

Ali sta tabelarični model in jezik DAX znanilca poslovnointeligenčnih rešitev za vse — študija primera

Igor Makovec, Ivan Erenda
TPV, d. d., Kandijška cesta 60, 8000 Novo mesto
i.makovec@tpv.si; i.erenda@tpv.si

Izvleček

Razvoj poslovnointeligenčnih rešitev je šel skozi tri obdobja. V prvem je bil še v domeni oddelkov za informatiko in osredinjen na orodja, v drugem je bil omogočen spletni dostop in pojavili so se hibridni uporabniki (poznavalci procesov in informacijske tehnologije) in v tretjem obdobju je osredinjen na aplikacije in je v domeni končnih uporabnikov. V analitično prezentacijski plasti poslovnointeligenčnih rešitev že dve desetletji kraljujejo kocke OLAP. Leta 2010 je Microsoft na trg poslal tabelarični model skupaj z jezikom DAX, ki je bil namenjen uporabnikom brez specializiranih analitičnih znanj, da bi lahko sami razvili podatkovne modele in prišli do iskanih informacij. Članek pokaže praktični primer uporabe tabelaričnega modela in uporabe jezika DAX na primeru izdelave poročila o letnem popisu materiala v podjetju. Moč jezika DAX se izkaže pri agregacijah, ki so potrebne za pravičen izračun dopustnega in davčno neobremenjenega kala oziroma izgube zaradi skladiščenja in prevoza blaga. V članku poizkušamo odgovoriti na vprašanje, ali je tak model že znanilec obdobja poslovne inteligence za vse in ali ima dovolj funkcionalnosti za veliko uporabnikov ob manjših stroških kot kadar koli prej.

Ključne besede: tabelarični model, BISM, Power Pivot, jezik DAX.

Abstract

Are the Tabular Model and the DAX Language Forerunners of Business Intelligence Solutions for Everyone? A Case Study

The development of business intelligence solutions is viewed as a three—stage process. In the first stage it was dedicated to IT experts and was focused on tools. The second stage offered web access and was placed into the hands of hybrid users. In the third stage it is increasingly focused on applications and intended for end users. In an analytical presentation layer of the last two decades, OLAP cubes have been playing a major role. In 2010 Microsoft published a tabular model with a brand new language DAX (Data Analysis Expressions), which was dedicated to users without special BI and analytical skills with the aim to allow self-building of models and analysing data. This article shows a practical example of a report by inventory account using the tabular model and the DAX language. The power of DAX can be seen in simple aggregated calculations of allowed tax free loss due to warehousing and transportation. We also try to answer the question whether the tabular model is a forerunner of »BI for everyone« and whether its functionalities are sufficient for a wider spectrum of users with lower costs than ever.

Key words: Tabular model, BISM, Power Pivot, DAX language.

1 UVOD

Načelo uspešnosti poslovanja je podjetju družbenoekonomsko določeno. V kapitalizmu je to dobičkonosnost ali rentabilnost. Bistveni sta dve meri, in sicer z vidika lastnika dobičkonosnost kapitala (angl. Return of Equity, ROE) in z vidika podjetja dobičkonosnost sredstev (angl. Return of Assets, ROA). Da bi bili ti dve meri čim večji, poteka v podjetjih neprestano odločevalski proces. Danes je ta bolj kot kadar koli prej podprt z informacijami, ki jih pridobimo iz podatkov.

To je povsem razumljivo, saj podjetja delujejo v vedno bolj zapletenih okoljih, kjer je na voljo veli-

ko podatkov. Podjetja jih generirajo tudi sama, saj je ravno zaradi težnje po dobičkonosnosti merjeno vedno več količin pri poslovanju podjetja z namenom, da lažje razpoznamo trenutno stanje in sprejmemo takojšnje korektivne ali prilagoditvene ukrepe za doseg ciljev ali pravilnega odgovora na spremembe na trgu. Pri tem nam najbolj pomagajo poslovnointeligenčni sistemi (angl. Business Intelligence, BI), brez katerih si danes ne znamo več predstavljati poslovanja.

Namen članka je prikazati možnost samopostrežnega poslovnointeligentnega sistema z uporabo dodatka Power Pivot v Excelu. V prvem delu navajamo prednosti in slabosti dveh podatkovnih modelov, večdimenzijskega in tabelarnega. Pri tem je pojasnjeno bistvo novega funkcijskega in poizvedbenega jezika DAX (angl. Data Analysis Expressions). V drugem delu je prikazana praktična uporaba orodja Power Pivot in jezika DAX na primeru izdelave poročila letnega popisa zalog, pri čemer je bil bistveni poudarek na pravilnem izračunu dopustnega uničenja, ki je neločljivo povezano s skladiščenjem blaga.

2 POSLOVNA INTEGENCA, KOT JO DANES VIDI MICROSOFT

2.1 Zgodovina

Leta 1998 je Microsoft predstavil SQL Server 7.0 kot sistem za upravljanje z relacijskimi podatkovnimi bazami, ki je vseboval OLAP Services za delo s podatkovnimi kockami OLAP. Dobili smo inovativno tehnologijo za hranjenje podatkov in agregacij. Leta 2000 je bil SQL Server napisan popolnoma na novo in je dobil znotraj OLAP Services tudi možnost podatkovnega rudarjenja. Konec leta 2005 je izšel SQL Server 2005, ki je vseboval postopke ETL (angl. Extract Transform Load), združene pod imenom SSIS (SQL Server Integration Services), strežnik za OLAP in podatkovno rudarjenje z imenom SSAS (SQL Server Analysis Services) ter strežnik za izdelavo poročil z imenom SSRS (SQL Server Reporting services). Leta 2010 smo dobili SQL Server 2008 R2 in samopostrežno poslovnointeligentno rešitev v sklopu kombinacije Excel 2010 – Power Pivot z jezikom DAX. Kreiranje modelov tako ni bilo več samo v domeni oddelkov za informatiko. Poslovni uporabniki so lahko končno sami kreirali podatkovne modele.

S SQL Server 2012 smo dobili tudi nov pojem semantični model (angl. Business Intelligence Semantic Model, BISM). Microsoft je s tem, ko je večdimenzijski in tabelarni model združil pod enim imenom, naredil tudi nekaj zmede, saj gre za dva različna modela in preden gremo v poslovnointeligentno rešitev, se je treba odločiti, kateri podatkovni model bomo izbrali.

2.2 Poslovnointeligentni semantični model

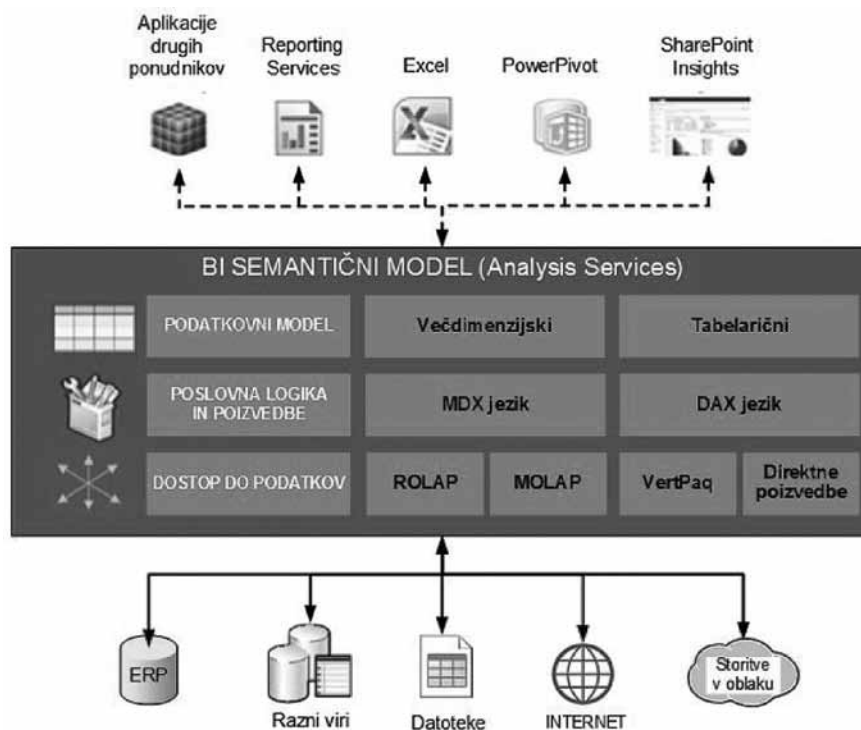
Poslovnointeligentni semantični model je nastal zaradi potreb časa. Imamo eno metodologijo dela in dve tehnologiji. Iz arhitekture vidimo, da je sestavljen iz starega večdimenzijskega modela in novega tabelarnega modela (slika 1). Danes prehod med njima še ni mogoč, čeprav razvoj poteka v tej smeri. O tem, kateri model izbrati, bomo več povedali v nadaljevanju.

Pri poslovni logiki imamo dva močna jezika, ki sta primerna za poizvedbe vsak za svoj model.

- MDX (angl. MultiDimensional eXpressions) je jezik za poizvedbe na podatkovni kocki OLAP. MDX kot jezik večdimenzijskih izrazov določa posebno sintakso za poizvedovanje nad množico večdimenzijskih podatkov, shranjenih v kockah OLAP. Na videz je podoben tradicionalnem jeziku SQL, zato je mnogo izrazov MDX mogoče prevesti v tradicionalni SQL. Dular in Godnov (2013) pri pregledu MDX ugotavljata, da je praviloma več vrstic kode v SQL mogoče zamenjati z eno samo vrstico v MDX.
- DAX (angl. Data Analysis Expressions) je nov jezik in je podoben funkcijam v Excelu. Omogočena je dinamična analiza, kar pomeni, da so rezultati prikazani v smislu pogleda oz. konteksta, ki ga izoblikujemo. DAX uporablja tabelarne konstrukte (tabele, stolpce in relacije) in vsebuje funkcionalnosti za podporo razvoju modelov. Njegove funkcije so razdeljene v več sklopov: datumske in časovne funkcije, informacijske, logične, matematične, statistične, besedilne in funkcije filtriranja ter t. i. funkcije časovne inteligence. Krivulja učenja jezika DAX je bistveno krajša kot pri jeziku MDX. Jezik ni zapleten, vendar tudi ni preprost. Že po enem tednu spoznavanja jezika lahko sestavimo tudi kompleksnejše modele in izvedemo zahtevnejše izračune.

Pri dostopu do podatkov je med modeloma velika razlika. Pri večdimenzijskem modelu so podatki v primeru MOLAP (angl. Multidimensional Online Analytical Processing) shranjeni v podatkovnem skladišču, ravno tako osnovni in agregirani podatki kocke. Pri ROLAP (angl. Relational Online Analytical Processing) pa kocka ni zgrajena, temveč je zgrajen in shranjen samo opis kocke.

Pri tabelarnem modelu je podatkovni model v celoti v spominu računalnika. Za to skrbi jedro xVelocity (bivši VertiPaq), ki zagotavlja hitre poizvedbe,



Slika 1: Arhitektura BI semantičnega modela (Vir: Harinath idr., 2012)

in sicer to izvaja na način grobe sile s pregledom podatkov v spominu. Podatki so shranjeni po stolpcih in ne kot je običajno po vrsticah. Hranjenje podatkov po vrsticah tabel je standard od sredine sedemdesetih let prejšnjega stoletja. Princip shranjevanja po stolpcih ni nekaj novega. Prva sta ga opisala že Estabrook in Brill leta 1969. Stonebraker idr. (2005) so v svoji raziskavi predstavili za branje optimirani design relacijskih sistemov za upravljanje podatkovnih baz. Posebno pri podatkovnih skladiščih je shranjevanje po stolpcih v nekaterih primerih bolj učinkovito, saj je treba pri obdelavi prebrati samo vrednosti stolpcev, ki so predmet zanimanja. S tem se izognemo branju nepotrebnih atributov. Za kompresiranje podatkov v stolpcih (spomina računalnika ni toliko, kot si bi ga želeli) jedro xVelocity uporablja dve tehniki, in sicer (Russo, 2013b):

- tako imenovano kodiranje RLE (angl. Run Length Encoding), s katerim v spomin ne shrani vseh vrednosti stolpca, ampak samo različne vrednosti in mesto, kjer se podatek pojavi prvič, ter število ponovitev, ki sledijo;
- slovarsko kodiranje, pri katerem v prvem koraku uredimo zapise po vrstnem redu, v drugem koraku pa sledi kodiranje RLE. Tako vsako vred-

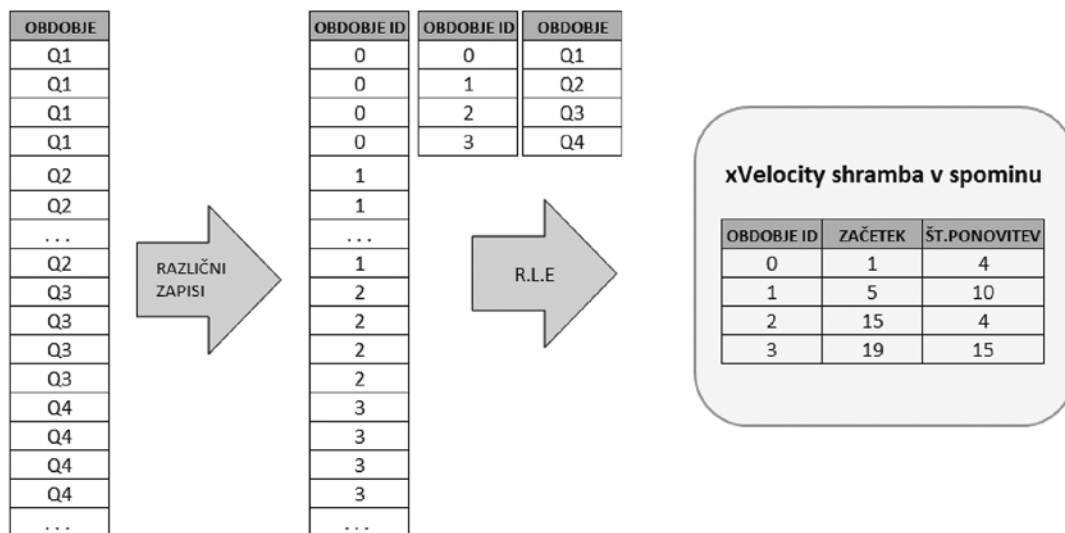
nost zapišemo samo enkrat in s tem bistveno zmanjšamo obseg zapisov (slika 2).

Stopnja kompresiranja tako doseže desetino velikosti nekomprimirane SQL baze podatkov.

Vitt in Cameron (2012) navedeta ključni značilnosti tabelaričnega modeliranja.

- Domačnost – delo s tabelaričnim modelom je domače veliko uporabnikom, ki redno delajo s podatki v tabelah, shranjenih v relacijskih podatkovnih bazah. DAX je jezik za pisanje formul in širi funkcionalnost Excela. Obstoječa znanja je mogoče hitro nadgraditi – dosti hitreje kot pri prehodu na večdimenzijske modele.
- 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 treba zaradi spremenjenih poslovnih zahtev spremeniti relacije in izračune.

Ob predstavitvi tabelaričnega modela je bilo veliko vprašanj, ali je to podatkovna kocka OLAP. Tabelarični model je dvodimenzijska predstavitev podatkov, čeprav je s stališča uporabnika pri kreiranju pogledov videti kot kocka. V prezentacijski plasti poslovno-inteligenčnega sistema infrastrukture poglede



Slika 2: V spominu se hranijo kompresirani podatki stolpcev (Vir: Povzeto po Russo, 2013b)

na kocke in na tabelarni model gradimo podobno kot pri vrtilnih tabelah – vmesnik je enak. To je dobro, uporabno, a lahko koga tudi malo moti. Ko poznamo ozadje, tak pristop lahko pozdravimo.

2.3 Moč jezika DAX

Funkcije DAX uporabimo z namenom, da definirajo polja, ki se prikažejo v Excelovi vrtilni tabeli. Poznamo dva tipa polj: izračunane stolpce in mere (Dickerman in Myers, 2011).

Reference v DAX niso v obliki *StolpecVrstica*, npr. A9, kot je v Excelu, temveč govorimo o popol-

nom kvalificiranih imenih v obliki *'Ime tabele'[Ime stolpca]*.

Izračunane stolpce dobimo tako, da v oknu Power Pivot prazen stolpec poimenujemo in vpišemo funkcijo DAX. Ta se takoj izračuna po vseh vrsticah tabele. Tako dobimo stolpec, kot so vsi drugi v tabeli. Edina razlika je, da so (neizračunani) običajno uvoženi iz eksternih podatkovnih virov, medtem ko se izračunani napolnijo po uvozu ali osvežitvi podatkov. Ti stolpci se obnašajo enako kot ostali – velja tako za prikaze v vrtilni tabeli kot za nadaljnja izračunavanja (slika 3).

The screenshot shows the Power Pivot interface. A calculated column is being created. The formula bar displays the DAX formula: `=if([Znesiv dom#val#]<0; [Znesiv dom#val#]; 0)`. The table below shows data for two rows. The calculated column is highlighted with a box and an arrow pointing to the formula bar.

Znesiv dom#val#	val#	dat#vnosa	Teden	MinusVredn1
-0,38 EUR	130901	30.10.2013 0...	WI	201344
-8,51 EUR	130901	30.10.2013 0...	WI	201344

Slika 3: Primer izračunanega stolpca v Power Pivotu

Mere so imenovani izrazi – formule. Dodamo jih v vrtilno tabelo na mesto izračunanih vrednosti. Pomembno je, da je rezultat izračunan dinamično za vsako celico v polju vrednosti glede na kontekst, ki ga določajo v vrtilni tabeli vrstice, stolpci in filtri. Na sliki 4 vidimo eno od novih mer. Rezultat ni odvisen samo od pogleda v vrtilni tabeli, temveč tudi od konteksta, ki je vgrajen kot filter v formuli DAX. V tem primeru gre za vsoto količine blaga, ki je bilo prejeto

med dvema datumoma. Za vsako blago sta bila datauma lahko različna.

Izpostaviti želimo novo funkcijo *Calculate*. Njena sintaksa je *CALCULATE (izraz DAX, Filter1, Filter2 itn.)*.

S to funkcijo (slika 4) določimo novo mero in nabor filtrov, pri katerih se izračuna. Kontekst, v katerem se prikaže rezultat, določajo vrstice, stolpci in filtri vrtilne tabele.

Številka materiala		PrejKolMedPopisi:=CALCULATE(sum('Premiki'[Količina]); or('Premiki'[VrP] = 101; 'Premiki'[VrP] = 102); DATESBETWEEN('Premiki'[Dat#knj#]; [PopisPredzad]; [PopisZad]))			
Obr#	Material	Številka materiala	SLok	VrP	Količina
1230	306226	PODSTAVEK FIKS. NOS ZUN. D-...	53	101	19
1230	306226	PODSTAVEK FIKS. NOS ZUN. D-...	53	101	19
		PrejKolMedPopisi: 3055122,5			
		PrejVredMedPopisi: 908866,65			

Slika 4: Primer mere v Power Pivotu

2.4 Kdaj uporabiti tabelarni model in jezik DAX

Serra (2013) ugotavlja, da je s tabelarnim modelom gradnja kock OLAP (v bistvu ne gre za kocke, samo isti vmesnik se uporablja) preprostejša kot v večdimenzijskem modelu, vendar uporaba tega modela še ni razširjena. Razlog vidi v potrebnem času učenja razvijalcev poslovnointeligenčnih sistemov, ki potrebujejo zadosten razlog za prehod na novo orodje. Tabelarni model ima še več pomanjkljivosti. Našteje jih kar 29. Če jih povzamemo samo nekaj:

- ni varnostnih mehanizmov na ravni zapisa,
- velikostna omejitev kocke OLAP, ki mora biti v celoti v spominu računalnika,
- relacije mnogo – mnogo zahtevajo delo z jezikom DAX,
- ne podpira podatkovnega rudarjenja,
- ne podpira uporabniškega formatiranja mer,
- omogoča samo eno kocko na podatkovni bazi,
- hierarhija starš – otrok zahteva delo z jezikom DAX,
- ne podpira raztrganih hierarhij (angl. ragged hierarchy),
- podpira samo poizvedbe v realnem času s t. i. directquery.

Tabelarni model je primeren za manjše preprostejše projekte, ki zahtevajo najboljše lastnosti ob uporabniški izkušnji. Trenutno ga še ne priporoča za kaj drugega.

Russo (2013a) potrdi te ugotovitve, vendar samo na delu manjših možnosti DAX proti MDX. Ugotavlja, da je tabelarni model dobra izbira v primerih:

- ko razvojni tim ne pozna dobro večdimenzijskega modela in jezika MDX; jezik DAX se je mogoče naučiti hitreje;
- ko niso potrebni izračuni, ki temeljijo na hierarhijah;
- ko obstaja potreba po izračunih, ki temeljijo na preštevanju različnih vrednosti mer;

- ko obstaja potreba po kompleksnih izračunih, ki temeljijo na relacijah mnogo proti mnogo.

Dodaja, da bi pri 80 odstotkih novih projektov lahko uporabili večdimenzijski ali tabelarni model in da pri tem igrajo najpomembnejšo vlogo sposobnosti razvojnega tima.

Temu je treba dodati, da je zelo verjetno, da bo Microsoft tabelarnemu modelu dodajal manjkajoče lastnosti in bodo razvojne investicije fokusirane na ta model (Wade, 2013).

Na sliki 5 vidimo, komu je namenjen kateri model: tabelarni model je za osebne, timske in delno korporacijske rešitve poslovnointeligenčnih sistemov, večdimenzijski model pa za zahtevne korporacijske rešitve poslovnointeligenčnih sistemov.

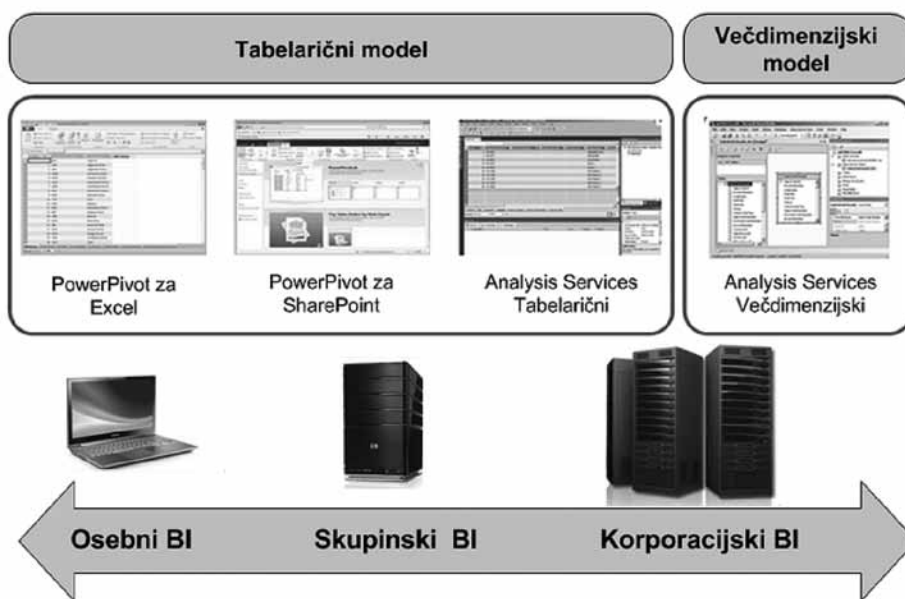
3 UPORABA POWER PIVOT PRI POROČILU POPISA ZALOG

Konec leta je bilo treba hitro sestaviti rešitev, s pomočjo katere bi pripravili poročila letnega popisa za vse poslovne enote TPV, d. d. Med drugim mora poročilo vsebovati podatke o:

- višku,
- manku,
- znižanjih davčne osnove:
 - dopustno uničenje zaradi skladiščenja,
 - poboti materiala,
 - druga dokazljiva znižanja,
- davčni obveznosti.

Najbolj zahteven je del izračuna dovoljenega kala oz. uničenje, ki ga kot zmanjšanje davčne osnove pri popisnih minusih kot dopustnega predvideva Pravilnik o stopnjah običajnega odpisa primanjkljaja in uničenja blaga (kalo, razsip, razbitje in okvara), ki sta neločljivo povezana s skladiščenjem in prevozom blaga (Ur. l. RS, 113/2008).

Uporabili smo programsko opremo Excel 2010 z dodatkom Power Pivot. Predvsem zato ker je po-

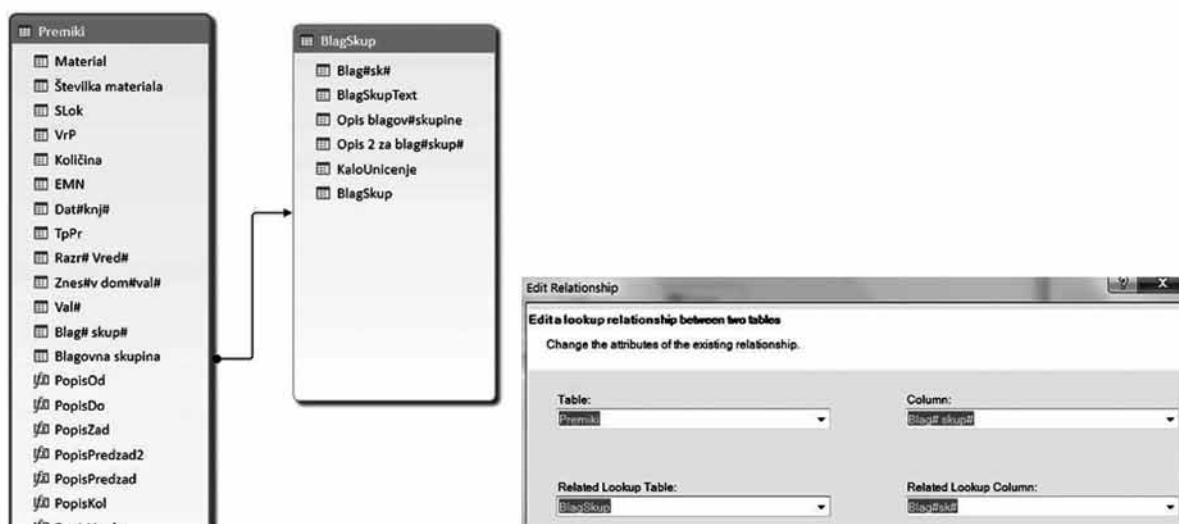


Slika 5: Uporabniki modelov (Vir: Prirejeno po Coates, 2012)

datkovni model vedno hranjen skupaj s poročilom in je tako poročilo. Uporabljen je bil agilni pristop predvsem zaradi ene od dvanajstih točk manifesta agilnega razvoja programske opreme, ki pravi, da je bistvena preprostost, tj. umetnost zmanjševanja količine nepotrebne dela.

Skupina TPV, d. d., kot celovito programsko rešitev uporablja SAP. V ozadju so bili izvoženi vsi premiki prejemov blaga (101, 102) in inventurni premiki (701–704, 707, 708) za leto 2013. Teh zapisov je okoli pol milijona. Izvozila se je tudi tabela vseh

blagovnih skupin, pri čemer smo skladno s pravilnikom dopisali odstotek dovoljenega uničenja blaga po posamezni blagovni skupini. Podatkovni model je zelo preprost, saj je sestavljen samo iz dveh tabel. V Power Pivotu sta bili narejeni povezavi na ti dve tabeli in določeni so bili stolpci, ki jih vidi uporabnik. Imena stolpcev se niso menjala, ker jih uporabniki poznajo iz izpisov SAP. Po uvozu podatkov v podatkovni model se je na način povleci in spusti kreirala edina relacija med tabelama (slika 6).



Slika 6: Enostaven podatkovni model in relacija

V oknu Power Pivot so bile na mestu za vnos izrazov DAX kreirane nove mere. Navajamo nekatere izmed njih.

- Zadnji datum popisa:

```
PopisZad:=CALCULATE(LASTDATE('Premiki'[Dat#knj#]); 'Premiki'[VrP] = 701 || 'Premiki'[VrP] = 702 || 'Premiki'[VrP] = 703 || 'Premiki'[VrP] = 704 || 'Premiki'[VrP] = 707 || 'Premiki'[VrP] = 708; FILTER('Premiki'; 'Premiki'[Dat#knj#] <= date(YEAR([PopisDo]);MONTH([PopisDo]);DAY([PopisDo])))
```

- Datum predhodnega popisa:

```
PopisPredzad2:=CALCULATE(MAX('Premiki'[Dat#knj#]); FILTER(ALL('Premiki'); COUNTROWS(FILTER('Premiki'; (EARLIER('Premiki'[Material])='Premiki'[Material] && EARLIER('Premiki'[Dat#knj#])<'Premiki'[Dat#knj#]) && (EARLIER('Premiki'[VrP]) = 701 || EARLIER('Premiki'[VrP]) = 702 || EARLIER('Premiki'[VrP]) = 703 || EARLIER('Premiki'[VrP]) = 704 || EARLIER('Premiki'[VrP]) = 707 || EARLIER('Premiki'[VrP]) = 708))))
```

- Če predhodnega popisa ni, kot predhodni datum popisa vzamemo datum prvega premika v letu:
PopisPredzad:=IF(ISBLANK([PopisPredzad2]); FIRSTDATE('Premiki'[Dat#knj#]); [PopisPredzad2])
- Prejeta količina blaga med popisom, za katerega delamo poročilo, in predhodnim popisom:
PrejKolMedPopisi:=CALCULATE(sum('Premiki'[Količina]); or('Premiki'[VrP] = 101; 'Premiki'[VrP] = 102); DATESBETWEEN('Premiki'[Dat#knj#]; [PopisPredzad]; [PopisZad]))

Sestavljena je bila vrtilna tabela (slika 7). Upoštevane so nove mere, kreirane v jeziku DAX. Za vsak material smo dobili potrebne podatke in izračunan dopustni ter upoštevan kaloz. Na vsaki poslovni enoti so v okviru popisa ugotavljali, ali je mogoče še dodatno znižati davčno osnovo. Te možnosti so poboti med materiali v okviru zakonodaje, označitev, da gre za potrošni material, če morda ni bil pravilno zajet v kosovnicah, in druga znižanja na podlagi zapisnikov z dokazili.

Mate	Številka materiala	Blagi	Blagovna skupina	KaloUnic	SLo	EN	PopisKol	PopisVredn	PopisPredzad	PopisZad
306606	NOSILEC ODBUJAČA LEVI - KRIVLJENE	130901	Odpreski	2.5	53	KOS	-7	-1,58	29.5.2013 0:00	23.12.2013 0:00

PrejKolMedPopis	PrejVredMedPopisi	DovoljKaloKol	UpostKaloKol	UpostKaloVredn	PobotKol (kontra predznak)	PobotVredn	PotrosMat (kontra predznak)	PotrosMatVredn	DrugaZnizKol (kontra predznak)	DrugaZnizVredn	KONTROLA
1741	393,81	43,525	7	1,58 €		- €		- €		- €	OK

Slika 7: Atributi analize z možnostjo korekcij (prikazani v dveh delih)

Sproti se kreira poročilo, ki je vključeno v zapisnik popisne komisije (slika 8). Od tu naprej pa so možne

analize vezane na lokalizacijo manka ali viška po vseh atributih, ki jih imamo v podatkovnem modelu.

ZAP. ŠT.	OPIS	VREDNOST
1	Višek	16.005,83 €
2	Manjko	16.005,83 €
3	Poboti	6.375,81 €
4	Dopusten primanjkljaj ali uničenje (Pravilnik o stopnjah običajnega odpisa primanjkljaja in uničenja blaga (kalo, razsip, razbitje in okvara), ki sta neločljivo povezana s skladiščenjem in prevozom blaga (Ur.l. RS 113/2008))	8.171,81 €
5	Potrošni material	5
6	Druga dokazljiva znižanja:	1.58 €
7	Znižanje davčne osnove skupno (3+4+5+6)	6.376,39 €
8	Osnova za DDV (2-7)	6.376,39 €
9	Znesek DDV (8*22%)	1.402,81 €

Slika 8: Kvantitativni del poročila popisa

4 SKLEP

Kot dobro velja pri tabelarnem modelu izpostavi-li preprostost gradnje podatkovnega modela in moč jezika DAX. Nedvomno je tabelarni model zaradi vsega navedenega zelo primeren pri osebnih in skupinskih poslovnointeligenčnih rešitvah. Pri korporacijskih je to odvisno od kompleksnosti problema. Ali je to prava pot, bo pokazal čas. Vse skupaj je odvisno od tega, kako bodo rešitev sprejeli uporabniki oziroma razvijalci. Menimo, da se moramo začeti učiti DAX čim prej, tudi zato, ker je jezik nezahteven. Povsem novo je le razumevanje novega konteksta filtrov pri uporabi izrazov DAX. V določenih primerih, ko moramo hitro priti do zahtevanih informacij, ne moremo čakati, da oddelek za informatiko k obstoječim večdimenzijskim modelom doda nove tabele dejstev ali tabele mer. V teh primerih smo s tabelarnim modelom in DAX zagotovo hitrejši.

Vključenost tabelarnega modela znotraj semantičnega modela je tisto, kar potrebuje resen poslovnointeligenčni sistem. Gre za tisto, kar manjka navezi Excel – Power Pivot, to je varstvo podatkov, osveževanje in izdelava varnostnih kopij.

Razvoj tabelarnega modela gre naprej in po dolgem času pomeni novost v analitično-prezentacijski plasti poslovnointeligenčnih sistemov in je zato resna alternativa večdimenzijskim modelom.

VIRI IN LITERATURA

- [1] Dickerman, H., Myers, P. (2011). *Data Analysis Expressions (DAX) in the Tabular BI Semantic Model*. White paper. Objavljeno na <http://www.microsoft.com/en-us/download/details.aspx?id=28572> (zadnji ogled 9. 12. 2013).
- [2] Dular, T., Godnov, U. (2013). *Analiza zalog*, 20. Konferenca Dnevi slovenske informatike, 15–17. 4. 2013, Portorož.
- [3] Coates, M. (2012). *Decisions: PowerPivot, SSAS Tabular, or SSAS Multidimensional Model in SQL Server 2012*. Objavljeno na <http://www.sqlchick.com/entries/2012/3/4/decisions-powerpivot-ssas-tabular-or-ssas-multidimensional-m.html> (zadnji ogled 22. 1. 2014).
- [4] Estabrook, W., G., Brill C., R. (1969). *The Theory of the TAXIR Accessioner*. Mathematical Biosciences 5 (1969), 327–340. Objavljeno na <http://home.comcast.net/~tolkin.family/taxir1.htm> (zadnji ogled 24. 1. 2014).
- [5] Harinath, S., Pihlgren, R., Lee, D., Sirmon, J., Bruckner, M. R. (2012). *Professional Microsoft® SQL Server® 2012 Analysis Services with MDX and DAX*, John Wiley & Sons, Inc., Indianapolis.
- [6] Pravilnik o stopnjah običajnega odpisa primanjkljaja in uničenja blaga (kalo, razsip, razbitje in okvara), ki sta neločljivo povezana s skladiščenjem in prevozom blaga. *Uradni list RS*, št. 113/2008.
- [7] Russo, M. (2013a). *Updates about Multidimensional vs Tabular #ssas #msbi*. Objavljeno na http://sqlblog.com/blogs/marco_russo/archive/2013/11/11/updates-about-multidimensional-vs-tabular-ssas-msbi.aspx (zadnji ogled 9. 12. 2013).
- [8] Russo, M. (2013b). *Optimizing Data Models for Tabular Solutions and PowerPivot*. Objavljeno na http://www.sqlpass.org/Portals/333/Marco%20Russo_Optimizing%20Data%20Models%20for%20Tabular%20Solutions%20and%20PowerPivot.pdf (zadnji ogled 22. 1. 2014).
- [9] Serra, J. (2013). *Tabular model: Not ready for prime time?* Objavljeno na <http://www.jamesserra.com/archive/2013/11/tabular-model-not-ready-for-prime-time/> (zadnji ogled 9. 12. 2013).
- [10] Stonebraker, M., Abadi, D. J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S., O'Neil, E., O'Neil, P., Rasin, A., Tran, N., Zdonik, S. (2005). *C-Store: A column-oriented DBMS*, *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, 2005, 557–564. Objavljeno na <http://db.lcs.mit.edu/projects/cstore/vldb.pdf> (zadnji ogled 9. 12. 2013).
- [11] Wade, C. (2013). *Multidimensional or Tabular*. Objavljeno na <http://christianwade.wordpress.com/2013/11/09/multidimensional-or-tabular/> (zadnji ogled 9. 12. 2013).
- [12] Vitt, L., Cameron, S. (2012). *Choosing a Tabular or Multidimensional Modeling Experience in SQL Server 2012 Analysis Services*, Microsoft Business Intelligence Technical Article. Objavljeno na <http://msdn.microsoft.com/en-us/library/h> (zadnji ogled 24. 1. 2014).

Makovec Igor je samostojni projektant logistike v direkciji logistike TPV, d. d. Svojo strokovno pot je začel kot mladi raziskovalec iz gospodarstva. Delal je kot načrtovalec regulacijskih sistemov pri podjetju iz tehnološkega parka Instituta Jožef Stefan. Kasneje je več let opravljal dela višjega svetovalca in vodje organizacijske enote v sklopu Carinske uprave Republike Slovenije. Delo v industriji je nadaljeval kot vodja logistike v podjetju skupine TPV. Sedaj načrtuje interno in eksterno logistiko s poudarkom na določitvi lastne cene logistike za nove projekte. Dejaven je na področju inovativnosti in razvoja. Zadnja leta redno sodeluje na strokovnih konferencah s prispevki s področja poslovne inteligence.

Ivan Erenda, direktor sektorja oskrba in proizvodnja in direktor PE Velika Loka TPV, d. d., je svojo strokovno pot začel kot tehnolog v podjetju Prevent, d. d., nadaljeval kot razvojni tehnolog in vodja sistemov kakovosti v podjetju IUUV, d. d., ter kot direktor kakovosti, direktor PE Velika Loka, direktor PE Suhor in direktor PE Ptuj v TPV, d. d. Raziskovalno se ukvarja predvsem s kriznim vodenjem in kompetencami intuitivnega delovanja.