerequisites Model for Data*, 1999.
 34. Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Pearson Education, Publishing as Addison Wesley Professional, 2004.
 35. Galai D., : *The Essentials of Risk Management*, internet izvor <http://ebooks.ebookmall.com/ebook/232174-ebook.htm> , (06.06.2008)
 36. Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns: Elements of reusable Object-Oriented Software*, Addison-Wesley, 1995.

-
37. Gardner, S. R. *Building the Data Warehouse*, Communications of the ACM , 41 (9), p. 52-60,1998.
 38. Grozdanić R., *Osnove knjigovodstva i računovodstva*, T.F. Čačak, 2004.
 39. Grozdanić R., Đorđević D., *Osnove Ekonomije*, Tehnički Fakultet Zrenjanin, 1999.
 40. Gucer V., Altaf N., Co I., Edwards J. A., Layton C., Vasconcelos D., Wiggett P., Zonin A., *Tivoli Management Services Warehouse and Reporting* , IBM Redbooks, January, 2007.
 41. Gunderloy M., Sneath T., *SQL Server Developer's Guide to OLAP with Analysis Services*, Sybex, USA, 2001.
 42. Hammer, M. & Champy, J., *Reengineering the Corporation: A Manifesto for Business Revolution*, New York: Harper Business. *Decision Support in an Uncertain and Complex World: The IFIP TC8/WG8.3 International Conference 2004*.
 43. Han J., Kamber M. *Data Mining: Concepts and Techniques* Morgan Kaufmann, Hawaii, 2000.
 44. Inmon, W. H., *Building the Data Warehouse*, 2nd Edition, New York: John Wiley and Sons, 1996.
 45. Inmon, W. H., *Building the Operational Data Store*, New York: John Wiley and Sons, 1996.
 46. Zahorjanski Mirosljub Veljovic Alempije , *Use of Analytical Data Bases and Gis for Monitoring Performance of Institutions of Higher Education*, METALURGIJA INTERNATIONAL, (2011), vol. 16 br. 12, str. 136-139
 47. Jovanović M., Veljović A.: *Nove tehnologije u procesnoj industriji: Projektovanje upravljačkih sistema*, Savez hemičara i tehničara Jugoslavije, Beograd, 1998. str. 115
 48. Kachur, R., *Data Warehouse Management Handbook*, Paramus: Prentice Hall, 2000.
 49. Keen, P., *Decision Support Systems: The Next Decade*, DSS Journal Vol. 3 pp. 253-265, 1997.
 50. Kelly S., *Data warehousing – The Route to Mass Customisation*, J. Wiley&Sons, Chicheser, England, 1996.
 51. Kimbal, R., *The Data Warehouse Toolkit*, J. Wiley&Sons, New York, USA, 2000.
 52. Kimball R., *A Dimensional Modeling Manifesto*, DBMS , 1977.
 53. Kimball R. , *The Data Warehouse Toolkit*, New York: John Wiley and Sons, 1996.
 54. Kimball R., *Slowly Changing Dimensions*, DBMS, 1996.
 55. Kimball, R., *The Data Warehouse Toolkit: Practical Techniques for Building Data Warehouses*, John Wiley & Sons, 1996.
 56. Kimball, Ralph, L. Reeves, M. Ross and W Thornthwaite, *The Data Warehouse Lifecycle Toolkit*, New York: John Wiley and Sons, 1998.
 57. Kombokall R. *The Data Warehouse*, Wiley, 1996.
 58. Larman C., *APPLYING UML and PATTERNS*, Prentice Hall, New Jersey, 1998.

-
59. Lazarević B., *Baze podataka*, FON, Beograd, 2005.
 60. Lazarević B., *Metodologija razvoja informacionih sistema*, FON-LABIS, Beograd, 2003
 61. Little, R. & Gibson, M., *Identification of Factors Affecting the Implementation of Data Warehousing*, 1999.
 62. Malhotra, Y. *Knowledge Management for the New Business World*, BRINT Institute , 2001, www.kmnetwork.com/whatis.htm (19.03.2009)
 63. Mattson, R. *Data Warehousing: Strategies, Technologies and Techniques*, McGraw-Hill, 1996.
 64. Mayer R., *A Framework and a Suite of Methods for Business Process Reengineering*, Texas A&M University, Knowledge Based Systems, Inc., 1997 (www.idef.com).
 65. Meyer, B., *Object-Oriented Software Construction*, Upper Saddle River, N.J., Prentice Hall, 1997.
 66. Meyer, M. & Winter, R., *Organization of Data Warehousing in Large Service Companies*, A Matrix, 2001.
 67. Meyer, M., *Organisatorische Gestaltung des unternehmensweiten Data Warehousing: Konzeption der Rollen, Verantwortlichkeiten und Prozesse am Beispiel einer Schweizer Universalbank*, Doctoral Thesis, University of St. Gallen, Bamberg: Difo-Druck, 2000 str. 88.
 68. Milićev, D., Zarić, M., Piroćanac, N., *Objektno orijentisano modelovanje na jeziku UML*, Skripta sa praktikumom, Mikro knjiga, Beograd, 2001.
 69. Mogin P. Luković I., *Principi baza podataka*, FTN, Novi Sad, 1996.
 70. Muller Pierre-Alain, *INSTANT UML*, Wrox press LTD., 1997.
 71. Naiburg E., Maksimchuk R., *UML za projektovanje baza podataka*, Cet, Beograd, 2002.
 72. Nešić Z., Veljović A., Radojičić M., *Neka razmatranja objektnog modeliranja u višekriterijumskoj analizi*, YUinfo 2004, Kopaonik, 2004.
 73. Nunamaker, J.; Brigs, R. , *Information Technology and Its Organizational Impact*, WITSA, March 2002.
 74. O'Donnell, P., Arnott, D., & Gibson, M., *Data warehousing development methodologies: A comparative analysis*, Working Paper, Melbourne, Australia: Decision Support Systems Laboratory, Monash University, 2002.
 75. Object Management Group, *UML 2.0 Superstructure Specification*, internet izvor <http://www.omg.org/cgi-bin/doc> , (16.01.2009)
 76. Oesterle, H., *Business in the Information Age: Heading for new Processes*, Berlin: Springer, 1995.
 77. Panian Ž., Klepac G., *Poslovna inteligencija*, Masmedia, 2004.
 78. Parker C., Thomas C., *Management Information Systems*, Second Edition, Mitchell Mc Graw-Hill, 2003.
 79. Dimic Gabrijela Kuk Kristijan Zahorjanski Miroljub , *Mining Student'S Data for Analyze Electronic Learning Materials Available*

-
- on the Moodle Course*, METALURGIJA INTERNATIONAL, (2011), vol. 16 br. 12, str. 78-82
80. Poe, V. *Building a Data Warehouse for Decision Support*, Prentice Hall, 1996.
 81. Poole, J., Chang, D., Tolbert, D., & Mellor, D., *Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration*, New York: John Wiley & Sons, 2002.
 82. Quatrani T., *Vizuelno modelovanje Rational Rose 2002 i UML*, Cet, Beograd, 2003.
 83. Radulović B., Kazi Z., *Informacioni sistemi-odabrana poglavlja*, Tehnički Fakultet Zrenjanin, 2006
 84. Riordan M. Rebecca, *Projektovanje baza podataka*, Mikro Knjiga, Beograd, 2006
 85. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy F., Lorensen, W., *Object-oriented Modeling and Design*, Prentice-Hall, New Jersey, 1991.
 86. Scheer, A.W. & Nuettgens, M., *ARIS Architecture and Reference Models for Business Process*, 2000.
 87. Selic, B., Gullekson, G., Ward, P., *Real-Time object-oriented modeling*, John Wiley & Sons, New York, 1994.
 88. Simon, A., *90 Days to the Data Mart*, New York: John Wiley & Sons, 2001.
 89. *Standardi IDEF0 i IDEF1X*, internet izvor, (www.idef.com), (14.01.2009)
 90. Stevens, L., *Knowing What Your Company Knows*, Knowledge Management, December 2000.
 91. Sydanmaanlakka P, *An Intelligent Organization: Performance, Competence and Knowledge Management*, Capstone, Oksford, 2002
 92. Stanojević I., Surla D., *Uvod u objedinjeni jezik modeliranja*, Grupa za informacione tehnologije, Novi Sad, 1999.
 93. Stanojević Lj., Veljović A., *Razvoj metodologije projektovanja poslovne inteligencije*, Monografija, Megatrend univerzitet primenjenih nauka, Beograd, 2008.
 94. Thomsen, E., and G. Spofford., *OLAP Solutions: Building Multidimensional Information Systems*, John Wiley & Sons, 1997.
 95. Turban, E. & Aronson, J. E., *Decision Support Systems and Intelligent Systems*, Sixth Edition, Upper Saddle River: Prentice Hall, 2003.
 96. Turban E., McLean E., *Wetherbe J. Informaciona tehnologija za menadžment*, ZUNS, Beograd, 2003
 97. *UML 1.3 standardi*, internet izvor, www.rational.com i www.omg.com (05.01.2009).
 98. Zahorjanski Miroljub Benkovic Sladjana S Veljovic Alempije , *Concept of Analytical Databases in the Creation of Financial*

99. Veljović A., *Objektno modeliranje informacionih sistema*, Megatrend, Beograd, 2005. str. 37-55
100. Veljović A., *Projektovanje informacionih sistema*, Kompjuter biblioteka, Čačak, 2003.
101. Veljović A., Radojičić, M., *Menadžment informacioni sistemi* Kompjuter biblioteka, Čačak, 2005.
102. Veljović A., *Menadžment razvojem*, Kompjuter biblioteka, Čačak, 2003
103. Veljović A., *Iformatičko upravljanje razvojem preduzeća*, Kompjuter biblioteka, Čačak, 2004.
104. Veljović A., *Kompjuter u sistemu kvaliteta*, Beograd, 2001
105. Veljović A. Njeguš A., *Osnove relacionih i analitičkih baza podataka*, Megatrend univerzitet primenjenih nauka, Beograd, 2004.
106. Veljović A., Lepojević B., Jovanović S., *Objektno-orijentisane baze podataka: Tehnologija koja dolazi*, Katedra za industrijski menadžment, YUinfo 2003, Kopaonik, 2003.
107. Veljović A., Radojičić M., *O nekim elementima strateškog pristupa upravljanja kvalitetom*, Katedra za industrijski menadžment, Tehnički fakultet Čačak, Časopis Strategijski menadžment, "SM, 1-2/2004" Subotica, str. 51-56, 2004.
108. Veljović A., Radojičić M., *Primena tehnika funkcionalnog i informacionog modeliranja i organizaciono okruženje*, Katedra za industrijski menadžment, Tehnički fakultet Čačak, Časopis Strategijski menadžment, "SM, 3/2003" Subotica, 2003.
109. Veljović A., Zahorjanski M., *Reinženjering informacionog sistema u oblasti knjigovodstva*, DQM, Beograd, 2006
110. Veljović A., Zahorjanski M., *Izrada izveštaja namenjenih menadžmentu preduzeća korišćenjem OLAP multidimenzionalnog pristupa podacima*, INFOFEST, septembar 2006, Budva
111. Veljović A., Eremija Z., Stanojević Lj., *Dokumenta sistema menadžmenta kvalitetom Megatrend univerziteta*, 2007.
112. Veljović A., Eremija Z., Stanojević Lj., *Informacioni podsistem za vrednovanje obrazovanja na Megatrend univerzitetu*, Beograd, 2005.
113. Veljović A., Eremija Z., Stanojević Lj., *Integralni informacioni sistem Megatrend Univerziteta*, 2007.
114. Veljović A., Zahorjanski M., *MS QUERY i OLAP multidimenzioni pristup podacima*, SYM-OP-IS oktobar 2006, Banja Koviljača.
115. Veljović A., *Informatički pristup upravljanju kao faktor jačanja konkurencije*, Treći međunarodni naučni skup na temu "Jačanje konkurentnosti preduzeća i privrede", Univerzitet Megatrend, 2005, Beograd, str 281–293, 2005.
116. Veljović A., *Modeliranje informacionih sistema*, Megatrend Univerzitet primenjenih nauka, Beograd, 2005.

-
117. Veljović A., Pišćević M., Dimitrijević N., *Višedimenziona analiza podataka kod informacionih sistema*, Katedra za industrijski menadžment, YUinfo 2003 Kopaonik, 2003.
 118. Veljović A., *Put ka integralnom informacionom sistemu na primeru MEGATREND Univerziteta*, Tehnički fakultet Čačak, Megatrend revija, međunarodni časopis za primenjenu ekonomiju, Vol 2, 2005, Beograd, 2005.
 119. Veljović A., Radojičić M., *Informatički pogled na upravljanje razvojem u preduzeću*, SymOrg 2004, Zlatibor, 2004.
 120. Veljović A., Radojičić M., *Menadžment informacioni sistemi*, Tehnički fakultet Čačak, 2005.
 121. Veljović A., Radojičić M., Tomić I., *Prikaz korišćenja UML na primeru poslova cirkulacije u biblioteci*, Symopis 2004 Fruška Gora, 2004.
 122. Veljović A., *Razvoj menadžment informacionih sistema zasnovanih na interaktivnom analitičkom procesiranju*, INFOFEST Budva, 2002.
 123. Veljović A., Stanojević Lj., *Identifikacija rizika u postupku projektovanja jedinstvenog informacionog sistema fakulteta Megatrend univerziteta*, Strategijski menadžment, "SM, 2007" Subotica, 2007.
 124. Veljović A., Stanojević Lj., *Razvoj informacionog sistema fakulteta i veza sa standardom ISO 9001:2000*, Tehnika, 2007.
 125. Veljović A., Tot I., Joksimović S., *Student success analyse system*, 8th BALKAN CONFERENCE ON OPERATIONAL RESEARCH, Bukurest, 2005.
 126. Veljović A., Živković D., *Prikaz korišćenja BPwin- u definisanju aktivnosti u projektovanju po zahtevima sistema kvaliteta*, Kvalitet- Poslovna politika, broj 5-6, 1998.
 127. Weetman, Pauline, - *Financial and management accounting*, London, Pitman Pitsburg, 1996.
 128. Winter, R. & Strauch, B., *A Method for Demand-driven Information Requirements Analysis in Data Warehousing Projects*, Journal of Data Warehousing, 2003.
 129. Wixom, B. H. & Watson, H. J. *An Empirical Investigation of the Factors Affecting Data Warehouse*, 2001.
 130. Walter.G. , *EAI Tools As Solutions for Management Software*, Industryweek, 2003.
 131. Aline Lucia Baroni, Yann-Gael Gueheneuc, Herve Albin-Amiot, *Design Patterns Formalization*, Research report 03/3/INFO, Nantes : Ecole Nationale Superieure des Techniques Industrielle et des Mines de Nantes-Computer Science Department 2008



CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

62
005.591.4:658.5
005.8

ZAHORJANSKI, Zahorjanski M., 1956-
Primena analitičkih baza podataka u finansijskom menažmentu
= Application of analytical databases in financial Management
/ Miroljub M. Zahorjanski: Fakultet tehničkih nauka,
2013 (Vrnjačka Banja : SaTCIP). - 157 str. :
graf. prikazi, tabele ; 25 cm. -
(Primena analitičkih baza podataka u finansijskom menažmentu;
#knj. #3 = Business Process Reengineering ; #vol. #3 /
urednik serije Alempije V. Veljović)

Na nasl. str.: Univerzitet u Kragujevcu. -
Tiraž 100.
Napomene i bibliografske reference uz
tekst. - Bibliografija: str. 151-157.

ISBN 978-86-7776-155-4

1. Уп. ств. насл.

а) Реинжењерство б) Пословни процеси -
Реинжењеринг с) Управљање пројектима
COBISS.SR-ID 200078348