

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**RAZVOJ POSLOVNO INTELIGENČNEGA SISTEMA ZA ANALIZO  
ZALOG PROIZVODNEGA PODJETJA Z UPORABO TABELARIČNEGA  
MODELA IN JEZIKA DAX**

Ljubljana, julij 2016

IGOR MAKOVEC

## IZJAVA O AVTORSTVU

Podpisani Igor Makovec, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Razvoj poslovno inteligenčnega sistema za analizo zalog proizvodnega podjetja z uporabo tabelaričnega modela in jezika DAX, pripravljenega v sodelovanju s svetovalcem prof. dr. Aleš Popovičem.

IZJAVLJAM,

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil/-a;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 11.07.2016

Podpis študenta: \_\_\_\_\_



# KAZALO

<b>UVOD .....</b>	<b>1</b>
<b>1 ODLOČANJE V POSLOVNEM PROCESU.....</b>	<b>2</b>
<b>2 POSLOVNA INTELIGENCA .....</b>	<b>6</b>
2.1 Definicija poslovne inteligence.....	6
2.2 Zgodovinski pregled razvoja poslovne inteligence .....	8
2.3 Arhitektura poslovne inteligence .....	10
2.4 Razvoj poslovno inteligenčnega sistema .....	14
2.4.1 Planiranje projekta.....	15
2.4.2 Definiranje poslovnih zahtev.....	16
2.4.3 Zasnova tehnološke arhitekture .....	17
2.4.4 Izbira produkta, programske opreme in namestitev .....	17
2.4.5 Dimenzijsko modeliranje.....	18
2.4.6 Fizična zasnova zadrževalnega dela in razvoj.....	21
2.4.7 Specifikacije za analitične aplikacije in razvoj analitičnih aplikacij.....	26
2.4.8 Postavitev, vzdrževanje in nadaljnji razvoj podatkovnega skladišča .....	26
2.5 Tabelarni model znotraj BI semantičnega modela.....	27
2.5.1 Hranjenje podatkov.....	28
2.5.2 Tabele .....	29
2.5.3 Jezik DAX .....	30
2.5.4 Optimizacija tabelarnega modela .....	33
2.6 Odločitev glede izbire ustreznega modela .....	34
2.7 Tržišče poslovne inteligence danes in trendi razvoja .....	36
<b>3 MANAGEMENT ZALOG .....</b>	<b>40</b>
3.1 Vrste zalog .....	42
3.2 Stroški zalog .....	44
3.3 Analiza zalog .....	46
3.3.1 ABC analiza zalog .....	49
3.3.2 XYZ analiza zalog .....	52
3.4 Kazalniki zalog .....	54
3.5 Ukrepi za zmanjšanje zalog .....	55
<b>4 TABELARNI MODEL ZA ANALIZO ZALOG SKUPINE TPV.....</b>	<b>57</b>
4.1 Skupina TPV in njen oskrbni proces .....	57
4.2 Izdelava prototipnega modela.....	59
4.2.1 Definiranje poslovnih potreb .....	59
4.2.2 Dizajn tehnološke arhitekture.....	60
4.2.3 Izbira programske opreme .....	60
4.2.4 Dimenzijsko modeliranje.....	61
4.2.5 Fizični dizajn, dizajn zadrževalnega dela in razvoj – ETL.....	64
4.2.6 Specifikacije za analitično aplikacijo in razvoj analitične aplikacije .....	71
4.2.6.1 Analize na osnovi mesečne granulacije podatkov.....	73

4.2.6.2	Analize na osnovi dnevne granulacije podatkov .....	76
4.2.7	SWOT analiza prototipa.....	80
<b>SKLEP.....</b>		<b>81</b>
<b>LITERATURA IN VIRI.....</b>		<b>84</b>

## **KAZALO TABEL**

Tabela 1: Primerjava definicij BI in DSS.....	6
Tabela 2: Primerjava OLTP sistema in podatkovnega skladišča .....	12
Tabela 3: Razredi ABC .....	50
Tabela 4: S pomočjo ABC analize lahko določimo parametre za za kontrolo zalog.....	51
Tabela 5: XYZ razredi glede na nihanje povprečne porabe.....	53

## **KAZALO SLIK**

Slika 1: Splošni model upravljanja.....	3
Slika 2: BI in DSS komponente .....	7
Slika 3: Trendi in vplivi na področju podatkovnega skladiščenja 1975-2000 .....	10
Slika 4: Tipična arhitektura sistema BI .....	11
Slika 5: Relacijski in večdimenzijski model .....	12
Slika 6: Diagram življenjskega cikla sistema BI.....	15
Slika 7: BUS matrika podatkovnega skladišča .....	19
Slika 8: Zasnova tabele dejstev kjer je potrebna točno časovno merjenje .....	20
Slika 9: Datumska dimenzija.....	21
Slika 10: Inkrementalno nalaganje z uporabo običajnih SSIS komponent .....	23
Slika 11: SCD komponente za primer spremenljivih in zgodovinskih atributov.....	25
Slika 12: Arhitektura BI Semantičnega modela .....	27
Slika 13: V pomnilniku se hranijo kompresirani podatki stolpcev .....	29
Slika 14: Microsoft BI semantični model .....	36
Slika 15: Gartnerjev magični kvadrant za BI in analitične platforme.....	37
Slika 16: Koraki aktivnosti pri analizi in zmanjšanju zalog podjetja.....	48
Slika 17: Splošna oblika grafične Pareto ABC analize zalog .....	51
Slika 18: Izbor politik na osnovi ABC in XYZ analize .....	53
Slika 19: Oskrbni proces v podjetju .....	58
Slika 20: BUS matrika podatkovnega skladišča .....	62
Slika 21: Preimenovanja atributov takoj po branju vira – izpis SAP transakcije MC.9 .....	63
Slika 22: Del metapodatkov za dimenzijo DimMaterial.....	64
Slika 23: Del metapodatkov za tabelo dejstev FaktPremiki.....	64
Slika 24: Nastavitev povezave do vira podatkov in nastavitev tipa atributa.....	65

Slika 25: ETL postopek za eno od dimenzij.....	66
Slika 26: ETL postopek za tabelo dejstev – vsi materialni premiki.....	67
Slika 27: Checksum – zgoščevalna funkcija po algoritmu MD5 .....	68
Slika 28: Transformacije zapisov .....	69
Slika 29: Lookup komponenta za iskanje nadomestnih ključev dimenzijske tabele.....	70
Slika 30: Generiranje manjkajočega dimenzijskega ključa.....	70
Slika 31: Transformacija števil.....	70
Slika 32: Premapiranje podatkov iz podatkovnega toka pred zapisom v v DW .....	71
Slika 33: Galaksija shema dimenzijskega modela.....	72
Slika 34: Osnovna nadzorna plošča.....	75
Slika 35: ABC analiza ROH materialov na izbrani poslovni enoti .....	76
Slika 36: Povprečna zaloga in obračanje zaloge pri različnih granulacijah podatkov .....	78
Slika 37: ROH materiali – razlike zaradi granulacije zajema podatkov.....	79
Slika 38: FERT materiali – razlike zaradi granulacije zajema podatkov .....	80



## UVOD

Današnje poslovno okolje označujejo hitre spremembe, oster konkurenčni boj med poslovnimi subjekti ter relativno velika negotovost. Podjetja si neučinkovitosti in neuspešnosti preprosto ne morejo več privoščiti. Zato je dan velik poudarek k vitkosti (angl. *lean*) v vseh segmentih: pri dobaviteljskih verigah, zalogah, proizvodnji in prodaji. Pritisk na stroške je zelo velik, odločilen. Zaradi sprememb na prodajni strani modela proizvodnje na zalogo skoraj ni več. Vedno več izdelkov, sploh dražjih, se proizvaja na zahtevo in ob upoštevanju želja kupcev. Posebej je to aktualno v avtomobilski industriji, saj se praktično vsa vozila izdelujejo za znane kupce in po njihovih individualnih željah.

Zaloge imajo pomemben vpliv na dobičkonosnost, ki je mera uspešnosti v sedanji družbeno-ekonomski ureditvi. Zaloge so zelo pomembno področje gospodarjenja v vsakem podjetju, saj omogočajo ustrezno delovanje številnih drugih funkcij, predvsem nabave, proizvodnje in prodaje, ker zagotavljajo njihovo nemoteno oskrbo ob planiranem času (Čižman, 2002). Zmanjšanje zalog povečuje dobičkonosnost, a ni možno kar tako in povprek. Potrebna je celovita analiza, ki obenem pokaže tudi najboljšo nabavno metodo.

Namen magistrskega dela je s pomočjo domače in tuje strokovne literature proučiti področje analize zalog in področje poslovne inteligence (angl. *business intelligence*, v nadaljevanju BI) ter preveriti uporabnost tabelaričnega modela. Ker je problemsko področje zalog izbrano kot ciljno področje, postavimo osnovno trditev, da je relevantne informacije potrebne za analizo zalog, mogoče dobiti iz množice podatkov v transakcijskem sistemu z uporabo novega tabelaričnega pristopa na področju BI. Primarni cilj magistrskega dela je izdelati BI prototip in preveriti navedeno trditev. Pri tem zasledujemo tudi več sekundarnih ciljev, ki spadajo k področju:

- identificirati potencialne za zmanjšanje zalog: ukrepi, nabavne metode;
- ugotoviti, ali je novi tabelarični pristop v analitsko - prezentacijski plasti sistema BI primeren za analizo zalog, in sicer na osnovi posnetkov stanja v skladiščih in na osnovi vseh materialnih premikov blaga;
- preveriti, ali pri analizi zalog pomembno vlogo igra podrobnost (granulacija podatkov).

Magistrsko delo pri zasledovanju zastavljenih ciljev uporablja kombinacijo deduktivnega in induktivnega raziskovalnega pristopa. V prvem deduktivnem delu – poglavja od 1 do 3 – z opisno metodo in metodo kompilacije obravnavamo pregled znanstvene in strokovne literature iz obravnavanega področja odločanja v poslovnem procesu, tehnologije BI in analize zalog. Pridobljena teoretična podlaga in pridobljena praktična znanja iz izobraževanj v tujini v več tovarnah družbe Faurecia služijo kot osnova za drugi empirični del, v katerem sestavimo prototipni sistem za analizo zalog proizvodnih podjetij skupine TPV. Ta prototip ponuja tudi odgovor, ali je postavljena trditev pravilna.

V prvem poglavju obravnavamo odločanje v poslovnem procesu. Pri odločanju potrebujemo informacije, ki jih zagotovi BI.

V drugem poglavju pregledamo definicije BI ob ugotovitvi, da se te stalno spreminjajo. Po kratkem zgodovinskem opisu BI prikažemo infrastrukturo BI in postopke razvoja sistema BI. Sledi opis večdimenzijskega in tabelaričnega modeliranja, ki od prve komercialne uvedbe v letu 2010 postaja vedno uporabnejše in predvsem se stalno in hitro razvija. K temu pregledu sodi tudi pregled koristnih napotkov, saj tabelarični model teče v pomnilniku računalnika in tega pomnilnika ni nikoli preveč. Izvedemo tudi pregled kriterijev, kdaj izbrati večdimenzijski in kdaj tabelarični model. Drugo poglavje zaključimo s pregledom BI tržišča danes ter v prihodnje.

V tretjem poglavju so obravnavane zaloge in njihov pomen pri poslovanju podjetja. Prikažemo pomen zalog v donosnosti lastniškega kapitala. Po razčlenitvi vrst zalog se osredotočimo na stroške zalog, saj je razumevanje teh potrebno za naslednji korak to je analiza zalog. Prikažemo sistemski napotek, kako ravnati pri analizi zalog, in opišemo osnovni analizi ABC in XYZ. Poglavje zaključimo s potrebnimi in priporočljivimi kazalniki zalog za proizvodno podjetje ter z izbranimi ukrepi za zmanjšanje zalog. Te ukrepe obravnavamo kritično, tudi glede na pridobljene izkušnje pri delu v logistiki.

Četrto poglavje je praktičen prikaz izgradnje podatkovnega skladišča, temelječega na teoretičnih osnovah iz prvega poglavja. Prikazana je tabelarična rešitev v analitsko-prezentacijski plasti BI. Na praktičnem primeru analize zalog in večjega števila podatkov preverjamo postavljeno trditev in iščemo optimalni pristop za delo s tabelaričnim modelom. Prikažemo več nadzornih plošč in naredimo primerjavo pomena granulacije vhodnih podatkov.

## **1 ODLOČANJE V POSLOVNEM PROCESU**

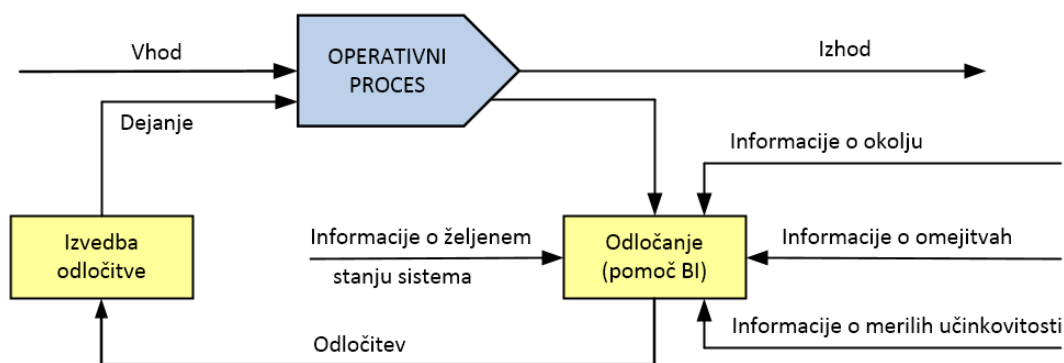
V podjetjih nenehno poteka odločevalski proces. Odločati se je potrebno o strategijah, taktikah, planu in o korektivnih aktivnostih za primere, ko dejansko stanje zaradi različnih razlogov odstopa od željenega. Danes so procesi kompleksni in zanašati se na intuicijo že dolgo ni več dovolj. Potrebujemo informacije o teku procesov tukaj in sedaj. Tem je potrebno dodati informacije iz okolja, da bi bila slika popolna (Makovec, 2013).

Faza odločanja je sestavni del splošnega modela upravljanja, ki ga lahko razumemo kot ciklično ponavljanje izbire in izvedbe upravljalnega dejanja (Gradišar, Jaklič & Turk, 2007). Upravljalna dejanja upravljalec uporablja za spreminjanje obnašanja sistema z namenom, da doseže cilj. Dejanja imajo dva koraka. V prvem koraku se upravljalec na podlagi pridobljenih informacij odloči za eno od možnih upravljalnih dejanj. V drugem koraku to dejanje izvede. To je možno izvesti pri sistemih, ki imajo cilj. Tak primer so organizacijski sistemi. Informacije zagotavljajo informacijski sistemi. Vsak upravljalni sistem lahko



razdelimo na dva dela, na del operativnih procesov in na del upravljalnih procesov. Oba dela sta med seboj povezana, tako da so izhodi enega dela vhodi v drug del. Tako pridemo do upravljalne zanke (Gradišar, Jaklič & Turk, 2007). Splošni model upravljanja je predstavljen na Sliki 1.

Slika 1: Splošni model upravljanja



Vir: Prirejeno po Gradišar, Jaklič & Turk, *Osnove poslovne informatike*, 2007, str. 214.

Gradišar, Jaklič in Turk (2007) ugotovljajo tudi, da je v literaturi najpogosteje srečati preprost klasičen model odločanja Nobelovca Herberta Simona, ki ga mnogi smatrajo očeta sodobne vede o procesu odločanja. Model odločanja iz leta 1960 sestavljajo tri faze:

1. spoznavanje,
2. oblikovanje in
3. izbira.

V prvi fazi spoznavanja zbiramo podatke o sistemu in pridobivamo podatke iz okolja z namenom, da zaznamo probleme ali priložnosti. Razumevanje problema, določanje organizacijskih ciljev in pravi podatki so ključni. Te je običajno težko dobiti in včasih tudi težko razumeti, kaj pomenijo.

V drugi fazi oblikovanja proučujemo probleme in priložnosti ter iščemo rešitve ter ugotovljamo možne posledice posameznih alternativ. Tukaj se določijo metode in sodila za ocenjevanje možnih rešitev.

V tretji fazi izbire na podlagi metod in sodil izberemo najboljšo oziroma optimalno možno rešitev. Iskanje te je pogosto neizvedljivo zaradi raznih razlogov: finančni, časovni, znanje, itd., zato takrat izberemo rešitev, ki ni najboljša, ampak je zadovoljiva.

Pri iskanju optimalne rešitve so nam v pomoč metode, ki vključujejo: linearno programiranje, dinamično programiranje, teorijo vrst, odločitvene tabele, analizo stroškov in koristi, teorijo iger itd. Kadar je možno rešitev dovolj dobro predstaviti z modelom, je lahko faza izbire izvedena z računalnikom. Ne sme se pozabiti na analizo občutljivosti, ki pokaže vpliv sprememb vhodnih podatkov na rešitev.

Pomoč BI je možna v prvi fazi pri zbiranju podatkov. S pravilnim prikazom in agregacijami pridemo do informacij, ki jih lahko v drugi fazi identificiramo, preučujemo probleme ter iščemo rešitve.

Pri tretji fazi pa je Simon razbil iluzijo: zmotno predstavo o »optimalni izbiri«, se pravi odločitvi, sloneči na kar največjem številu podatkov. Klasična teorija o odločanju je slonela na koncepciji o »optimizaciji«, ki sta jo leta 1945 razvila von Neumann in Morgenstern v delu Teorija iger in gospodarno vedenje. Sprejeti optimalno rešitev po tej teoriji pomeni, da se glede na vse podatke o problemu odločimo za najboljšo pot. Simon pravi, da je tako ukrepanje nestvarno, ker zahteva sposobnost racionalizacije in predvidevanja, ki ga človek nima. V realnosti nikoli ne upoštevamo vsega, kar je povezano s problemom, ker nimamo vsega na voljo. Zadovoljiti se moramo samo z »dovolj dobrimi odločitvami« oziroma s principom »omejene racionalnosti«. Ta princip je definiral: »Sposobnost človekovega duha, da formulira in rešuje zapletene probleme, je zelo majhna v odnosu z razsežnostjo problemov, katerih rešitev je v realnem svetu odvisna od objektivno racionalnega vedenja...« (Delo, 1978). To pomeni, da kadar sprejemamo kako racionalno odločitev, ne upoštevamo realnega položaja, temveč poenostavljen model resničnosti. Šahist ne razmišlja v okviru realnega rezultata partije, kajti tega rezultata ne pozna; pozornost namenja omejenim ciljem, denimo obvladovanju središčnih polj, prednosti v figurah. Skuša seveda predvidevati nasprotnikove poteze. Pri tem pa je toliko uspešnejši, kolikor bolj doume nasprotnikovo psihologijo, kolikor boljši model njegovega vedenja si naredi (Delo, 1978).

Odločanje je lahko tudi intuitivno. Ta model je v preteklosti veljal za manj priporočljivega. Danes znanstveniki že priznavajo, da je lahko tudi učinkovit. Erenda (2014) je intuicijo in intuitivno odločanje raziskoval pri vršnih in srednjih menedžerjih v slovenski avtomobilski industriji. Ena izmed pomembnejših ugotovitev je, da se vršni in srednji menedžerji v avtomobilski industriji, ki vsakodnevno sprejemajo odločitve, pri svojem delu in odločanju v veliki meri (79,3 %) zanašajo na intuicijo. Pri tem je ugotovil, da obstajajo statistično značilne razlike glede na vodstvene izkušnje. Namreč anketiranci z najmanj vodstvenimi izkušnjami so bili mnenja, da jih intuicija ne usmerja, medtem ko so vsi ostali (takih je bilo tri četrtine) menili, da jih intuicija usmerja in jim predstavlja pomembno vodilo pri svojem delu oziroma sprejemanju odločitev. S pomočjo metode Pajek je prav tako prišel do ugotovitev o statistično značilnih spremenljivkah (lastnostih), ki najznačilneje vplivajo na intuitivno odločanje in je na osnovi le-teh oblikoval model intuitivnega odločanja. Navedeni model intuitivnega odločanja pri vršnih in srednjih menedžerjih v avtomobilski industriji je na osnovi ugotovitev raziskave opredelil z dvema osnovnima komponentama, in sicer: s čustveno in časovno komponento, ter dvema vplivnima faktorjema, in sicer: z izkušnjami in z ozaveščanjem intuitivnega delovanja.

O tem, kako odločanje poteka v praksi, so Rozman, Kovač in Koletnik (1993) ugotovili, da managerji dajejo prednost kratkoročni uspešnosti in da so uspehi kratkoročnih odločitev razmeroma majhni. Prednost naj bi imele celovite rešitve, a raziskave kažejo, da so bolj redke. Namen informacij je zmanjšanje tveganja pri odločanju. Odločamo namreč o

prihodnosti. Ni težko ustvariti veliko podatkov, težko pa je pravočasno pripraviti ustrezne informacije. Čakanje nanje ovira in odmika odločanje.

Če je izbira modela odločanja stvar odločevalca, ki je pri tem lahko uspešen ali neuspešen, velja, da se da odločanje tudi izboljšati. Napotke za to dajeta Kopačkova in Škrobačkova (2006), ki ugotavljata, da je kar 90% odločitev managementa vnaprej programiranih, saj gre za ponavljajoče se ali rutinirane odločitve. Neprogramirane odločitve srečamo običajno pri kriznih situacijah. Pravita tudi, da je možno najti skupne principe in orodja za boljše odločanje, in predlagata nekaj tehnik:

- **Povečati je potrebno znanje.** Pomanjkanje izkušenj, vezanih na problem, lahko povzroči slabe odločitve.
- **Potrebno je biti kreativen.** Kreativnost je proces razvijanja novih originalnih odzivov na problem. Je bistvena pri odločanju pri odkrivanju alternativ ali definiranju problema. Dosežemo jo s preverjanjem predpostavk in z razmišljanjem med procesom korak po koraku.
- **Uporabljati je potrebno intuicijo.** Intuicija se lahko definira kot miselni proces, pri katerem se nezavestno odločimo na osnovi akumuliranega znanja in izkušenj.
- **Odločati se je potrebno v ustreznem času,** ne v času slabega razpoloženja.
- **Ni si potrebno preveč prizadevati za dokončnost odločitve,** zelo malo odločitev je dokončnih.

Rozman, Kovač in Koletnik (1993) k temu ob navedbi potrebnih lastnosti managerjev dodajo še:

- Logično sklepanje je potrebno v fazi ugotavljanja problemov. Gre za sklepanje o vzrokih in posledicah in o medsebojnih povezavah.
- Sistemski pristop zahteva usklajevalno naravnost ob gledanju na stvari in dogajanja z vidika celote. Prav to slednje je razmeroma redka sposobnost, saj je le malo ljudi, ki so sposobni gledati z vidika celote in videti stvari in dogodke v njihovi povezanosti.

Mihelčič (2008) pravi: »...kako priti do takih informacij, ko pa vemo, da je znotraj in v okolju združb o številnih dejstvih na razpolago mnogo podatkov, ki so ali pa niso »prave surovine« za pripravo zaželenih informacij. Kako torej vedeti, katere informacije potrebujemo pri odločanju in kako jih zaznati (v večjih dragih ter težko pridobljivih in) razpoložljivih podatkih? Odgovor na to nam daje le dobra povezava znanj s področij analiziranja poslovanja in organizacije ter orodij informacijske tehnologije...Od dobrega analitika torej ne pričakujemo samo, da sestavi primerne razpredelnice s podatki, ampak tudi to, da je sposoben na podlagi teh podatkov izrekat kakovostne sodbe in predlagati možne smeri prihodnjega poslovanja. Tu se izkaže kot izjemno pomembna pot analiziranja. Ko si jo enkrat zastavimo, se namreč pravzaprav že delno odločamo za izložke analiziranja -

analizo in sklepe. Način členitve in povezovanja korakov analiziranja je zato pogosto že odločilen za njegov uspeh.«

## 2 POSLOVNA INTELIGENCA

O pomenu BI danes je napisanega veliko. Poslovna inteligenca ne samo analizira preteklost z namenom, da izboljšamo prihodnost, ampak nanjo gledamo predvsem z vidika koristi, ki jih prinese pri podpori odločevalskim procesom, ki so stalnica pri poslovanju podjetij.

### 2.1 Definicija poslovne inteligence

Vsi informacijski sistemi neposredno ali posredno podpirajo odločevalske procese. Če so informacijski sistemi temu posebej namenjeni govorimo o sistemih za podporo odločanju (angl. *decision support systems*, v nadaljevanju *DSS*). Pregled zgodovinskega razvoja BI (poglavje 2.2) jasno kaže, da je BI izšel iz razvoja *DSS* sistemov in je prepletenost očitna in meja zabrisana. Kopačkova in Škrobačkova (2006) sta primerjali definicijo BI in *DSS* (Tabela 1).

*Tabela 1: Primerjava definicij BI in DSS*

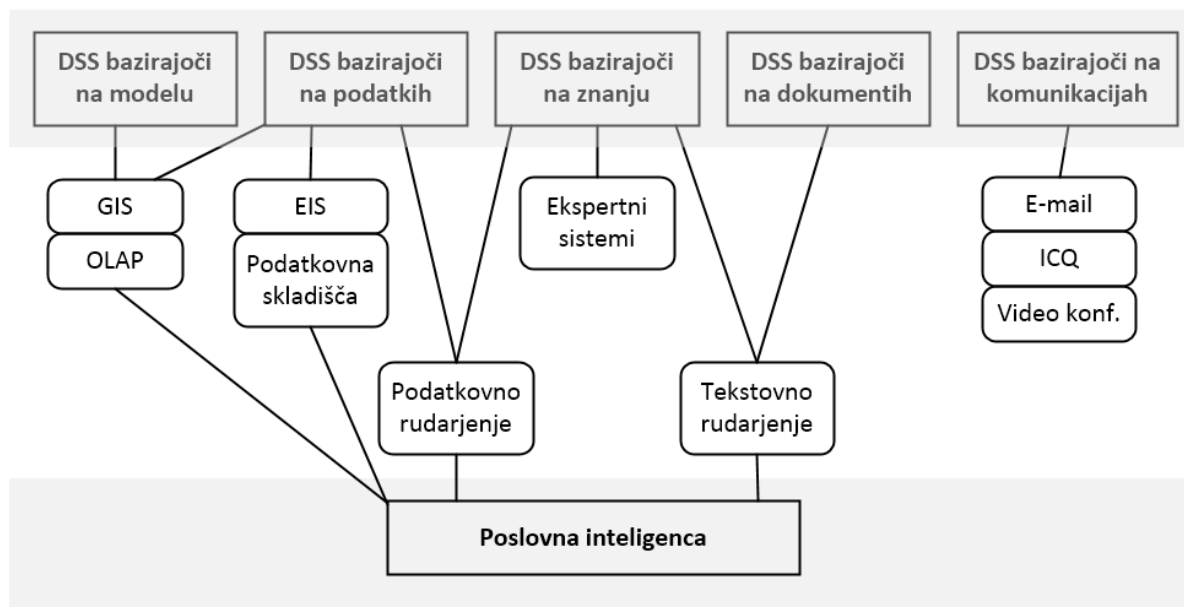
<b>BI</b>	<b>DSS</b>
BI je široka vrsta aplikacij in tehnologij za zbiranje, hranjenje, analiziranje in zagotavljanje dostopa do podatkov z namenom pomagati uporabnikom podjetja pri sprejemanju boljših poslovnih odločitev.	DSS je računalniški program, ki analizira poslovne podatke in jih tako predstavi, da lahko uporabniki lažje sprejemajo poslovne odločitve.

*Vir: Kopačkova & Škrobačkova, Decision support systems or business intelligence: what can help in decision making, 2006, str 99.*

Ugotavljata bistvene enakosti v »sprejemanju poslovnih odločitev« in »poslovnih podatkih«. Oba koncepta sta fokusirana na boljše in lažje sprejemanje poslovnih odločitev. Razliko vidita v zgodovinskem nastanku ter pravita, da znanstvena srenja operira s terminom sistemi za podporo odločanju, ko govorijo o orodjih uporabnih pri odločitvenih procesih, medtem ko proizvajalci programske opreme uporabljajo termin poslovna inteligenca za orodja s podobno funkcionalnostjo. Struktura BI tudi ni stabilna in lahko rešitve uporabijo samo nekatere komponente, ki jih sicer BI pokriva, pač glede na kupčeve potrebe. BI se razvija in prihodnje komponente bodo lahko drugačne. Podata pregledno shemo (Slika 2), ki razloži podobnosti med *DSS* in BI.

Pri pregledu današnjih definicij BI je opaziti, da danes skoraj vsak avtor podaja svojo definicijo BI. Te se tudi hitro spreminjajo in dopolnjujejo skladno z razvojem orodij BI.

Slika 2: BI in DSS komponente



Vir: Kopačkova & Škrobačkova, *Decision support systems or business intelligence: what can help in decision making*, 2006, str. 102.

Howson (2008) definira BI kot niz tehnologij in procesov, ki omogoča ljudem na vseh nivojih v organizaciji dostop in analizo podatkov. Brez ljudi, ki interpretirajo informacije in ukrepajo na njihovi osnovi, poslovna inteligenca ne doseže ničesar. Kasneje tej definiciji doda še del tehnologije: »Poslovna inteligenca je niz tehnologij in procesov, ki omogoča uporabnikom dostop, analizo in raziskovanje podatkov za odločanje. Te tehnologije vključujejo v ozadju sisteme podatkovnega skladiščenja in v ospredju uporabniška orodja.« Pri The Data Warehousing Institute (2014) navedejo, da: »Poslovna inteligenca združuje podatke, tehnologijo, analitiko in človeško znanje z namenom optimizirati poslovne odločitve in voditi uspeh podjetja. BI programi običajno kombinirajo podatke podatkovnega skladišča podjetja in BI platformo ali set orodij za transformiranje podatkov v uporabne poslovne informacije, ki omogočajo ukrepanje.« Mihelčič (2008) pravi, da je poslovna inteligenca: »Spoznavanje - tudi z veliko pomočjo orodij informacijske tehnologije - značilnosti (ter vzrokov) poslovnih stanj, procesov in funkcij v združbah z analiziranjem ter ustrezno podprto obveščanje nosilcev poslovnih odločitev z informacijami, potrebnimi za uveljavljanje potrebnih ukrepov za izboljšanje poslovanja »danes« (taktika) in »jutri« (strategija).« Mihelčič (2008) tudi izpostavlja vlogo pravih in pravočasnih informacij za učinkovito in uspešno odločanje. Takšne informacije odločevalci potrebujejo, in če pridejo do njih pred tekmeci, imajo prednost. Pridobitev in usmeritev takih informacij, vključno z že razpoložljivimi znanjem, k pravim uporabnikom je tisto, kar je lahko odločilno v poslovni tekmi.

Več avtorjev pri definiciji BI označuje kot krovni termin. Eno takih navaja Zaman (2005),

ki pravi, da: »BI ni niti produkt niti sistem. Je krovni termin, ki združuje arhitekture, aplikacije in podatkovne baze. V realnem času omogoča interaktivni dostop, analize in manipulacijo informacij, ki zagotavljajo poslovni skupnosti lahak dostop do poslovnih podatkov. BI analizira zgodovinske podatke, ki ji poslovanje generira preko transakcij ali z ostalimi poslovnimi aktivnostimi – in pomaga z analiziranjem preteklih in sedanjih poslovnih situacij in performans. BI pomaga odločevalcem, da se odločajo bolj informirani in dostavlja končnim uporabnikom kritične informacije o kupcih in partnerjih, vključujoč informacije o obnašanju in trendih.« Kasneje je Zaman (2005) že pretiraval, ko je BI označil kot umetnost vedenja in pridobivanja poslovne prednosti iz podatkov. Leksikon Cankarjeve založbe (1984) pravi, da je umetnost obče ime za različne vrste človekovega oblikovanja in izražanja na raznih področjih (glasba, likovna umetnost, književnost, gledališče, ples itd).

Vsem tem definicijam je skupno to, da vključujejo podatke, informacije in dostavo teh odločevalcem. Redko katera se dotakne tehnologije BI, če že, potem le površno kot »set tehnologij in procesov« ali »kombinira arhitekture, aplikacije in podatkovne baze«. Omemba podatkovnih skladišč je še bolj redka, posebej v novejšem času, saj podatkovna skladišča niso več nujni del infrastrukture BI.

Smatramo, da bi najenostavneje BI definirali kot pretvarjanje podatkov v informacije s ciljem identifikacije novih priložnosti in dostavo informacij pravim osebam v pravem času ob neposredni pomoči informacijske tehnologije. Lahko tudi rečemo, da prodajalci BI danes večinoma uporabljajo kot sinonim za orodja, primerna podpori odločitvenim procesom.

## **2.2 Zgodovinski pregled razvoja poslovne inteligence**

Običajno je povzeti zgodovino kakšne tehnologije relativno enostavno. Ko smo to poizkusili pri BI, se je izkazalo, da temu ni tako, saj v glavnem avtorji opisujejo zgradbo sistemov in delovanje. Verjetno se ravno zaradi raznolikosti procesov, nabora tehnologij in orodij, ki tvorijo BI, avtorji opisu zgodovine izognejo oziroma je o tem napisanega zelo malo. Podobno ugotavlja tudi Quarles van Ufford (2002) v svojem poročilu, ko pravi, da s primerjavo zapisov raznih avtorjev ni prišla do enakih zgodovinskih podatkov. Za ponazoritev problematike obravnave zgodovine BI je po njenem mnenju zelo dober pregled trendov in medsebojnih vplivov na področju podatkovnega skladiščenja (Slika 3).

Vendar nekaj mejnikov vsekakor kaže omeniti. Termin poslovna inteligenca je prvi uporabil Luhn (1958) in sicer kot: »sposobnost razumeti medsebojne odnose med dejstvi na način, da lahko vodimo akcije za doseg željenega cilja«. Razvoj sistemov za podporo odločanju najceloviteje, vsaj do poznih 90-tih, opisuje Power (2007). Povzemamo samo nekaj ključnih mejnikov:

1960 - 1965 Simon napiše knjigo o novi znanosti pri odločitvah v managementu  
»The New Science of Management Decision«. Računalniki IBM 360

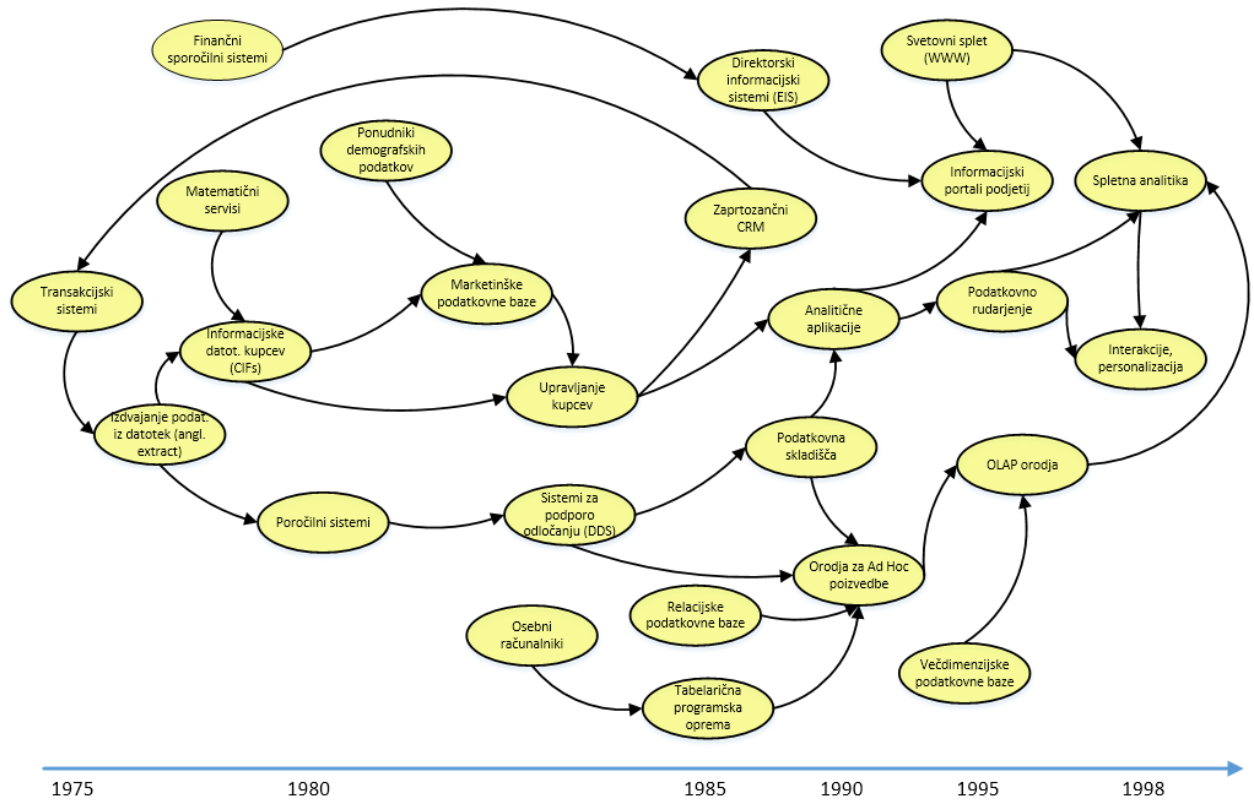
in nasledniki so omogočili razvoj upravljalsko informacijskih sistemov (angl. *management information systems*) za velika podjetja. Ti sistemi so zagotavljali managementu periodična poročila s področij računovodstva in iz transakcijskih procesnih sistemov. Poročila niso bila interaktivna.

- 1966 - 1970 Ferguson in Jones predstavita študijo o sistemu za podporo odločanju pri planiranju proizvodnje ob uporabi računalnika IBM 7094. Predstavljeno je prvo večdimenzijsko analitično orodje IRI Express.
- 1971 - 1975 Meador in Ness objavita članek o sistemu za podporo odločanju pri korporativnem planiranju »DSS Application to Corporate Planning«.
- 1976 - 1980 Sprague in Watson objavita članek o sistemu za podporo odločanju v bankah »A decision Support System for Banks«. Začne se razvoj na managementu informacij in odločitvene podpore (MIDS). Objavljen je prvi program za delo s tabelami, imenovan Visicalc
- 1981 - 1985 Inmon objavi članek o relacijskem sistemu za upravljanje s podatkovnimi bazami. Izide program za delo s tabelami Lotus 123.
- 1986 - 1988 Dobimo program Excel, ki ima že možnost dodajanja grafov. Programska oprema EIS pri Analog Devices omogoči izračune uravnoveženih kazalnikov.
- 1989 Celih 30 let po prvi omembi BI Dresdner predlaga termin BI, ki po principu dežnika pokriva koncepte in metode za izboljšanje procesa poslovnega odločanja z uporabo podpornega sistema, ki temelji na podatkih.
- 1991 Inmon izda knjigo »Building the Data Warehouse and Database Machines and Decision Support Systems«
- 1993 Codd definira 12 temeljnih točk za OLAP procesiranje.
- 1995 Ustanovljen je Data Warehouse Institute (TDWI).
- 1997 Wal-Mart in Teradata kreirata največjo produkcijsko podatkovno skladišče, ki je z velikostjo 24 TB kar trikrat presevalo velikostni razred podatkov na svetovnem spletu.
- 1998 Power definira spletno orientiran sistem za podporo odločanju kot računalniški sistem, ki managerjem ali poslovnim analitikom dostavlja informacije za odločanje ali orodja za podporo odločanju s pomočjo klienta kot Netscape Navigator ali Internet Explorer.

Nadaljnji razvoj je bil zelo hiter. Na trgu se je pojavljalo vedno več ponudnikov BI. Celovitega opisa nismo našli. Še največ lahko bralec izve iz letnih pregledov področja BI, ki ga vsako leto pripravijo pri Gartnerju kot največji analitski hiši na področju informacijskih

tehnologij, raziskav in svetovanj. Zadnji dve poročili so objavili Sallam et al (2015) in Parenteau et al, (2016).

Slika 3: Trendi in vplivi na področju podatkovnega skladiščenja 1975-2000



Vir: Quarles van Ufford, *Business Intelligence-The Umbrella Term*, 2002, str. 13.

## 2.3 Arhitektura poslovne inteligence

Na Sliki 4 vidimo tipično arhitekturo sistema BI. Hiter pogled na sliko pokaže, da so na levi strani različni viri podatkov, na desni strani pa iskane informacije, ki smo jih pridobili iz podatkov. Za podjetja je večinoma najvažnejši vir lastna celovita programska rešitev – transakcijski sistem (angl. *OLTP - online transaction processing*). Vsaj v našem primeru proizvodnega podjetja (poglavje 3) je temu tako. Ne moremo pa to trditi za ves spekter podjetij, sploh ne za spletna prodajna podjetja. Dejstvo je, da se do različnih virov dostopa različno in da so predvsem podatki različnih formatov in strukture, tudi nestrukturirani.

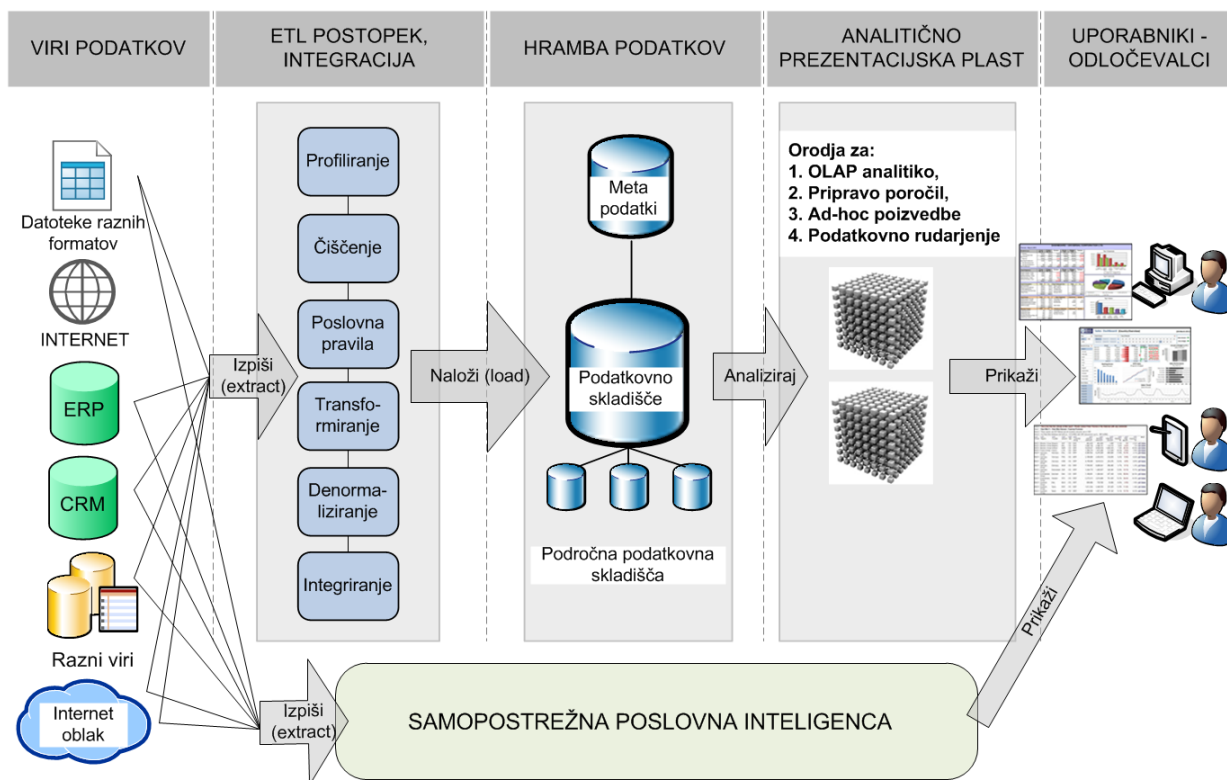
Izpisani podatki iz virov gredo najprej skozi tako imenovani ETL (ang. *Extract, Transform, Load*) postopek, ki ima več faz (Slika 4).

Na sredini imamo podatkovno skladišče (angl. *data warehouse*, v nadaljevanju DW). Kimball in Caserta (2014) ga definirata kot sistem, ki: »Izpiše (angl. *extract*), očisti, prilagodi



(ang. conform) in dostavi izvorne podatke v dimenzijsko hrambo podatkov in potem podpre in izvede poizvedbe in analize za namen odločanja«. Ozko gledano pa je podatkovno skladišče podatkovna baza, ki hrani podatke, namenjene analiziranju in poročanju. Samo po sebi se postavlja vprašanje:

Slika 4: Tipična arhitektura sistema BI



Zakaj podatkovno skladišče sploh rabimo? Del odgovora na to da Howson (2008) z navedbo razlik med OLTP sistemom in podatkovnim skladiščem (Tabela 2).

Drugi del odgovora tiči v dejstvu, da so povečini OLTP sistemi zelo zasedeni. Vsako poizvedovanje jih zaradi veliko povezanih tabel zelo obremenjuje. Zato je bolje, da relevantne podatke prenesemo v podatkovno skladišče in jih tam obdelujemo. Tudi model podatkovnega skladišča je lažje razumljiv za izvajanje poizvedb.

Glede podatkovnega skladišča sta možna dva konceptualna modela: relacijski in dimenzijski (Slika 5). Relacijski model bolj zagovarja Inmon (2005) in dimenzijskega Kimball (2002). Pri relacijskem modelu imamo praviloma normalizirane tabele s kar najpodrobnejšimi - atomičnimi podatki. Tabele so povezane preko ključev in tujih ključev. Pogled na tako povezane podatke je možen iz različnih perspektiv, in to šteje v plus temu modelu. Govorimo o entitetah, atributih in povezavah med tabelami ali o relacijah. Entitete so opazovane stvari ali dogodki, atributi so lastnosti entitete in relacije kreirajo hierarhijo tabel. Poenostavljeno bi lahko tudi povedali, da so entitete vrstice v tabelah, atributi imena stolpcev in relacije

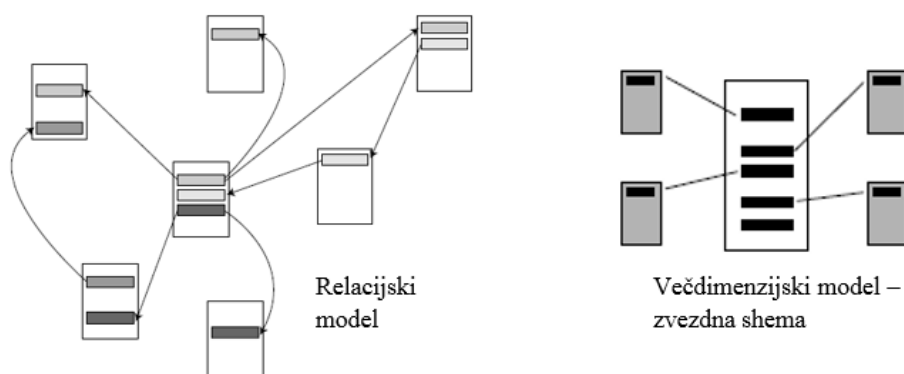
povezave med tabelami. Pridemo lahko do zelo velikih modelov, ki imajo na stotine, tisoče ali celo na stotisoče tabel kot npr. pri eni najbolj znanih celovitih programskih rešitev za podjetja SAP.

Tabela 2: Primerjava OLTP sistema in podatkovnega skladišča

Razlika	OLTP operativni sistem	Podatkovno skladišče
Namen	Izvedba operativnih nalog	Zagotavlja dostop do informacij
Zgodovina	Tekoči podatki z malo zgodovine	Veliko zgodovinskih podatkov. Omogočene analize po obdobjih.
Pravočasnost	Podatki v realnem času	Podatki periodično izpisani. Običajno dnevno. Lahko bolj pogosto, tudi v realnem času.
Nivo podrobnosti	Podrobni podatki	Agregirani podatki z različno stopnjo agregacije
Odzivnost	Hiter vnos a počasne poizvedbe	Hitre poizvedbe. Samo za branje podatkov.
Struktura tabel	Ogromno normaliziranih tabel – v tisočih	Del je denormaliziran. Zvezdna ali snežinkasata oblika. Manj tabel kot OLTP
Dimenzije	Redka uporaba hierarhij	Hierarhije običajne: čas, proizvodi, kupci itd
Poročanje in analiziranje	Fiksna poročila po eni podrobni dimenziji.	Fiksna, ad-hoc poročila in analize po več dimenzijah in po vseh poslovnih funkcijah.

Vir: Povzeto po Howson, *Successful Business Intelligence*, 2008, str. 29.

Slika 5: Relacijski in večdimenzijski model



Vir: Povzeto po Inmon, *Building the data Warehouse 4<sup>th</sup> Edition*, 2005, str. 358 in 360.

Pri večdimenzijskem modelu imamo tabele s podatki dejstev (angl. *fact tables*), kjer hranimo mere oz. vrednosti, ki jih lahko agregiramo in ki nas zanimajo. Te vrednosti opisujejo dimenzijske tabele (angl. *dimension tables*). Te tabele omogočajo kontekst analitičnega pogleda na podatke v tabelah dejstev. Poznamo več dimenzijskih shem. Osnovna in največkrat implementirana je zvezdna shema. Ostale vrste shem dobro opisujeta Moody in

Koritnik (2000). Dimenzijski model lahko naredimo iz relacijskega. Nabor literature za ta namen in predstavitev tehnike opisujeta Rizzi in Golfarelli (2001).

Na Sliki 5 posamezen pravokotnik predstavlja tabelo, potemnjeni vodoravni pravokotniki znotraj pa posamezne attribute.

Pomembno je, da razumemo, kje so razlike med obema modeloma. Najvažnejši razliki sta pri fleksibilnosti in performanci. Relacijski model je zelo fleksibilen, vendar zaradi kompleksnosti ni hiter pri poizvedbah. Možno ga je enostavno razširiti z dodajanjem tabel. Zaradi teh lastnosti se bolj uporablja v OLTP sistemih. Večdimenzijski model je zelo hiter glede poizvedb, vendar ima slabosti pri fleksibilnosti. Ko enkrat zasnujemo strukturo večdimenzijskega modela, predstavlja širitev te strukture težavo. Inmon (2005) predvsem na osnovi fleksibilnosti ugotovlja, da je relacijski model idealen za centralno podatkovno skladišče in večdimenzijski za področna podatkovna skladišča.

Glede pristopa se Inmonov in Kimballov pogled močno razlikujeta. Inmon zagovarja pristop od zgoraj navzdol (angl. *top down approach*). Predlaga izgradnjo centralnega vseobsegajočega podatkovnega skladišča, ki mu lahko sledijo področna podatkovna skladišča kot neke vrste sateliti z namenom pokrivanja področnih potreb posameznih oddelkov podjetja. Podrobnost podatkov naj bo kar največja, saj se nikoli ne ve, kaj se bo potrebovalo za namen poizvedb. Tak pristop zahteva veliko resursov. Na drugi strani Kimball zagovarja pristop od spodaj navzgor (angl. *bottom up approach*) s kreiranjem neodvisnih področnih podatkovnih skladišč glede na potrebe uporabnikov in s kasnejšo integracijo preko skupnih dimenzij. Tak pristop je bistveno hitrejši. To je seveda zagotovo pogodu sponzorju projekta. V nadaljevanju se bomo pri konceptu podatkovnega skladišča omejili na večdimenzijski model, ker je to v praksi najbolj uporabljan model in ker je tako tudi modeliran naš prototip v poglavju 4.

Ko imamo podatkovno skladišče naloženo, lahko izvajamo poizvedbe in analitiko. V analitično prezentacijski plasti zadnji 20 let kraljuje sprotna analitična obdelava podatkov - OLAP (angl. *online analytical processing*). Thomsen (2002) pravi, da ima termin OLAP več pomenov, saj je izražen v več tehnoloških nivojih, od hranjenja do dostopa do podatkov. Na grobo imamo OLAP koncepte, OLAP jezike in OLAP produkte. Howson (2008) razloži, da OLAP premakne fokus z vprašanja »Kaj se dogaja?« na odkrivanje »Zakaj se dogaja«. Uporabniki iščejo informacije in zato vrtijo in vrtajo podatke, da bi odkrili podrobnosti in vzorce. Nekaj bistvenih karakteristik OLAP orodja loči od orodij za poročanje in poizvedovanje:

- večdimenzijskost (Analiziramo lahko numerične vrednosti po več dimenzijah. Poročila so enodimenzijska);
- hitrost (Sprememba pogleda agregatov iz npr. leta na četrtoletje je praktično trenutna.);
- interaktivnost (Vrtenje omogoča pogled na numerične vrednosti iz različnih perspektiv. Tu so še druge OLAP operacije.);

- različni nivoji agregacij;
- navzkrižno dimenzijsko preračunavanje.

Jaklič (2009) navaja OLAP operacije, ki omogočajo odkrivanje informacij iz podatkov v dimenzijskih tabelah in tabelah dejstev:

- vrтанje v globino (angl. *drill down*) (Podatke pogledamo bolj podrobno navzdol po hierarhiji ali dodamo pogled po dimenziji.);
- zviјanje (angl. *roll-up/down*) (Podatke pogledamo manj podrobno navzgor po hierarhiji ali odvzamemo dimenzijo.);
- rezanje (angl. *slice and dice*) (V bistvu filtriramo. Podkocka z izbiro ene dimenzije je *slice* in podkocka z izbiro dveh ali več dimenzij je *dice*.);
- vrtenje (angl. *pivot*) (V pogledu zamenjamo dimenzijo – vrtime kocko);
- vrтанje čez (angl. *drill accross*) (Iz ene tabele dejstev oz. iz ene kocke v drugo. Preko skupne dimenzije.);
- vrтанje skozi (angl. *drill through*) (Pridemo do podrobnih podatkov.);
- primerjave;
- izračuni.

O OLAP obstaja veliko literature, saj je tehnologija v zreli fazi uporabe z veliko dobre prakse. Za poglobljeno razumevanje pa priporočamo pregled del: Harinarayan, Rajaraman in Ullman (1996), Thomsen (2002) ter Wrembel in Koncilia (2007).

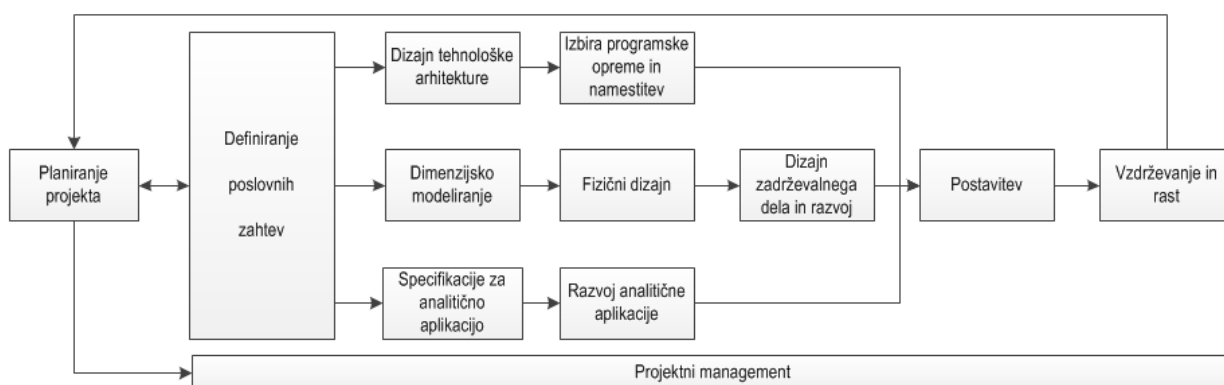
V praksi ni vedno potrebno, da podjetja implementirajo celotno arhitekturo BI sistema. Sodobna BI orodja omogočajo zajem podatkov direktno pri virih. Podatkovna skladišča niso več pogoj. ETL postopek, hrambo podatkov in analitično prezentacijsko plast nadomestimo s tako imenovano samopostrežno poslovno inteligenco (Slika 4). Sodobna orodja so dovolj prijazna ter intuitivna pri uporabi in računalniki so dovolj zmogljivi, da lahko napredni uporabniki sami dostopajo do podatkovnih virov in sestavijo poizvedbe, nadzorne plošče ter poročila oz. pridobijo informacije iz podatkov. Tipični predstavniki danes so: MS PowerPivot, MS Power BI, Tableau Desktop, Qlik Sense, Domo, Loocker, Zoho Reports. Pri Logi Analytics (2014) so v raziskavi o vplivu samopostrežne BI ugotovili, da ima samo 22% poslovnih uporabnikov dostop in uporablja samopostrežno BI. Kar 90% poslovnih in IT uporabnikov se pa strinja, da je za poslovne uporabnike pomembno, da dostopajo do podatkov brez pomoči IT.

## 2.4 Razvoj poslovno inteligenčnega sistema

Razvoj večdimenzijskega sistema BI opisujeta sistematično Kimball in Ross (2002), Slika 6. V tem načrtu so vsebovane tudi faze, ki sta jih pred tem za razvoj podatkovnega skladišča predlagala Golfarelli in Rizzi (1998).

Na Sliki 6 vidimo, da se po dveh uvodnih pomembnih fazah, kot sta planiranje projekta in definiranje poslovnih potreb, razvoj nadaljuje po treh poteh. Zgornja je tehnološka, srednja pot se osredotoča na podatke in podatkovno skladišče in spodnja pot na razvoj analitičnih aplikacij. Ravno ta spodnja pot je z vidika končnega uporabnika zelo pomembna in lahko pomeni tudi uspeh ali neuspeh sistema BI. Ne pomaga niti, da so ostale poti perfektno izvedene. Za časa vseh faz teče projektni management, saj je potrebno projekt spremljati in ga usmerjati.

Slika 6: Diagram življenjskega cikla sistema BI



Vir: R. Kimball & M. Ross, *The Data Warehouse Toolkit, Second Edition, 2002, str. 332.*

V nadaljevanju opisujemo posamezne faze, deloma opise združimo.

### 2.4.1 Planiranje projekta

Kot v večini projektov je tudi pri BI projektih planiranje projekta zelo pomembna faza. Veljajo enake predpostavke. Zaradi specifičnosti kaže opozoriti samo na ključne razlike. Uvedba sistema BI ima lahko na podjetje zelo velik vpliv, saj BI zmanjša čas, potreben za pridobivanje informacij, in tako sprostijo človeške resurse, ki se lahko sedaj ukvarjajo s produktivnejšimi zadevami. Ne nazadnje tudi s korektivnimi ukrepi, če nadzorne plošče kažejo odstopanja. Kimball in Ross (2002) opozarjata na pomembne zadeve:

- močan poslovni sponzor projekta je ključen, ta mora imeti vizijo oz. mora biti prepričan v korist BI;
- motivacijo vseh vpletenih, posebej pa poslovnih uporabnikov pri postavitvi podatkovnega skladišča;
- razpoložljivost tehničnih sredstev, človeških virov in predvsem kvalitetnih podatkov, saj tukaj ni hitre rešitve;
- razumevanje med IT delom organizacije in poslovnimi uporabniki, IT mora razumeti potrebe poslovnih uporabnikov, velja tudi v nasprotno smer;

- analitično kulturo v podjetjih je potrebno naravnati v smer uporabe informacij pri odločanjih.

Loshin (2003) k temu dodaja zelo pomembno določitev diskretnih kriterijev uspeha in ustreznih metrik uspeha. Te vidi na področjih: funkcionalnosti, dostopnosti, hitrosti izvajanja, realizacije zagotovljenih funkcij, kvalitete informacij, dosegljivosti sistema in poslovne ustreznosti. Howson (2008) med skrivnosti uspeha projektov BI šteje vzpostavitev zavedanja med uporabniki, kaj BI prinaša. Ko začne BI sistem že dostavljati informacije iz podatkov, je potrebno pri uporabnikih povečati znanje o tem, kdaj in kako BI uporabljati. Kot tretjo fazo vidi povečanje uporabe BI po vseh nivojih podjetja. Hvala (2011) pa med pogoji za uspešno izvajanje projekta navaja še potrebno razumevanje vodstva podjetja, ki mora: »Razumeti, da se vseh zadev ne da ovrednotiti v denarju in da je potrebno kdaj kaj izvesti, tudi če ni v naprej jasno, kako se bo izšlo. Razumeti morajo, da se bo zelo slabo izšlo, če določenih stvari ne izpeljemo.« in v nadaljevanju: »Kdor ne bo nikoli tvegala, namreč tudi ne bo nikdar ničesar naredil.« To slednje je pri BI projektih še kako pomembno. Izračun donosnosti investicije v BI (angl. *return of investment*, v nadaljevanju ROI) ni povsem brez predpostavk. V skupin TPV smo izračun naredili na osnovi:

- direktnih koristi oz. prihranka v času dela pri: kreiranju poročil, integraciji virov podatkov, administraciji podatkovnih baz, razvoju aplikacij in pri administraciji raznih sistemov za namene pridobivanja informacij,
- posrednih koristih preko oportunitetnih prihrankov,
- pričakovanih koristi: povečanje produktivnosti zaradi hitrejše dosegljivosti, zbiranja in analiziranja podatkov, avtomatizacije pridobivanja in osveževanja podatkov, lažjega planiranja in razvoja BI aplikacij, hitrejšega odzivnega časa in zagotavljanja varnosti. Uvedba BI odpre nove možnosti analiziranja podatkov, ki jih pred tem nismo imeli.

Pri tem nismo ovrednotili pomena ene verzije resnice, ki jo omogoča uvedba BI. Ta je dostikrat lahko odločilna. Je lahko tudi glavni problem nasprotnikov uvedbe BI, saj je po uvedbi skrivanje informacij oteženo.

#### **2.4.2 Definiranje poslovnih zahtev**

Kimball in Ross (2002) podajata nekaj napotkov glede evidentiranja poslovnih zahtev uporabnikov. Najprej je optimalno izvesti uvodni sestanek z uporabniki, na katerem mora poslovni sponzor odigrati svojo vlogo in uporabnikom predstaviti pričakovanja in njihovo vključenost. Nadalje predlagata uporabo hibridnega pristopa z intervjuji, da zberemo zahteve do podrobnosti, vključno s kriteriji za uspeh projekta, in skupne sestanke z namenom dosega konsenza. Snemalnike intervjujev ni potrebno uporabljati, ker motijo dinamiko izvedbe. Na intervjuju je dobro dodati enega ali dva člana tima kot opazovalca. Vse je potrebno zapisati oz. dokumentirati. Predno se pride do odločanja o prioritetah, je potrebno opraviti tudi sestanke s poznavalci izvornih podatkov zaradi določitve izvedljivosti zahtev. Seznam

zahtev je potrebno predstaviti sponzorju in predstavnikom vodstva. Potrebno se je odločiti glede prioritete in ta odločitev ne sme biti sprejeta brez poslovnih uporabnikov. Kimball in Ross (2002) predlagata celo zahteve umestiti v enega od štirih kvadrantov. Pri tem gre x os od majhne do velike možnosti izvedljivosti in y os od majhnega do velikega vpliva na poslovanje. Zahteve, ki bodo prve na vrsti, so tako v zgornjem desnem kvadrantu, saj imajo največji vpliv na poslovanje in jih je možno izvesti.

Da dobro definirane poslovne zahteve zagotavljajo razumljivost in potrebno zgoščenost informacij, ugotavljajo tudi Popovič, Hackney, Coelho in Jaklič (2012). V raziskavi o vplivu zrelosti in kulture pri analitičnem odločanju ugotavljajo, da je relevantnost in točnost informacij (angl. *content quality*) pomembnejša od hitrega procesiranja in dostave informacij (angl. *access quality*).

### **2.4.3 Zasnova tehnološke arhitekture**

Kimball in Ross (2002) sta predvidela 7 korakov:

1. določitev arhitekturnih zahtev, pri čemer so vhod poslovne potrebe zbrane v predhodni fazi, način dela so tehnološko naravnani intervjuji znotraj IT;
2. dokumentiranje arhitekturnih zahtev, ne sme se pozabiti na poslovne zahteve glede dosegljivosti BI. Morda 24/7 (celi dnevi, vse dni v tednu);
3. groba definicija podrobnosti arhitekture;
4. podrobnejša definicija podrobnosti arhitekture, pogosto je že potrebno pregledati tržišče in ponudbo, vključiti je potrebno tudi zahteve po varnosti in razširljivost ter fleksibilnost sistema;
5. določitev faz implementacije;
6. dokumentiranje celotne arhitekture in faz implementacije, da se preda neposrednim izvajalcem;
7. pregled tehnične arhitekture s člani tima, IT in sponzorji.

Rezultat te faze je seznam za nakup tehnične opreme. Pod to so mišljeni predvsem serverji in diskovne kapacitete. Če ustreza, so lahko serverji tudi virtualizirani. V tem delu se lahko ugotovijo morebitne težave pri integriranju v celoto informacijskega sistema podjetja.

### **2.4.4 Izbira produkta, programske opreme in namestitev**

V tem delu je potrebno določiti sodila skupaj z utežmi, po katerih bomo produkte primerjali med sabo. Bolj kot so ta sodila specifična, boljša so, saj splošnim kriterijem verjetno zadosti velik del ponudnikov ali pa kar vsi. Predhodno se je potrebno dobro informirati preko spleta, na konferencah, pri dobaviteljih opreme in opraviti ogled uspešnih in primerljivih, arhitektur kot jih potrebujemo sami. Potem ko v prvem krogu izbora izločimo nekonkurenčne ponudbe, se lahko osredotočimo na podrobnejše preverjanje. Preverljive

reference dobaviteljev pri podobnih projektih so pomembne in jih pregledamo in po možnosti preverimo. Iz dobrih praks izhaja, da je priporočljivo opraviti ogled pri sorodnih podjetjih, kjer so BI že implementirali. Izkušnje iz prakse nam lahko zelo izboljšajo verjetnost pravilne odločitve.

S ponudniki se dogovorimo za izdelavo prototipa na manjšem obsegu podatkov, saj to veliko pove o načinu dela ponudnika in o sposobnostih.

Pri izboru programske opreme moramo biti previdni. Napačen izbor ima lahko odločilne posledice pri uvedbi BI. Prekomplicirana orodja bodo povzročila, da se bodo uporabniki vrnili na star način dela s preglednicami. Po Howson (2008) ni treba stremeti k temu, da bi vsi uporabniki uporabljali ista orodja. Pristop, ki so ga ubrala podjetja z uspešnimi implementacijami BI, je uporabiti pravo orodje za pravega uporabnika. Za managerje so primerne interaktivne nadzorne plošče in interaktivna poročila, naprednim uporabnikom orodja za sestavo poizvedb, drugim uporabnikom pa objavljena poročila.

#### **2.4.5 Dimenzijsko modeliranje**

Kljub temu, da sta Moody in Koritnik (2000) ugotovila, da Kimball metode dimenzijskega modeliranja ni razvil na osnovi teorije, ampak na osnovi opazovanj prakse in brez empiričnega testiranja, je metoda zelo uspešna v praksi. V praksi Kimball in Ross (2002) predlagata dimenzijsko modeliranje izvesti z majhnim timom, v katerem naj bodo: analitik poslovnega sistema, napredni analitik in podatkovni arhitekt (angl. *data modeler*). Naloga tima je določiti potrebno podrobnost podatkov, dejstva in dimenzije. Dizajn je potrebno validirati z drugimi člani tima z IT člani in z uporabniki. Nastala dokumentacija naj definira: tabele, imena stolpcev, računsko pravila za dejstva in pravila za počasi se spreminjajoče dimenzije. Te informacije so tudi prvi vhod za katalog metpodatkov. Osnova, ki jo zagovarjata kot: »dolgoročni uspeh podatkovnega skladišča«, je sestaviti BUS matriko podatkovnega skladišča (Slika 7). Ta matrika je »močno sredstvo za planiranje in komunikacijo«. Arhitektura pri tem ni odvisna od tehnologije in je »racionalni pristop pri dekompoziciji podatkovnega skladišča podjetja«. BUS matriko sestavimo tako, da poiščemo dejstva (poslovni proces), ki nas zanimajo. Poiščemo tudi dimenzijske podatke. Ti niso nič drugega kot podrobnosti o dejstvih. Kot primer, če je dejstvo nabavno naročilo z navedbo proizvoda, količine, datuma naročila, datuma dobave itd., so pri tem v dimenziji zapisani podrobni podatki o prodajalcu: naslov, pariteta, kontaktne osebe in drugo.

Kaj so dejstva in kaj potrebe je odvisno od uporabniških zahtev definiranih v predhodni fazi definiranja poslovnih zahtev. Po Kimball in Ross (2002) BUS matriko kreiramo v 4 korakih:

1. določimo poslovni proces. Če poznamo poslovne zahteve in dostopnost podatkov, ta odločitev ni težka;



2. določimo nivo podrobnosti. Več kot je teh podrobnosti več je kasnejših možnosti različnih agregacij oz. pogledov na podatke. Naraste analitična zmožnost a na drugi strani tudi potreba po resursih: strojna in softverska oprema;
3. izberemo dimenzije. Vpliv prejšnjega koraka je tu velik;
4. izberemo dejstva, podatke, katerih agregacije preko dimenzij nas zanimajo.

Slika 7: BUS matrika podatkovnega skladišča

POSLOVNI PROCES	DIMENZIJE							
	Datum	Proizvod	Trgovina	Pospeševanje prodaje	Skladišče	Prodajalec	Pogodba	Pošiljatelj
Prodaja na drobno	X	X	X	X				
Zaloga prodaje na drobno	X	X	X					
Dobave prodaji na drobno	X	X	X					
Zaloga centralnega skladišča	X	X			X	X		
Dobave centralnemu skladišču	X	X			X	X		
Nabavna naročila	X	X			X	X	X	X

Vir: R. Kimball & M. Ross, *The Data Warehouse Toolkit, Second Edition, 2002, str. 79.*

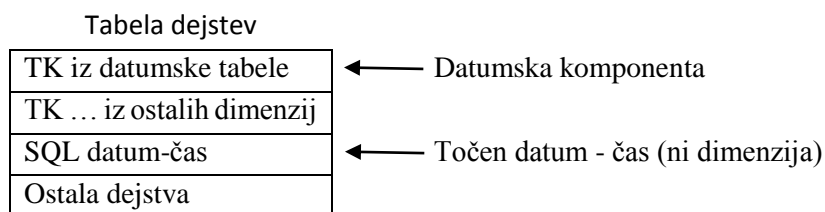
Od vseh dimenzij izstopa **časovna dimenzija**, saj jo najdemo v veliki večini podatkovnih skladišč. Zato jo kaže pogledati podrobneje.

Podatkovna skladišča v tabelah dejstev običajno hranijo kronološke podatke. Tudi večina analiz oziroma vprašanj pri analizah vsebuje kronološko komponento. Zanima nas, koliko smo v določenem času prodali kupcem, koliko porabili materiala, kakšne zaloge smo imeli itd. Kimball in Caserta (2004) pravita, da je najbolj običajna in uporabna datumska dimenzija z granulacijo posameznega dne. Podajata tudi nekaj napotkov, ki jih je potrebno upoštevati, namreč, pri kreiranju datumske dimenzije ne smemo pozabiti vsaj na en zapis, ki bo povezan z zapisi v tabeli dejstev, ki so brez datuma, ki imajo vpisan neveljaven datum ali pa se datum še ni zgodil. Ker je primarni ključ datumske tabele idealno celoštevilčni nadomestni ključ, je smiselno, da ima tak datumski zapis nadomestni ključ, ki gre na konec datumske tabele, npr. 99999999. Razlog je v tem, da če pride do razdelitve datumske tabele (angl. *partitioning*) zaradi hranjenja starejših datumov, je ta zapis za neveljavne ali manjkajoče datume vedno v zadnji aktivni particiji. Dopusčata možnost, da je nadomestni ključ datumske tabele celoštevilčni brez posebnega pomena ali pa neke vrste »samogovoreči«, saj veliko ETL razvijalcev uporablja celoštevilčni ključ, ki je sestavljen iz datuma, npr. 20160216 za datum 16.02.2016. Če je granulacija podatkov v tabeli dejstev mesec,

potrebujemo datumsko mesečno dimenzijo. Izbrati je potrebno ali bo baza za mesečno dimenzijo prvi ali zadnji dan v mesecu.

Če obstaja zahteva po granulaciji znotraj dneva po urah, minutah ali celo sekundah, pa priporočata zasnovo kot na Sliki 8. V tabelo dejstev vključimo podatek o točnem času (angl. *date time stamp*). Časovna dimenzija, ki bi vsebovala vsako sekundo, ni priporočljiva, ker bi bila prevelika. Tudi ni priporočljiva dodatna dimenzija, ki bi vsebovala vsako sekundo znotraj enega dneva, saj je delo z dvema dimenzijama na nivoju dneva in na nivoju sekunde znotraj dneva preveč nepregledno, posebej za več dni.

*Slika 8: Zasnova tabele dejstev kjer je potrebna točno časovno merjenje*



*Vir: R. Kimball & J. Caserta, Data Warehouse ETL Toolkit, Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data, 2004, str. 173.*

Datumske dimenzije kreiramo na začetku projekta na različne načine: Excel, T-SQL ali pa jih uvozimo iz spleta. Število in raznolikost atributov lahko varira in je odvisno od namena in zgradbe našega podatkovnega skladišča npr., če časovna dimenzija nima podatka, kateri dan je praznik, ne moremo pri analizi postaviti vprašanja koliko smo prodali izdelkov na praznične dni od datuma x do datuma y. Običajno vsebujejo attribute:

- primarni ključ,
- datum,
- ime dneva,
- ime meseca,
- zaporedna številka dneva v tednu (1-7),
- zaporedna številka dneva v mesecu (1-21),
- številka tedna v koledarskem letu letu (1-53),
- številka četrletja (1-4),
- indikator, ali je dan vikend,
- indikator, ali je dan praznik.

Če imamo opraviti še s fiskalnim koledarjem, ki je različen od koledarskega leta, se lahko doda še fiskalne attribute kot npr.: številka tedna v fiskalnem letu, številka meseca v fiskalnem letu, fiskalno četrletje in druge potrebne attribute, po katerih bi pri analizah lahko

spraševali. Primer enostavne datumske dimenzije, uporabne na prototipu (poglavje 4) vidimo na Sliki 9.

Slika 9: Datumska dimenzija

	DatumPK	Datum	Datum_SI	DanVMesecu	DanVTednu	DanVLetu	DanIme	JeVikend	Teden	Mesec	Meseclme	Cetrletje	Leto
5890	20160214	2016-02-14	14.02.2016	14	7	45	nedelja	1	7	2	februar	1	2016
5891	20160215	2016-02-15	15.02.2016	15	1	46	ponedeljek	1	8	2	februar	1	2016
5892	20160216	2016-02-16	16.02.2016	16	2	47	torek	0	8	2	februar	1	2016
5893	20160217	2016-02-17	17.02.2016	17	3	48	sreda	0	8	2	februar	1	2016
5894	20160218	2016-02-18	18.02.2016	18	4	49	četrtek	0	8	2	februar	1	2016

## 2.4.6 Fizična zasnova zadrževalnega dela in razvoj

Kimball in Ross (2002) za ETL postopek uporabljata tudi ime zadrževalni del (angl. *staging*). V tem delu preberemo podatke iz virov podatkov in jih pripravimo za nalaganje v podatkovno skladišče. Omeniti je potrebno bistvene korake, ki se v tem delu izvedejo.

1. Pridobivanje, tudi **izpis podatkov** (angl. *extract*) iz virov podatkov. ETL orodja omogočajo branje podatkov iz različnih virov. Tipično iz raznih baz podatkov, tekstovnih datotek, Excel datotek, XML datotek, spleta in drugih virov. S pomočjo tako imenovanih ekstraktorjev pa tudi iz ERP sistemov.
2. **Profiliranje** je pregled kakovosti in razpoložljivosti podatkov. Podatki morajo biti enolični, konsistentni in morajo odražati resnično stanje.
3. **Čiščenje** je korak, ki, kot pravita Kimball in Caserta (2004), »doda vrednost«. Jaklič (2008) opozarja, da podcenjevanje kakovosti lahko privede do tega, da bi podjetje po uvedbi BI lahko odločalo slabo hitreje kot pred uvedbo. Podatke popravimo. Za primerjavo lahko uporabimo validirane sete podatkov. Podatke naredimo konsistentne, da so vedno v enaki obliki. Podatke kompletiramo in jih dopolnimo.
4. **Uvedba poslovnih pravil** pomeni, da neka kombinacija vhodnih podatkov pogojuje specifičen vpis v podatkovno skladišče. Tukaj je potrebna pazljivost, ker se poslovna pravila s časom spreminjajo. Že zapisani podatki v podatkovnem skladišču lahko potem niso več v skladu z novimi pravili, saj so bili procesirani na podlagi starih poslovnih pravil.
5. **Transformiranje** vključuje spremembo tipov podatkov, poenotenje enot mer, aritmetična preračunavanja, poenotenje podatkov o času (12/24 urni format), delo s podatki, ki predstavljajo vrednost nič, ali pa ni podatka.
6. **Denormaliziranje** je korak, v katerem zmanjšamo število tabel. Dimenzijske tabele so lahko denormalizirane, in to je tudi eno od bistev večdimenzijskega modela. Dobimo razumljiv model, ki je primeren za izvajanje poizvedb.
7. **Integriranje** je korak, v katerem povežemo tabele iz raznih virov. Dimenzijske tabele in tabele dejstev povežemo tako, da nadomestni ključ iz dimenzijskih tabel vstavimo kot tuji ključ v tabele dejstev.

8. **Nalaganje v podatkovno skladišče** je poseben izziv pri prvem nalaganju, saj gre večinoma za veliko količino podatkov. Glede na poslovne zahteve potem podatke nalagamo v večinoma periodičnih časovnih intervalih. Nalagamo običajno prirastno – samo nove podatke od zadnjega nalaganja. Prvi razlog je v potrebnem času za to opravilo in drugi je v ohranitvi zgodovinskih podatkov. Posebej je potrebno omeniti popraviljanje podatkov dejstev, kar so čisto običajni primeri. Kimball in Caserta (2004) navajata tri načine: z negiranjem dejstev, s prepisom dejstev in z brisanjem ter ponovnim nalaganjem dejstev. Postopek z negiranjem dejstev ima dobro stran, ker je vedno možen pregled zgodovine in analiza popraviljanja podatkov. Za brisanje in ponovno nalaganje pa pravita, da je »akademsko gledano prepovedano v podatkovnem skladiščenju«. Sicer vseeno postopek podrobno opisujeta z rešitvami, ki to pogojno dovoljujejo.

### **Prirastno nalaganje podatkov v podatkovno skladišče**

Prirastno (angl. *incremental load*) nalaganje podatkov v podatkovno skladišče je po Kimball in Caserta (2004, str 228) proces, ki se izvaja periodično z namenom, da držimo podatkovno skladišče sinhronizirano s podatkovnimi viri. Proces se lahko izvaja intervalno, lahko pa tudi v realnem času. Običajno pa zadostuje dnevno zaradi konsistentnosti poročil znotraj istega dneva. S prirastnim polnjenjem naložimo podatke, ki so novi ali spremenjeni od prejšnjega nalaganja. Večina prirastno nalaganje povezuje s periodičnim časovnim nalaganjem – naložimo samo podatke od zadnjega nalaganja naprej. Pri tem Leonard et al (2012) opozarjajo, da ni vedno tako enostavno, ker so včasih spremembe narejene na zgodovinskih podatkih in včasih tudi ni možno ugotoviti, kdaj so bile spremembe narejene.

Pri nalaganju podatkov je pristop med dimenzijami in tabelami dejstev različen. Razlog je v naravi entitet. Na eni strani so dimenzije manjše in se entitete lahko lahko spremenijo, ali jim poteče veljavnost in jih nadomestimo z novimi ali pa so fiksne. Pri tabelah dejstev pa je največji izziv popraviljanje podatkov.

Leonard et al (2012) opišejo metodologije prirastnega nalaganja podatkov, ki so implementirane v SSIS (SQL Server Integration Services). SCD (Slowly Changing Dimension) komponento podrobneje razloži še Nanda (2011). Vse te metodologije so učinkovite, če so uporabljene na pravi način in pri pravem scenariju. V nadaljevanju bomo opisali pri prototipu (poglavje 4) uporabljena pristopa. Namenoma izpustimo druge možnosti, ki imajo tudi svoje prednosti in slabosti. Gre za uporabo komponente Merge in jezika T-SQL ter za uporabo komponent CDC.

### **Prirastno nalaganje - uporaba običajnih SSIS komponent *Lookup* in *Conditional Split***

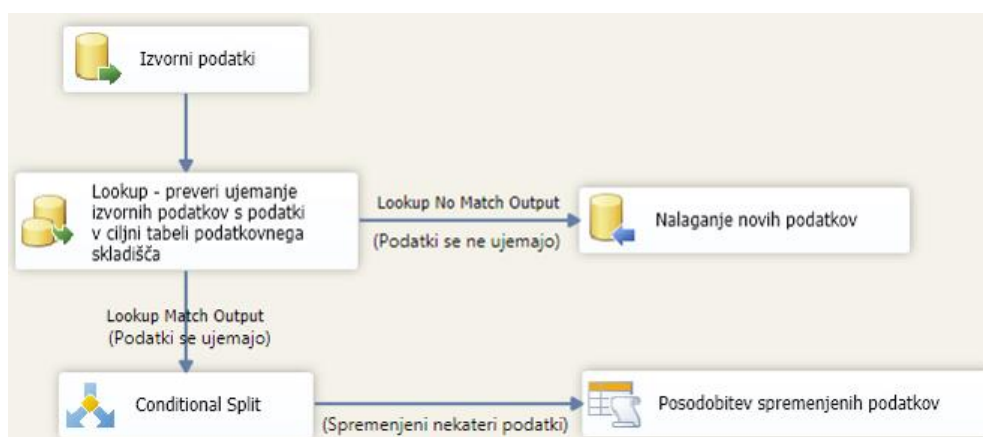
Postopek je znan od uvedbe SSIS v letu 2005 in je velikokrat preverjen (Slika 10). Je fleksibilen in se v praksi, če je pravilno nastavljen, dobro obnaša. Ne zahteva posebnih nastavitvev sistema za upravljanje podatkovnih baz in ne zahteva zunanjih komponent.

Postopek se sestoji iz naslednjih korakov:

1. Preberemo izvirne podatke. Praktično ni omejitev glede virov podatkov.
2. S komponento *Lookup* preverimo ujemanje izvornih podatkov s podatki v ciljni tabeli podatkovnega skladišča. Preverjanje se izvede na atributih, ki sestavljajo naravni ključ.
3. Neujemajoči podatki se naložijo v ciljno tabelo podatkovnega skladišča – gre za nove podatke.
4. Pri ujemajočih podatkih se preverijo še drugi izbrani atributi, ki se lahko spremenijo. Če se ne ujemajo z zapisanimi vrsticami v ciljni tabeli podatkovnega skladišča, se te vrstice posodobijo z novimi podatki.

Postopek je posebno uporaben, če ni možna shramba vmesnih podatkov. Ker se preverjanje podatkov (*Lookup*) izvaja v pomnilniku računalnika, je pri zelo veliko podatkih priporočljivejša uporaba komponente *MERGE* in jezika *T-SQL*.

*Slika 10: Inkrementalno nalaganje z uporabo običajnih SSIS komponent*



*Vir: Prirejeno po Leonard et al, SQL Server 2012 Integration Services Design Patterns, 2012, str. 230.*

### **Prirastno nalaganje - uporaba SCD čarovnika oziroma komponent za počasi se spreminjajoče dimenzije**

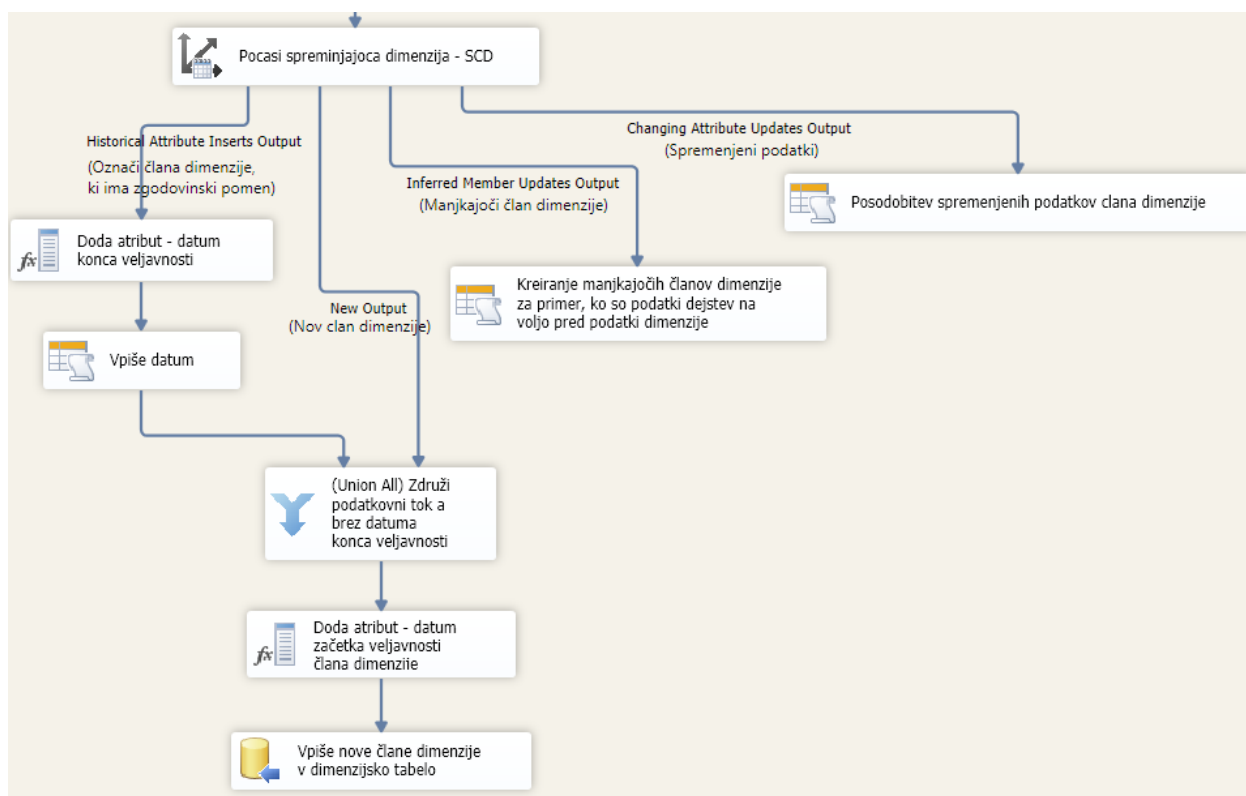
Tudi ta postopek je znan od uvedbe SSIS v letu 2005. Zaradi performančnih lastnosti je primernejši za dimenzijske tabele vendar lahko deluje tudi na tabelah dejstev. Ker smo ta postopek uporabili praktično pri vseh dimenzijskih tabelah prototipa, ga bomo opisali malo podrobneje. Čeprav v nadaljevanju govorimo o dimenzijah, velja enako za tabele dejstev. Osnovna ideja postopka je, da ločimo 3 načine dela pri spremembah vrednosti atributov:

- Tip 0 – **Fiksni atribut.** Ni dovoljena sprememba vrednosti. V praksi bi bila npr. to EMŠO številka. SCD komponenta pregleda prispelo vrstico podatkov oz. njen naravni ključ z vsemi naravnimi ključi v vrsticah dimenzijske tabele. Če ne najde ujemanja, spusti podatke na izhod za nove podatke. Če najde ujemanje, pa preveri, ali ima nova vrstica stolpec s fiksnim atributom. V tem primeru sprememba in zapis v dimenzijsko tabelo ni mogoč. Čarovnik ne kreira toka podatkov naprej. Obstaja pa možnost, da ga ujamemo, ker se taka vrstica pošlje na izhod za fiksne attribute. Komponento lahko nastavimo, da javi napako, če naleti na zahtevo po spremembi fiksnega atributa.
- Tip 1 – **Spremenljivi atribut.** Dovoljena je sprememba vrednosti atributa. Ob spremembi posodobimo vrednost atributa oziroma staro vrednost atributa prepisemo z novo vrednostjo. Zgodovina se ne ohrani, imamo vedno samo zadnjo vrednost. V praksi je to npr. telefonska številka kontaktne osebe. Običajno stare številke ni potrebno hraniti. V primeru, ko SCD komponenta najde ujemanje med naravnim ključem prispele vrstice podatkov in naravnim ključem ene od vrstic v dimenzijski tabeli ter ima nova vrstica stolpec s spremenljivim atributom, pošlje podatkovni tok na izhod za spremenljive attribute. Čarovnik doda še OLE DB komponento, ki izvede posodobitev spremenjenih vrednosti atributa (Slika 11). V OLE DB komponenti je v bistvu zapisan SQL ukaz za posodobitev vrednosti atributa člana dimenzije.
- Tip 2 – **Zgodovinski atribut.** Ob spremembi ohranimo staro vrstico v tabeli dimenzije in dodamo novo vrstico, ki ima isti naravni ključ vendar drugi nadomestni ključ, ki je kar zaporedno celo število vrstice v dimenzijski tabeli – za to skrbi kar SQL Server. Stara vrstica se ohrani. Da je njena veljavnost potekla se naredi vpis datuma v stolpec »veljavnost do« ali naredi vpis v stolpec z indikatorjem veljavnosti. V praksi je to npr. sprememba teže izdelka, ko je treba star podatek o teži ohraniti. V primeru, ko SCD komponenta najde ujemanje med naravnim ključem prispele vrstice podatkov in naravnim ključem ene od vrstic v dimenzijski tabeli ter ima nova vrstica stolpec z zgodovinskim atributom, pošlje podatkovni tok na izhod za zgodovinske attribute in na izhod za nove vrstice (Slika 11). V podatkovni tok po izhodu za zgodovinske attribute čarovnik doda komponento, ki doda in vpiše v stolpec datum konca veljavnosti člana dimenzije. V podatkovni tok za nove vrstice pa čarovnik doda komponento, ki združi oba podatkovna tokova in še komponento, ki doda stolpec datum začetka veljavnosti in ga tudi vpiše. Na koncu imamo še in komponento OLE DB, ki vse vpiše v dimenzijsko tabelo.

SCD komponenta ima kar 6 izhodov in sicer za vrstice podatkov s:

- spremenljivimi atributi (*angl. changing attributes*) – prepíšejo se stare vrednosti atributa;
- fiksnimi atributi (*angl. fixed attributes*) – sprememba vrednosti atributa ni dovoljena;
- zgodovinskimi atributi (*angl. historical attributes*) – ohranimo stare vrstice in dodamo nove;
- manjkajoče člane (*angl. inferred member*) – primeri, ko so podatki tabel dejstev na voljo pred podatki tabele dimenzije;
- nove vrstice (*angl. new output*) – gre za popolnoma nove vrstice ali vrstice, ki imajo zgodovinske attribute in je na teh atributih prišlo do spremembe vrednosti. Vrstica s starimi vrednostmi se ohrani in doda nova vrstice z novimi podatki;
- nespremenjen izhod (*angl. unchanged output*) – ni spremembe vrednosti atributov. Dejansko gre za vrstico za katero SCD komponenta v dimenzijski tabeli najde povsem enako. V dimenzijsko tabelo se ne vpiše ponovno, ker bi to pomenilo podvojitev naravnega ključa.

*Slika 11: SCD komponente za primer spremenljivih in zgodovinskih atributov ob podpori prezgodaj prispelih podatkov tabele dejstev*



V praksi uporaba komponente SCD poteka tako, da nas skozi postopek vodi čarovnik, ki pomaga nastaviti SCD parametre, od katerih potem zavisi kombinacija komponent SSIS. Važno je, da pravilno izberemo, kateri atributi so fiksni (tip 0) in se sprememba manifestira

kot napaka, kateri atributi se lahko spreminjajo (tip 1) in imamo vedno samo zadnjo veljavno vrednost in kateri atributi so zgodovinski (tip 2) in se zapiše nova vrstica ter stara ohrani.

Uporaba SCD je relativno preprosta in primerna za manjšo količino podatkov. Največja zamera so tako performanse in destruktivno obnašanje v primerih sprememb. Če npr. prekonfiguriramo SCD komponento (npr. spremenimo tip atributov), se vsi deli ponastavijo in zgubimo morebiti vgrajene svoje komponente v podatkovni tok. Upajmo, da bo to obnašanje Microsoft kmalu odpravil.

Pri sestavi prototipa podatkovnega skladišča (poglavje 4) smo komponento uporabili pri vseh dimenzijah ravno zaradi fleksibilnosti saj pokriva vsa možne lastnosti atributov.

#### **2.4.7 Specifikacije za analitične aplikacije in razvoj analitičnih aplikacij**

Največje delo glede razvoja BI je do tega trenutka narejeno. Kimbal in Ross (2002) predlagata najprej narediti standarde za aplikacije kot tudi videz menijev in nato 10-15 analitičnih aplikacij, skladno s prioriteta. Barve v grafih so zelo pomembne, tudi videz, saj je važno podajanje informacij, ne pa nepotrebni okraski.

Razvoj analitičnih aplikacij je možen takoj ko imamo vse pripravljeno: zahteve, naloženo podatkovno skladišče in pripravljene metapodatke. Sicer tukaj Kimball in Ross (2002) priporočata z razvojem analitičnih aplikacij začeti že po končnem zadrževalnem delu podatkovnega skladišča, saj se lahko pokažejo še kakšni podatkovni problemi.

#### **2.4.8 Postavitev, vzdrževanje in nadaljnji razvoj podatkovnega skladišča**

Postavitev je mišljena kot polnjenje podatkovnega skladišča in aktiviranje uporabniških orodij. Mora biti dobro planirana in zahteva potem ustreznosti, najprej od projektnega tima in potem še od omejenega števila naprednih uporabnikov. V to fazo se všteva tudi potrebno izobraževanje uporabnikov.

Glede vzdrževanja sodobna orodja že omogočajo spremljanje uporabe posameznih nadzornih plošč, poročil, dostope do podatkovnega skladišča. Izkaže se, da lahko kasneje tudi optimiziramo podatkovno skladišče, če se kake izračunane mere niso potrebne – jih ne uporabljamo. Velja tudi za dimenzije. Če na drugi strani ugotovimo veliko frekvenco uporabe, se lahko posvetimo ciljnim optimizacijam. Podatkovno skladišče lahko smatramo kot infrastrukturo, za katero običajno skrbijo oddelki informatike. Mora imeti določeno vzdrževanje, tudi v smislu stalnega merjenja performans. BI projekt praktično nikoli ni končan, saj se bodo vedno kazale nove potrebe po informacijah. Zato je povratna zanka v diagramu življenjskega cikla sistema BI še kako smiselna (Slika 6).

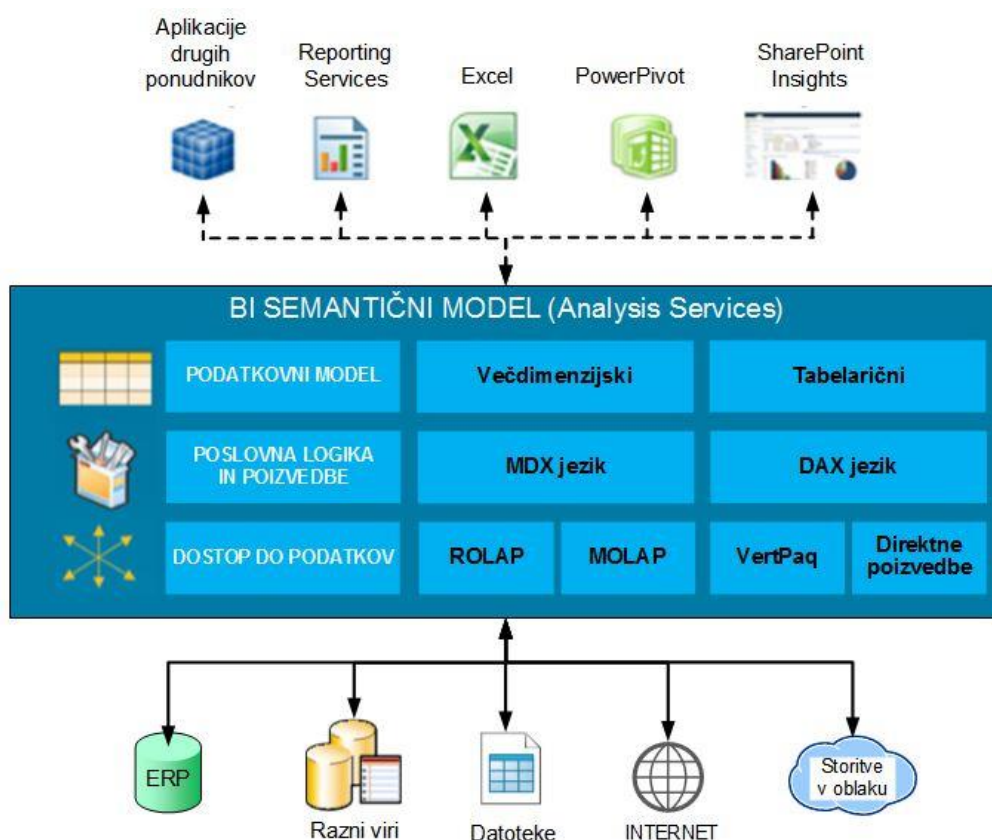
Pri rasti je pomembno, da kontinuirano nadgrajujemo znanja BI in spremljamo razvoj BI. V to je potrebno vključiti posebej uporabnike, saj so večinoma oni generator novih uporab sistema BI. Rast je v bistvu ciklično ponavljanje vseh faz v diagramu življenjskega cikla sistema BI.



## 2.5 Tabelarični model znotraj BI semantičnega modela

Zaradi vedno hitreje spreminjajočih se potreb in sprememb v okolju je prišlo do spoznanja, da so pri pridobivanju informacij oddelki informatike ozko grlo. Microsoft je na potrebo po samopostrežni BI odgovoril leta 2010, ko je predstavil PowerPivot dodatek (angl. *add on*) k Excel in jezik DAX (*Data Analysis Expressions*). Kreiranje modelov tako ni bilo več samo v domeni oddelkov informatike. Napredni poslovni uporabniki so lahko končno enostavno sami kreirali podatkovne modele. Če jo šlo na začetku bolj za tipanje tržnega potenciala, je resnično pravo vrednost prinesla izdaja SQL Server 2012 s katero smo dobili tudi nov pojem **BI semantični model** (angl. *BI Semantic Model* ali krajše BISM). Microsoft je s tem, ko je večdimenzijski in tabelarični model združil pod enim imenom BISM, naredil tudi nekaj zmede, saj imamo eno ime za dva različna modela. O tem, katerega bomo uporabili, se moramo odločiti čisto na začetku, saj prehod med modeloma še ni mogoč. Tako imamo eno tehnologijo in dve metodologiji, večdimenzijsko in tabelarično. Arhitekturo BI semantičnega modela vidimo na Sliki 12.

Slika 12: Arhitektura BI Semantičnega modela



Vir: Harinath et al, *Professional Microsoft SQL Server 2012 Analysis Services with MDX and DAX*, 2012.

Model je konceptualno sestavljen iz treh nivojev:

### 1. Nivo dostopa do podatkov.

Pri večdimenzijskem modelu imamo dve možnosti hranjenja podatkov. Pri MOLAP (angl. *Multidimensional Online Analytical Processing*) načinu so podatki shranjeni v podatkovnem skladišču, ravno tako osnovni in agregirani podatki kocke. Pri ROLAP (angl. *Relational Online Analytical Processing*) pa kocka ni zgrajena, ampak je zgrajen in shranjen samo opis kocke. Poizvedbe gredo direktno do podatkov.

Pri tabelaričnem modelu jedro xVelocity (bivši VertiPaq) vse podatke drži v pomnilniku računalnika, tipično kompresirane do faktorja 10x (več v poglavju 2.5.1), zato omogoča hitre poizvedbe. Ni potrebe po indeksiranju ali agregiranju vnaprej. Omogočeno je tudi tako imenovano *DirectQuery* poizvedovanje, če je podatkov preveč za v pomnilnik računalnika. Pogoji pri tem pa je, da so podatki v SQL Serverju.

### 2. Nivo poslovne logike in poizvedb.

Pri poslovni logiki in pri poizvedbah imamo dva jezika, ki sta primerna za poizvedbe vsak za svoj model:

- a) MDX (angl. *MultiDimensional eXpressions*) je jezik za poizvedbe na zbirki podatkov OLAP. MDX kot jezik večdimenzionalnih izrazov določa posebno sintakso za poizvedovanje nad množico večdimenzionalnih podatkov shranjenih v OLAP kockah. Na videz je podoben tradicionalnem SQL jeziku, zato je mnogo MDX izrazov mogoče prevesti v tradicionalni SQL. Vendar se podobnost kmalu neha, kajti praviloma je več vrstic kode v SQL-u zamenjamo z eno samo vrstico v MDX. (Dular in Godnov, 2013);
- b) DAX (angl. *Data Analysis Expressions*) je nov jezik, navidezno podoben funkcijam v Excel. DAX uporablja tabelarične konstrukte (tabele, stolpce in relacije) in vsebuje funkcionalnosti za podporo razvoju modelov. DAX je posebej opisan v poglavju 2.5.3.

### 3. Nivo podatkovnega modela.

V razvojnem okolju SQL Server Analysis Services (v nadaljevanju SSAS) lahko zgradimo večdimenzijski model, ki ga sestavljajo dimenzije in mere ali tabelarični model iz tabel, stolpcev in relacij med tabelami. Za uporabnika, če uporablja Excel kot odjemalec, praktično ni razlik, saj do obeh modelov pristopa preko vrtilnih tabel.

#### 2.5.1 Hranjenje podatkov

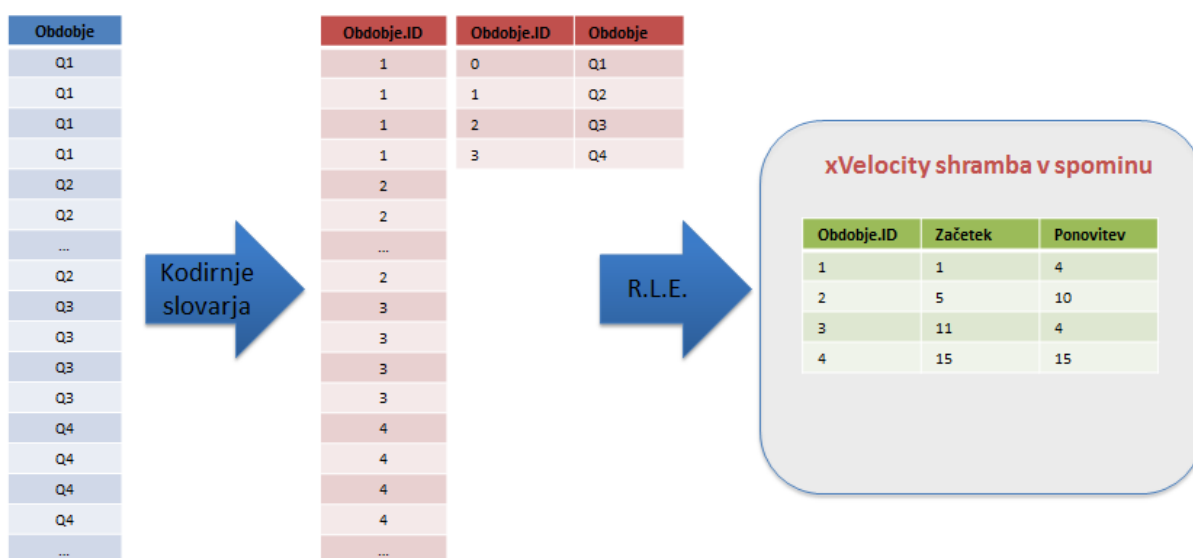
Tabelarični model je ob izvajanju poizvedb v pomnilniku računalnika (angl. *in-memory*). Jedro imenovano xVelocity zagotavlja hranjenje podatkov in hitre poizvedbe. Hranjenje podatkov je izvedeno po stolpcih in tako optimizirano namenu BI, to je agregiranju podatkov. Večinoma se sprašujemo po merah v stolpcih tabel, npr. kakšen je bil znesek prodaje, pri čemer je znesek v istem stolpcu. Imamo tako imenovano podatkovno bazo, ki hrani podatke po stolpcih (ang. *columnar database*). Princip ni nov, izvira iz 60. let prejšnjega stoletja. Prva sta ga predstavila Estabrook in Brill (1969). Hinkka (2012) razlaga,

da agregiranje vrednosti v stolpcih zahteva, da sistem prebere samo stolpec, namesto, da prebere vse vrstice in za vsako izpiše zahtevano vrednost stolpca in šele potem naredi agregatni izračun. Lahko se uporabi tudi kompresiranje podatkov in tako premakne obremenitev iz počasnih operacij dostopa (angl. *I/O*) na obremenitev hitrega procesorja. Kako so podatki shranjeni v tabelaričnem modelu, točneje razlaga Russo (2013b). Gre za dve tehniki (Slika 13):

- Tako imenovano *Run Length Encoding* - R.L.E. metodo, s katero se v pomnilnik ne shranijo celi stolpci, ampak samo različne vrednosti ter začetek in konec, ko se pojavi ta vrednost.
- Kodiranje slovarja, bistveno se zmanjša število tekstovnih zapisov, saj se vsak zapiše samo enkrat.

Stopnja kompresiranja tako doseže ca 10 glede na velikost nekomprimirane SQL baze podatkov. Vitt in Cameron (2012) pojasnjujeta, da stopnja kompresiranja zavisi kardinalnosti podatkov. Visoka kardinalnost pomeni, da so podatki znotraj stolpca zelo različni. Nizka kardinalnost na drugi strani pa pomeni, da se podatki znotraj stolpca lahko ponavljajo. Nižja kot je kardinalnost, večja je stopnja kompresiranja

Slika 13: V pomnilniku se hranijo komprimirani podatki stolpcev



Vir: Povzeto po Russo, *Optimizing Data Models for Tabular Solutions and PowerPivot*, 2013, str.2.

## 2.5.2 Tabele

V tabelaričnem modelu ne obstaja koncept dimenzij in mer. Namesto tega so podatki organizirani v tabele, ki imajo med sabo relacijske povezave. (Vitt & Cameron, 2012). Torej vse tabele so enakovredne. Samo naša odločitev je, kaj so mere in kaj podatki dejstev.

Obstaja pa nekaj posebnosti, ki jih je potrebno upoštevati pri tabelaričnem modelu. Te navajajo Russo, Ferrari in Webb (2012b):

1. Prva posebnost je, da mora za uporabo funkcij časovne inteligence (več v poglavju 2.5.3) datumska tabela vsebovati vse dneve v letu.
2. Druga posebnost je, da je v trenutni verziji tabelaričnega modela (SQL 2014) pri več povezavah med dvema tabelama lahko aktivna samo ena. Kot aktivno je smotrno izbrati tisto, ki najbolj uporabljana. Neaktivne povezave je moč aktivirati s funkcijo USERRELATIONSHIP. Vendar je potrebno za vsako kalkulacijo, vezano na neaktivne povezave, narediti svojo mero. To pomeni toliko dodatnih mer, kot je preračunov raznih agregatov po vseh datumskih stolpcih. Rešitev je lahko, da se naredi toliko tabel, kot je povezav – multipliciranje tabel. Velja opozorilo, da primerno poimenujemo tabele in stolpce tabel tako, da so imena razpoznavna v poročilih. Priporoča se uporaba prefiksov npr. poimenovati datumsko tabelo, ki je povezana z datumom pošiljanja kot »Datum posiljanja«, in datumsko tabelo, ki je povezana z datumom naročila kot »Datum narocila«.
3. Pri delu z manjkajočimi ali napačnimi datumi predlagajo v datumsko tabelo dodati poseben zapis in zapis NULL v stolpec z datumom.

K točki 2 naj dodamo, da v primeru datumskih tabel z multipliciranjem letnih dobimo še možnost selekcije zapisov, ki so presek med različnimi datumskimi obsegi. Pri primeru Slika 33 so med tabelama DimDatum in FactMC9 tri povezave. Aktivna je samo ena. V tabeli *FactMC9* imamo namreč tri datumske attribute. Če bi vsaka povezava imela svojo datumsko tabelo, bi lahko enostavno naredili poizvedbo, koliko materiala smo izdali v določenem obdobju, katerega zadnji prevzem je bil v nekem drugem obdobju.

K točki 3 povemo, da smo to v ETL postopku poizkusili a se potem datumske tabele ni dalo v SSAS označiti, da gre za datumsko tabelo, saj stolpec ni imel zvezno tekočih datumov. NULL ni datum. Torej bi odpadle DAX funkcije časovne inteligence. Te se sicer da zapisati s standardnimi DAX funkcijami, a izrazi lahko hitro postanejo nepregledno dolgi. Več o rešitvi tega problema v poglavju 2.5.3.

### **2.5.3 Jezik DAX**

Jezik DAX (angl. *Data Analysis Expressions*) je poizvedovalni jezik. Za izvedbo operacij uporablja funkcije. Na prvi pogled so videti kot razširitev Excel-ovih funkcij a tu se tudi vsa podobnost neha. Funkcije so namenjene analiziranju podatkov pri povezanih tabelah in dinamični analizi, ki zavisi od konteksta pogleda. Seznam funkcij je najti na veliko mestih. Tudi v literaturi, ki jo ocenjujemo kot potrebno za razumevanje jezika: Russo in Ferrari (2011), Dickerman in Myers (2011) ter na zahtevnejši stopnji Russo, Ferrari in Webb (2012b). Priporočljivo je tudi spletišče [www.sqlbi.com](http://www.sqlbi.com), ki nudi veliko praktičnih primerov in opisov zahtevnejših rešitev.

Izpostavimo naj samo nekaj lastnosti jezika DAX, saj je za delo in razumevanje potrebno pregledati uvodoma priporočeno literaturo.

V DAX se večinoma uporablja popolnoma kvalificirane reference. Te niso v obliki naslova celice kot v Excel, npr. B6, ampak kot kvalificirana imena stolpcev 'Ime tabele'[ImeStolpca] ali kot kvalificirana imena mer 'Ime tabele'[ImeMere]. Nekvalificirana imena – samo ime znotraj oglatega oklepaja so možna. Pri stolpcih je dopustno, če je izračun znotraj iste tabele. Pri merah pa običajno niso potrebna kvalificirane reference, vsaj dokler meram dajemo različna imena.

DAX formule se uporabijo z namenom, da definirajo polja, ki se prikažejo v Excel vrtilni tabeli. V osnovi poznamo dva tipa polj: izračunane stolpce in izračunane mere. **Izračunani stolpci** so stolpci, ki jih definiramo v SSAS ali v PowerPivot oknu na način, da stolpcu določimo ime in vpišemo DAX formulo, ki se izračuna za vse vrstice tabele. Izračunani stolpci se izračunajo ob nalaganju podatkov v model (ang. *processing time*) takoj, ko so naloženi regularni stolpci. Obnašajo se kot vsi drug stolpci, v vrtilnih tabelah jih lahko uporabimo v vrsticah, stolpcih, filtrih ali pa kot sestavni del izračunavanja, ki definira mere. **Izračunane mere** so formule, ki jim dodelimo ime in so sestavljene iz DAX funkcij. V vrtilni tabeli sodijo na mesto, kjer prikazujemo vrednosti. Rezultat formule se izračuna za vsako celico v območju vrednosti in sicer glede na kontekst, ki ga določajo v vrtilni tabeli filtri, vrstice in stolpci. Je pa zadeva dosti bolj fleksibilna in na rezultat lahko vplivajo še filtri vgrajeni v DAX formuli.

Poglavitni pogoj za delo s tabelaričnim modelom je dobro razumevanje funkcionalnosti vrtilnih tabel. Pri DAX moramo razumeti dva bistvena pogleda – konteksta, ki določata kako se izračunajo DAX izrazi. O **kontekstu vrstice** (angl. *row context*) govorimo, kadar se formula izračuna za vsako vrstico tabele. Tipičen tak primer so izračunani stolpci. Bolj zahteven je **kontekst filtrov** (angl. *filter context*). Vsaka celica v območju vrednosti se po formuli, s katero zapišemo izračunano mero, izračuna glede na filtre, ti so: vrstice, stolpci, filtri in razčlenjevalniki.

Jezik DAX ima pri verziji SQL 2012 naslednje število funkcij: statistične (28), funkcije za transformacijo tabel (7), logične (8), informacijske (15), matematične (24), za obdelavo teksta (17), časovne in datumske (17), filter in funkcije za določanje vrednosti (19) in funkcije časovne inteligence (35). Pri SQL 2016 smo dobili še 50 dodatnih funkcij.

Od celotnega nabora naj na tem mestu poudarimo samo funkcijo CALCULATE, ki ima sintakso: CALCULATE (<izraz>; <filter1>; <filter2>;...<filterN>)

Funkcija izračuna <izraz> - formulo za mero. Lahko je <izraz> neka druga DAX funkcija (npr. SUM) nad podatki, ki jih določijo – izberejo filtri <filter1>... Rezultat se preračuna glede na kontekst filtrov prikaza vrtilne tabele. Če povemo drugače, kontekst filtrov prikaza v vrtilni tabeli izbere podatke, nad katerimi se izračuna <izraz> pri tem se pa še dodatno upoštevajo filtri določeni v <filter1>...

Jezik DAX je poizvedovalni jezik. Izvaja se zelo hitro, lahko se ga je naučiti in krivulja učenja je ugodna. Po drugi strani pa ga je težko res obvladati. Zahteva tudi veliko truda za doseg optimalnih rezultatov. Jezik je nov in nov je tudi koncept. Na drugi strani pa zahteva manj znanja modeliranja podatkovnih baz. DAX formule so dinamične in se izvajajo glede na kontekst pogleda. Argumenti funkcij so večinoma stolpci. To dvoje zahteva nov pogled v pisanje formul in dejansko traja nekaj časa, da se tega navadimo. Potrebno je razmišljati drugače.

Tako imenovane **funkcije časovne inteligence** so ene najpomembnejši funkcij, saj nam omogočajo primerjave ali izračune vrednosti v različnih časovnih obdobjih. V SSAS 2012 lahko naštejemo 35 funkcij časovne inteligence. Ločimo:

- funkcije, ki vrnejo samo en datum, običajno se uporabijo kot argument k drugim funkcijam, primera funkciji:
  - LASTDATE (ImeStolpcaZDatumi) in
  - ENDOFMONTH (ImeStolpcaZDatumi) ter
- funkcije, ki vrnejo tabelo datumov, običajno se uporabijo kot filter argument pri drugih funkcijah, večinoma pri funkciji CALCULATE, primera:
  - DATESBETWEEN (ImeStolpcaZDatumi; ZacetniDatum; KoncniDatum) in
  - DATEADD (ImeStolpcaZDatumi; SteviloIntervalov; Interval).

Prodajo predhodnjega leta povezanih tabel *Prodaja* in *DimDatum* izračunano s pomočjo funkcije DATEADD, ki je v bistvu filter in v tem primeru premakne kontekst pogleda datumov od nastavljenega v filtru, ali razčlenjevalniku ali na časovni osi na isti obseg datumov ampak eno leto prej. Funkcija SUM sešteje vrednosti prodaje v stolpcu [Znesek].

ProdajaPredhLeta:=

```
CALCULATE(SUM(Prodaja[Znesek]); DATEADD(DimDatum[Datum]; -1; Year))
```

Pogoj za delovanje funkcij časovne inteligence je kontinuiran potek datumov v datumski tabeli ali v datumskem stolpcu. To je obenem tudi ena od omejitev pri tabelaričnem modelu. Ta problem se posebej pokaže pri delu z manjkajočimi ali napačnimi datumi, ko običajno za tak datum v datumsko tabelo vpišemo nadomestni ključ -1 in kot datum postavimo vrednost NULL. To v tabelaričnem modelu ne gre. Izbor možnosti predstavlja Russo (2013a), predlaga:

- namesto NULL vpis realnega datuma, ki je izven obsega dejanskih datumov v tabeli dejstev, npr. 01.01.1900, a to rešitev ne vidi kot dobro, saj bo končni uporabnik datum videl kot dejanski čeprav ga v tabeli dejstev ni;
- v tabelarični model prenesti datumsko tabelo a brez NULL zapisa. Pri prenosu predlaga uporabo poizvedbe SELECT \* FROM DimDatum WHERE Datumi IS NOT NULL. Pri tem je *DimDatum* ime datumske tabele in *Datumi* ime stolpca z datumi. Grupiranje po

takih datumih se v Excel vrtilih tabelah vidi kot zapisi *blank*. Mesta takega zapisa pri prikazu ni možno kontrolirati.

#### 2.5.4 Optimizacija tabelaričnega modela

Ker se tabelarični model izvaja v pomnilniku računalnika in pomnilnika ni nikoli dovolj, je potrebno poznati in uporabljati optimizacijske pristope, predvsem tiste, ki zmanjšujejo porabo pomnilnika ali pa procesorski čas za izvedbo poizvedbe in preračunavanj. Nabor dobrih napotkov se stalno povečuje. Sirmon, Galloway, Gross in Gulati (2013) natančno razlagajo ozadje tabelaričnega modela in obsežen nabor ukrepov. Navedemo naj samo nekatere, ki jih večinoma upoštevamo tudi pri razvoju prototipa v magistrskem delu:

- Zvezdna shema upoštevajoč modeliranje po Kimball, je optimalni model. Modeli na podatkih iz OLTP sistema v tretji normalni formi (3NF) se tipično obnašajo performančno slabše.
- V model naložimo samo stolpce, ki so analitično relevantni. Zmanjšanje potrebnega pomnilnika za model omogoči več pomnilnika za poizvedbe, procesiranje in predpomnilnik (angl. *cache*).
- Poenostavimo kompleksne izračune s premikom v izračunane stolpce. Tako v primeru počasnih poizvedb krivce – kompleksne izračune prenesemo v izračunane stolpce, ki se izračunajo v času procesiranja oziroma nalaganja podatkov modela. Ta predlog se v primeru prototipa ni obnesel.
- Izogibati se je potrebno funkcijam, namenjenim detekciji napak ISERROR in IFERROR, saj zelo poslabšata performance, saj se po detekciji napake naprej poizvedba odvija po celica-po-celica načinu.
- Uporaba funkcije ADDCOLUMNS je učinkovitejša kot uporaba funkcije SUMMARIZE, razen, če je potrebno dodati filtre na več mest v poizvedbi.
- Uporaba funkcije CALCULATE je učinkovitejša kot uporaba funkcije FILTER, saj postavi kontekst filtra za vse operacije znotraj oklepaja.
- Potrebno je spremeniti podatkovne tipe v takšne, ki porabljajo manj pomnilnika. Če gre za številčni stolpec z vrednostmi, ki nimajo več od 4 decimalk, je učinkovit ukrep sprememba tipa podatkov v *Currency*. Na prikazu pa izpisati brez znaka valute.
- Uporaba naravnih hierarhij je bolj učinkovita glede na nenaravne.

Na strani Excel odjemalca so prav tako možne optimizacije:

- Razčlenjevalnike (angl.  *slicer* ) je primerno uporabiti samo za dimenzijske tabele, pa še to takrat, ko naslovov v razčlenjevalnikih ni veliko. Opcija je uporabiti filtre v vrtilnih tabelah.
- Razčlenjevalniki za filtriranje, ki jih redko uporabljamo, niso potrebni. Opcija je uporabiti filtre v vrtilnih tabelah.

- Z onemogočanjem (angl. *disable*) navzkrižnega filtriranja (angl. *cross-filter*) pri razčlenjevalnikih se prepreči vrednotenje po naslovih razčlenjevalnika v primeru, če se izbor spremeni v drugih razčlenjevalnikih (Alexander, Decker in Wehbe, 2014). Opazili smo, da je to izjemno učinkovito. Ocenjujemo, da je se čas obnovitve posamezne nadzorne plošče pri prototipu (poglavje 4) lahko skrajšal tudi za 5-krat. Razčlenjevalniki so zelo dobrodošla funkcionalnost in omogočajo, da takoj vidimo, kateri pogoji filtriranja so nastavljeni a s sabo prinesejo tudi veliko upočasnitev, če jih je preveč in če dopuščamo navzkrižno filtriranje, saj se preverjajo vsi filtri - xVelocity jedro namreč preračuna vsako izračunano mero za vse naslove v razčlenjevalnikih.

## 2.6 Odločitev glede izbire ustreznega modela

Primerjalni seznam karakteristik večdimenzijskega, tabelaričnega in modela v PowerPivot dobro predstavljajo: Serra (2012), Sindol (2013) in Faulkner (2016).

Naj navedemo nekaj ključnih razlogov in predlogov kdaj in zakaj uporabiti tabelarični model:

- Navajenost – Delo s tabelaričnim modelom je domače mnogim uporabnikom, ki redno delajo s podatki v tabelah, shranjenih v relacijskih podatkovnih bazah. DAX je podaljšek funkciji Excel. Ta obstoječa znanja je možno hitro nadgraditi (Vitt & Cameron, 2012).
- Fleksibilnost – Ker ni toge organizacije podatkov v merah in dimenzijah, tabelarični model pospeši razvojne cikle, saj zahteva manj predpriprave podatkov. Taka podatkovna arhitektura je bolj prilagodljiva, ko je potrebno zaradi spremenjenih poslovnih zahtev spremeniti relacije in izračune (Vitt & Cameron, 2012);
- Razvoj modela je cenejši glede na čas, vire in potrebna znanja (Serra, 2012).
- V primerih ko večdimenzijski model zahteva uporabo posnetkov stanja (angl. *snapshots*), je tabelarični model boljši, saj posnetkov stanja ne potrebuje. Izračun dnevnega stanja je izveden v času poizvedbe. Hitrost omogoči dejstvo, da so vsi podatki v pomnilniku računalnika (Serra, 2012). To lastnost smo uporabili tudi pri našem prototipu – poglavje 4.
- V primerih ko imamo na voljo kratke razvojne cikle in ko je čas izvajanja poizvedb kritičen je tabelarični model zaradi teka v pomnilniku računalnika hitrejši od večdimenzijskega. Za primere, ko je podatkov veliko in presegajo kapaciteto pomnilnika se predlaga uporaba načina poizvedb *DirectQuery*, ki je nekako ekvivalenten ROLAP načinu pri tradicionalnem večdimenzijskem modelu – poizvedbe potekajo na podatkih hranjenih v relacijski podatkovni bazi in niso v pomnilniku. PowerPivot je primeren dokler velikost podatkov ne preseže 2GB (Sindol, 2013).
- Če razvojni tim ne pozna dobro večdimenzijskega modela in jezika MDX, se DAX jezik da naučiti hitreje (Russo 2013a).
- Ne obstaja potreba po kompleksnih izračunih, ki temeljijo na relacijah mnogo proti mnogo (Russo 2013a).



Russo (2013a) tudi pravi, da bi pri 80% novih projektov lahko uporabili večdimenzijski ali tabelarni model in da se razlika lahko pojavi samo pri sposobnosti razvojnega tima.

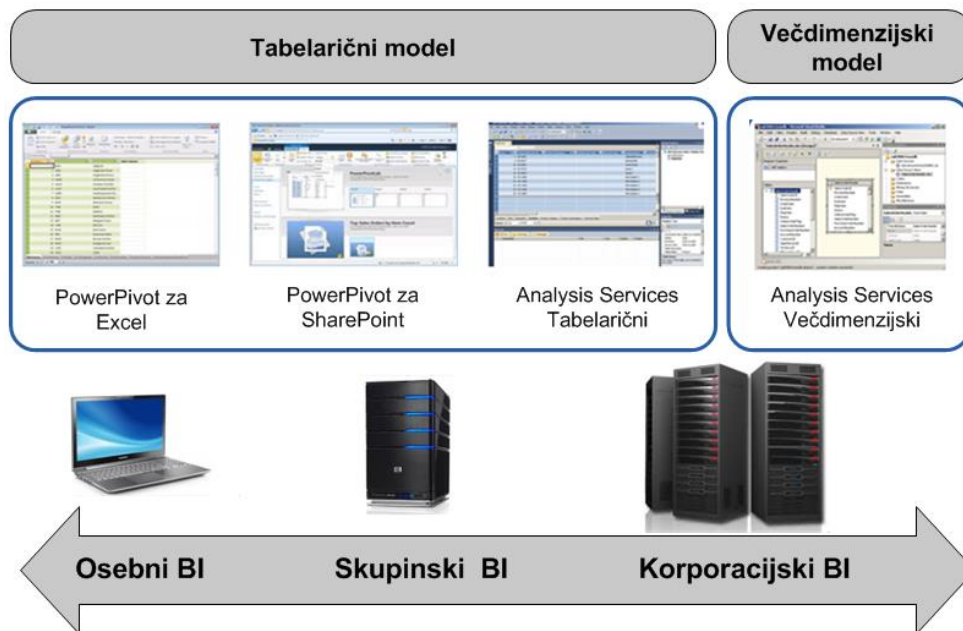
Russo, Ferrari in Webb (2012b) navajajo nekaj manjkajočih funkcionalnosti tabelarnega modela ob navedbi, da niso potrebne za vse projekte in pripombi, da, če jih nismo uporabljali v večdimenzijskem modelu, jih ne bomo pogrešali:

- Zapisovanje v podatkovno bazo (angl. *writeback*). Končni uporabniki, predvsem pri planiranju, želijo vpisati vrednosti.
- Prevodi (angl. *translations*). Metapodatki so uporabnikom dosegljivi v različnih jezikih.
- Varnost podatkov na nivoju celice (angl. *cell security*). V resnici redko uporabljena tudi v večdimenzijskih modelih.
- Nepravilne hierarhije (angl. *ragged hierarchies*). V večdimenzijskem modelu lahko hierarhija narejena na način, da se skriva člane pri določenih pogojih. Ragged hierarhijo dobimo, ko vsaj en član hierarhije nima nima vseh nivojev.
- Dimenzije v več vlogah (angl. *role-playing dimensions*). V večdimenzijskem modelu lahko enkrat kreirana dimenzija nastopi večkrat pod različnimi imeni in z različnimi relacijami do tabel mer. V tabelarnem modelu moramo tako dimenzijo naložiti večkrat to poveča zahteve pri vzdrževanju.
- Napredne kalkulacije (angl. *scoped assignments, unary operators*). Uporabljajo se predvsem pri finančnih aplikacijah. V tabelarnem modelu je z DAX to težko implementirati. Omejitev je pri zapisovanju v podatkovno bazo.

Ugotavljajo, da je zaradi navedenih pomanjkljivosti tabelarni model manj primeren za finančne aplikacije. »Zelo verjetno, da bo Microsoft k tabelarnemu modelu dodajal manjkajoče lastnosti in bodo razvojne investicije fokusirane na ta model« (Wade, 2013). To se že uresničuje, saj SQL Server 2016 prinaša veliko novosti, ki povečujejo funkcionalnosti tabelarnega modela in omogočajo lažje delo. Možno je tudi že poizvedovanje z MDX po tabelarnem modelu (De Jong, 2015). Pri tabelarnem modelu je ugodno tudi to, da je za hiter razvoj mogoče uporabiti PowerPivot, ki je sestavni del Excel. Model je potem enostavno prenosljiv v SSAS. Osnovno informacijo, komu je kak model namenjen, lahko vidimo na Sliki 14.

Vseeno moramo navesti nekaj bistveni razlik med obema predstavljenima modeloma. Prva zelo pomembna razlika je, da je pri tabelarnem modelu pristop z izgradnjo podatkovnega skladišča po principih zvezdnih shem zelo učinkovit. Vendar zvezdne sheme niso nujne. Vse tabele so v modelu enake in ni razlik med dimenzijami in merami, če odmislimo vsebinske razlike. To dejstvo odpre precej poti v fleksibilnejše modeliranje, a lahko tudi hitro pripelje v situacijo relacij mnogo : mnogo, ki pa v tabelarnem modelu niso podprte in je potrebno dodatno delo. Druga bistvena razlika je v obvladujoči velikosti modela. Tabelarni model teče v pomnilniku računalnika, tu pa sta velikost in hitrost pomnilnika (RAM) odločilna.

Slika 14: Microsoft BI semantični model



Vir: Microsoft, Blog Series: Creating BI Solutions with BISM Tabular, 2012.

Iz navedenega izhaja, da še ni pravega recepta, kateri model uporabiti. Tabelarični model je novejši, se hitreje razvija in odpravlja pomanjkljivosti ter ima ugodnejšo krivuljo učenja. Na drugi strani so OLAP rešitve v zreli fazi z veliko know-how in velikim repozitorijem rešitev. Verjetno ni zgrešena ocena, da je za hitre prototipne rešitve primernejši tabelarični model. Enako velja pri samopostrežni BI. Sobivanje dveh modelov pod krovom iste tehnologije pri Microsoft BISM je odprlo možnost preizkušanja obeh modelov, ne da bi se morali učiti uporabe novih uporabniških vmesnikov, saj poteka navsezadnje vse delo znotraj razvojnega okolja Visual Studio.

## 2.7 Tržišče poslovne inteligence danes in trendi razvoja

Današnje stanje BI dobro pokaže raziskava analitske hiše Gartner (Parenteau et al, 2016), ki že od leta 2006 objavlja raziskave imenovane *Magic Quadrant*. Področje BI ima svoj kvadrant z imenom *Magic Quadrant for Business Intelligence and Analytics Platforms* (Slika 15).

Opažajo, da se je v zadnjih letih težišče BI premaknilo od podprtega s strani informatike na stran poslovnih uporabnikov, kajti vedno večja je potreba po hitrem analiziranju podatkov in po avtonomiji poslovnih uporabnikov.

Slika 15: Gartnerjev magični kvadrant za BI in analitične platforme



Vir: Parenteau et al, *Magic Quadrant for Business Intelligence and Analytics Platforms*, 2016.

Temu je zelo pripomogla evolucija orodij samopostrežne BI, ki ne zahtevajo več vpletanja informatikov pri definiranju podatkovnih modelov. Danes je izražena potreba po hitrosti, raziskovanju in hitrem prototipiranju. Strateška predvidevanja navajajo, da bo do 2018 imela večina poslovnih uporabnikov dostop do orodij samopostrežne poslovne inteligence. Pri ocenjevanju ponudnikov za leto 2016 so zajeli 4 sklope oz. skupno 14 najpomembnejših zmožnosti BI:

- infrastruktura: administracija BI skupaj s povrnitvijo pri napakah, delo v oblaku, varnost in administracija uporabnikov in povezovanje s strukturiranimi in nestrukturiranimi viri podatkov;
- management podatkov: upravljanje z metapodatki, ETL postopek, samopostrežna priprava podatkov;
- analiziranje: vključena napredna analitika, visoko interaktivne nadzorne plošče z vključena geo-analitika, interaktivno vizualno raziskovanje, zmožnost prikaza na mobilnih napravah, zmožnost poizvedb v naravnem jeziku;
- deljenje informacij: vključenost analitike v poslovne procese in portale, zmožnost objave analitičnih informacij preko različnih tipov in metod distribucije, vključujoč možnost

iskanja, pripovedovanja zgodb, razporejanja in alarmiranja, zmožnost deljenja preko forumov.

Pri Tableau (2014) so že leta 2014 napovedovali:

- Konec obdobja specialistov. Podatkovne analize bodo postale del znanj poslovnih uporabnikov.
- Zrelost servisov v oblaku bo pomagala IT oddelkom vzpostaviti BI storitve v oblaku. Prišlo bo do novih scenarijev sodelovanja s kupci in do novih dostopov z mobilnimi napravami.
- Samopostrežna analitika bo postala običajna. Poslovni uporabniki pričakujejo uporabnost in fleksibilnost od svojih nadzornih plošč.
- Analitika bo vedno bolj vključena v transakcijske sisteme, najprej na področju upravljanja odnosov s strankami, kjer bo zagotovljena podpora prodajnikom pri dnevnih odločitvah. BI bo pripeljal informacije do oddelkov, ki so tipično pri tem zaostajali, do proizvodnje in prodaje na drobno.
- Prioriteta bo "pripovedovanje zgodb". Poplava nadzornih plošč brez vsebine ne pomaga. Zgodbe so način, kako skomunicirati ideje in pogled v podatke.
- Mobilna BI bo v ospredju. Poslovni uporabniki začenjajo zahtevati dostop do informacij, ne da pridejo v pisarno za računalnik.
- Podjetja bodo resno začele analizirati podatke iz socialnih omrežij. Podjetjem je važno, da razumejo kako so rangirana pri svojih kupcih.

Opozorimo naj, da je pri vseh BI raziskavah potrebna določena mera kritičnosti in previdnosti. Rezultati raziskav ne smejo biti edina izbira pri odločitvi, za katero BI rešitev se kaže odločiti. Optimalna izbira pri specifičnih potrebah je lahko drugačna in ni nujno, da so v takih primerih vodilne rešitve najboljše. Te so najboljše so pri kriterijih, ki so bili upoštevani v raziskavah. In upoštevano je bilo veliko splošnih kriterijev.

Vezano na trende razvoja Sallam, Tapadinhas, Parenteau, Yuen in Hostmann (2014) ugotavljajo, da je tržišče BI sredi pospešene transformacije BI sistemov, ki so se prvotno uporabljali za merjenja in poročanja, v sisteme, ki podpirajo analize napovedovanja in optimizacije. Večina poslovnih uporabnikov, analitikov in managerjev ni nikoli sprejela ad hoc poizvedb, parameteriziranih poročil in sprotne analitične obravnave (OLAP), ker smatrajo te pretežke za uporabo pri velikem številu primerov. Ponudniki so danes svoje platforme že dopolnili s poslovnimi uporabniško vodenimi tehnikami odkrivanja podatkov, s katerimi so omogočili enostavnejšo analitiko, primerno širšemu krogu uporabnikov in primerno za širok spekter realnih primerov.

Tržišče BI vidijo kot enega najhitrejših rastočih tržišč programske opreme. Predvidevajo letno rast 7% tudi po letu 2017.

Aktivnosti bodo predvsem potekale na naslednjih področjih:

- Naslednjih 10 let se bo veliko vlagalo v »Internet stvari«. Najhitreje narašča ravno količina podatkov, ki jih pridobivamo v realnem času, od senzorjev, strojev in dogodkov, generiranih od oseb ali transakcijskih poslovnih sistemov. Novo prilagojeno poslovno analitiko po pogojevala ta multistrukturnost podatkovnih tipov.
- Področje velike količine podatkov (angl. *big data*) je gledano s stališča operativne izvedbe prešlo v krivulji razvoja tehnologije skozi čas (angl. *hype cycle*) v fazo streznitve, kjer je rast majhna in bo to upočasnjevalo rast BI.
- Rast BI v oblaku ima potencial za generiranje novih aplikacij, ki bodo podatke industrije integrirale z analitičnimi orodji in bodo analize na prodaj kot servisi.
- Mobilni BI ostaja pomembno področje. Predvideva se možnost rasti.
- Trend je v integraciji BI in performančnega managementa, povezanega s planiranjem in izvedbo.
- Razvoj ETL orodij se bo nadaljeval, saj je priprava podatkov trenutno časovno zelo zahtevna.

Zelo pomembno je sporočilo Gartnerja (2014), ki pravi, da morajo vodje informatike BI obravnavati vsaj do 2017 kot top tehnološko prioriteto, kajti prednosti odločanja na osnovi podatkov so managerjem jasne za širok spekter področij: dobaviteljske verige, proizvodnja, prodaja, marketing, management rizika, finance in človeški viri. Predvidevanja za strateško planiranje so naslednja:

- z letom 2015 bo večina ponudnikov BI odkrivanje podatkov vključila v osnovno BI platformo s premikom iz poročilnih sistemov v analitične sisteme;
- do 2016 bo zmeda na področju velikih podatkov rast BI zadrževala na stopnji pod 10 % letno;
- do 2017 bo preko 50 % implementacij analitike zmožno uporabljati dogodkovne podatke od strojev, aplikacij ali posameznikov, analitičnih aplikacij ponudnikov programske opreme ne bo več mogoče ločiti od aplikacij ponudnikov storitev.

Vežano na trende razvoja BI ugotovimo, da je zaradi hitrega razvoja tehnologij, hitrega naraščanja količine podatkov ter predvsem spreminjanja potreb, smiselno gledati samo najnovejše raziskave in dognanja. Pri informacijskih potrebah se vse bolj zavedamo, da tisto, kar je pomembno danes jutri morda ne bo več. Vrzel med dogodkom in informacijo se vedno bolj zmanjšuje in je informacija praktično potrebna takoj. To pomeni, da bodo morala biti podatkovna skladišča zelo fleksibilna, hitro sestavljiva in tudi odveč, če bo do zgodovinskih podatkov možno priti on-line. Ker je kompleksnost vedno večja, bo potreba po informatikih ostala še naprej, a bo njihova funkcija verjetno v ozadju pri pripravi in vzdrževanju infrastrukture. Selitev BI v oblak je neizogibna, a varnostno še ne čisto prepričljiva. Dvomi ostajajo in številna podjetja se raje zanašajo na privatni oblak.

### 3 MANAGEMENT ZALOG

Management (kontrola) zalog uvrščamo med najpomembnejše logistične dejavnosti v proizvodnih in tudi v neproizvodnih podjetjih in je neposredno povezan z nabavno logistiko (Čižman, 2002). Bistvo je v določitvi nivoja zalog in ostalih parametrov dobavnega sistema tako, da so stroški, povezani z dobavami in zalogami, najmanjši. Z zalogami se je potrebno ukvarjati in brez njih ne moremo. Pučko (1993) pravi: »Zaloge se nam v podjetju pojavljajo predvsem zato, da si omogočimo sorazmerno neodvisnost posameznih faz poslovnega procesa in dogajanja v njih. To omogoča opravljanje vsake faze na najbolj učinkovit način. S tem je mogoče stroške vsake faze znižati.«

Najprej je potrebno zaloge opredeliti. V Slovenskih računovodskih standardih (2015) je navedeno, da so zaloge praviloma sredstva v opredmeteni obliki, ki bodo porabljena pri ustvarjanju proizvodov ali opravljanju storitev oziroma pri proizvodnji za prodajo ali prodana v okviru rednega poslovanja. Zakaj pa se o zalogah sploh treba pogovarjati, če pa Čižman (2002) pravi, da zaloge niso »niti popolnoma koristne niti popolnoma škodljive«. Bistvo se skriva v kapitalizmu – v družbeno ekonomskem sistemu, v katerem živimo. Zanj je značilno, da so proizvodna sredstva v zasebni lasti, izrazita individualna odgovornost za dejanja, pridobitništvo kot motiv na eni strani in trg kot poglobljen selekcijski in regulacijski mehanizem proizvodnje (obseg, vrsta in diverzifikacija), dohodka (velikost in razdelitev) in uporabe dohodka (zasebna poraba in investicije) na drugi strani. Od druge svetovne vojne je predvsem v evropskih državah uveljavljen koncept socialnotržnega gospodarstva, za katerega je značilno predvsem varovanje zasebne lastnine proizvodnih sredstev in konkurenčno oblikovani blagovni trgi s temeljnim zakonskim načelom prepovedi oblikovanja kartelov na eni strani in pravica do sindikalnega združevanja ter sistem kolektivnih pogajanj o osnovnih delovnih razmerah, o plačah ipd. na drugi. Pomembna je tudi vloga države pri vodenju ustrezne gospodarske politike za doseganje ciljev gospodarskega razvoja in v oblikovanju trdne socialne politike za zagotavljanje ustreznih življenjske ravni zaposlenih. Socialno tržno gospodarstvo je kot demokratičen tristranski sporazum med delodajalci, delavci in državo (Veliki splošni leksikon, 1998).

V ekonomskem procesu kapitalizma dominira zakon vrednosti (angl. *law of value*) (Marx, 1894). Gre za to, da se investira finančni kapital z namenom realiziranja dobička, ki se ponovno reinvestira v proizvodnjo. Tako pridemo do kontinuiranega procesa akumulacije. Važna je tako dobičkonosnost (rentabilnost, profitabilnost).

Bodie, Kane in Marcus (2008) podrobneje predstavljajo meri dobičkonosnosti:

- donosnost lastniškega kapitala - ROE (angl. *return of equity*), enačba 1 in

$$ROE = \frac{\text{neto dobiček}}{\text{kapital}} \quad (1)$$

- donosnost premoženja - ROA (angl. *return of assets*), enačba 2.

$$ROA = \frac{\text{dobiček pred davki}}{\text{in obrestmi sredstva}} \quad (2)$$

Za razumevanje faktorjev, ki vplivajo na ROE, in za primerjavo s konkurenco analitiki pogosto razcepijo ROE v serijo kvocientov po DuPontovem sistemu, enačba 3.

$$ROE = \frac{\text{neto dobiček}}{\text{kapital}} = \frac{\text{neto dobiček}}{\text{dobiček pred davki}} \times \frac{\text{dobiček pred davki}}{\text{dobiček in obrestmi}} \times \frac{\text{dobiček pred davki}}{\text{in obrestmi}} \times \frac{\text{prodaja}}{\text{prodaja}} \times \frac{\text{sredstva}}{\text{sredstva}} \times \frac{\text{kapital}}{\text{kapital}} \quad (3)$$

(1)                      (2)                      (3)                      (4)                      (5)

Pomen posameznih kvocientov enačbe 3:

- (1) Predstavlja davčno obremenitev in politiko prizadevanj podjetja, da bi zmanjšalo davčno obremenitev. Faktor ni odvisen od kapitalske strukture podjetja.
- (2) Faktor bo večji, če dolžnikom ni treba plačati obresti na dolg. Največja vrednost faktorja je 1. Faktor je odvisen od kapitalske strukture podjetja.
- (3) Faktor je poznan kot operativna marža. Vrednost je med 0 (0 %) in 1 (100 %). Faktor ni odvisen od kapitalske strukture podjetja.
- (4) Faktor je poznan tudi kot obrat sredstev – ATO (angl. *assets turnover*). Kaže na učinkovitost porabe sredstev podjetja in meri letno prodajo, ki jo generira vsak € v sredstvih. Faktor ni odvisen od kapitalske strukture podjetja.
- (5) Faktor meri stopnjo finančnega vzvoda. Zavisi od dolga podjetja, enačba 4.

$$\frac{\text{sredstva}}{\text{kapital}} = \frac{\text{kapital} + \text{dolg}}{\text{kapital}} = 1 + \frac{\text{dolg}}{\text{kapital}} \quad (4)$$

Iz faktorja (4) v enačbi 3 vidimo, zakaj je višina zalog pomembna pri dobičkonosnosti. Zaloge namreč spadajo med sredstva. Čim manjše bodo, večji bo faktor (4) in posledično donosnost lastniškega kapitala.

Bodie, Kane in Marcus (2008) posebej poudarjajo, da visok ROE, izračunan na preteklih podatkih, še ne zagotavlja visokega ROE v prihodnosti. Na drugi strani pa padajoči ROE kaže na to, da nove investicije podjetja niso tako uspešne kot pretekle. Če primerjamo ROE med sabo, je to potrebno storiti med podjetji v isti panogi.

### 3.1 Vrste zalog

Razdelitev zalog je možna po več kriterijih, dva osnovna sta:

- po vrstah proizvodov in
- po funkcijah, ki jih zaloge opravljajo.

Po prvem kriteriju je dobra členitev, ki jo predvidevajo slovenski računovodski standardi (2015), konkretno v SRS 4, so vrste zalog opredeljene takole:

- »Zaloge so praviloma sredstva v opredmeteni obliki, ki bodo porabljena pri ustvarjanju proizvodov ali opravljanju storitev oziroma pri proizvodjanju za prodajo ali prodana v okviru rednega poslovanja.
- Zaloga materiala zajema količine v skladišču, dodelavi in predelavi pa tudi na poti od dobavitelja, če jih je kupec že prevzel oziroma prevzel tveganja in koristi v zvezi z njimi. Kot material se lahko šteje tudi drobni inventar z dobo koristnosti do leta dni, lahko pa tudi tisti z dobo koristnosti več kot leto dni, če njegova posamična nabavna cena ne presega 500 evrov.
- Zaloga v postopku proizvodjanja zajema nedokončano proizvodnjo in polproizvode. Nedokončana proizvodnja lahko obsega tudi opravljanje storitev, ki so do konca obračunskega obdobja dokončane, od naročnika pa še ne prevzete (potrjene).
- Zaloga, namenjena prodaji, zajema dokončane proizvode in trgovsko blago v skladišču ter količine na poti do kupca, dokler jih ne prevzame, trgovsko blago pa tudi količine na poti od dobavitelja, če jih je kupec že prevzel.«

Ob tem so ključni pojmi definirani:

- »**Material v širšem pomenu** so osnovni in pomožni material, polproizvodi, deli, nadomestni deli, drobni inventar ter gorivo in mazivo. Odpadki pri odpisanih stvareh ali lastni proizvodnji se štejejo za material, če so namenjeni prodaji.
- **Drobni inventar v širšem pomenu** so orodje, naprave in druga oprema ter ločljiva embalaža in podobne stvari z dobo koristnosti do leta dni. Organizacija lahko uvrsti med drobni inventar tudi stvari drobnega inventarja, katerih doba koristnosti je daljša od leta dni in katerih posamična nabavna cena ne presega 500 evrov.
- **Nedokončana proizvodnja v širšem pomenu** obsega nedokončano proizvodnjo na proizvodjalnih mestih, lastne polproizvode in dele, namenjene za dokončevanje proizvodov, ter odpadke, namenjene uporabi v proizvodnji.
- **Proizvodi** so dokončani proizvodi v obliki stvari, namenjeni neposredni prodaji, ki pa so še vedno last organizacije. Vključujejo tudi tista biološka sredstva, ki so zaloga.«



Po drugem kriteriju, po funkcijah, ki jih opravljajo, pa zaloge opredeljuje Rusjan (2009):

- **Serijske zaloge**, te so posledica nabave in proizvodnje v ekonomsko optimalnih količinah. Omogočajo razporeditev stroškov nabave in priprave proizvodnje na večje število enot.
- **Varnostne zaloge** so potrebne v primeru negotovosti: dobave, proizvodnje in tudi povpraševanja. Zaloge vhodnih materialov varujejo pred negotovostmi dobaviteljev, npr. glede dobavnih rokov, kvalitete dobavljenih materialov in podobno. Imamo lahko tudi varnostno zalogo nedokončane proizvodnje zaradi morebitnih okvar strojev, interne kakovosti in varnostno zalogo dokončane proizvodnje zaradi negotovosti potreb kupcev.
- **Sezonske zaloge** so povezane s sezonskimi nihanji v povpraševanju. Posledično niso usklajene proizvodne zmogljivosti in povpraševanje. Eden izmed načinov kratkoročnega usklajevanja je povečanje zalog v času, ko proizvodne zmogljivosti presegajo povpraševanje.
- **Razbremenilne zaloge** oblikujemo, da bi napravili posamezna delovna mesta neodvisna od dogajanj na drugih delovnih mestih. Značilne so predvsem za montažno linijo. Z njimi blažimo razlike v časih izvajanja med delovnimi mesti. Lahko bistveno povečajo učinkovitost montažne linije a je povečevanje smiselno samo do neke mere.
- **Tranzitne zaloge** nastanejo zaradi prevozov vhodnih materialov od dobaviteljev in dokončanih proizvodov do distributerjev ali kupcev. Odvisne so od lokacije proizvodnje in lokacije dobaviteljev in kupcev ter od izbranega načina transporta.
- **Špekulativne zaloge** uporabljamo jih v primeru, ko pričakujemo večje spremembe na trgu, npr. zvišanje cen ali pomanjkanje določenega materiala.

Zelo podobno ugotavlja tudi Ljubič (2000), ki pravi, da zaloge

- blažijo nihanja:
  - porabe materiala, sestavnih delov in gradnikov v proizvodnji,
  - povpraševanja po gotovih izdelkih na trgu;
- kompenzirajo napake zaradi:
  - neprimernih metod planiranja,
  - netočnih podatkov o stanju,
  - odklonov dobavljenih količin,
  - odklonov dobavnih rokov;
- imajo ekonomske učinke:
  - sezonski nakup blaga, ko je poceni, in poraba takrat, ko je drago,
  - popusti pri nakupu večjih količin blaga.

## 3.2 Stroški zalog

Čižman (2002) ugotavlja, da je v zalogah investiran pomemben del vseh sredstev in da so skladišča lahko prava zakladnica kapitala. Vukovič in Završnik (2011) sta bolj konkretna in navajata, da v večini proizvodnih podjetij zaloge tvorijo drugo največjo premoženjsko kategorijo v bilanci stanja. Večja so le materialna sredstva in oprema. Zaloge predstavljajo približno 30 % vloženih sredstev podjetja. Stroške zalog Završnik (2008) z vidika upravljanja deli v tri skupine:

1. **Stroški vzdrževanja zalog** se gibljejo med 20 % in 40 % vrednosti zalog letno:
  - a) Oportunitetni stroški investiranih sredstev (12-20 %): Denar, ki je vezan v zalogah, ni na voljo za druge namene. Če ga bi investirali v donosno opremo ali v vrednostne papirje, bi od tega lahko imeli korist. Govorimo o stroških zavržene priložnosti.
  - b) Stroški zavarovanja (2-4 %): Podjetja imajo svoje premoženje večinoma zavarovano proti možni izgubi: požar, poplava, kraja, elementarne nesreče. Zaloge spadajo med premoženje in je tudi zanje potrebno plačati zavarovalno premijo.
  - c) Davek na premoženje (1-3 %): Odmeri se na osnovi ocenjene vrednosti premoženja podjetja. Večje kot bodo zaloge, večja bo vrednost premoženja in posledično davek.
  - d) Stroški skladiščenja (1-3 %): Skladišče, v katerem je material, se vsako leto amortizira za določen znesek vrednosti. V primeru najema se plačuje najemnina. Stroški so izraženi v € na m<sup>2</sup>. Odvisni so od prostora, ki ga zaloge zavzemajo.
  - e) Stroški zastaranja, poškodovanja in uničenja (4-10 %): Zaloge se dostikrat poškodujejo, zastarajo in pride tudi do odtujitve. Čeprav se preprečitvam tega namenja posebna skrb, se to vseeno zgodi. Po drugi strani pa material lahko tudi zastara zaradi upada prodaje, saj se na trgu pojavljajo stalno novi produkti.

Stroški vzdrževanja zalog naraščajo in padajo proporcionalno z naraščanjem in padanjem ravni zalog. Velja enačba 5

$$LSVZ = \frac{K}{2} * SN * SVZ \quad (5)$$

kjer je: LSVZ – letni strošek vzdrževanja zalog  
K – nabavljena količina materiala  
SN – stroški nabave na enoto materiala  
SVZ – stroški vzdrževanja zalog v % od vrednosti zalog

### 2. Nabavni stroški naročanja:

- a) Predhodni stroški. Nastanejo v času definiranja potreb, izbire dobaviteljev, obiskov dobaviteljev, v času pogajanj, pogovorov z dobaviteljem. Sem se prištevajo tudi

pisarniške potrebščine, čas uporabe računalnika, telefonov itd.

- b) Stroški dobave blaga. Gre za stroške prevoza in dostave, stroške carine, zavarovanja itd.
- c) Stroški prevzema. Sem se šteje prevzem materiala, vhodna kontrola količin in kakovosti, potrjevanje in plačilo računov itd.

Ker je delo nabavnega oddelka v določenem času razporejeno na številna naročila, se uporabi povprečen strošek na naročilo. Nabavni stroški za razliko od stroškov vzdrževanja niso povezani z velikostjo zalog, temveč so števila naročil in dobav v določenem času. Celotni stroški za čas, ki ga je nabavnik porabil, bodo enaki ne glede na to, ali gre za naročilo za 20 ali 200 kosov.

Največji del nabavnih stroškov sestavljajo indirektni stroški dela in režijski stroški, ki nastanejo v nabavnem oddelku in drugih oddelkih v katerih se odvijajo aktivnosti v zvezi z nabavo. Stroški dobave in manipulacije z naročilom se spreminjajo s številom naročil. Ti stroški so precej manjši od režijskih stroškov in stroškov, povezanih s človeškimi viri. Letne nabavne stroške lahko izračunamo po enačbi 6.

$$LNS = \frac{P}{K} * N \quad (6)$$

- kjer je:
- LNS – letni nabavni stroški
  - P – pričakovana letna uporaba materiala
  - K – naročena oz. dobavljna količina materiala
  - N – nabavni stroški na naročilo

### 3. Stroški pomanjkanja zalog

Ti stroški so lahko včasih zelo veliki in imajo velike posledice za poslovanje podjetja predvsem velja to za podjetja z veliko serijsko proizvodnjo. Pogosto so skriti v režijskih stroških in jih je težko vključiti v modele zalog. Posledice so lahko:

- a) pomanjkanje končnih izdelkov,
- b) stroški zaradi zastojev v proizvodnji in večji režijski stroški na enoto končnega izdelka,
- c) stroški dodatnih aktivnosti zaradi nadomestitve zalog, gre za stroške povezane z naročanjem, dodatne stroške dobave in morebitne višje stroške materiala, ki ga je morda potrebno kupiti pri drugem dobavitelju,
- d) izguba zaupanja kupcev zaradi zamujene dobave.

Čeprav zaloge pomenijo strošek, pogledi različnih služb nanje v podjetju niso enotni. Prodaja hoče čim hitreje ustreči kupcem, zato si želi imeti čim več zalog vseh vrst končnih izdelkov, če ne izdelkov pa vsaj gradnikov in materialov, iz katerih se lahko hitro izdelajo končni izdelki. Proizvodnja hoče imeti velika proizvodna naročila, velike serije, da je čim

bolj učinkovita. Prav tako želi imeti čim večje zaloge nedokončane proizvodnje, da ne bi zastajala zaradi pomanjkanja materiala. Prodaja ima raje malo velikih naročil kot veliko majhnih. Zaloge tudi razume kot zavarovanje pred podražitvami ali izpadi dobav. Na drugi strani pa finance želijo zmanjšati vse oblike vlaganj v zaloge zaradi stroškov kapitala. Tudi razvojne službe hočejo majhne zaloge, da tehnične spremembe ne bi čakale, dokler niso izčrpane zaloge starih izvedb (Ljubič, 2000).

### 3.3 Analiza zalog

Da bi ugotovili, kaj se nam dogaja z zalogami, da bi lahko sprejeli ustrezne odločitve in aktivnosti, je potrebno zaloge analizirati, sprotno ali občasno, zavisi od potreb in dinamike poslovanja. Če analiza pomeni »...razčleniti, ugotoviti sestavne dele nečesa,« (Veliki splošni leksikon, 1998), je to v primeru zalog ugotavljanje količine, vrste in gibanja zalog. Čeprav lahko zaloge analiziramo na različne načine, so pomembni samo tisti, ki nam pomagajo sprejemati ključne odločitve. Walters (2003) pravi, da se je potrebno skoncentrirati samo na tri osnovna vprašanja:

1. Katere materiale naj imamo na zalogi?

Podjetje mora zagotavljati minimalno zalogo, ki še zagotavlja potrebno oskrbo. To pomeni:

- a) zaloga materialov mora biti v razumni količini,
- b) v zalogo ne dodajati nepotrebnih materialov,
- c) odstraniti nepotrebne materiale, ki se ne potrebujejo več.

Če zaloge niso kontrolirane, obstaja tendenca po povečanju. Znani so argumenti: »Smo, ker zagotavljamo dobro oskrbo, za katero pa potrebujemo velike zaloge.« S širjenjem se vedno dodaja tudi nove materiale brez posebnega razmisleka, medtem ko stari ostajajo na zalogi ob pričakovanju, da se bodo porabili. V primeru rezervnih delov je to še posebej očitno. Z novim strojem pridejo novi rezervni deli, medtem ko se rezervni deli starega stroja še vedno hranijo, saj je pogosto težko sprejeti dejstvo, da so danes brez vrednosti, čeprav so bili ob nabavi dragi. Zaradi naraščanja zalog in potrebnega prostora managerji v nekem trenutku zahtevajo racionalizacijo. Pogosto hočejo hitre rezultate in zahtevajo npr. zmanjšanje zalog za 10%. Najlažji način, kako to doseči, je, da zalog po porabi ne dopolnjujemo. Rezultat je, da se zaloge pogosto uporabljenih materialov znižajo, zaloge redko uporabljenih materialov pa ne. Da bi se izognili takim situacijam, je potrebno racionalno dodajati materiale ob primerjavi stroškov in prednosti skladiščenja glede na stanje, če te materiale ne skladiščimo.

2. Kdaj naročiti material?

V osnovi gre za tri različne pristope. Prvi je naročanje variabilnih količin v fiksnih časovnih intervalih. Ta pristop se pogosto uporablja v trgovinah, ko se konec delovnega

dne pregleda zaloge in do naroči prodane količine. Drugi pristop je naročanje fiksnih količin. Nadzor stanja zalog se izvaja stalno, in ko zaloge dosežejo določen nivo, se naroči fiksna količina. Tako se lahko spreminja čas med dvema naročiloma. Tretji pristop je glede na potrebe. Menja se lahko količina in čas med dvema naročiloma.

### 3. Koliko naj naročimo?

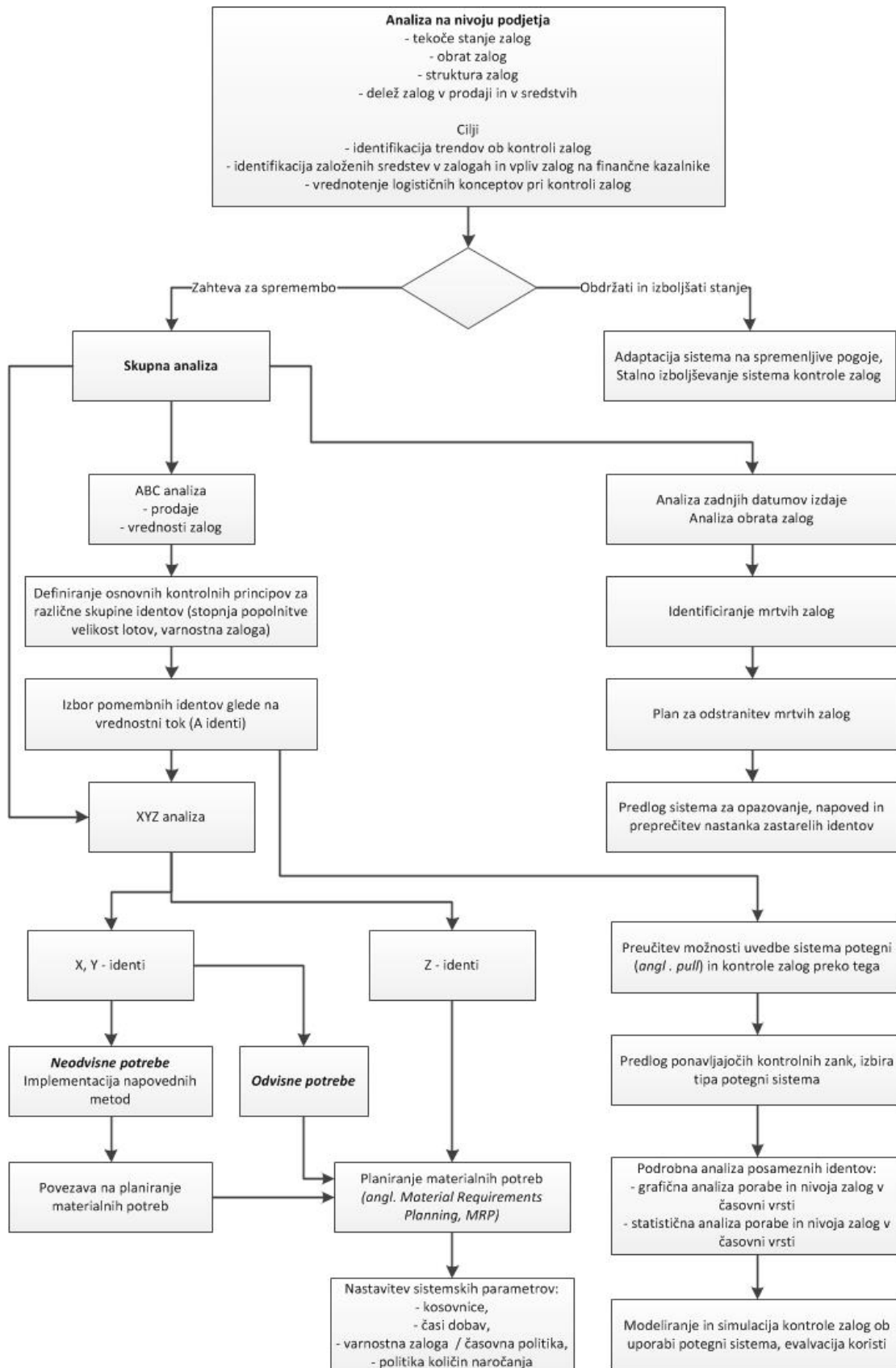
Vsako naročilo spremljajo stroški administracije in dobave. Pri velikih redkih naročilih so stroški naročanja in dobave majhni, ampak imamo zato veliko zalogo. Če pa na drugi strani naročamo pogosto v majhnih količinah, so stroški naročanja in dobav veliki, ampak imamo majhno zalogo. Poiskati je potrebno kompromis med prej navedenima ekstremoma.

Na prvi pogled se zdi, da smo izpustili druga važna vprašanja: Kakšna naj bo povprečna zaloga? Koliko naj investiramo v zalogo? Kakšno razpoložljivost naj zagotavljamo kupcem? Koliko zastojev bo pri tem? Vendar v praksi odgovori na ta vprašanja izhajajo iz odgovorov na prejšnja tri osnovna vprašanja. Ko smo enkrat odločitev o frekvenci in količini naročil, smo s tem avtomatično določili povprečno zalogo, razpoložljivost in verjetnost zastojev.

Krajčovič in Plinta (2012) sta opisala metodologijo analize zalog in zmanjšanja zalog, ki je bila razvita na univerzi Žilina na Slovaškem. Metodologija bazira na najboljših praksah in na inženirskem pristopu od zgoraj navzdol ter ima tri faze (Slika 16):

1. Analiza zalog na nivoju podjetja. Cilj je določiti nivo zalog, obračanje zalog, strukturo in gibanje zalog po vhodnih materialih, polizdelkih in gotovih izdelkih.
2. Skupna analiza skupin materialov. Cilj je materiale razdeliti v več skupin glede na pomembnost, vzorec porabe, obračanje in drugo z namenom določitve pristopov k analizi zalog in k optimizaciji. Pri tem se uporabi ABC metodo po več parametrih (vrednosti prodaje, vrednosti zalog), XYZ analizo in analizo obračanja zalog.
3. Posamična analiza materialov. Uporablja se na materialih z največjim potencialom prihranka, običajno v celoti na A materialih. Predlaga se:
  - a) analiza porabe in nivoja zalog, ki se izvede grafično s pregledom gibanja zalog in porabe,
  - b) statistična analizo zalog, ki vsebuje pregled zgodovinske porabe, določitev osnovnih parametrov za upravljanje zalog: varnostne zaloge, minimalnega in maksimalnega nivoja zalog ter nivo ponovnega naročanja, če se uporablja,
  - c) simulacija porabe in simulacijo dopolnjevanja zalog,
  - d) primerjava simulacije in realnega stanja.

Slika 16: Koraki aktivnosti pri analizi in zmanjšanju zalog podjetja



Vir: M. Krajčovič & D. Plinta, *Comprehensive approach to the inventory control system improvement*, 2012, str. 36.

Osnovni pristop k upravljanju z zalogami z izdelavo kataloga zalog in predvsem z ABC ter XYZ analizo so opisali tudi drugi avtorji: Čižman (2002) , Ljubič (2000), Rusjan (2009), Završnik (2008), Vukovič in Završnik (2011). Če je katalog zalog pomemben pri komunikaciji med udeleženci, sta ABC in XYZ analiza pomembni za izbiro sistema nadzora zalog.

S **katalogom zalog** opišemo vse materiale (postavke) v zalogah. Namenjen je identifikaciji materialov po: proizvajalčevi številki, uporabnikovi številki, klasifikacijski številki in drugih podatkih. Slika materiala dostikrat zelo olajša razpoznavanje. Katalog je bistvenega pomena pri komunikaciji, da vsi vemo, o katerem materialu iz zaloge se pogovarjamo, in v procesu nadzora kot npr. ob letnem popisu. Predlaga se tudi združevanje identičnih materialov različnih dobaviteljev pod eno oznako, a to vedno ni mogoče, sploh ne v primerih kjer je potrebno sledenje porabe materialov.

### 3.3.1 ABC analiza zalog

S pomočjo ABC razvrstitve postavke v zalogi razvrstimo v tri skupine glede na osnovo povprečnih investicij v vsako postavko ali na osnovi letne porabe vsake postavke v Eur.

Proučevanja mnogih podjetij so pokazala, da majhen odstotek vseh postavk v zalogah zajame večino sredstev, investiranih v zaloge. Kakšen delež postavk obsega določen odstotek zalog, sicer od podjetja do podjetja nekoliko odstopa, a ne veliko. Ker je ta značilnost silno podobna pravilu, ki ga je pri preučevanju ekonomskih zakonitosti leta 1906 ugotovil italijanski ekonomist Vilfredo Pareto, ugotovil je, da ima 20 % prebivalstva v lasti 80 % zemlje, rečemo ABC analizi tudi Paretova analiza. Paretova ugotovitev se je kasneje izkazala kot pravilna na mnogih področjih. Gre za to, da je 80 % učinkov posledica 20 % vzrokov ali drugače z 20 % truda pridemo do 80 % rezultata. Analiza ABC je pomembna,

- ker nam pomaga odkriti, s katerimi materiali v zalogi se moramo ukvarjati in kateri sistem naročanja je pravi za posamezen material,
- ker je enostavna in uporabna predvsem pri večjem številu postavk,
- ker je nazorna in grafično pregledna,
- ker nam pomaga določiti osnovna pravila in politike glede zalog (Tabela 4).

ABC analizo izvedemo z naslednjimi koraki:

1. za vse materialne postavke ugotovimo porabo v opazovanem obdobju, poraba je produkt količine z ustrezno ceno;
2. postavke razvrstimo po padajoči vrednosti porabe,
3. dodamo kumulativni odstotek postavk od 0 do 100 %,
4. dodamo k vsaki postavki kumulativni delež porabe ( $Y_j$ ) v % glede na celoto. Kumulativni delež je vsota porabe te postavke ( $X_i$ ) in vseh predhodnih postavk glede

na celoto (enačba 6),

$$Y_j(\%) = \frac{X_j}{X} * 100 = \frac{\sum_{i=1}^j X_i}{\sum_{i=1}^n X_i} * 100 ; J = 1, 2, \dots, n; \quad (6)$$

kjer pomeni:  $Y_j$  – kumulativni delež porabe glede na celoto

$X_i$  – poraba postavke  $i$

$n$  – število postavk

5. določimo meje za razrede A, B, in C, različni avtorji jih postavljajo različno (Tabela 3),
6. lahko narišemo tudi diagram kumulativnih deležev (Slika 17), diagram ima ime tudi Lorenzova krivulja.

ABC analiza tako materiale dejansko razdeli v tri skupine:

- **Materiale A**, tem je potrebno nameniti največjo pozornost, saj ima tukaj že procentualno manjše zmanjšanje zalog velik vpliv. Materiali A imajo največji potencial pri zmanjšanju zalog. Nazorno lahko to ponazorimo s primerom: Zmanjšanje zalog za 5 % v skupini A, ki predstavlja 70 % vrednosti, bi bilo enakovredno 35 % zmanjšanju zalog v skupini C, ki predstavlja 10 % zaloge.
- **Materiale B**, ki so na sredi med materiali A in C, in zato je potrebna večja pozornost kot na skupino C materialov.
- **Materiale C**, ki v vrednosti ali v porabi predstavljajo najmanjšo skupino, a po zastopanosti jih je največ. Glede prihranka pri zalogah je ta skupina najmanj pomembna.

*Tabela 3: Razredi ABC*

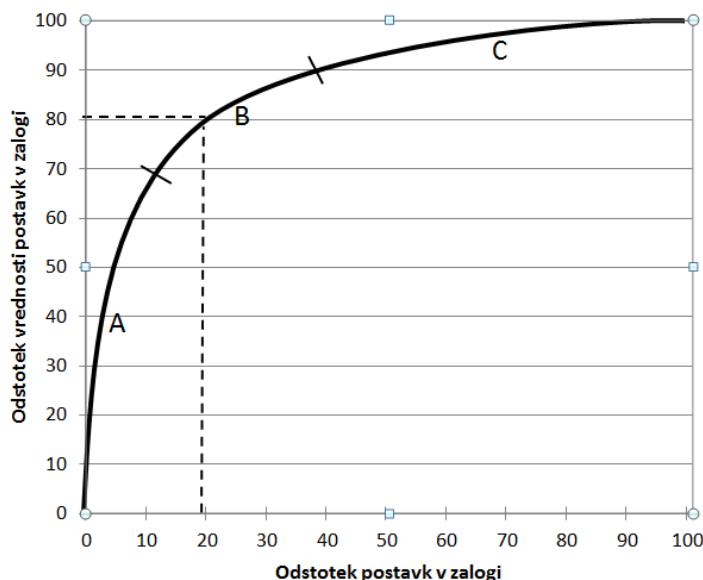
Avtor	A	B	C
Čižman (2002)	10-20 % postavk 60-80 % skupne vrednosti	20-30 % postavk 15-30 % skupne vrednosti	Ostalo 5-10 % skupne vrednosti
Završnik (2008)	10% postavk 75% skupne vrednosti	15 % postavk 15 % skupne vrednosti	75 % postavk 10 % skupne vrednosti
Rusjan (2009)	20-30 % postavk 45-55 % skupne vrednosti	20-30 % postavk 20-30 % skupne vrednosti	50 % postavk 25 % skupne vrednosti
Kobert (1992)	Navaja celo različne primere od strmih krivulj kjer zelo malo materialov predstavlja večino zaloge do praktično linearnih krivulj kjer je delež postavk enakomeren.		

Vukovič in Završnik (2011) omenjata pri ABC analizi še eno dimenzijo, in sicer pomen klasifikacije. Da bi nabavnim menedžerjem priskrbeli dodatna navodila, bi bilo potrebno vsako postavko klasificirati z vidika pomembnosti v vzdrževanju in delovanju na trištevilčni



skali: 1-kritično, 2-srednje, 3-nekritično. Na ta način bi manj pomembno postavko A označili z A-3. V določenih primerih lahko postavka C-1 zahteva večjo pozornost kot postavka A-3.

Slika 17: Splošna oblika grafične Pareto ABC analize zalog



Vir: Prirejeno po Kobert, *Managing inventory for cost reduction*, 1992, str. 199.

Politike glede zalog po posameznih skupinah materialov so zbrane v tabeli 4.

Tabela 4: S pomočjo ABC analize lahko določimo parametre za za kontrolo zalog

	<b>A</b>	<b>B</b>	<b>C</b>
Karakteristika	Velik obrat	Srednji obrat	Nizek obrat / počasi premikajoče
Število dobaviteljev	Malo - eden		
Pristop k kontroli zalog	Minimalno zalog	Optimalne zaloge	Dovolj zalog
Frekvenca naročanja	Velika (dnevi, tedni)	Srednja (tedni, meseci)	Nizka (2-4 krat letno)
Količina naročanja	Majhna	Srednja	Velika
Varnostna zaloga	Majhna točno določena	Srednja, enostavno določena	Velika
Določanje materialnih potreb	Natančno, deterministični pristop Naročanje na podlagi optimalne količine	Deterministični in stohastični pristop Periodično naročanje.	Stohastični pristop / v skladu z direktnimi zahtevami
Evidenca	Natančna evidenca količine zalog, vhodov in izhodov, cen in kakovosti	Evidenca količine zalog, vhodov in izhodov in ostalih podatkov glede na materialne skupine	Vizualna kontrola nivoja zalog in premikov
Mesto v skladišču	Najbližje mestu izdaje	Na sredini	Najdlje od mesta izdaje

Vir: Krajčovič & Plinta, *Comprehensive approach to the inventory control system improvement*, 2012, str. 38; Rusjan, *Management proizvodnih in storitvenih procesov*, 2009, str. 315.

Bulinski, Waszkiewicz in Buraczewski (2013) opozarjajo na napačne zaključke iz ABC analize, če se vsa pozornost nameni materialom iz skupine A in skupini B ter C obravnava občasno. V takem primeru podjetje lahko pride v situacijo, ko ostane brez materialov B ali C. Sčasoma se lahko struktura povpraševanja zamenja in se lahko materiali iz skupine C v prihodnosti prodajajo v večjih količinah. Opozarjajo na potrebo periodičnem izvajanju analiza ABC.

### 3.3.2 XYZ analiza zalog

Pri tej analizi, ki je dejansko modifikacija ABC analize, materiale razvrstimo v tri skupine glede na vzorec porabe, ki ga predstavljata zanesljivost in stalnost:

- **Materiale X** katerih poraba teh materialov je ustaljena, približno enaka v časovnih obdobjih. Mogoče jo je napovedovati zelo zanesljivo.
- **Materiale Y** pri katerih poraba varira. Pojavljajo se trendi in sezonske komponente. Napoved porabe je srednje zanesljiva.
- **Materiale Z** z občasno (naključno, sporadično) porabo. Napovedovanje je težko, celo nemogoče z uporabo standardnih napovednih metod.

Čižman (2002) opisuje postopek analize, ki jo izvedemo za vsako materialno postavko posebej:

1. Ugotovimo porabljeno količino po mesecih in izračunamo povprečje za ustrezno število terminskih enot, običajno mesecev (enačba 7),

$$\bar{R} = \frac{\sum_{i=1}^n R_i}{n} \quad (7)$$

kjer pomeni:  $\bar{R}$  – povprečna količina porabe v obravnavanem časovnem obdobju  
 $R_i$  – dejanska količina porabe  
 $i$  – indeks terminskih enot  
 $n$  – število terminskih enot v obravnavanem časovnem obdobju

4. Za vsako terminsko enoto določimo absolutno vrednost odstopanja porabe od povprečne porabe in nato izračunamo povprečno odstopanja porabe v ustreznem časovnem obdobju (enačba 8),

$$\bar{D} = \frac{\sum_{i=1}^n |(R_i - \bar{R})|}{n} \quad (8)$$

kjer pomeni:  $\bar{D}$  – povprečno odstopanje v časovnem obdobju

5. Izračunamo še povprečno nihanje porabe (enačba 9),

$$\bar{N} = \frac{100 * \bar{D}}{\bar{R}} \quad (9)$$

kjer pomeni:  $\bar{N}$  – povprečno nihanje porabe v časovnem obdobju v odstotkih

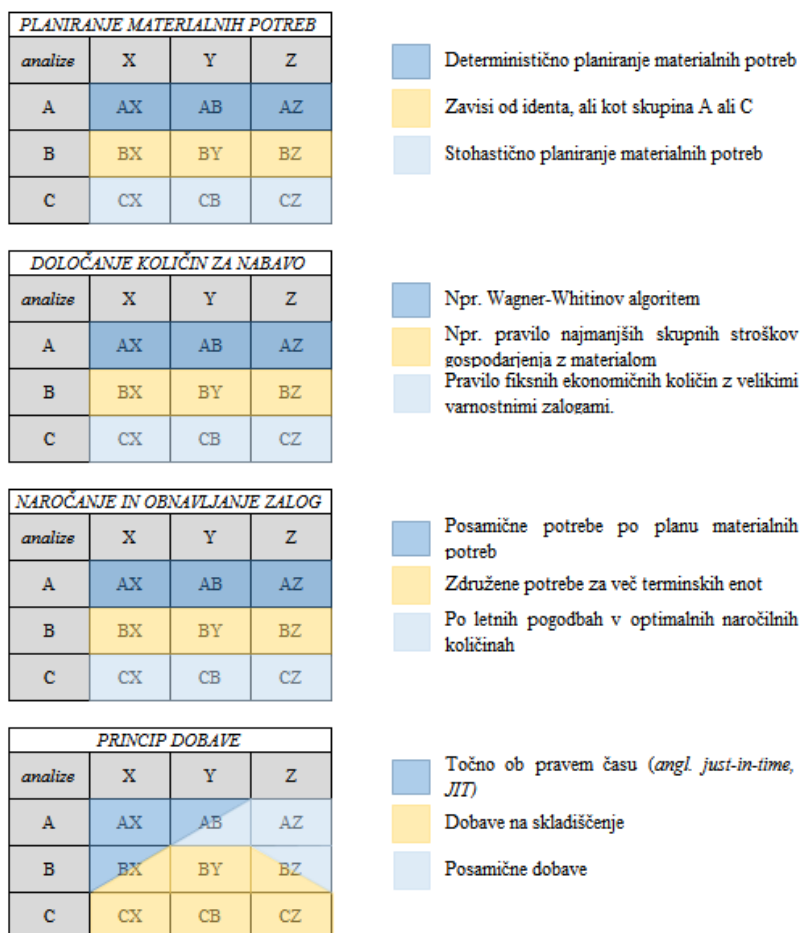
Kot pri ABC analizi tudi tukaj v literaturi ni čiste enakosti pri razvrstitvi v razrede (Tabela 5).

Tabela 5: XYZ razredi glede na nihanje povprečne porabe

Avtor	X	Y	Z
Čižman (2002)	ca 50 % postavk <20 % nihanje	20-30 % postavk 20-50 % nihanje	20-30 % postavk >50 % nihanje
Ljubič (2008)	<50 % nihanje	50-100 % nihanje	>100 % nihanje

S kombinacijo ABC in XYZ analize materiale razvrstimo kar v 9 razredov od AX do CZ. Uvrstitev posameznega materiala v določen razred pogojuje izbor politik glede: planiranja materialnih potreb, določanja količin za nabavo, naročanja in obnavljanja zalog, izdajanja materialov v proizvodnjo, knjiženja in obračuna porabe. Nekatere politike so predstavljene na Sliki 18.

Slika 18: Izbor politik na osnovi ABC in XYZ analize



Vir: Povzeto po T. Ljubič, Planiranje in vodenje proizvodnje: modeli, metode, podatki, 2000, str. 361.

### 3.4 Kazalniki zalog

Danes brez omembe koncepta vitke proizvodnje ne gre. Ta koncept vsebuje številne tehnike za izboljševanje poslovanja. Med njimi je tudi zmanjševanje zalog v proizvodnji. Schmenner (1993) razlaga, da zaloge v proizvodnji zgolj prikrivajo probleme in pomanjkljivosti kot so: slabo planiranje, dolgi časi priprave strojev, zastoji zaradi okvar in podobno.

Za merjenje uspešnosti upravljanja z zalogami uporabljamo kazalnike. To so številčni podatki, ki pokažejo stanje in so v pomoč pri odločanju o zalogah. Vprašanje je, kateri so primerni za zaloge. Splošno usmeritev lahko povzamemo kar iz Slovenskega računovodskega standarda 29 (2015). Ta med temeljne kazalnike obračanja šteje:

- koeficient obračanja zalog materiala

$$\frac{\text{porabljeni material v letu dni (po nabavnih cenah)}}{\text{povprečna zaloga materiala (po nabavnih cenah)}} \quad (10)$$

- koeficient obračanja zalog proizvodov

$$\frac{\text{prodani proizvodi v letu dni (praviloma po proizvajalnih stroških)}}{\text{povprečna zaloga proizvodov (praviloma po proizvajalnih stroških)}} \quad (11)$$

- koeficient obračanja zalog trgovskega blaga

$$\frac{\text{prodani blago v letu dni (po nabavni vrednosti)}}{\text{stanje zalog blaga (po nabavni vrednosti)}} \quad (12)$$

- koeficient mesečne zadostnosti zaloge proizvodov

$$\frac{\text{stanje zalog proizvodov (po proizvajalnih stroških)}}{\text{povprečna mesečna prodaja (po proizvajalnih stroških)}} \quad (13)$$

Pri kazalnikih obračanja zalog ne smemo pozabiti na dejstva, ki jih navaja Duhovnik (2003), ki pravi, da na stanje zalog poleg metod vrednotenja zalog vplivajo tudi naslednja dejstva:

- Hitrejše ali počasnejše obračanje zalog je lahko posledica tudi različnih pogojev poslovanja, npr.: povečanje cen proizvodov podjetja, poslabšanje ponudbe na nabavnem trgu, uvajanje novih tehnologij, sprememba optimalnega obsega nabave zaradi spremenjenih nabavnih pogojev.
- Nihanje v hitrosti obračanja zalog je lahko posledica sprememb cen, ne da bi se obenem spreminjale tudi količine. Posebno pozornost je potrebno nameniti nihanju cen zalog v primerjavi z nihanjem cen gotovih proizvodov. Kazalnik obračanja zalog se spremeni

tudi pri nespremenjenih cenah in količinah, če se spremeni računovodska usmeritev glede vrednotenja zalog.

- Stanje zalog na določen dan je odvisno tudi od dejavnosti podjetja. Pri podjetjih, ki opravljajo več dejavnosti, se strukturne spremembe izražajo v hitrosti obračanja zalog. Če so dejavnosti sezonske narave, so nihanja toliko večja.

### 3.5 Ukrepi za zmanjšanje zalog

Rusjan (2009) ugotavlja, da moramo za vsako zalogo v podjetju vedeti, zakaj je nastala, šele potem lahko sprejemamo ustrezne rešitve glede zmanjševanja zalog. Zalog ne smemo zmanjšati, če zanje obstajajo utemeljeni razlogi. Pravi, da bomo morali najprej odpraviti razloge za zaloge in šele potem lahko zmanjšamo zalogo. Krajčovič in Plinta (2012) navajata, da z zmanjšanjem zalog dosežemo še

- prihranek prostora. Skladišča dosegajo 15-30 % površin podjetja,
- prihranek pri delavcih, vsaka zaloga zahteva pisarniško in skladiščno delavno silo;
- prihranek pri skladiščni opremi: skladiščna oprema za hranjenje, premikanje, embalaža,
- poenostavitev kontrole logističnega toka,
- poenostavitev evidenc in povečanje transparentnosti,
- izboljšanje podobe podjetja.

Standardnega pristopa k zmanjšanju zalog, ki bi ga potrjevalo več avtorjev, v literaturi nismo našli. Veliko pa je opisov upravljanja zalog, ki vsebujejo analize zalog, določanje ekonomične naročilne količine in opise oblik sistemov naročanja. Glede potrebnih aktivnosti za zmanjšanje že obstoječih zalog kaže povzeti nekatere ugotovitve avtorjev: Cox (2013), Krajčovič in Plinta (2012), Rusjan (2009). Če te ugotovitve strnemo v seznam, razdeljen po področjih, in dodamo še nekaj praktičnih ugotovitev v avtomobilski industriji, lahko zaloge zmanjšamo z naslednjimi ukrepi:

1. Ukrepi, vezani na skladiščni prostor:
  - a) Zmanjšati stroške prostora. V poštev pride leasing, javna skladišča.
  - b) Zagotoviti optimalno shranjevanje v skladišču, uporaba opreme za manipulacijo v ozkem prostoru.
2. Ukrepi, vezani na dobaviteljsko verigo:
  - a) Vpeljati zbirne Transporte (angl. *milkrun*). To vodi k zmanjšanju transportnih stroškov, saj je lažje dosežemo bolj polne kamione, ki obenem opravijo običajno krajšo pot, kot jo bi sicer, če bi blago vozili od vsakega dobavitelja posebej.
  - b) Posluževati se zbirnih skladišč, ki so blizu več dobaviteljem. Tako racionaliziramo prevoze.
  - c) Varnostne zaloge lahko zmanjšamo z odpravo negotovosti, zaradi katerih so določene.

3. Ukrepi vezani na lastno proizvodnjo:
  - a) Serijske zaloge lahko zmanjšamo z zmanjšanjem proizvodnih serij. Osnovni pogoj je skrajšanje časov priprave strojev. Druga možnost je zmanjšanje minimalnih nabavnih količin na podlagi dogovorov z dobavitelji.
  - b) Sezonske zaloge lahko zmanjšamo s povečanjem prilagodljivosti proizvodnega procesa in z najemom sezonskih delavcev.
  - c) Varnostne zaloge zaradi možnih napak v proizvodnji ali strojeloma lahko zmanjšamo z zmanjšanjem verjetnosti nastanka. Napake v proizvodnem procesu zmanjšuje dvig kakovosti. Glede strojeloma moramo povečati reaktivnost serviserjev in izboljšati hitro dobavljivost rezervnih delov.
  - d) Nekurantna zaloga je mrtev kapital in moramo njen nastanek preprečiti, saj pomeni izgubo za podjetje. To vrsto zaloge enostavno ugotovimo ob analizi obračanja zaloge. Koeficient obračanja (enačba 14) je takrat 0 ali pa zelo majhen. Vendar je, ko tako zalogo ugotovimo, vse že prepozno. Ukrepi so potrebni pred tem, saj je potrebno preprečiti nastanek nekurantne zaloge. K ukrepom bi šteli dobro sodelovanje celotne dobaviteljske verige in pravočasno napoved koncev serij izdelkov. Druga možnost je še načrtovanje konca serij s pravočasnimi popusti pri prodaji. Seveda pa ne gre brez uvedb metod napovedovanja porabe.
  - e) V Skupini TPV smo ob koncu serije izdelkov za vozilo Twingo (Renault oznaka X44), ko se je končala življenska doba 440 serijskih materialov, razvili aplikacijo, narejeno v PowerPivot s pomočjo jezika DAX, ki je ob vsaki tedenski napovedi končnega kupca o količinah izdelkov, ki jih še rabijo do zaustavitve serije, izvedla analizo vseh zalog in odvisnih ter neodvisnih potreb. Rezultat je bil pričakovani ostanek za vsak material. Sledile so aktivnosti: brisali smo nastavitve, ki so povzročale varnostne zaloge, nastavili smo parametre MRP, ki so zagotavljali, da smo materiale naročali prej in smo s tem navidezno zagotavljali varnostne rezerve, brisali smo lahko fiksne potrebe, ki so sčasoma nastale pri posameznih materialih. Neverjetno je, koliko različnih anomalij v praksi skriva previsoka varnostna zaloga.
4. Drugi ukrepi:
  - a) Redno izvajati popis materiala. Nepravilno stanje lahko povzroči nedobavo kupcu. Da bi to preprečili, imamo zaradi možnosti nepravilnega stanja običajno večje varnostne zaloge, ki jih ne bi potrebovali, če bi se dokumentarno stanje ujemalo s stanjem s skladišču. Ob popisih je nujno ugotavljati razloge za nastanek nepravilnega stanja in jih odpravljati s korektivnimi ukrepi. Dostikrat so razlogi samo v nepravilnem knjiženju izmeta.
  - b) Redno spremljati izbrane kazalnike, jih objavljati na ustreznih mestih, pregledovati in analizirati rezultate z delavci, ki so neposredno udeleženi v tem logističnem procesu in sprejemati ustrezne korektivne ukrepe za stalni napredek. Pri izrazito sezonskem poslovanju je treba primerjati rezultate z istim obdobjem predhodnega leta, saj bi primerjava mesec proti mesecu dala napačno sliko.

Kaltnekar (1998) navaja, da je stroške možno znižati predvsem z odpravljanjem vseh nepotrebnih elementov v proizvodnji. Toyota sistem pogosto opisujejo kot »izcejanje vode iz suhe brisače«. V seznam izgub navaja nepotrebne zaloge surovin in gotovih proizvodov ter dolg čas zadrževanja materiala v samem proizvodnem procesu.

## **4 TABELARIČNI MODEL ZA ANALIZO ZALOG SKUPINE TPV**

Uporabnost tabelaričnega modela preverimo na podatkih, ki prikazujejo realno kompleksnost. Za ta test izberemo področje zalog Skupine TPV za obdobje od 01.01.2015 do 17.02.2016. Letno gre za skoraj 3.600.000 blagovnih premikov. Kot celovito programsko rešitev Skupina TPV uporablja ERP (angl. *enterprise resource planning*) sistem SAP. V SAP je vpisanih 160 skladišč, bodisi fizičnih bodisi logičnih, in beleži se 241 različnih vrst premikov blaga. Tako je podatkov dovolj in kompleksnost je primerna za potrditev ali zanikanje trditve, da je s tabelaričnim modelom mogoče pridobiti relevantne informacije, potrebne za analizo zalog. Obenem se zaradi zasnove DW opisanega v nadaljevanju lahko preveri tudi druge sekundarne cilje naloge. Cilj, vezan na pomen granulacije vira podatkov, je zelo pomemben. Referenčni podatki izvirajo iz SAP transakcije MC.9, ki se tudi sicer v Skupini TPV uporablja za analizo zalog. Ker transakcija v svetu podatkovnih baz pomeni zaporedje opravil sistema za upravljanje podatkovnih baz, moramo takoj pojasniti, da gre pri SAP transakcijah za 4 mestno kodo sestavljeno iz črk in števil. Ta koda je bližnjica do SAP opravila (angl. *task*) oziroma funkcije, do katere pridemo običajno preko menija.

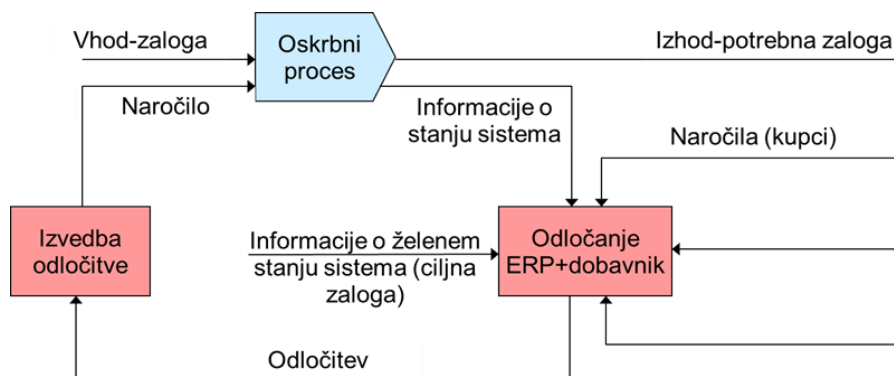
### **4.1 Skupina TPV in njen oskrbni proces**

Skupina TPV je razvojni dobavitelj, delujoč na globalnem trgu avtomobilske industrije. Produkti programi Skupine TPV so povezani v dveh poslovnih divizijah, AvtoIN ter Vozila. V okviru divizije AvtoIN se razvija in proizvaja sklope karoserije ter podvozja, avtomobilske sedeže, elemente avtomobilskih sedežev ter tesnila motorjev za potrebe avtomobilskih proizvajalcev in sistemskih dobaviteljev. Ob tem se razvija in ponuja celovite stroškovno učinkovite rešitve za potrebe upravljanja proizvodne logistike, kjer so avtomatsko vodena vozila eden izmed ključnih produktov. Na drugi strani divizija Vozila povezuje program proizvodnje lahkih tovornih prikolic, program prodaje novih in rabljenih avtomobilov in program celovitih storitev za uporabnike vozil. Skupina je imela v letu 2014 130 milijonov € prihodka in je zaposlovala 1071 delavcev (Skupina TPV, 2015). Poleg omenjenih divizij v sodelovanju z nemškim partnerjem Boeckmann Skupina TPV razvija in proizvaja lahke tovrne prikolice, s katerimi je po tržnem deležu drugi največji proizvajalec v Evropi. Skupina TPV je tudi eden od družbenikov v Razvojnem centru SiEVA, v sklopu katerega razvija tehnološke kompetence in deluje na strateških raziskovalnih projektih. Skupina TPV je s svojimi proizvodnimi zmogljivostmi prisotna na 5 lokacijah v Sloveniji. Poslovne enote so tako v krajih: Novo mesto, Brežice, Suhor, Velika Loka in Ptuj. Ena poslovna enota je še v srbskem Kragujevcu. V SAP ima vsaka poslovna enota svojo štirimestno šifro.

Vsaka poslovna enota ima več fizičnih skladišč in še bistveno več logičnih skladišč. Zaloge, ki se skladiščijo, so planirane na osnovi večinoma tedenskih naročil kupcev. Vsi veliki kupci tedensko ali dnevno po elektronski izmenjavi podatkov (angl. *EDI – electronic data interchange*) pošiljajo obdobje fiksnih količin in različno dolga obdobja predvidevanj. Obdobja fiksnih količin so lahko od nekaj dni do enega meseca. Obdobja predvidevanj pa sežejo tudi pol leta vnaprej ali še dlje. V času priprave poslovnih načrtov so dosegljive tudi letne napovedi. Kot v vsej avtomobilski industriji je v predvidevanjih pričakovana nedoločenost, saj nihče ne pozna točno obnašanja avtomobilskega trga, na katerega vpliva vsa gospodarska situacija. Nenapisano pravilo pravi, da lahko napovedi odstopajo +/- 20%. V stabilnih razmerah so nihanja dosti manjša. Na drugi strani imajo poslovne enote svoje dobaviteljske verige. Predvidevanja kupcev se preslikajo v predvidevanja po dobaviteljskih verigah navzdol. Dobavni časi so zelo različni, od enega dneva do treh mesecev. To zadnje predstavlja velik izziv.

Proizvodnja poteka serijsko ali na podlagi delovnih nalogov. Naročilo kupca ni tisto, ki sproža naročilo proizvodnji, ampak je to niveliran proizvodni plan. Niveliranje po principih *heijunka* je uvedeno zaradi doseganja vitke proizvodnje. Serijska proizvodnja gre v zaloge končnih proizvodov. Tu nas zanima kakšna je optimalna raven zalog posameznih proizvodov. Cilji so določeni izkustveno in manj empirično. Vpliv odzivnosti dobaviteljev v primeru strojeloma in variabilnost kupcev sta ocenjena po najboljših močeh. Oskrbni proces v podjetju poteka skladno s Sliko 19.

Slika 19: Oskrbni proces v podjetju



Vir: I. Makovec, *Uporaba orodja poslovne inteligence pri upravljanju z zalogami v podjetju avtomobilske industrije*, 2012, str. 7.

Poslovne enote za izračun odvisnih potreb uporabljajo MRP (angl. *material requirements planning*), ki je funkcionalnost SAP. Izračun potrebnih količin materialov se začne s podatki o načrtovanih količinah proizvodnje iz glavnega proizvodnega načrta in se na osnovi podatkov iz kosovnic določi količino potrebnih sestavnih sklopov ter materialov za izdelavo načrtovanih izdelkov glede na datume. Časovno naravnano načrtovanje materialov se doseže s t.i. eksplozijo kosovnic in upoštevanjem obstoječih zalog in že naročenih količin ter z



zamikanjem ustreznih začetnih časov posameznih proizvodnih operacij. Kot vedno je rezultat MRP transakcij skladen z vhodnimi podatki. Da pa podatki za naročanje materiala niso optimalni, je več razlogov: naročila ni niso zaokrožena na embalažne enote in končni kupec količine samo napove, in ne fiksira (Makovec, 2012).

Skupina TPV v skrbi za dolgoročno uspešnost poslovanja, stalno izvaja več konceptov vitke proizvodnje. Cilj je neprekinjeno izboljševanje poslovanja. Obseg zalog je pri tem pomemben, saj vežejo veliko denarja. Zavedanje, da brez njih ni možno obstaja, a prav tako obstaja stalna težnja po zmanjševanju in s tem po povečevanju učinkovitosti poslovnih enot v Skupini TPV. Pričakuje se, da bo z uvedbo BI to področje lažje pokriti in da bo možno izkoristiti vrednost poslovnih podatkov v ERP sistemu.

## **4.2 Izdelava prototipnega modela**

Najhitreje do rešitve vodi prototipiranje s sprotnim preverjanjem rezultatov. S prototipom se tudi učimo in ga popravljamo do končne rešitve. Življenjski cikel prototipiranja ima tako naslednje faze (Kovačič, 2009):

1. ugotavljanje informacijskih potreb,
2. razvoj osnovnega prototipa,
3. uporaba in beleženje sprememb,
4. izboljšanje prototipa in vrnitev na tč. 3 (če je potrebno),
5. ocena delujočega sistema,
6. izvedba sprememb,
7. instalacija, delovanje in vzdrževanje.

Faze dejansko sovpadajo s fazami življenjskega cikla razvoja sistema BI kot je bil predstavljen v poglavju 2.4. Razvoj sistema BI je moč razumeti kakor večni postopek prototipiranja, saj se skladno z novimi potrebami vedno spreminja in izboljšuje. Ker v nadaljevanju razvijamo sistem BI, se bomo ravnali po fazah, kot so bile navedene v poglavju 2.4. Spustili bomo faze: planiranje projekta, uvedba v produkcijo in vzdrževanje ter rast, saj jih ne bomo izvedli, ker niso potrebne za test hipoteze in za realizacijo v nalogi postavljenih ciljev.

### **4.2.1 Definiranje poslovnih potreb**

Namen naloge je izdelati prototip BI rešitve za analizo zalog z uporabo tabelaričnega pristopa. Med cilji je pregled zalog in identifikacija potencialov za zmanjšanje zalog oz. pridobitev informacij za izbor ustreznih nabavnih metod za posamezne materiale. Pri tem je bistvena ABC analiza. Kazalnik obračanja zalog izračunamo po enačbi 14.

$$Dnevi\ vezave\ zalog = \frac{\text{Število\ dni\ v\ mesecu}}{\text{Koeficient\ obračanja\ zalog}} = \frac{\text{Število\ dni\ v\ mesecu}}{\frac{\text{Poraba\ (€)}}{\text{Povprečna\ zaloga\ (€)}}} \quad (14)$$

#### 4.2.2 Dizajn tehnološke arhitekture

Preverimo, katero informacijsko infrastrukturo imamo na voljo. Skupina TPV kot poslovno informacijski sistem oziroma kot celovito programsko rešitev od leta 2004 uporablja SAP, sedaj v verziji ECC 6.0 EHP. Za SAP produkcijsko instanco R3P sta v uporabi dva strežnika IBM x3650 M2 z Windows 2008/64 SP2, ki sta povezana v cluster. Baza MS SQL 2008 R2 in aplikacija SAP sta ločeni instanci. Baza R3P je velika 2,5 TB. Za SAP instance: test R3Q, razvoj R3D, SOLMAN in PI se uporabljajo virtualni strežniki. Virtualizacija je izvedena z opremo VMware in dvema fizičnima strežnikoma IBM x3650 M4. Kot diskovni sistem se uporablja diskovni sistem IBM z SVC kontrolerji, na katere sta priključena dva diskovna podsistema V3700. V vseh poslovnih enotah je v uporabi preko 500 PC. Standardno so v uporabi Office 2013 programska orodja.

Za izvedbo prototipa imamo na voljo PC HP ProBook 6560B z 8 GB RAM, procesorjem Inter Core i5-2450M, 2,50 GHz s 64-bitnim operacijskim sistemom Windows 7. Ocenjujemo, da mora zadostovati za namen izvedbe prototipa. Imamo še opcijo povečati RAM do 16 GB, če ne bi zadostovala obstoječa kapaciteta.

#### 4.2.3 Izbira programske opreme

Skupina TPV še nima vzpostavljenega korporacijskega sistema BI, vendar se na to pospešeno pripravlja. V uporabi je SQL Server 2012 Express in SharePoint 2013 portal. Poizvedbe za poročila so napisane v T-SQL jeziku in se objavljajo na portalu. Vir podatkov so izvozi SAP transakcij »Z«, ročni izvozi standardnih SAP transakcij in drugi viri kot Access in Excel. Osebne in timske BI rešitve se izvajajo z Excel, Access in v zadnje času s PowerPivot. Veliko analitik je narejenih v SAP z namenskimi tako imenovanimi »Z« transakcijami, ki so v bistvu SAP poizvedbe ali ABAP programi. Namensko SAP transakcijo spoznamo po tem, da se začne na črko »Z«. »Z« transakcije je možno nastaviti tako, da je možen periodični samodejni izvoz rezultatov v določeno mapo. Možno je tudi nastaviti ločilo (angl. *delimiter*) med podatki.

Za prototip izberemo programsko rešitev Microsoft SQL Server 2012 Developer X64 zaradi nekaj pragmatičnih razlogov:

- kot uporabniški vmesnik v analitično - prezentacijski plasti se lahko uporabi vsem poznani Excel in je faza učenja zelo skrajšana,
- znanja so lahko dosegljiva na internetu,
- programska oprema je dosegljiva preko Microsoft MSDNAA (*Microsoft Developer Network Academic Alliance*).

Najtežji del odločitve je, katero verzijo SLQ Serverja izbrati za doseglo namena. Microsoft pri novih izdajah pogosto spremeni poimenovanja verzij in je potreben vsakokratni skrbni pregled funkcionalnosti. Verzije: Web, Standard, Business Intelligence in Enterprise so namenjene poganjanju na serverjih in za produkcijske namene organizacije. Za poganjanje na osebnih računalnikih so primernejše verzije: Express, Express z naprednimi servisi, Express z orodji in Developer. Prava odločitev je nedvomno verzija Developer, ker vsebuje vse, kar vsebuje tudi verzija Enterprise. Razlika je samo v tem, da licenca ne dovoljuje produkcijskega dela, omogoča pa razvoj. Pomembne komponente programske rešitve so:

- SSIS (SQL Server Integration Services) je orodje, namenjeno izgradnji podatkovnih skladišč in za izvedno ETL postopka, poleg tega skrbi za avtomatizacijo nalaganja podatkov in administrativne funkcije;
- SSAS (SQL Server Analysis Services) je orodje za OLAP in tabelarične analize ter podatkovno rudarjenje, v bistvu je analitična podatkovna baza, ki omogoča dostop klientom;
- Excel 2013 je odličen odjemalec (angl. *client*), je sestavni del Office 2013 nabora programskih orodij.

#### 4.2.4 Dimenzijsko modeliranje

Poslovne potrebe imamo znane in lahko takoj zapišemo BUS matriko podatkovnega skladišča, kot je to opisano v poglavju 2.4.5 (Slika 20). Vsebinsko gre za en poslovni proces in sicer za vodenje zaloga, ki spadajo pod logistiko. V našem primeru gre za dve tabeli dejstev z različno granulacijo in tudi deloma različnimi pripadajočimi dimenzijami.

V ta del spada tudi podatkovna analiza: pregled razpoložljivosti podatkov, pregled konsistentnosti, validacija vrednosti, pregled podrobnosti, računski pravila za podatke iz tabel dejstev, pravila za počasi se spreminjajoče dimenzije itd. Te informacije so pomembne za sestavo kataloga metapodatkov.

Pri pregledu razpoložljivosti virov podatkov ugotovimo, da lahko uporabimo izpise več »Z« transakcij in izpise več tabel. Izpise je možno prožiti po urniku, npr. vsak dan, kar povsem zadostuje. Vsi izpisi so v obliki ER (angl. *entity relationship*) tabel, a v različnih formatih: tekstovni, Excel. Zadoščajo naslednji viri:

- Za dimenzijske tabele:
  - izpis vseh dobaviteljev – SAP transakcija ZDOBAVITELJI,
  - izpis vseh kupcev - SAP transakcija ZKUPCI,
  - izpis matičnih podatkov materialov – SAP transakcija ZMAT\_LISTA2,
  - tabela vseh vrst premikov – SAP tabela H\_T156,
  - tabela vseh vrst dokumentov – SAP tabela T003,

- tabela skladišč z nazivi – SAP tabela T001L,
- tabela obratov oz. poslovnih enot z nazivi – SAP tabela T001W.
- Za tabeli dejstev:
  - izpis vseh materialnih premikov – SAP transakcija ZALOG2. Inicialni izvoz iz SAP za opazovano obdobje slabih 14 mesecev je trajal v ozadju 3,6 ure in tekstovna datoteka je bila velika skoraj 800 MB. Seveda je dnevni prirast ustrezno manjši;
  - izpis rezultata SAP transakcije MC.9, ki je logistično poročilo LIS (angl. *Logistics Information System*) z granulacijo enega meseca in uporablja tabeli S031 ter S032. Podatki se zbirajo mesečno na koncu meseca. Gre v bistvu za neke vrste mesečno popisovanje stanja zalog (angl. *inventory snapshot*). V tabeli S031 so zapisi o materialnih premikih na mesečni osnovi in v tabeli S032 trenutna zaloga in vrednost zaloge.

Slika 20: BUS matrika podatkovnega skladišča

POSLOVNI PROCES	DIMENZIJE							
	Datum	Matični podatki materiala	Skladišča	Poslovne enote - obrati	Kupci	Dobavitelji	Vrsta dokumenta	Vrsta premika
Zaloge na osnovi mesečnega zajema podatkov	X	X	X	X				
Materialni premiki	X	X	X	X	X	X	X	X

Pri večini navedenih SAP transakcij je bil izziv katere tako-imenovane kazalnike (v bistvu attribute) izbrati za izpis. Npr. SAP transakcija MC.9 ima v naboru preko 50 kazalnikov. Za izpis aktiviramo samo relevantne. Rezultate »Z« transakcij izvozimo kot tekstovne datoteke (ekstenzija txt) s TAB razmejljnikom med posameznimi podatki (angl. *tab delimited*). Navedene »Z« transakcije niso standardni del SAP, ampak so razvite v Skupini TPV za razne namene, tudi namene poročanja. SAP tabele izvozimo s SAP transakcijo SE16.

Pri pregledu validacije podatkov ugotovimo več nekonsistentnosti in podrobnosti, ki vplivajo na nadaljnje delo, predvsem na ETL postopek:

- Format izpisa podatkov ni vselej enak. Lep primer je številka materiala, ki je enkrat lahko tekstovnega formata z vodilnimi ničlami do 6 mest in drugič številka.
- Števila so v tabeli vseh materialnih premikov so zapisana na način, da je ločilo tisočic vejica, decimalno ločilo pika in minus na desni strani.

Takojšnja in pragmatična odločitev je, da vse podatke v SSIS uvozimo v tekstovnem formatu [DT\_STR] različnih dolžin 10, 18 ali 50 znakov. Mere se lahko potem v številke pretvori v

fazi transformacije. Enako se lahko pretvori datume in čas iz tekstovnega zapisa v datumski oz. časovni zapis.

Russo (2012a) trdi, da za delo z metapodatki v SQL Serverju 2012 ni podpore, vsaj ne vgrajene podpore. Pregled rešitev tretjih ponudnikov pokaže, da je možno dobiti poročilo oz. seznam metapodatkov, a šele po postavljeni strukturi v SSIS. To nam v ničemer ne pomaga, saj potrebujemo plan sprememb imen in tipov podatkov sočasno ali pred postavljanjem strukture v SSIS. V našem primeru, če seštejemo vse tabele, gre skupno za skoraj 300 atributov.

Ker izvedemo več:

- preimenovanj iz nerazumljivih imen v »govoreča« imena tako, da ime kaže na vsebino (Slika 21). Ta del je zelo pomemben, ker so originalna imena atributov kot npr. pri SAP transakciji MC.9 lahko celo zavajajoča. »SkVredZa« ni skupna vrednost zalog ampak vrednost porabe obdobja in preimenovanje v »PorabaVredn« je pravzaprav nujno
- sprememb formatov zapisov,
- uvedb dodatnih atributov v podatkovni tok ali pa uvedemo primarne kjuče iz dimenzijskih tabel, ki postanejo tuji ključi v tabelah dejatev,

naredimo katalog metapodatkov na način, kot je prikazan na Sliki 22 in na Sliki 23. Katalog nam je v izredno pomoč pri premapiranjih ob zapisu podatkov v SQL bazo. Obenem služi kot dokumentacija te faze.

*Slika 21: Preimenovanja atributov takoj po branju vira – izpis SAP transakcije MC.9*

External Column	Output Column
PovVrVrZal	PovZalogaVredn
PovVrVrZal enota	PovZalogaVrednEnota
Celot zal	ZalKonecObdKolicina
Celot zal enota	ZalKonecObdKolicinaEnota
Celot por	PorabaKolic
Celot por enota	PorabaKolicEnota
SkVredZa	PorabaVredn

Vsako spreminjanje imen po postavitvi SSIS strukture je zelo nerodno opravilo, saj je potrebno prekontrolirati celoten tok podatkov, celo več, SSIS paket lahko celo deluje, a na koncu podatkov ni, saj smo jih preimenovali ali niso bili ustrezno premapirani.

Slika 22: Del metapodatkov za dimenzijo DimMaterial

Ime tabele v DW:		DimMaterial		Checksum: Uporabi SCD							
SAP Query		ZMAT_LISTA2									
Datoteka		Zmat_lista2.txt									
VIR (SAP, Excel, drugo)				SSIS		SQL DW				Opomba	
Atribut	Primer entitete	Tip na vhodu (ekstrakcija)	Atribut (novo ime)	Tip po transformu	KLUČ	SCD tip	Atribut	Tip	KLUČ		
							MaterialID	int	PK	Generira SQL	
			ZacetekVeljavnosti				ZacetekVeljavnosti	date	TK	Tuj ključ	
			KonecVeljavnosti				KonecVeljavnosti	date	TK	Tuj ključ	
			ManjkClanDim_Ind				ManjkClanDim_Ind	bit	TK	Tuj ključ	
Material	6736	[DT_STR], 10	Material			BK	Material	varchar (10)			
Številka materiala	PLOŠČA NOSILNA D	[DT_STR], 50	NazivMateriala			2	NazivMateriala	varchar (50)			
Kratki tekst materiala	PLOŠČA NOSILNA D										
TipPr	ROH	[DT_STR], 10	TipMateriala			2	TipMateriala	varchar (10)			
Stara št. materiala		[DT_STR], 50	StaraStMateriala			2	StaraStMateriala	varchar (50)			
Klasifikacija	5238869	[DT_STR], 50	Klasifikacija			2	Klasifikacija	varchar (50)			
EMN	KOS	[DT_STR], 10	EnotaMateriala			2	EnotaMateriala	varchar (10)			
Neto teža	0,16	[DT_STR], 18	NetoTeza			2	NetoTeza	decimal(18,3)			
En	KG	[DT_STR], 10	EnotaNetoTeze			2	EnotaNetoTeze	varchar (10)			
Obr.	1220	[DT_STR], 10	SifraObrata			BK	SifraObrata	varchar (10)			

Slika 23: Del metapodatkov za tabelo dejstev FaktPremiki

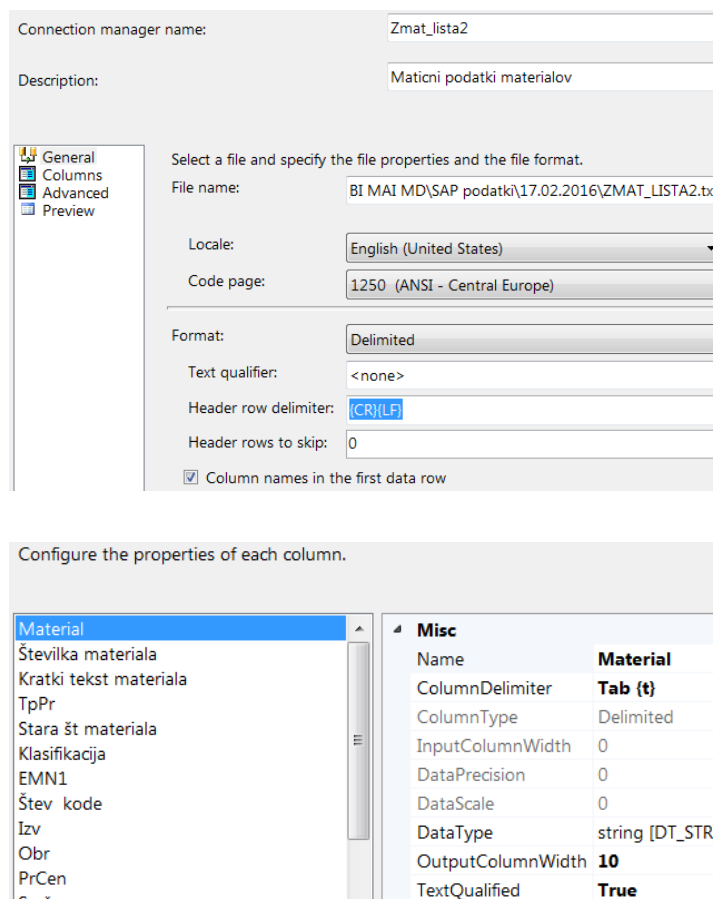
Ime tabele v DW:		FaktPremiki		Checksum: Na attribute BK							
SAP Query		ZALOG2									
Datoteka		zaloge2.txt		SAP izpiši z izgledom: " \Za Data Warehouse"							
VIR (SAP, Excel, drugo)				SSIS		SQL DW				Opomba	
Atribut	Primer entitete	Tip na vhodu (ekstrakcija)	Atribut (novo ime)	Tip po transformu	KLUČ	SCD tip	Atribut	Tip	KLUČ		
							PremikID	int	PK	Generira SQL	
			SkLokacijaTK	int			SkLokacijaTK	int	TK	Tuj ključ	
			ObratTK	int			ObratTK	int	TK	Tuj ključ	
			KupecTK	int			KupecTK	int	TK	Tuj ključ	
			DobaviteljTK	int			DobaviteljTK	int	TK	Tuj ključ	
			VrstaPremikaTK	int			VrstaPremikaTK	int	TK	Tuj ključ	
			VrstaDokumentaTK	int			VrstaDokumentaTK	int	TK	Tuj ključ	
			MatPodMaterialaTK	int			MatPodMaterialaTK	int	TK	Tuj ključ	
			DatumKnjizenjaTK	int			DatumKnjizenjaTK	int	TK	Tuj ključ	
			DatumVnosaTK	int			DatumVnosaTK	int	TK	Tuj ključ	
Obr.	1260	[DT_STR], 10	SifraObrata			BK	SifraObrata	varchar (10)			
Material	405431	[DT_STR], 10	Material			BK	Material	varchar (10)			
Številka materiala	NOSILEC REDUKTOR	[DT_STR], 50	NazivMateriala				NazivMateriala	varchar (50)			
Slok	76	[DT_STR], 10	SifraSkladisca			BK	SifraSkladisca	varchar (10)			
VrP	101	[DT_STR], 10	SifraVrstePremika			BK	SifraVrstePremika	varchar (10)			
Količina	12	[DT_STR], 50	Kolicina	(DT_NUMERIC,18,3)			Kolicina	decimal(18,3)			
EMN	KOS	[DT_STR], 10	EnotaKolicine				EnotaKolicine	varchar (10)			
Neto teža	8,5	[DT_STR], 50	NetoTeza	(DT_NUMERIC,18,3)			NetoTeza	decimal(18,3)			
En1	KG	[DT_STR], 10	EnotaNetoTeze				EnotaNetoTeze	varchar (10)			
Skupna teža	102	[DT_STR], 50	SkupnaNetoTeza	(DT_NUMERIC,18,3)			SkupnaNetoTeza	decimal(18,3)			
En2	KG	[DT_STR], 10	EnotaSkupneNetoTeze				EnotaSkupneNetoTeze	varchar (10)			
Dat.knj.	8.1.2015	[DT_STR], 10	DatumKnjizenja	[DT_DBDATE]		BK	DatumKnjizenja	date			
Čas	7:16:15	[DT_STR], 10	CasVnosa	[DT_DBTIME]		BK	CasVnosa	time			

#### 4.2.5 Fizični dizajn, dizajn zadrževalnega dela in razvoj – ETL

V konkretnem primeru ne namenimo pozornosti kreiranju preagregiranih tabel in pozornost, ki ju Kimball (2002) razume kot bistvena sestavna dela pri fizičnem dizajnu. Takoj preidemo na razvoj ETL strukture. Postopek v SSIS je razviden iz Slike 25, ki kaže tek ETL postopka za primer dimenzije DimMatPodMateriala, ki ima 42 atributov. Kot je razvidno, je bilo skozi ETL postopek v podatkovno skladišče zapisanih 67.203 materialov z matičnimi podatki. V SSIS je potrebno vsako komponento nastaviti. Najprej v *Connection Manager* nastavimo povezavo (Slika 24) do vira podatkov, ki je v našem primeru tekstovna datoteka (angl. *flat file*). V tem delu moramo omeniti pomen ekstraktorjev. Gre za

programska orodja – vmesnike med ETL postopkom in ERP sistemom. Lep tak primer je ekstraktor Theobald Xtract IS, ki poveže SAP ERP in SSIS tako, da v SSIS doda komponente, ki so namenjene medsebojni povezavi. Iz SAP tako lahko prenesemo tabele, SAP poizvedbe, SAP ABAP poročila, SAP BW info kocke, SAP BW poizvedbe in drugo. Ekstraktorja nimamo na voljo, zato so vsi viri izpisi iz SAP transakcijskega sistema v datoteke.

Slika 24: Nastavitev povezave do vira podatkov in nastavitev tipa atributa

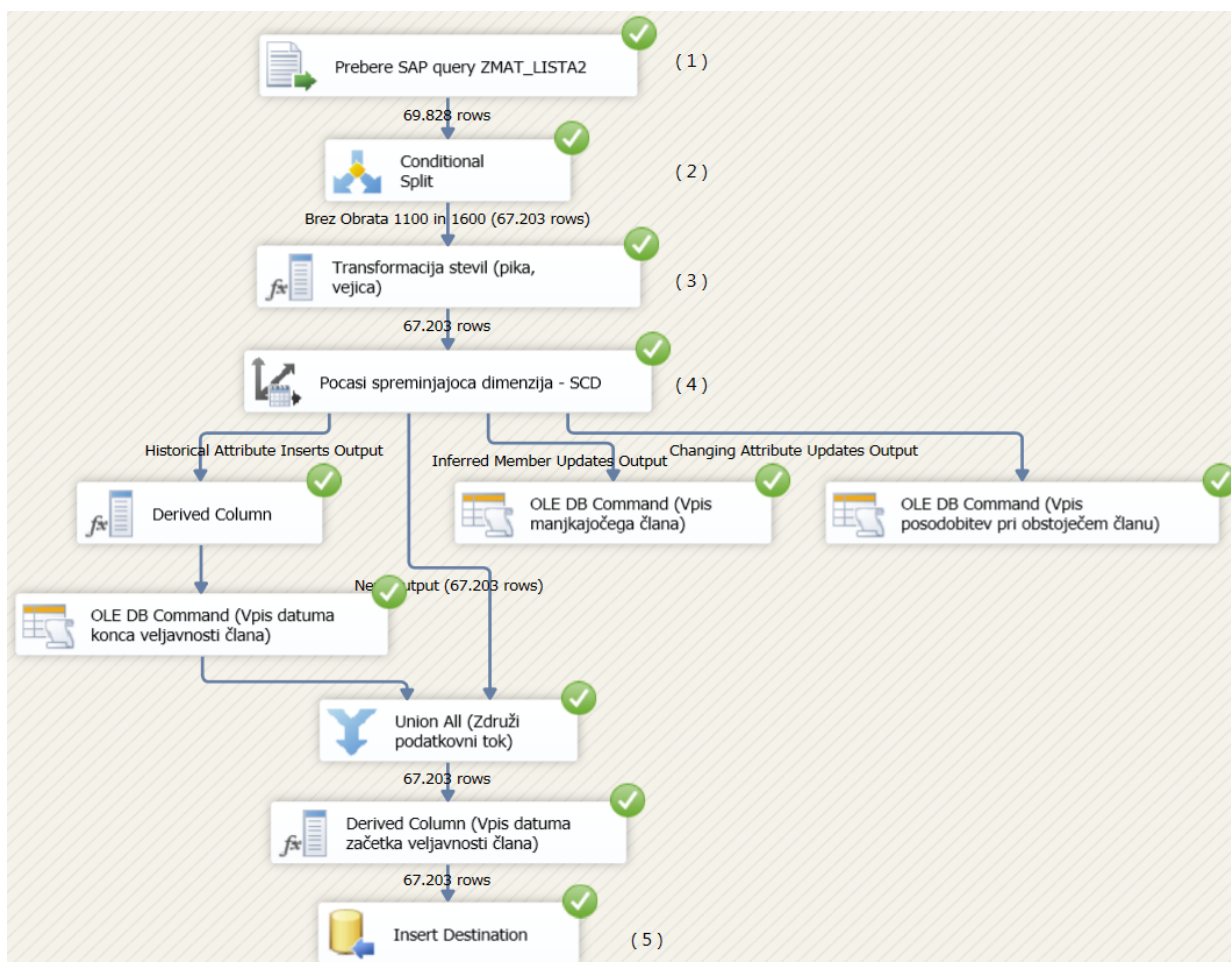


Sledi nastavitve vsake komponente posebej (številke ob komponentah na Sliki 25):

1. Branje podatkov iz vira podatkov je skladno s prej nastavljenjo povezavo. V komponenti nastavimo, kateri stolpci iz vira podatkov se spustijo v podatkovni tok, ki poteka naprej in izvedemo preimenovanja atributov (primer na Sliki 21).
2. Izločimo neobstoječi poslovni enoti – samo optimalizacija dimenzije. Podatki ne bodo potrebni, ker sta enoti zaprti pred leti.
3. V komponenti izvedemo transformacijo zapisov števil. Zamenjamo tekstovni zapis, kjer so ločila med tisočicami vejice in pred decimalkami pika, tako da vejice odstranimo in piko zamenjamo z vejico, nato zapis preformatiramo v številčni zapis, v SSIS format (DT\_NUMERIC, 18,3).

4. Funkcija in namen komponente za počasi se spreminjajoče dimenzije sta podrobneje opisana v poglavju 2.4.6. V primeru te dimenzije nastopajo samo atributi, za katere spremljamo zgodovinske podatke, dodana sta datum začetka in datum konca veljavnosti podatkov.
5. Nastavimo premapiranje iz atributov v podatkovnem toku v ustrezne attribute tabele v podatkovnem skladišču (SQL Server). Izvede se vpis podatkov v prej definirano tabelo v SQL serverju. Definicija tabele je predhodno izvedena z jezikom T-SQL.

Slika 25: ETL postopek za eno od dimenzij



Po istem principu izvedemo ETL postopek za vse dimenzije.

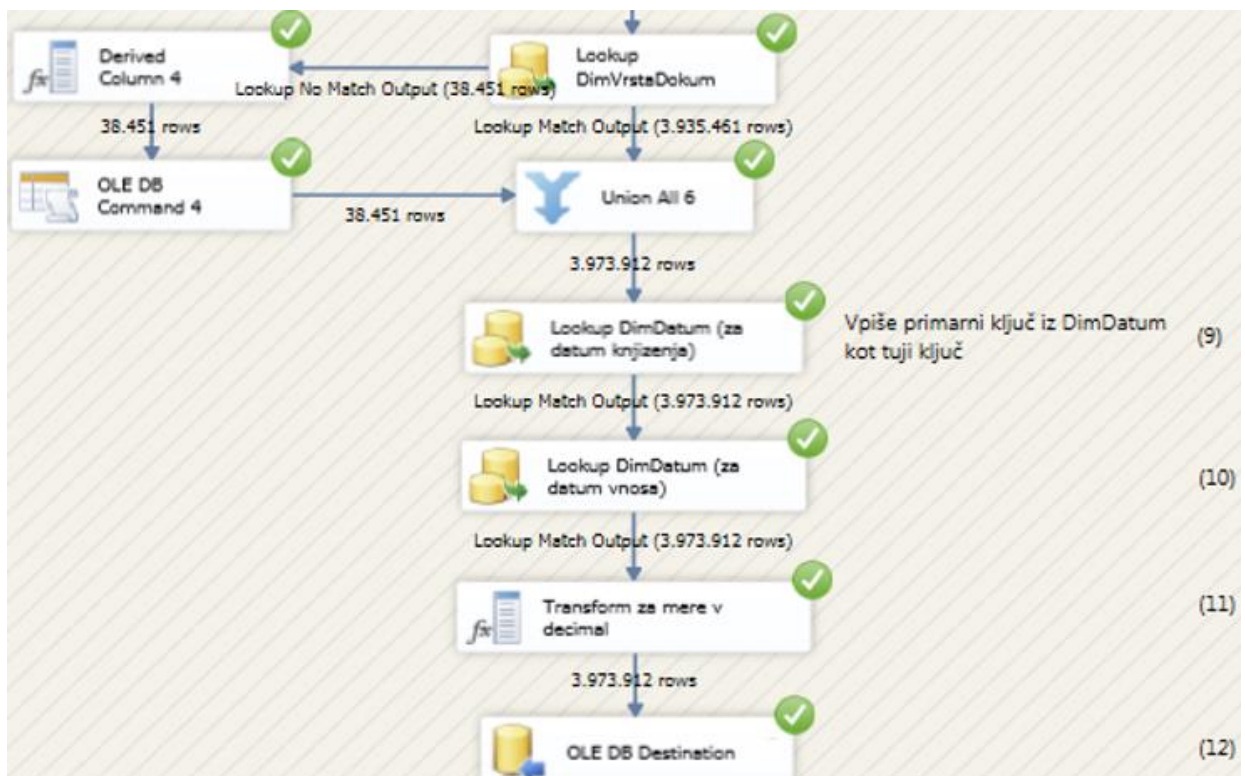
Ko so dimenzije zapisane v DW, sledi ETL postopek za vse tabele dejstev. Kimball in Ross (2002) predlagata naslednje faze: izvoz podatkov iz izvornega sistema, polnitev dimenzij s podatki, ločitev podatkov glede na granularcijo, transformacije, nadomestitev naravnih ključev z nadomestnimi, dodelitev nadomestnih ključev za manjkajoče dimenzijske podatke, zagotovite kvalitete, nalaganje podatkov, informiranje o uspešnem nalaganju ali o napaki. Pojasnimo posamezne komponente (številke ob komponentah na Sliki 26):



Slika 26: ETL postopek za tabelo dejstev – vsi materialni premiki.



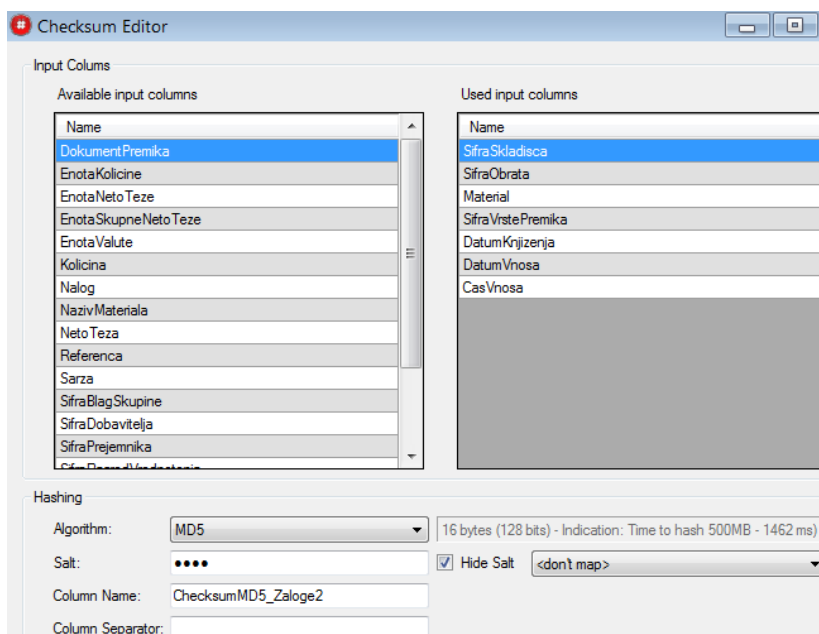
Zaradi velikosti je del slike izpuščen. Kombinacija komponent 5-8 se ponovi 7-krat in sicer za vsako dimenzijo



1. Branje podatkov iz vira podatkov je skladno s prej nastavljeno povezavo.
2. Izračun zgoščevalne funkcije. Namenjena je ugotovitvi, ali zapis v DW že obstaja, saj se podatke enega materialnega premika vnese samo enkrat. Podvojevanje je seveda

napaka. Izvede se tako imenovano nalaganje novih podatkov (angl. *incremental load*). To se lahko naredi s pregledom atributov, ki sestavljajo glavni ključ. Ta je običajno sestavljen ključ iz več atributov. Pregled bi bilo potrebno opraviti s komponento Lookup tolikokrat, kot je atributov, ki sestavljajo sestavljen ključ. To je zamudno, zato je Green (2010) predlagal uporabo zgoščevalne funkcije CRC-32 in objavil ustrezno SSIS komponentno *Checksum Transformation*. Komponenta izračuna iz izbranih atributov zgoščevalno funkcijo CRC-32, in sicer v celoštevilčnem formatu s predznakom z dolžino zapisa 4 byte. Obseg je tako od  $-2^{31}$  do  $+2^{31} - 1$  oz. je možnih 4.294.967.295 različnih zapisov. Če imamo v našem modelu 4.000.000 zapisov letno in zgoščevalno funkcijo računamo iz 7 atributov, je verjetnost kolizije ali iste vrednosti zgoščevalne funkcije pri 0,65% ( $7 \cdot 4.000.000 / 4.294.967.295$ ) primerov. To je enostavno nesprejemljivo. Zato za ta namen dokončno raje izberemo SSIS komponento *Checksum*, ki jo je predlagal Rossum (2012). Izberemo zgoščevalno funkcijo MD5 z dolžino zapisa 16 byte. S Slike 27 je razvidno, da se zgoščevalna funkcija izračuna iz 7 atributov, ki tvorijo sestavljen ključ.

Slika 27: Checksum – zgoščevalna funkcija po algoritmu MD5



3. Ker vrednost funkcije (angl. *checksum*) pri vsakokratnem zapisu zapišem v DW, je dovolj samo primerjava tega stolpca z izračunano vrednostjo zgoščevalne funkcije iz novih podatkov na vhodu. Tako zadostuje pregled in ugotavljanje ujemanja (angl. *lookup*) po samo enem stolpcu namesto po sedmih. Po podatkovnem toku nadaljujejo samo zapisi, ki se še ne nahajajo v DW.
4. Izvedemo transformacije: iz tekstovnih zapisov datumov in časa v datumske in časovne zapise in dodamo ali odvzamemo vodilne ničle – odvisno od vira. Ker je kakovost podatkov iz SAP dobra, ne potrebujemo popravkov. Tudi uvedba posebnih poslovnih pravil ni potrebna (Slika 28).

Komponente 5-8 se ponovijo za vsako dimenzijo posebej.

5. S komponento *Lookup* poiščemo, ali v dimenzijski tabeli obstaja zapis z ustreznim naravnim ključem. Zaradi učinkovitosti za časa pregleda, dimenzijsko tabelo držimo v pomnilniku računalnika. Ker pa je v našem primeru ta dimenzija večja - več kot 60.000 zapisov s po 42 atributi, v pomnilnik spravimo samo veljavne (datum poteka veljavnosti še ni postavljen) relevantne attribute naravnega ključa in nadomestni ključ. Če komponenta zapis v dimenzijski tabeli najde, v podatkovni tok prepisemo nadomestni ključ (Slika 29), ki ga shranimo kot tuji ključ v tabelo dejstev ob zapisu v DW.
6. Če zapisa z enakim naravnim ključem v dimenzijski tabeli ni, v podatkovni tok dodamo manjkajoči nadomestni ključ.
7. V dimenzijsko tabelo zapišem nov ključ manjkajoče dimenzije. V podatkovni tok zapišem nadomestni ključ manjkajoče dimenzije in iz tabele dejstev oba podatka o naravnem ključu, ki manjka – v konkretnem primeru *Material* in *SifraObrata*. Druge attribute dimenzijske tabele napolnimo z vrednostjo NULL. Pravo vrednost bodo dobili, če oz. ko bo ustrezen dimenzijski zapis na voljo (Slika 30).
8. Združimo podatkovni tok.
9. Prepisemo nadomestni ključ iz datumske dimenzije – za datum knjiženja.
10. Prepisemo nadomestni ključ iz datumske dimenzije – za datum vnosa.
11. Naredimo transformacijo vseh številčnih zapisov (mer) iz tekstovnega formata v številčni. Izpis iz SAP ima negativni predznak desno za številom in med tisočicami vejice in pred decimalnimi ciframi piko. Za vsak številčni zapis uporabim SSIS transformacijski zapis kot na Sliki 31.
12. Podatke zapišemo v DW. Pazimo pri premapiranju (Slika 32). Velikost DW v SLQ Serverju znaša kar 1,36 GB.

Slika 28: Transformacije zapisov

Derived Column Name	Derived Column	Expression	Data Type
DatumVnosa_DT_DATE	<add as new column>	DatumVnosa == "" ? NULL(DT_DBDATE) : (DT_DBDATE)(SUBSTR...	database date [DT_DBDATE]
DatumKnjizenja_DT_DATE	<add as new column>	DatumKnjizenja == "" ? NULL(DT_DBDATE) : (DT_DBDATE)(SUBS...	database date [DT_DBDATE]
Cas_DT_DBTIME	<add as new column>	(DT_DBTIME)CasVnosa	database time [DT_DBTIME]
PrejemnikZVodilNidami	<add as new column>	(DT_STR,10,1250)RIGHT("0000000000" + SifraPrejemnika,10)	string [DT_STR]
NaliPrOrgIzObrata	<add as new column>	(DT_STR,10,1250)(LEFT([SifraObrata ],2) + "00")	string [DT_STR]
SifraPrejemnika	Replace 'SifraPrejemnika'	REPLACE(LTRIM(REPLACE(SifraPrejemnika,"0","")),",","0")	string [DT_STR]
SifraDohavitelia	Replace 'SifraDohavitelia'	REPLACE(LTRIM(REPLACE(SifraDohavitelia,"0","")),",","0")	string [DT_STR]

Slika 29: Lookup komponenta za iskanje nadomestnih ključev dimenzijske tabele

Use results of an SQL query:

```
SELECT [MaterialID], [Material], [SifraObrata]FROM
[DimMatPodMateriala]WHERE [KonecVeljavnosti] IS NULL
```

Lookup Column	Lookup Operation	Output Alias
MaterialID	<add as new column>	TK_MaterialID

Slika 30: Generiranje manjkajočega dimenzijskega ključa

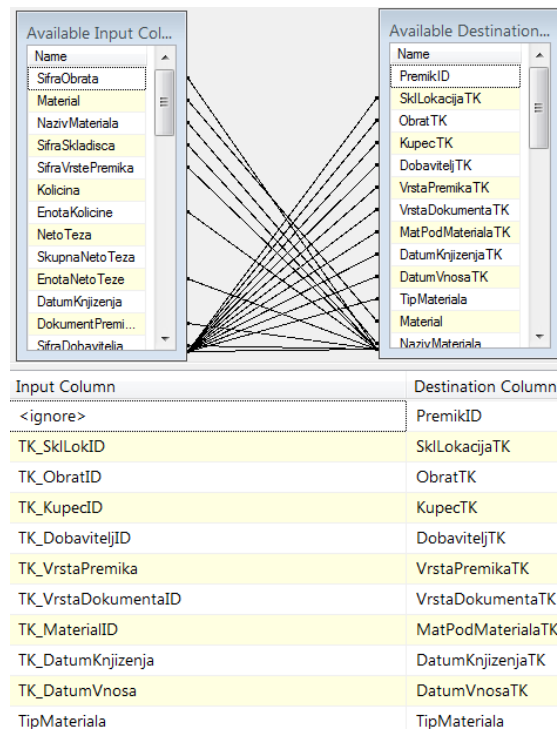
Name	<b>Generira Inferred TK DimMatPodMateriala</b>
PipelineVersion	0
UsesDispositions	True
ValidateExternalMetadata	True
Version	2
<b>Custom Properties</b>	
CommandTimeout	0
DefaultCodePage	1250
SqlCommand	EXEC GeneratePK_DimMatPodMateriala ?, ?, ? output

Input Column	Destination Column
Material	@Material
[SifraObrata ]	@SifraObrata
TK_Inferred_DimMatPodMateriala	@ReturnID

Slika 31: Transformacija števil

(DT\_NUMERIC,18,3)(RIGHT(Kolicina,1) == "-" ? "-" + REPLACE(REPLACE(SUBSTRING(Kolicina,1, LEN(Kolicina) - 1),".", ""),",", ".") : REPLACE(REPLACE(Kolicina,".", ""),",", "."))

Slika 32: Premapiranje podatkov iz podatkovnega toka pred zapisom v v DW



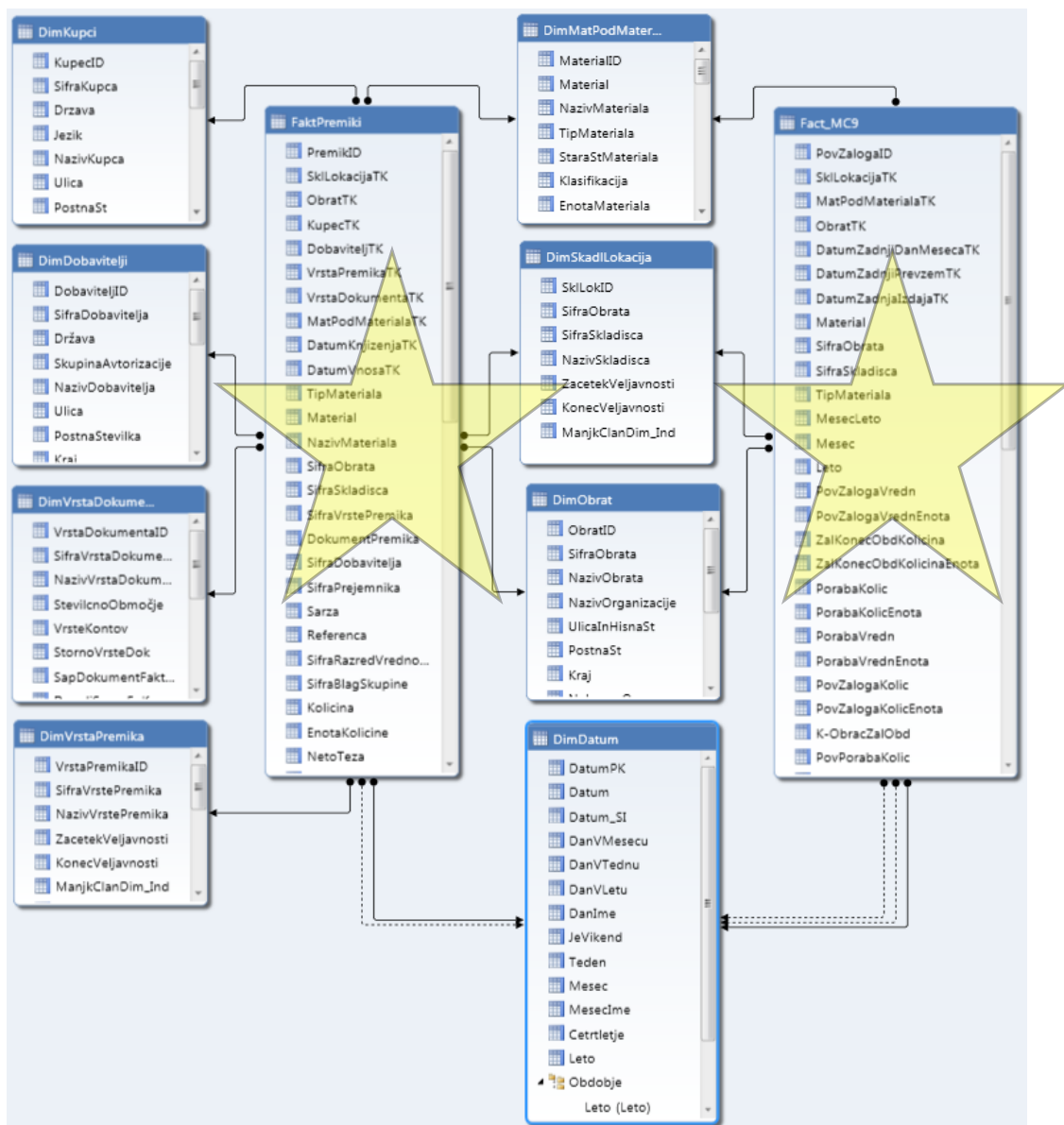
#### 4.2.6 Specifikacije za analitično aplikacijo in razvoj analitične aplikacije

V tem delu se osredotočamo na analitične potrebe. Ker je namen pokazati zaloge, obračanje zalog, ABC analizo in narediti primerjavo med rezultati različnih granulacij pri zajemu podatkov, mora biti prikaz informacij številčen in grafičen. Obenem se v tem delu tudi izdelajo standarde za menije in izgled. Kot odjemalec uporabimo Excel 2013 z možnostmi prikaza časovne vrste in z razčlenjevalniki za namen filtriranja. Številčni podatki se prikazujejo z vrtilnimi tabelami, grafični prikazi pa so lahko z linijski ali stolpčni diagrami. Krožni diagrami niso primerni zaradi več razlogov. Osnovni razlog je težko ločevanje velikosti pri podobnih deležih.

Potem ko je ETL postopek izveden in so podatki v področnem podatkovnem skladišču, programska oprema izbrana in specifikacije za aplikacijo načrtane, izvedemo razvoj analitične aplikacije. V SSAS Tabular iz DW naložimo vse tabele, tabele so različnih velikosti. Najmanjša dimenzija DimObrat ima samo 20 vrstic, največja DimMatPodMateriala pa 67.255. Dosti večji sta tabeli dejstev Fact\_MC9 ima 694.171 vrstic in FaktPremiki 3.973.912. Izvedemo še povezave med tabelami, saj se žal ne prenesejo samodejno. Povezave so narejene preko tujih ključev v tabelah dejstev. Nadomestni ključ v dimenzijskih tabelah je v tabele dejstev vstavljen kot tuji ključ. Dobimo galaksija shemo, ki jo tvorita dve zvezdni shemi. Imamo namreč dve tabeli dejstev (zvezdi), ki si delita nekaj skupnih dimenzij (Slika 33). Pri tem upoštevamo posebnost tabelaričnega modela, da je med dvema tabelama lahko istočasno aktivna samo ena povezava. Imamo samo eno datumsko dimenzijo DimDatum, ki je povezana s tabelama dejstev preko več povezav, saj imamo v

tabelah dejstev več datumskih atributov. Nismo se odločili za podvojevanje datumske tabele DimDatum, saj namen modela ni bil vezan na sočasne poizvedbe po več datumskih stolpcih tabel dejstev. Kot aktivno povezavo nastavimo tisto, ki se bo pri izračunih največ uporabljala. Za primer tabele dejstev FaktPremiki je to datum knjiženja materialnega premika. Je relevanten praktično pri vseh izračunih v nadaljevanju. Datum vnosa materialnega premika v ERP pa je pomemben, če bi hoteli npr. ugotavljati, kateri premiki se knjižijo z zamudo ali predčasno. Da bi uporabil neaktivno povezavo, ima tabelarični model nekaj posebnosti. Več o specifikih povezovanja tabel pri tabelaričnem modelu je navedeno v poglavju 2.5.2. Pomembno je, da v datumski tabeli atribut, ki vsebuje zvezno tekoče datume, označimo kot datumski (več v poglavju 2.5.3). To je pomembno zaradi uporabe tabelaričnih funkcij časovne inteligence. Kreiramo še datumsko hierarhijo z nivoji leto, četrtletje, mesec, datum. Poimenujemo jo Obdobje.

Slika 33: Galaksija shema dimenzijskega modela



Tabelarični model je praktično narejen in že lahko delamo poizvedbe. Kot odjemalec uporabim Excel. Lahko ga zaženem kar iz SSAS ali pa izven s tem, da ga moram povezati na tabelarični model v SSAS. Poizvedbe se delajo preko funkcionalnosti vrtilnih tabel. Seveda po merah, ki so vsebovane v tabelah dejstev. V mojem primeru tabele dejstev vsebujejo tudi dimenzijske podatke. Gre za degenerirane dimenzije, ki niso tuji ključi in niso povezava do pravih dimenzij (Kimball & Caserta, 2004). Tak primer atribut NazivMateriala v tabeli FaktPremiki. Povezava do dimenzijske tabele DimMatPodMateriala je sestavljen ključ iz atributov Material in SifraObrata in v ETL postopku je v tabelo dejstev FaktPremiki vpisan pripadajoči nadomestni ključ iz dimenzijske tabele kot tuji ključ. Gledano s stališča zmanjšanja modela in optimizacije, bi bilo smiselno v DW in tabelarični model zapisati tabelo dejstev brez atributov: Material, NazivMateriala. Niso potrebni, ker smo v ETL postopku v tabelo dejstev vpisali kot tuji ključ nadomestni ključ iz dimenzijske tabele. Povezava je ustrezna in mnogo : 1 (tabela dejstev : dimenzijska tabela). Za namen prototipa smo pustili tabele, kot so na viru podatkov. Namen so performančna in druga testiranja kasneje, saj so pri tabelaričnem modelu vse tabele enakovredne. Tabelarični model namreč ne loči tabel dimenzij in tabel dejstev. Važne so samo povezave.

Model je pripravljen in agregacije so možne. Ker pa potrebujemo še kaj več kot zgolj seštevanje po obstoječih stolpcih, je potrebno z DAX izrazi dodati nove mere. Tu se šele pokaže parava moč tabelaričnega modela. Naprej gre postopek po dveh poteh, saj želimo primerjati dve vrsti granulacije na osnovi mesečnih zajemov podatkov in na osnovi vseh premikov, iz katerih izpeljemo dnevna stanja zalog.

#### 4.2.6.1 Analize na osnovi mesečne granulacije podatkov

Vir je izpis SAP transakcije MC.9. Ta je pri osnovni nastavitvi SAP v Skupini TPV nastavljena (angl. *customized*) na granulacijo enega meseca. Znotraj tekočega meseca se ažurira vsak dan. Vrednosti statističnih atributov se praviloma izračunajo po enačbi 15.

$$\text{Povprečna vrednost} = \frac{\text{začetna vrednost} + n \text{ vrednost na koncu meseca}}{n + 1} \quad (15)$$

Ker se zaloga spreminja dnevno, lahko enačba generira napako 0-100%. Predpostavimo, da je zaloga v začetku in na koncu meseca enaka 0. Po enačbi 10 se izračuna povprečna zaloga 0. Če je bila vmes zaloga vsaj en dan v mesecu različna od nič, je napaka enačbe že 100%. Če je dan dobave materiala ravno na prvega in zadnjega v mesecu, je povprečna zaloga prevelika, in če smo prvega in zadnjega v mesecu ravno pričakovali dobavo, je povprečna zaloga prenizka. Praksa je pokazala, da pri serijski proizvodnji, pri periodičnih dobavah in v daljšem časovnem obdobju lahko dobimo še zadovoljive rezultate. Tudi če so rezultati absolutno napačni, so še vedno primerljivi med sabo in je možno videti vsaj trend. Pri velikih materialih se skupno povprečje tudi nekoliko poravnava, ker je pri nekaterih materialih napaka v plus in drugje v minus. Vendar na drugi strani boljše SAP transakcije ne poznamo. Možno

je dobiti izpis povprečne zaloge s SAP transakcijo MC.49, vendar ta za izvajanje zahteva ogromno resursov in že pri par materialih preseže dovoljen čas izvajanja, ki je pri nastavitvi SAP v Skupini TPV 10 minut.

V prototipnem projektu imamo rezultat SAP transakcije MC.9 zapisane v DW v tabeli dejstev *Fact\_MC9*. Izpisano je 30 bistvenih atributov. Analiza podatkov in konteksti pogledov so narejeni v odjemalcu Excel, ki ga povežemo na SSAS Tabular, v katerem predhodno dodamo še nekaj izračunanih mer s pomočjo jezika DAX:

- mero za izračun koeficienta obračanja zalog  
KoefObrZal:=IFERROR( calculate(sum(Fact\_MC9[PorabaVredn])/sum(Fact\_MC9[PovZalogaVredn])); blank()),
- mero za izračun obračanja zalog v dnevih  
DniObrZal:=if([KoefObrZal]>0;MAX(Fact\_MC9[DniVMesecu])/[KoefObrZal]; blank())

Večji problem je pri ABC analizi. Po pregledu literature in po preverjanju rešitev ugotovljamo, da so rešitve, vezane na izračune v dodatnih izračunanih stolpcih, kjer je v DAX izrazih uporabljena funkcija EARLIER (Russo in Ferrari, 2011), delovale na manjšem testnem številu podatkov (ca 1000), ko pa smo hoteli v SSAS tabular naložiti tabelo *Fact\_MC9*, ki ima skoraj 700.000 vrstic, to pri 8 GB RAM v računalniku ni uspelo oz. je jedro xVelocity javilo pomanjkanje RAM. Izračunani stolpci se namreč izračunajo v času nalaganja podatkov v model (angl. *processing time*). Torej ta rešitev ni bila dobra. Tako smo ugotovili nedelujočo pot. Tudi to je del raziskovalnega procesa.

Prava rešitev je dokaj težko razumljiva dinamična rešitev z izračunom skupin ABC v času izvedbe povpraševanja (angl. *query time*). Sledili smo predlogu, ki ga je v osnovi pripravil Brueckl (2013). Dve leti za prvo rešitvijo je objavil optimalnejšo in hitrejšo rešitev. Opis rešitve je daljši in bralcu priporočamo pregled zapisa Brueckl (2015). Ustrezna koda v DAX je:

```
PovpZalogaVrednABC:=CALCULATE (
    [PovpZalogaVrednS];
    VALUES (DimMatPodMateriala[Material] );
    FILTER (
        CALCULATETABLE (
            ADDCOLUMNS (
                ADDCOLUMNS (
                    VALUES (DimMatPodMateriala[Material] );
                    "ZunanjaVrednost"; [PovpZalogaVrednS] );
                "KumuliranaVrednProcent"; DIVIDE (
                    SUMX (
```



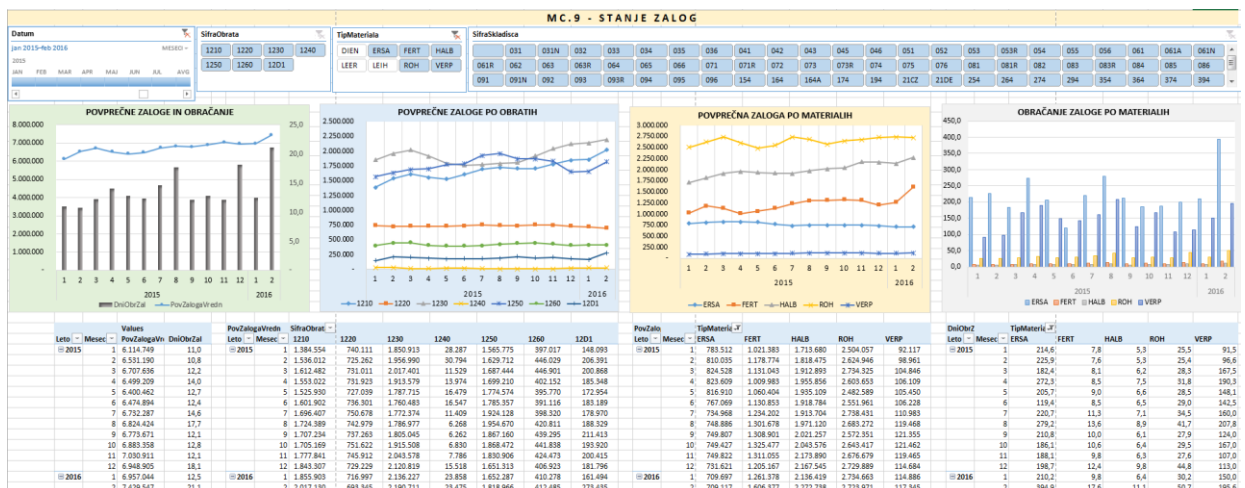
```

FILTER (
    ADDCOLUMNS (
        VALUES (DimMatPodMateriala[Material] );
        "NotranjaVrednost"; [PovpZalogaVrednS] );
    [NotranjaVrednost] >= [ZunanjaVrednost]
);
[NotranjaVrednost]
);
CALCULATE (
    [PovpZalogaVrednS];
    VALUES (DimMatPodMateriala[Material] )
)
)
);
ALL ( 'DimMatPodMateriala' )
);
[KumuliranaVrednProcent] >+[MinABCMeja]
&& [KumuliranaVrednProcent] <=[MaxABCMeja]
))

```

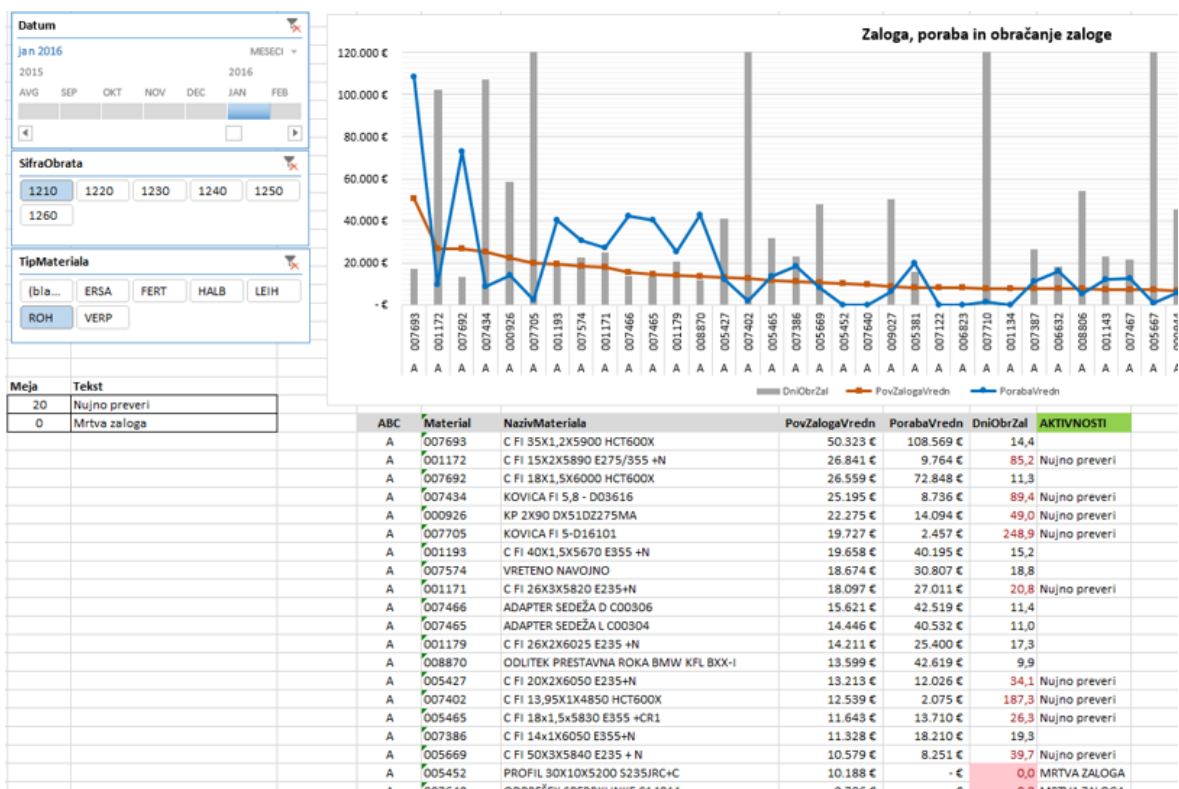
Na osnovi tako postavljenega modela prikažemo potrebne informacije za analizo in upravljanje z zalogami. Naredimo nekaj primerov. Na Sliki 34 je prikazana osnovna interaktivna nadzorna plošča za vse poslovne enote Skupine TPV. Prikazuje: skupne povprečne zaloge in obračanje zalog v dnevih, povprečne zaloge po poslovnih enotah, povprečne zaloge po tipih materialov in obračanje zalog po materialih. Preko razčlenjevalnikov lahko vrtamo do prikaza stanja v posameznem skladišču posamezne poslovne enote. Če bi dodali še cilje, bi prišli do celovite nadzorne plošče za prikaz kazalnikov uspešnosti poslovanja.

Slika 34: Osnovna nadzorna plošča



Za konkretno analizo po materialih in za sprejemanje korektivnih ukrepov pa je primernejša nadzorna plošča na Sliki 35, na kateri prikazujemo porabo, povprečno zalogo in dneve obračanja zaloge posameznega materiala. Na sliki je izbrana poslovna enota, ki se v Skupini TPV ukvarja z razrezom in preoblikovanjem cevi in žic. Izbran je prikaz ROH materialov (v SAP so pod to oznako vodeni vhodni materiali). Že bežen pogled pokaže, s kako lahko identificiramo morebitne materiale, na katere moramo biti pozorni. Na grafu zelo hitro identificiramo, s katerimi materiali se je potrebno ukvarjati najprej. Potrebno je iti od leve (največje povprečne zaloge, A materiali) proti desni in gledati tiste materiale, ki imajo počasno obračanje zalog (sivi stolpec v grafu) in nizko porabo (plava krivulja). Smiselno in pravilno je začeti pri materialih klasifikacije A po ABC analizi.

Slika 35: ABC analiza ROH materialov na izbrani poslovni enoti



#### 4.2.6.2 Analize na osnovi dnevne granulacije podatkov

Vir podatkov je izpis »Z« transakcije ZALOGES. Gre za SAP poizvedbo, s katero lahko izpišemo vse materialne premike. V SAP je uporabljena logika, da se popravki knjižijo kot nasprotna knjižba, pri kateri sta količina in vrednost negativni. Vsak tip premika ima pripadajoči premik za popravek ali negativno knjižbo. Kot primer tip premika 601 je »GD izdaja blaga:dob« in 602 je »VR Stor.dobave Blaga«. Na tej logiki tudi gradimo matematiko izračuna stanja v skladiščih. Dnevno stanje zalog, mera [DnevnaZalKol], je vsota stanja zaloge, pridobljena in vseh materialnih premikov in začetnega stanja:

```

DnevnaZalKol :=
CALCULATE (
    SUM ( FaktPremiki[Kolicina] );
    FILTER (
        ALL ( 'DimDatum' );
        DimDatum[Datum] <= MAX ( DimDatum[Datum] )
        && DimDatum[Datum] >= [DatumZac]
    )
)
+ CALCULATE (
    SUM ( Fact_MC9[ZalKonecObdKolic] );
    FILTER (
        ALL ( DimDatum[Datum] );
        DimDatum[Datum] >= EDATE ( FaktPremiki[DatumZac]; -1 )
        && DimDatum[Datum] < FaktPremiki[DatumZac]
    )
))

```

DAX izraz je dinamičen in od konteksta pogleda zavisi, ali računamo dnevno stanje zalog cele Skupine TPV, posameznega obrata, posameznega skladišča. Prva funkcija CALCULATE je vsota vseh premikov. V funkciji SUM seštejemo premike, in sicer tiste, ki jih določi funkcija FILTER. Najprej izničimo kontekst pogleda s funkcijo ALL tako, da zajamemo vse datume, potem damo v obseg za računanje tiste premike, ki imajo datum premika manjši od datuma, nastavljenega v kontekstu pogleda (funkcija MAX). V odjemalcu Excel je to nastavljen datum v časovni vrsti ali v razčlenjevalniku ali kar v vrtilni tabeli (filter). Druga funkcija CALCULATE je začetna količina, ki jo pridobimo kar iz tabele dejstev Fact\_MC9 z DAX izrazom. Gre za količino konec meseca pred začetnim datumom premikov v tabeli dejstev FaktPremiki. Bolj nazorno lahko to opišemo s primerom. Če želimo prikazati dnevne zaloge npr. za januar 2016, v časovni vrsti nastavimo mesec januar 2016. Ker zaloge izračunavamo iz začetnega stanja, ki mu prištevamo prejeme in odštevamo izdaje, moramo začetnemu stanju prišteti vse premike za vse dneve. Ko časovna vrsta pride do prvega datuma, ki je nastavljen v kontekstu pogleda, to zalogo izpiše, potem se doda premike tega dne in izpiše naslednji datum in tako do zadnjega datuma v kontekstu pogleda. Dnevne zaloge tako imamo in moramo izračunati še povprečje. Tukaj je nujno iterirati po vseh datumih. Zato uporaba funkcije AVERAGEX namesto funkcije AVERAGE, saj moramo, da bi dobili pravo povprečje, sešteti vrednosti po vseh datumih časovnega obsega, za katerega računamo povprečje. Meseci namreč nimajo enakega števila dni. V jeziku DAX vse funkcije, ki imajo na koncu X, zahtevajo dva parametra: tabelo in izraz, ki se izvede nad vrstico te tabele. Pri tem je tabela lahko rezultat neke funkcije. Da bi dosegli pravo povprečje, uporabimo funkcijo VALUES, ki vrne vse datume – vsak dan samo 1x in posledično se za vsak dan 1x izračuna dnevna zaloga. Pravilen izraz DAX je:

PovpZalKol:=AVERAGEX ( VALUES ( DimDatum[Datum] ); [DnevnaZalKol] )

Analitika na osnovi materialnih premikov močno razširi možnosti analiz in poda točna stanja povprečnih zalog in posledično obrata zalog. Koliko zgrešimo s SAP transakcijo, ki uporablja mesečne agregacije, kaže Slika 36. Na levi strani slike vidimo izračun povprečne zaloge in obračanja zaloge A materialov na osnovi dnevne posnetke stanja, ki ga izračunano iz vseh materialnih premikov in na desni strani slike vidimo podatke o povprečni zalogi iz SAP transakcije MC.9 ter izračun obračanja zaloge. Na sredini slike je pojasnilo zakaj zakaj je povprečna zaloga izračunana napačno pri mesečni granulaciji podatkov.

Pregled obkroženih podatkov pokaže:

- Pri materialih 007514, 007534, 007535, 009127, 009128, 009140 in 009141 SAP transakcija poda od 65 % 135 % prevelik podatek o povprečni zalogi. Za toliko je prevelik potem tudi izračun obračanja zaloge v dnevih. Pri materialu 007514, ki je po ABC analizi porabe drugi po porabi, bi pri izračunu obrata zaloge 12,97 dni šli morda v agresivno zmanjševanje zaloge ali spremembo nabavne metode. To bi bila napaka, saj je dejanski obrat zaloge 5,53 dni. Lahko bi povzročili podzaloženost.
- Pri materialu 007601 SAP transakcija poda za 62 % premajhen izračun obračanja zaloge,
- Pri materialu 001112 SAP transakcija poda celo nično povprečno zalogo, napačno bi lahko sklepali, da gre za mrtvo zalogo. Pravo obračanje zaloge je 3,92 dneva in je potrebno preveriti frekvenco dobav in se vprašati ali nimamo morda premalo zaloge.

Slika 36: Povprečna zaloga in obračanje zaloge pri različnih granulacijah podatkov

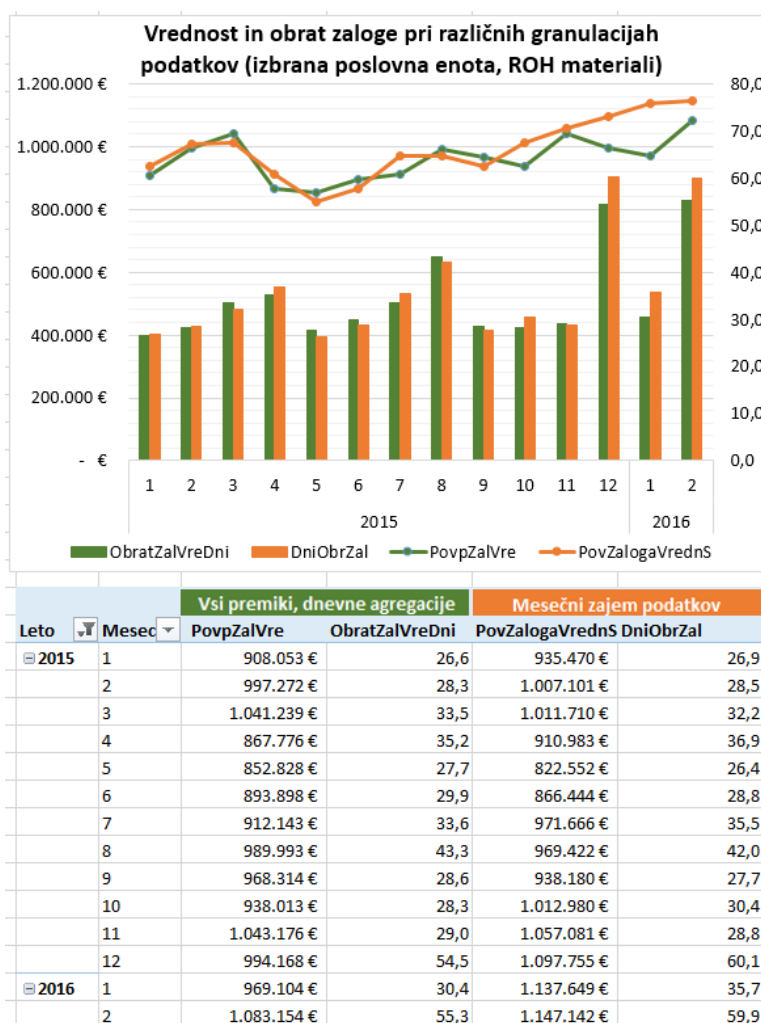
Granulacija: vsi premiki, dnevne agregacije				Granulacija: mesečni zajem podatkov (SAP MC.9)				
Material	NazivMater	PovpZalVre	PorabaVre	ObratZalVre	PovZalogaVrednS	PorabaVrednS	DniObrZal	NAPAKA:
008661	K 1,5X370 HX2	12.405 €	25.328 €	15,18	12.664 €	25.328 €	15,50	2%
007514	K 2,5X280 S42	4.373 €	24.503 €	5,53	10.255 €	24.503 €	12,97	135%
008715	PUŠA NORTON	17.319 €	24.078 €	22,30	15.802 €	24.078 €	20,34	-9%
007597	K 4X313 S500M	16.300 €	23.650 €	21,37	18.008 €	23.650 €	23,61	10%
007598	K 3X259 S500M	6.948 €	23.518 €	9,16	7.135 €	23.518 €	9,41	3%
008663	K 1,1x590 XE3	3.471 €	19.330 €	5,57	5.403 €	19.330 €	8,67	56%
009381	PLF 1,25X100C	13.524 €	19.275 €	21,75	11.394 €	19.275 €	18,32	-16%
007601	K 3X276 S500M	3.972 €	17.073 €	7,21	1.505 €	17.073 €	2,73	-62%
001097	K 4X214 S550M	3.072 €	15.942 €	5,97	3.126 €	15.942 €	6,08	2%
009140	KOTNIK POVEZ	2.770 €	15.851 €	5,42	5.404 €	15.851 €	10,57	95%
009141	KOTNIK POVEZ	2.760 €	15.851 €	5,40	5.403 €	15.851 €	10,57	96%
007336	K 2,5X209 S50	23.039 €	15.713 €	45,45	21.401 €	15.713 €	42,22	-7%
001112	KP 2X430 HX4	1.979 €	15.648 €	3,92	- €	15.648 €	-	-100%
007535	K 1,4X624 HC3	4.248 €	14.053 €	9,37	7.027 €	14.053 €	15,50	65%
009127	NASLON VZME	5.226 €	13.553 €	11,95	8.911 €	13.553 €	20,38	71%
009128	NASLON VZME	4.976 €	13.494 €	11,43	8.631 €	13.494 €	19,83	73%
008668	K 2X368 S355M	3.774 €	13.429 €	8,71	4.949 €	13.429 €	11,42	31%
009539	Zapirni elemen	6.469 €	13.333 €	15,04	7.303 €	13.333 €	16,98	13%
007534	K 2X368 S355M	4.334 €	13.291 €	10,11	7.386 €	13.291 €	17,23	70%
009538	ZAPIRNI ELEME	4.907 €	13.114 €	11,60	6.710 €	13.114 €	15,86	37%
007691	K 2,5X228 HX3	11.253 €	13.023 €	26,79	11.949 €	13.023 €	28,44	6%
001089	KP 2,5X430 HX	6.179 €	12.924 €	14,82	6.142 €	12.924 €	14,73	-1%
001096	K 4X210 S550M	6.845 €	12.899 €	16,45	8.126 €	12.899 €	19,53	19%

Ugotovili smo, da je pri mesečni zajem podatkov ne zagotavlja pravilnega podatka o povprečni zalogi. Posledično je tudi izračun obračanja zaloge napačen. Na ravni posameznega materiala je ta napaka lahko tudi zelo velika.

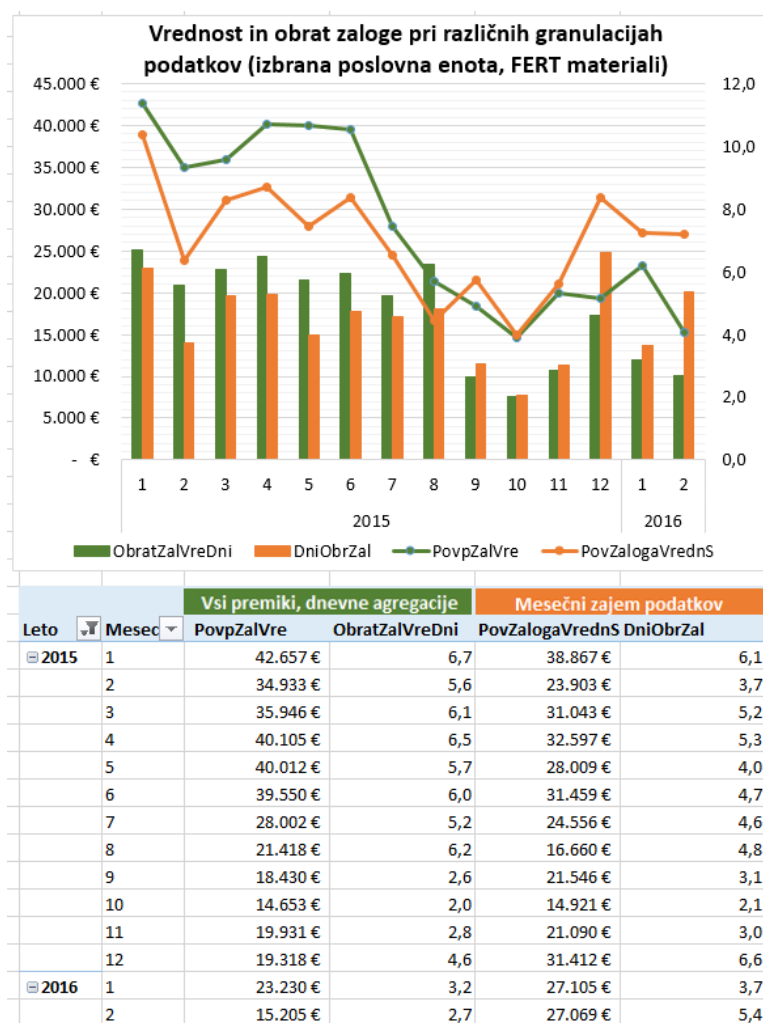
Vendar, če je vpliv granulacije podatkov na mikro ravni - posamezen material, lahko zelo velik, se to pri velikem številu materialov (tudi več 1000) v posamezni poslovni enoti spremeni. Slika 37 kaže povprečje zalog in dneve obračanja zalog na eni od poslovnih enot za ROH materiale in Slika 38 za FERT materiale (v SAP so pod to oznako vodeni materiali, ki se prodajajo), je razvidno, da je pravo povprečje zalog (zelena krivulja) lahko v nekem obdobju zelo blizu (odstopanje do 10 %) povprečju zalog iz mesečnega zajema podatkov (oranžna krivulja). Vendar lahko tudi odstopa v plus ali minus.

Naveden je prepričljiv dokaz, da je pri analizi zalog proizvodnega podjetja potrebno imeti pravi vir podatkov in pravo granulacijo podatkov. Drugače informacije sicer dobimo, a so lahko napačne. Rezultat BI analiz je treba vedno preveriti. Trditev, da tisto, kar znamo izračunati »na roke«, z lahkoto implementiramo tudi v BI, drži. Seveda obstajajo področja BI, kjer to ne velja absolutno, saj gre lahko tudi za zelo kompleksna izračunavanja.

Slika 37: ROH materiali – razlike zaradi granulacije zajema podatkov



Slika 38: FERT materiali – razlike zaradi granulacije zajema podatkov



#### 4.2.7 SWOT analiza prototipa

Za izdelani prototip lahko naredimo SWOT (*angl. strengths, weaknesses, opportunities, threats*) analizo s katero ugotovimo prednosti in slabosti, ki izhajajo iz notranjega okolja ter priložnosti in nevarnosti iz zunanjega okolja. Barone et al (2010) ugotavljajo, da do konkurenčne prednosti pridemo takrat, ko se začno ujemati prednosti in priložnosti. Analiza narejena na našem prototipu pokaže, da je prednosti in priložnosti več od slabosti in nevarnosti. Jezik DAX je omenjen med prednostmi in priložnostmi, saj obstaja priložnost uporabe samopostrežne BI v določenih primerih. Navedene rešitve z jezikom DAX so enake.

SWOT analiza prototipa:

- PREDNOSTI
  - s pridobljenimi informacijami je mogoče boljše odločati pri upravljanju z zalogami,

- analize na ravni materiala omogočajo optimizacijo zalog in izbiro optimalne nabavne metode. Posledično možno povečanje dobičkonosnosti lastniškega kapitala,
- manj poizvedb in s tem manjša obremenitev OLTP sistema. Vsi relevantni podatki so preneseni v DW,
- integriranost podatkov,
- hitrost dostopa do podatkov dosežena z uporabo tabelaričnega modela (cel je v pomnilniku računalnika) in z uporabo jezika DAX (za izračunane nove mere),
- možna primerjava med časovnimi obdobji,
- zanesljivost in enostavnost uporabe.
- **SLABOSTI**
  - stroški izdelave rešitve,
  - nezadostna kvaliteta podatkov. Težave pri ETL postopku zaradi nepoznavanja podatkovnega modela SAP,
  - administratorska in vzdrževalna dela.
- **PRILOŽNOSTI**
  - uporaba rešitev na drugih področjih in pri drugih podjetjih,
  - pri manj zahtevnih primerih in pri kvalitetnih podatkih je možno tabelarični model in analitične rešitve uporabiti tudi brez podatkovnega skladišča – uporaba samopostrežne BI (orodja PowerPivot in Power BI).
- **NEVARNOSTI**
  - licenčni stroški za programsko opremo,
  - slabo sodelovanje med člani projekta v primeru postavitve sistema BI,
  - pomanjkanje izkušenj in znanja.

## **SKLEP**

Danes smo priča izjemno hitremu naraščanju podatkov. Pri IDC (2014) predvidevajo, da bomo leta 2020 na svetu imeli 44 ZB podatkov. Podatek za leto 2013 je bil 4,4 ZB. Rast je tako 40 % letno. Veliko podatkov producirajo organizacije same z beleženjem vedno več količin. Tudi število senzorjev na strojih in napravah se povečuje. Odločanja v podjetjih na pamet niso več možna, saj visoko konkurenčno okolje napak ne tolerira. Prav tako ni možno na pamet obvladovati velikega števila podatkov. Tu se izkaže poslovna inteligenca, ki omogoča pridobiti informacije iz podatkov. Vendar samo po sebi tudi to ni dovolj, saj je vrednost informacij lahko tudi nič, če jih ne uporabimo ali ne uporabimo prav. Analiziranje podatkov je smiselno le, če za to obstajajo poslovne potrebe. In te v primeru zalog obstajajo, saj velikost zalog vpliva na mero dobičkonosnosti. Osnovi pristop k analizi zalog je analiza ABC, ki se lahko nadgradi še z analizo XYZ. Pri analizi je pomemben tudi sistematičen pristop, kot je opisan v poglavju 3.3. Samo po sebi se postavlja vprašanje, ali praktično potrebujemo veliko raznih analiz. V razmislek je zapis iz časopisa Delo (1978), kjer so citirali članek iz dnevnika La Monde, v katerem je Michel Croizier opisal prispevek nobelovca Herberta Simona: »Ljudje, odgovorni za gospodarstvo, živijo v zmotnem

prepričanju, da je za sprejemanje dobre odločitve dovolj, če kar najbolj izkoristijo podatke, ki jih imajo na voljo. Simon je dokazal, da je človek za tak postopek nesposoben in da se v resnici zadovolji s prvo rešitvijo, ki se mu zdi zadovoljljiva«. Torej veliko informacij iz podatkov ni dovolj, treba je pripraviti prave informacije in jih prav izkoristiti. Ali je bilo prav, pa lahko pove samo razvoj dogodkov, ki sledijo potem. Na kratko povedano, manj je lahko tudi več. Na področju BI moramo težiti k temu, da pripravimo prave informacije, ne pa k temu, da pripravimo goro informacij. In dejanski kriteriji pri tem so poslovne potrebe, ki so pa lahko manjše od želja poslovnih uporabnikov, ki bi radi še to in ono pa v resnici tega sploh ne potrebujejo. Pomembno je pripraviti informacije, ki jih potrebujemo pri odločanju o aktivnostih ali pri sprejemanju odločitev.

V magistrski nalogi je v analitično – prezentacijski plasti uporabljen nov tabelarični pristop, ki ga je predstavil Microsoft s SLQ Server 2012 in v omejenem obsegu s PowerPivot leta 2010. Kljub že dolgo znanim principom je šele razvoj računalnikov omogočil izvajanje analiz velikega števila podatkov v pomnilniku računalnika. Uvodoma postavljeno trditev je potrdil razvoj prototipa, ki je bil narejen na realnih podatkih in na verodostojno veliko podatkih. Pri sestavi področnega podatkovnega skladišča je bilo čutiti pomanjkanje standarda modeliranja. Vendar so bila priporočila, ki sta jih pripravila Kimball in Ross (2002), zelo dobro izhodišče in vodilo. Vsako programsko orodje pa ima seveda nekaj svojih posebnosti, ki jih je potrebno upoštevati.

Razviti prototip za analizo zalog je že v tem ciklu prototipiranja pokazal:

- da je tabelarični model lahko alternativa OLAP kockam, ki pri analizi podatkov kraljujejo že 20 let. Gradnja tabelaričnega modela je enostavna in krivulja učenja jezika DAX je ugodnejša od učenja jezika MDX. To velja na žalost samo v začetni fazi. Pri kompleksnejših izračunih pa nič več. Večina takih izračunov niti ne potrebuje. Oteževalna okoliščina je, da je potrebno spremeniti razmišljanje v stolpično, saj so argumenti funkcij po večini stolpci, in ne vrstice tabel.
- da je tabelarični model hiter pri analiziranju podatkov. Količina in hitrost RAM sta poleg hitrosti procesorja najpomembnejša faktorja.
- da je pri analizi zalog pravilna granulacija podatkov odločilna oz. zelo pomembna. Na primeru celo ugotovimo, da je SAP transakcija za analizo zalog MC.9 samo pogojno uporabna. Na ravni materiala lahko generira veliko napako, tudi preko 100 %.

Potrebno je tudi opozoriti na omejitve naloge, saj je rešitev pripravljena glede na trenutne okoliščine obravnavanega podjetja in zato morda vključuje nekatere specifične, ki niso neposredno uporabljive na drugih primerih. Metode, principi in DAX funkcije pa so prenosljive tudi na druge primere.

Ker je prototipiranje iterativni postopek, bi bilo v nadaljevanju raziskovanja primerno narediti primerjavo z OLAP modelom analize zalog, kot sta ga opisala Dular in Godnov



(2013). Podatkovno skladišče lahko uporabimo enako in z istimi podatki. Prototip se lahko še izboljša in razvije naprej do končnega izdelka. Podjetje tako pridobi verodostojne KPI, ki so podlaga za aktivnosti zmanjševanja in optimizacije zalog ter podlaga za pravilno izbiro nabavnih metod.

Iz pridobljenih izkušenj lahko tabelarični model samo priporočimo zaradi enostavnosti in hitrosti. Tudi modeliranje je manj zahtevno, saj so vse tabele enakovredne. Kljub temu je poznavanje večdimenzijskega modeliranja zelo velika prednost. Aktualnost tabelaričnega modela povečuje dejstvo, da moramo v današnjem okolju, ki se spreminja praktično nenehno, dostikrat hitro priti do informacij na osnovi velike količine podatkov. Sploh, če gre za nove vire podatkov, dostikrat ne moremo čakati oddelka informatike, da k obstoječim večdimenzijskim modelom doda nove mere ali dimenzije in sestavi OLAP kocke. V takih primerih smo lahko s tabelaričnim modelom učinkovitejši. Ker pa je tabelarični model preko PowerPivot že vgrajen v Excel 2013, je praktično na voljo širšemu krogu uporabnikov. Za večjo razširjenost poskrbijo še Power BI zastonska orodja. Tako je trditev, da smo s tabelaričnim modelom že v obdobju BI za vse (angl. *BI for everyone*), čisto na mestu. Pričujoča analiza zalog pokaže, da je tabelarični model z jezikom DAX primeren za reševanje tudi bolj kompleksnih zadev z veliko podatki.

## LITERATURA IN VIRI

1. Alexander, M., Decker, J., & Wehbe, B. (2014). *Business Intelligence Tools for Excel Analysts*. New York, Hoboken: John Wiley & Sons , Inc.
2. Barone, D., Yu, E., Won, J., Jiang, L., & Mylopoulos, J. (2010, November). Enterprise modeling for business intelligence. *In IFIP Working Conference on The Practice of Enterprise Modeling*, 31-45. Springer Berlin Heidelberg.
3. Bodie, Z., Kane, A., & Marcus, A. (2008). *Essentials of investments* (7<sup>th</sup> ed.). New York; McGraw-Hill Companies, Inc.
4. Brueckl, G. (2013, 28. januar). *Dynamic ABC Analysis in PowerPivot using DAX*. Najdeno 5. februarja 2016 na spletnem naslovu <http://blog.gbrueckl.at/2013/01/dynamic-abc-analysis-in-powerpivot-using-dax/>
5. Brueckl, G. (2015, 19. januar). *ABC Classification – Dynamic*. Najdeno 5. februarja 2016 na spletnem naslovu <http://www.daxpatterns.com/abc-classification-dynamic/>
6. Bulinski, J., Waszkiewicz, C., & Buraczewski, P. (2013). Utilization of ABC/XYZ analysis in stock planning in the enterprise. *Agricultural and Forest Engineering*, (61), Warsaw university of life sciences press. 1 – 115.
7. Cox, R. (2013). *25 Ways to Lower Supply chain Inventory costs*. Najdeno 19. maja 2014 na spletnem naslovu <http://www.tompkinsinc.com/article/2004/25-ways-lower-inventory-costs/>
8. Čižman, A. (2002). *Logistični management v organizaciji*. Kranj: Moderna organizacija v okviru FOV.
9. De Jonge, K. (2015, 28. oktober). *What's new for SQL Server 2016 Analysis Services in CTP3*. Najdeno 21. marca 2016 na spletnem naslovu <https://blogs.msdn.microsoft.com/analysisservices/2016/02/03/whats-new-for-sql-server-2016-analysis-services-in-ctp3-3/>
10. Dickerman, H., & Myers, P. (2011). *Data Analysis Expressions (DAX) in the Tabular BI Semantic Model*. Najdeno 9. decembra 2013 na spletnem naslovu <http://www.microsoft.com/en-us/download/details.aspx?id=28572>.
11. Duhovnik, M. (2003). *Obračunski in denarni tokovi kot podlaga za ugotavljanje uspešnosti*. Ljubljana: Zveza računovodij, finančnikov in revizorjev Slovenije.
12. Dular, T., & Godnov, U. (2013) Analiza zalog. *Zbornik 20. konferenca Dnevi slovenske informatike, 15.-17. april 2013* [zgoščenka]. Portorož: Društvo slovenskih informatikov
13. Estabrook, G. F., & Brill, R. C. (1969), *The Theory of the TAXIR Accessioner*. Mathematical Biosciences (5), 327-34, American Elsevier Publishing Company, Inc.
14. Faulkner, A. (2016, 12. januar). *SSAS Multidimensional vs. SSAS Tabular Which one do I choose*. Najdeno 10. marca 2016 na spletnem naslovu <http://pragmaticworks.com/Training/Details/SSAS-Multidimensional-v-SSAS-Tabular>

15. Gartner. (2014). *Predicts 2014: Business Intelligence and Analytics Will Remain CIO's Top Technology Priority*, Najdeno 19. aprila 2014 na na spletnem naslovu [http://www.gartner.com/technology/media-roducts/newsletters/datawatch/vol1\\_issue1/gartner.html](http://www.gartner.com/technology/media-roducts/newsletters/datawatch/vol1_issue1/gartner.html)
16. Gradišar, M., Jaklič, J., & Turk, T. (2007). *Osnove poslovne informatike*. Ljubljana: Ekonomska fakulteta.
17. Golfarelli, M., & Rizzi, S. (1998). *A Methodological Framework for Data Warehouse Design*. Najdeno 8.januarja 2013 na spletnem naslovu <http://libra.msra.cn/Publication/357890/a-methodological-framework-for-data-warehouse-design>
18. Green. D. (2010). *Checksum Transformation*. Najdeno 10 januarja 2014 na spletnem naslovu <http://www.sqlis.com/post/Checksum-Transformation.aspx>
19. Harinarayan, V., Rajaraman, A., & Ullman J.D. (1996). Implementing data cubes efficiently. *SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data* (str. 205-216). New York: ACM.
20. Harinath, S., Pihlgren, R., Lee, D., Sirmon, J., & Bruckner, M.R. (2012). *Professional Microsoft SQL Server 2012 Analysis Services with MDX and DAX*. Indianapolis: John Wiley & Sons, Inc.
21. Hinkka, A. (2012, 1. januar). *Query Execution in Columnar Databases*. Najdeno 20. januarja 2014 na spletnem naslovu [https://www.cs.helsinki.fi/webfm\\_send/1005/atte\\_final.pdf%3C](https://www.cs.helsinki.fi/webfm_send/1005/atte_final.pdf%3C)
22. Howson, C. (2008). *Successful Business Inteligence Secrets to Making BI a Killer App*. USA: McGraw-Hill
23. Hvala, D. (2011). Projekti in napovedovanje prihodnosti. *Monitor PRO revija*, 11(2), 30-32.
24. IDC. (2014, april). *Data Growth, Business Opportunities, and the IT Imperatives*. Najdeno 15. januarja 2016 na spletnem naslovu <http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>
25. Inmon W. (2005). *Building the Data Warehouse* (4<sup>th</sup> ed.), Indianapolis: Wiley Publishing, Inc.
26. Jaklič, J. (2008). *Gradivo pri predmetu Baze podatkov*. Ljubljana: Ekonomska fakulteta.
27. Jaklič, J. (2009). *Gradivo pri predmetu Tehnologija poslovne inteligence*. Ljubljana: Ekonomska fakulteta.
28. Kaltnekar, Z. (1988). *Organizacija delovnih procesov*. Kranj: Moderna organizacija.
29. Kimball, R., & Caserta, J. (2004). *The Data Warehouse ETL Toolkit, Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Indianapolis: Wiley Publishing inc.
30. Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit, Second Edition*. United States of America: John Wiley & Sons Inc.

31. Kobert, N. (1992). *Managing inventory for cost reduction*. Prentice Hall Direct.
32. Kovačič, A. (2009), Prosojnice pri predmetu strategija informatizacije poslovanja. Ljubljana: Ekonomska fakulteta
33. Krajčovič, M., & Plinta, D. (2012). Comprehensive approach to the inventory control system improvement. *Management and Production Engineering Review*, 3(3), 34-44.
34. *Leksikon Cankarjeve založbe* (1984). Cankarjeva založba, Ljubljana.
35. Leonard, A., Masson, M., Mitchell, T., Moss, J., & Ufford M. (2012). *SQL Server 2012 Integration Services Design Patterns*. New York: Springer Science+Business Media.
36. Ljubič T. (2000). *Planiranje in vodenje proizvodnje: modeli, metode, podatki*. Kranj: Moderna organizacija.
37. Logi Analytics (2014). *2014 State of Self-Service BI Report*. Najdeno 7. aprila 2016 na spletnem naslovu <http://learn.logianalytics.com/2014-state-self-service-bi-report>
38. Loshin D. (2003). *Business Intelligence The Savvy Managers's Guide*. San Francisco: Morgan Kaufmann Publishers.
39. Lukawiecki, R., (2014). *Business Intelligence Trends & Strategies*. Najdeno 14 decembra 2014 na spletnem naslovu <http://www.efocus.sk/images/uploads/1.lukawiecki.pdf>
40. Luhn, H. P. (1958). A Business Intelligence System, *IBM Journal of Research and Development* , 2(4), 314-319.
41. Makovec, I., (2012). Uporaba orodja poslovne inteligence pri upravljanju z zalogami v podjetju avtomobilske industrije. *19. konferenca Dnevi slovenske informatike [zgoščenka]*. Portorož: Slovensko društvo informatika.
42. Makovec, I., (2013). Od podatkov do informacij v primeru proizvodnega podjetja avtomobilske industrije. *5. industrijski forum IRT 2013*. Škofljica: Profidtp.
43. Marx, K., (1894). *Capital Vol. III, The process of capitalist production as a whole*. New York: International Publishers.
44. Microsoft BI Team, (2012, 11. oktober). *Blog Series: Creating BI Solutions with BISM Tabular (Part 1 of 4)*. Najdeno 28. februarja 2016 na spletnem naslovu [http://blogs.msdn.com/b/microsoft\\_business\\_intelligence1/archive/2012/10/11/blog-series-creating-bi-solutions-with-bism-tabular-part-1-of-4.aspx](http://blogs.msdn.com/b/microsoft_business_intelligence1/archive/2012/10/11/blog-series-creating-bi-solutions-with-bism-tabular-part-1-of-4.aspx)
45. MicroStrategy, (2011). *MicroStrategy 9, Project Design Guide, Eleventh Editio*. Najdeno 6. julija 2015 na spletnem naslovu <http://www2.microstrategy.com/producthelp/manuals/9.2.1/en/projectdesign.pdf>
46. Mihelčič, M., (2008). *Analiziranje kot pogoj oziroma temeljna sestavina »poslovne inteligence«*. Najdeno 10. februarja 2014 na spletnem naslovu <http://www.fri.uni-lj.si/file/100688/Clanek---analiziranje-kot-pogoj-oziroma-temeljna-sestavina-poslovne-inteligence.doc>

47. Moody, L. D., & Koritnik A.R., M. (2000). *From enterprise Models to Dimensional Models: A Methodology for Data warehouse and Data Mart Design.* , 2000. Najdeno 08. januarja 2013 na spletnem naslovu <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.511>
48. Nanda, A., (2011). *Hands-on Microsoft SQL Server 2008 Integration Services.* New York: Mc Graw Hill Companies.
49. Odločamo, ne da bi vedeli, kako to počnemo. (1978, 23. december). *Delo, sobotna priloga*, št. 296.
50. Parenteau, J., Sallam, L. R., Howson, C., Tapadinhas, J., Schlegel, K., & Oestreich, W. T. (2016, 04. februar). *Magic Quadrant for Business Intelligence and Analytics Platforms.* Najdeno 28. februarja 2016 na spletnem naslovu <https://www.gartner.com/doc/reprints?id=1-2XXET8P&ct=160204&st=sb>
51. Popovič, A., Hackney, R., Coelho, P. S., & Jaklič, J. (2012). Towards business intelligence systems success: Effects of maturity and culture on analytical decision making, *Decision Support Systems*, 54(1), 729 – 739
52. Pučko Danijel (1993). *Planiranje v podjetjih.* Ljubljana: Ekonomska fakulteta.
53. Power, D. J. (2007, 10. april). *A Brief History of Decision Support Systems.* Najdeno 27. aprila 2013 na spletnem naslovu <http://dssresources.com/history/dsshistory.html>
54. Rizzi, S., & Golfarelli, M. (2001). *Data Warehouse Design.* Najdeno 10. januarja 2014 na spletnem naslovu <http://www-db.deis.unibo.it/~srizzi/PDF/DWtutorial.pdf>
55. Rossum, J. (2012). *Custom SSIS Component: Checksum Transformation* Najdeno 10. januarja 2014 na spletnem naslovu <http://microsoft-ssis.blogspot.com/2012/04/custom-ssis-component-checksum.html>
56. Rozman, R., Kovač, J. & Koletnik, F. (1993). *Management.* Ljubljana: Gospodarski vestnik.
57. Rusjan, B. (2009). *Management proizvodnih in storitvenih procesov.* Ljubljana: Ekonomska fakulteta.
58. Russo, M. (2012). *Metadata for BI Solutions in the Microsoft stack.* Najdeno 21. februarja 2016 na spletnem naslovu [http://sqlblog.com/blogs/marco\\_russo/archive/2009/05/17/metadata-for-bi-solutions-in-the-microsoft-stack.aspx](http://sqlblog.com/blogs/marco_russo/archive/2009/05/17/metadata-for-bi-solutions-in-the-microsoft-stack.aspx)
59. Russo, M. (2013a). *Handling Wrong or Missing Dates in Tabular.* Najdeno 11. februarja 2016 na spletnem naslovu <https://www.sqlbi.com/articles/handling-wrong-or-missing-dates-in-tabular/>
60. Russo, M. (2013b). *Optimizing Data Models for Tabular Solutions and PowerPivot.* Najdeno 22. januarja 2014 na spletnem naslovu [http://www.sqlpass.org/Portals/333/0Marco%20Russo\\_Optimizing%20Data%20Models%20for%20Tabular%20Solutions%20and%20PowerPivot.pdf?ver=2013-11-18-140112-790](http://www.sqlpass.org/Portals/333/0Marco%20Russo_Optimizing%20Data%20Models%20for%20Tabular%20Solutions%20and%20PowerPivot.pdf?ver=2013-11-18-140112-790)
61. Russo, M., & Ferrari, A. (2011). *PowerPivot for Excel 2010 Give Your Data Meaning.* Redmont, Washington, USA: Microsoft Press.

62. Russo, M., Ferrari, A. & Webb, C. (2012). *Microsoft SQL Server 2012 Analysis Services The BISM Tabular Model*. California: O'Reilly Media Inc.
63. Sallam, L. R., Hostmann, B., Schlegel, K., Tapadinhas, J., Parenteau, J., & Oestreich, W. T. (2015, 23. februar) *Magic Quadrant for Business Intelligence and Analytics Platforms*. Najdeno 28. februarja 2016 na spletnem naslovu
64. Sallam, L. R., Tapadinhas, J., Parenteau, J., Yuen D., & Hostmann B. (2014, 20. februar). *Magic Quadrant for Business Intelligence and Analytics Platforms*. Najdeno 4. marca 2014 na spletnem naslovu <http://www.tableausoftware.com/gartner-magic-quadrant-2014>
65. Schmenner, R. W. (1993) *Production operation management*. New York: Macmillan
66. Serra, J. (2012, 4. april). *SQL Server 2012: Multidimensional vs tabular*. Najdeno 9. decembra 2015 na spletnem naslovu <http://www.jamesserra.com/archive/2012/04/sql-server-2012-multidimensional-vs-tabular/>
67. Sindol, D. (2013, 04. september). *Choosing between Analysis Services Multidimensional and Tabular Models*. Najdeno 4. marca 2016 na spletnem naslovu <http://blog.aditi.com/data/choosing-between-analysis-services-multidimensional-and-tabular-models-part-2/>
68. Sirmon, J., Galloway, G., Gross, C., & Gulati, K. (2013). *Performance Tuning of Tabular Models in SQL Server 2012 Analysis Services*. Microsoft SQL Server Technical Article. Najdeno 16. februarja 2016 na spletnem naslovu [https://msdn.microsoft.com/en-us/library/dn393915.aspx?tduid=\(fd21f467e769b50b2382465ce608ebe0\)\(256380\)\(2459594\)\(TnL5HPStwNw-Up\\_xU7HSIYY\\_\\_8vSmXzF0g\)\(\)](https://msdn.microsoft.com/en-us/library/dn393915.aspx?tduid=(fd21f467e769b50b2382465ce608ebe0)(256380)(2459594)(TnL5HPStwNw-Up_xU7HSIYY__8vSmXzF0g)())
69. Skupina TPV. (2015). *Skupina TPV*. Najdeno 5. februarja 2016 na spletnem naslovu <http://www.tpv.si/si/wp-content/uploads/sites/2/2015/09/PredstavitevTPV-si.pdf>
70. Slovenski računovodski standardi. *Uradni list RS* št. 95/2015.
71. Škrobáčková, M., & Kopáčková, H. (2006). *Decision support systems or business intelligence: what can help in decision making?*. Scientific papers of the University of Pardubice. Series D, Faculty of Economics and Administration. 10 (2006).
72. Thomsen, E. (2002). *OLAP Solutions Building Multidimensional Information Systems* (2<sup>nd</sup> ed.), New York: John Wiley & Sons Inc.
73. Van Ufford, D. Q. (2002). *Business Intelligence-The Umbrella Term*. University of Amsterdam, Facultatea de atiinte Exacte, Noiembrie.
74. Vitt, L., & Cameron S. (2012). *Choosing a Tabular or Multidimensional Modeling Experience in SQL Server 2012 Analysis Services*. Microsoft white paper. Najdeno 9. novembra 2013 na spletnem naslovu <https://msdn.microsoft.com/en-us/library/hh994774.aspx>
75. Vukovič, G., & Završnik, B. (2011). *Obvladovanje nabave*. Celje: Fakulteta za komercialne in poslovne vede.

76. Tableau. (2014). *Top 10 Trends in Business Intelligence for 2014*. Najdeno 17. marca 2014 na spletnem naslovu <http://www.tableausoftware.com/asset/top-10-business-intelligence-trends-2014?cid=701600000000YuYO&ls=Advertisement&lsd=AdRoll%20-%20Remarket%20-%20Top%2010%20Trends%202014&adgroup=&kw=&adused=&distribution=AdRoll>
77. The Data Warehousing Institute. (b.l.). *Business Intelligence*. Najdeno 15. februarja 2014 na spletnem naslovu <http://tdwi.org/portals/business-intelligence.aspx>
78. Veliki splošni leksikon. (1998). Ljubljana: DZS d.d..
79. Vitt, L. & Cameron, S. (2012). *Choosing a Tabular or Multidimensional Modeling Experience in SQL Server 2012 Analysis Services*, Microsoft Business Intelligence Tehnical Article. Najdeno 17. marca 2014 na spletnem naslovu <http://msdn.microsoft.com/en-us/library/hh994774.aspx>
80. Wade, C. (2013). *Multidimensional or Tabular*. Najdeno 9. decembra 2013 na spletnem naslovu <http://christianwade.wordpress.com/2013/11/09/multidimensional-or-tabular/>
81. Waters, D. (2003). *Inventory Control and Management*, West Sussex: John Wiley & Sons.
82. Wrembel, R. & Koncilia, C. (2007). *Data Warehouses and OLAP Concepts, Architectures and Solutions*. USA: IRM Press.
83. Zaman, M.. (2005). *Business Intelligence: Its Ins and Outs*. Najdeno 12. februarja 2014 na spletnem naslovu [http://www.ism.co.at/analyses/Business\\_Intelligence/Ins\\_and\\_Outs.html](http://www.ism.co.at/analyses/Business_Intelligence/Ins_and_Outs.html)
84. Završnik, B. (2008). *Management nabave in oskrbnih verig*. Maribor: Ekonomsko-poslovna fakulteta.