

## SPRENDIMŲ RĖMIMO SISTEMOSE NAUDOJAMŲ DUOMENŲ SAUGYKLŲ PROJEKTAVIMO METODŲ ANALIZĖ

Audrius Rima<sup>1</sup>, Aidas Šmaižys<sup>2</sup>, Olegas Vasilecas<sup>3</sup>

Vilniaus Gedimino technikos universitetas

El. paštas: <sup>1</sup>audrius@delfi.lt; <sup>2</sup>asmaizys@gmail.com, <sup>3</sup>olegas@fm.vgtu.lt

**Santrauka.** Duomenų saugyklos projektavimas skiriasi nuo tradicinių duomenų bazių projektavimo proceso. Operatyvaus valdymo informacinių sistemų reliacinės duomenų bazių valdymo sistemos yra grindžiamos normalizacija ir yra pritaikytos efektyviai vykdyti paprastas transakcijas. Kai tuo tarpu duomenų saugyklos grindžiamos denormalizacija ir yra skirtos sudėtingoms užklausoms vykdyti ir daugiamatei duomenų analizei atlikti. Be to, duomenų saugyklose, skirtingai nuo tradicinių duomenų bazių valdymo sistemų, skirtų operatyviam verslo procesų valdymui, yra saugomi konsoliduoti duomenys. Todėl duomenų saugyklų projektavimui taikomi metodai savo paskirtimi iš esmės skiriasi, bet jie dar nėra pakankamai išstbulinti ir neatitinka realių šiandienos inžinerinių poreikių. Šiame straipsnyje nagrinėjami egzistuojantys sprendimų rėmimo sistemų duomenų saugyklų projektavimo metodai, analizuojami tokių saugyklų kūrimo ypatumai, inžineriniai poreikiai ir verslo reikalavimai.

**Reikšminiai žodžiai** Duomenų saugykla, daugiamatė analizė, verslo taisyklės, sprendimo rėmimo sistemos.

### Įvadas

Organizacijose šiandien naudojamas keletas informacinių sistemų, kurių kiekviena skirta palaikyti tam tikrą verslo veiklą. Operatyvaus (angl. *operational*) duomenų apdorojimo sistemos yra orientuotos į transakcijos greitį ir dažniausia yra naudojamos reliacinės duomenų bazių valdymo sistemos (RDBVS). Duomenys yra normalizuojami ir mažinamas duomenų pertekliškas. Duomenų saugyklos (DS) yra skirtos sudėtingoms užklausoms įvykdyti, duomenys juose saugomi denormalizuoti, taip pat dažniausia duomenų saugyklose saugomi istoriniai duomenys iš įvairių sistemų. Dėl šių savybių duomenų saugyklų projektavimo metodai turi skirtis nuo įprastų operatyvaus duomenų apdorojimo bazių kūrimo metodų.

DS dažniausia kuriama kaip centrinė istorinių duomenų saugykla, kuri vartotojams suprantamai pateikia duomenis. DS suteikia tradicinį, lengvai valdomą duomenų centrą sprendimo rėmimo sistemoms ir užtikrina, kad duomenys būtų lengvai prieinami vartotojams. DS nėra tik duomenų kopija, tai unikalus duomenų rinkinys, kuris yra pritaikytas sprendimų priėmimui.

Straipsnio tikslas apžvelgti DS projektavimo metodus, taip pat atkreipti dėmesį į problemas su kuriomis susiduriama kuriant duomenų saugyklą. Egzistuojantys reliacinių duomenų bazių projektavimo metodai, netinka projektuojant DS. Straipsnyje išryškinti

skirtumai tarp operatyvaus valdymo sistemų ir duomenų saugyklų.

### Susijusių publikacijų analizė

Duomenų saugykla – tai dalykinis (angl. *subject-oriented*), integruotas, nekintantis (angl. *nonvolatile*) ir skirtingas laike (angl. *time-variant*) duomenų rinkinys naudojamas valdymo sprendimų rėmimui (Inmon *et al.* 2005).

DS poreikis atsirado kai prireikė patikimų, konsoliduotų, unikalių ir integruotų atskaitų. DS yra vienas iš perspektyvesnių sprendimų, galintis verslui suteikti strateginę informaciją. Vienas iš pateikiamų duomenų saugyklos apibrėžimų (Ponniah *et al.* 2001) – tai informacinė aplinka (angl. *informational environment*), kuri:

- Suteikia integruota ir bendrą vaizdą apie įmonę.
- Sprendimų priėmimui suteikia galimybę paprastai pasiekti dabartinę (aktualią) ir istorinę informaciją.
- Įgalina atlikti sprendimo priėmimo transakcijas nestabdant produkcinės sistemos.
- Pateikia organizacijos suderintą (angl. *consistent*) informaciją.
- Suteikia lankstų ir interaktyvų šaltinį strateginei informacijai.

DS turi keletą privalumų prieš reliacines duomenų bazes. Pirmiausia duomenys atskirti nuo veikiančios sistemos ir skirti analizės tikslams. Antra, saugomi istorinių duomenų skirtingų lygių apibendrinimai (Seok

*et al.* 2007). Toks apibendrintų istorinių duomenų saugojimas padarė duomenų saugyklų sistemas greitas ir efektyvias analitiniu sistemų požiūriu. Taip pat DS duomenys gali būti surenkami iš skirtingų ir įvairiapusių sistemų – senų informacinių sistemų (IS), reliacinių bazių, XML failų ir pan.

Visos šiuolaikinės duomenų saugyklos suteikia aplinką daugiamatei analizei (angl. *multidimensional analysis*). Versle tai tampa neatskiriama ir pakankamai svarbia informacijos pateikimo dalimi. DS vartotojams leidžia atlikti daugiamatę duomenų analizę, todėl jie gali analizuoti verslą iš įvairių aspektų. Daugiamatės analizės sinonimas būtų OLAP.

OLAP (On-Line Analytical Processing) – tai programinės įrangos technologija (Spofford *et al.* 2006, Thomsen E. *et al.* 2002), leidžianti greitai ir efektyviai peržiūrėti ir analizuoti didelius informacijos kiekius, atlikti prognozavimą. Duomenų atvaizdavimui naudojamas daugiamatis duomenų modelis, atspindintis realų kompanijos vaizdą, kuris yra lengviau suprantamas vartotojui.

OLAP sistemos tikslas yra patenkinti analitiko lūkesčius. Svarbi vartotojo elgsena analizuojant duomenis (Sapia *et al.* 1999), pagal ją galima nustatyti kokius duomenys jam reikalingi. Nagrinėjant analitiko elgseną, analizuojant duomenis, galima nustatyti taisykles, pagal kurias atliekama analizė.

DS yra dinaminės esybės, kurios nuolat vystosi. Laikui bėgant versle atsiranda naujų klausimų į kuriuos DS turi atsakyti. Į kai kuriuos iškilusius naujus klausimus gali atsakyti egzistuojanti DS. O kitas atvejais reikia sukurti naujas DS struktūros schemas, pritaikant DS ir naujiems, ir seniems klausimams atsakyti. Vienas iš šios problemos sprendimo būdų yra perkurti DS iš naujo pritaikant naujiems reikalavimams. Tokiu atveju reikėtų atlikti visą DS kūrimo ciklą: projekto planas, reikalavimų apibrėžimas, projektavimas, konstravimas, diegimas, augimas ir priežiūra (Inmon *et al.* 2005). Toks statinis kūrimo procesas užima daug laiko ir yra komplikotas procesas, todėl sunku reaguoti į pasikeitusius verslo reikalavimus. Šią problemą bandoma spręsti pasitelkiant kitus metodus. Kuriant DS svarbu sukurti tinkama daugiadimensinį struktūros modelį, kuris, pagal tradicinį kūrimo ciklą, kuriamas projektavimo fazėje, kurią sudaro – sudarymas multidimensinio modelio, duomenų transformavimo projektavimas, architektūrinis ir projektavimas ir metaduomenys, informacijos pristatymo projektavimas (Ponniah *et al.* 2001).

Kitos galimos DS projektavimo problemos – sistemos architektūra bus netobula, DS modelis bus

sukurtas netinkamas, t.y. neatsižvelgiant į verslo reikalavimus, taip pat išskiriama, kad negalima tiksliai nustatyti kokius duomenų apibrėžimus naudoti ir koks duomenų modelis leis vartotojams veiksmingai įvykdyti užklausas visos teminėse srityse (Theodoratos *et al.* 1999).

Atlikus analizę paaiškėjo, kad egzistuoja pakankamai daug problemų projektuojant duomenų saugyklas. Literatūros šaltiniuose siūlomi ir nagrinėjami įvairūs duomenų saugyklų projektavimo metodai. Toliau panagrinėsime, kuom DS skiriasi nuo tradicinių duomenų ba-zių valdymo sistemų, taip pat apžvelgsime egzistuojančius DS projektavimo metodus.

### **RDBVS ir DS duomenų struktūros skirtumai**

Operatyvinė duomenų bazės duomenų struktūra dažniausia yra normalizuota ir tokios duomenų bazės yra pritaikytos transakcinėms operacijoms. DS saugomi nenormalizuoti duomenys, kurie dažniausia pritaikyti sudėtingoms užklausoms atlikti.

Faktas (angl. *fact*) – tai kolekcija susijusių duomenų elementų, kuriuos sudaro matavimai ir konteksto duomenys. Pavyzdžiui, tai galėtų būti pardavimai. Matavimas (angl. *measure*) vaizduoja stulpelį, kuriame yra kiekybiniai duomenys, dažniausia skaičiai, kurie gali būti agreguoti. Matavimai paprastai yra atributas faktų lentelėje. Dimensija (angl. *dimension*) yra grupė matavimų susietų pagal tam tikrą hierarchiją, pvz. laiko dimensija - diena, mėnuo ir metai. Dimensija apibrėžia kontekstinį foną faktui.

Dažniausia DS išskiriamos tokios struktūros schemas: plokštės, terasos, žvaigždės, snaigės. Modeliuojant dimensijomis dažniausia naudojami du baziniai modeliai žvaigždės ir snaigės. Žvaigždės modelis yra bazinė struktūra dimensiniam modeliui. Ji turi vieną didelę centrinę lentelę – faktų lentelę- ir aibę mažesnių lentelių – dimensijų – surikiuotų aplink centrinę lentelę. Žvaigždės modelis yra dekompozicija vienos ar daugiau dimensijų.

Nuo tradicinės reliacinės schemas, žvaigždės schema skiriasi dviem dalykais: lentelių pasirinkimu ir būdu kaip konstruojami ryšiai tarp lentelių. Tradiciniu reliacinės schemas ryšiai yra grindžiami loginiu ryšiu tarp šių lentelių, tačiau ypatingo dėmesio neskiriant vienai esybei ar ryšiui. Žvaigždės schema yra labiau suvaržyta, ji grindžiama ryšiu tarp dimensijos lentelės ir fakto lentelės.

## RDBVS ir DS duomenų manipuliavimo kalbų skirtumai

Reliacinėse duomenų bazėse manipuliavimui su duomenimis, dažniausia naudojama SQL užklausų kalba. Norint duomenimis manipuluoti OLAP bazėse reikalinga kalba skirta, manipuliavimui su daugiadimensiniais duomenimis. Viena iš tokių kalbų būtų MDX (Multi-dimensional Expressions) kalba, kuri realizuota Microsoft™ Analysis Services. Naudojant MDX kalbos sintaksę galima kurti daugiamačio kubo objektus, formuoti užklausas ir manipuluoti duomenimis. Analysis Services suteikia galimybę modifikuoti duomenis, t.y. vartotojai gali keisti kubo duomenis, analizės, priklausomai nuo tikslų. Šiuo atveju duomenys tiesiogiai nepakeičiami, tačiau saugomas pats pakeitimo faktas ir užklausos atveju vartotojui pateikiami jau pakeisti duomenys (Spofford *et al.* 2006).

## RDBVS ir DS projektavimo metodų palyginimas

Duomenų saugyklos projektavimas skiriasi nuo tradicinių reliacinių duomenų bazių projektavimo. DS projektavime dažniausia naudojamas dimensinis modeliavimas, kuris leidžia duomenų dubliavimą ir taip pagerina duomenų analizės užklausas.

DS kūrimo metodai skirstomi į šiuos požiūrius: į vartotoją orientuotas DS kūrimas (angl. *User oriented warehouse development*), į operacijas orientuotas DS kūrimas (angl. *Operational oriented warehouse development*) ir į verslo procesus orientuotas DS kūrimas (angl. *Business process oriented warehouse development*). Teisingiausias požiūris yra į verslo procesus orientuotas DS kūrimas, nes čia siūloma pradėti DS kūrimą nuo verslo procesų analizės, tai leidžia sukurti DS atitinkančią aktualius verslo reikalavimus.

Straipsniuose (Husemann *et al.* 2000, Phipps *et al.* 2002) siūlomas duomenų saugyklos kūrimas panašus į tradicinės duomenų bazės kūrimo procesą. Išskiriami tokie kūrimo žingsniai: reikalavimų analizė ir specifikacija, konceptualusis projektavimas (angl. *Conceptual design*), Loginis projektavimas (angl. *Logical design*), fizinis projektavimas (angl. *Physical design*). Reikalavimų analizės žingsnyje nustatomi tikslai ir veikiančiose IS saugomi duomenys, konceptualus projektavimo – faktai, dimensijos, agregacijos, metaduomenys, loginio projektavimo – kuriama loginė DS schema, fizinio projektavimo - kuriamas fizinė DS schema.

Toks kūrimo procesas reikalauja daug laiko išteklių, nes pasikeitus reikalavimams DS, reikia įvykdyti visus

žingsnius – iš naujo nustatyti reikalavimus, parinkti faktus, dimensijas, matavimus, galiausia sukurti DS struktūros modelį. Kadangi šiuolaikinio verslo poreikiai nuolat kinta, priimti sprendimams reikia duomenų vis iš įvairių perspektyvų, todėl dažnai sukurtas DS struktūros modelis nebetenkina iškeltų verslo reikalavimų ir tai būtų patogu DS struktūra keisti dinamiškai pasikeitus verslo reikalavimams. Vienas iš tokių būdų siūlomas straipsnyje (Kaula *et al.* 2009) metodas žvaigždės schemą sujungti verslo taisyklėmis (angl. *business rules*). Susiejimas vykdomas: kiekviena dimensija apibrėžiama verslo taisykle, kiekvienas faktas yra verslo taisyklės konstrukcija, kiekvienas veiksmas remiasi fakto ir dimensijos kombinacija, kuri susijusi su žvaigždės schemos atveju. Apibrėžus žvaigždės modelį verslo taisyklėmis galima greitai reaguoti į kintančius verslo reikalavimus, nes dažniausia verslo taisyklės galima užrašyti įprasta kalba ir transformuoti jas į kitus darinius, kas jau buvo parodyta straipsniuose (Rima *et al.* 2007, Šmaižys *et al.* 2007).

Straipsniuose (Watson *et al.* 2004, Prakash *et al.* 2004) siūloma pirmiausia atsižvelgti į verslo poreikius, o ne į organizacijoje veikiančių informacines sistemas esančius duomenis. Nes gali būti atvejų kai organizacijoje veikiančiose IS nesaugomi ir neapdorjami reikiami duomenys, kurie reikalingi duomenų analizei, todėl gali prireikti modifikuoti versle esančias IS, kad būtų saugomi ir apdorjami reikalingi duomenys analizei.

Straipsnyje (Chowdhary *et al.* 2003) siūloma sudaryti duomenų saugyklos metamodelį, kuris automatizuotų paprastos ir prisitaikančios duomenų saugyklos sukūrimą, taip pat išlaikytų ryšius tarp verslo modelių. Siūloma saugoti du abstrakcijos lygius – loginį, kuriame būtų informacija apie matavimus ir dimensijas DS ir jų sąryšius ir fizinis lygis, kuriame pateikiama fizinė realizacija. Siūlomas projektavimo metodas suteikia galimybę lanksčiai prisitaikyti prie naujų verslo poreikių informacijai.

Lentelėje (1 lentelė) pateiktos pagrindinės savybės, kuriomis duomenų saugykla skiriasi nuo tradicinės duomenų bazės.

**1 Lentelė.** DS ir RDBVS palyginimas.

	<b>RDBVS</b>	<b>DS</b>
Tipas	Orientuota į dalykinę programą	Dalykinė (angl. <i>subject-oriented</i> )
Duomenys	Normalizuoti	Istoriniai, konsoliduoti, denormalizuoti
Taikymo sritys	Verslo procesų įgyvenimui	Analizei, analitikai, sprendimų rėmimo palaikymui

Modeliavimas	ER	Dimensinis modeliavimas
Paskirtis	Optimizuota tranzakcijoms	Optimizuota sudėtingoms užklausoms
Manipuliavimo kalbos	struktūrizuota užklausų kalba (pvz. SQL)	Multidimensinių užklausų kalba (pvz. MDX)
Atsako laikas	Labai svarbus	nėra kritiškai svarbus
Naudotojai	operatoriai	Analitikai, strategai sprendimų priėmėjai

Atlikus tyrimą paaiškėjo, kad DS skiriasi nuo tradicinių duomenų bazių valdymo sistemų. Pagrindinis skirtumas tas, kad duomenys DS saugomi denormalizuoti. Tai pat svarbu atsižvelgti į tai, kad skiriasi taikymo sritys, saugomi duomenys, naudotojai ir kt., tai lemia, kad tradicinių bazių kūrimo metodai netinka kurti DS, todėl tyrinėjami nauji metodai. Šie metodai, kaip parodė tyrimas šiuo metu nėra pakankamai tobuli.

### Išvados

Tradicinis duomenų bazių kūrimo metodai netinka kurti saugykloms, nes DS pagal savo paskirtį ir kitomis savybėmis skiriasi nuo operatyvaus valdymo duomenų bazių. Kuriant duomenų saugyklos modelį, dažniausia pradedama nagrinėti veikiančiose informacinėse sistemose saugomi duomenys, pagal kuriuos ir modeliujama duomenų saugykla.

DS skiriasi nuo tradicinių RDBVS – savo struktūra, taikymo sritimi, paskirtimi, todėl DS kūrimo ciklas skiriasi nuo tradicinių duomenų bazių kūrimo ciklo. Be to, DS projektavimas dažniausia yra neautomatizuotas darbas, reikalaujantis daug laiko sąnaudų.

Kadangi verslas nuolat kinta ir tokiam kintančiam verslui nuolat reikia priiminėti sprendimus. Naujiems sprendimams priimti kartais reikia visiškai kitos informacijos, net jei yra sukurtoje DS. Toks naujos informacijos poreikis priverčia perprojektuoti DS, todėl yra siūlomi įvairūs projektavimo metodai, kurie leistų greitai modifikuoti esamą duomenų saugyklą.

Egzistuojantys DS projektavimo metodai yra dar nepakankamai lankstūs ir tobuli. Dažniausia pradedama nagrinėti nuo duomenų, saugomų organizacijoje veikiančiose informacinėse sistemose ir į verslo reikalavimus atsižvelgiama tik atlikus tokių duomenų analizę. Todėl nėra patenkinami verslo reikalavimai norimai gauti informacijai iš duomenų saugyklos ir tokios saugyklos tampa netinkamos priiminėti sprendimus, nes negali pateikti visos reikiamos verslui informacijos.

Tolimesniuose darbuose bus pasiūlytas metodas, skirtas duomenų saugyklos projektavimui, kuris remtųsi

trijų lygmenų karkasu, kuris pasiūlytų (Vasilecas *et al.* 2006, Vasilecas *et al.* 2007) straipsniuose. Metodas pasiūlo pirmiausia atsižvelgti į verslo poreikius ir reikalavimus ir tik tolimesniuose žingsniuose atsižvelgti į duomenų šaltinius.

### Literatūra

- Chowdhary P.; Bhaskaran K.; Caswell N. S. ir kt. 2006. Model Driven Development for Business Performance Management. *IBM Systems Journal*, Volume 45 , Issue 3 (July 2006): 587 – 605.
- Husemann B.; Lechtenböcker J.; Vossen G. 2000. Conceptual Data Warehouse Design. In Proc. of the International Workshop on *Design and Management of Data Warehouses (DMDW 2000)*: 3-9.
- Inmon W. H. 2005. *Building the Data Warehouse*, 3th Edition, Wiley Publishing, Inc., Indianapolis: 412.
- Kaula R. 2009. Business Rules for Data Warehouse. *International Journal of Information Technology*, Volume 5 Number 1 Winter: 58-66.
- Phipps C.; Davis K. C. 2002. Automating Data Warehouse Conceptual Schema Design and Evaluation. In Proc. of 4th Int. *Workshop on Design and Management of Data Warehouses*, volume 58, CEUR-WS.org: 23—32.
- Ponniiah P. 2001. *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*. John Wiley & Sons: 516.
- Prakash N.; Gosain A. 2003. Requirements Driven Data Warehouse Development. Proc. in *CAiSE Short Paper*: 1-4.
- Rima A.; Šmaižys A.; Vasilecas O. 2007. Intelektualizuota duomenų analizė verslo taisyklių transformacijų pagrindu. *Informacinės technologijos 2007*, Kaunas: Technologija: 202-206.
- Sapia C. 1999. On Modeling and Predicting Query Behavior in OLAP Systems. Proc. of the International Workshop on Design and Management of Data Warehouse, Germany: 2-10.
- Seok Kp. I.; Abdullaev S.R. 2007. A Study on Successful Business Intelligence Systems in Practice. Lecture Notes In *Computer Science*; Vol. 4490. Proc. of the 7th international conference on Computational Science, Part IV: ICCS 2007 Beijing, China: 729 – 732.
- Spofford G., Harinath S., Webb C., Hai Huang D., Ci-Vardi F. 2006. MDX Solutions Second Edition With Microsoft® SQLServer™ Analysis Services 2005 and Hyperion® Essbase. Wiley Publishing: 716,
- Šmaižys A.; Rima A.; Vasilecas O. 2007 Verslo taisyklių panaudojimas duomenų analizės metamodelių transformacijų pagrindu. 10-osios Lietuvos jaunųjų mokslininkų konferencijos "Mokslas – Lietuvos ateitis" pranešimų rinkinys, Vilnius: Technika: 2-8.
- Theodoratos D.; Sellis T. 1999. Dynamic Data Warehouse Design. *Data warehousing and knowledge discovery*. International conference No1, Florence , ITALIE (30/08/1999), vol. 1676: 1-10.
- Thomsen E. 2002. *OLAP Solutions Building Multidimensional Information Systems*. Second Edition. John Wiley & Sons, Inc., 2002: 661.
- Vasilecas O.; Smaizys A. 2006. The framework: an approach to support business rule based data analysis. In O. Vasilecas et al. (eds.), Proc. of 7th *International IEEE Baltic Conference*

on *Databases and Information Systems*, July 3, Vilnius, Technika: 141 – 147.

Vasilecas O.; Smaizys A. 2007 The Framework for Business Rule Based Software Modeling: An Approach for Data Analysis Models Integration. *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, V 155, IOS Press, 2007, ISSN 0922-6389:175-188.

Watson H. J.; Fuller C.; Ariyachandrat T. 2004. Data warehouse governance: best practices at Blue Cross and Blue Shield of North Carolina. *Decision Support Systems*, Volume 38 , Issue 3 (December 2004): 435 – 450.

## ANALYSIS OF DECISION SUPPORT SYSTEMS DATA WAREHOUSE DESIGN METHODS

**A. Rima, A. Šmaizys, O. Vasilecas**

### S u m m a r y

The paper introduces to the existing data warehouse design methods, their advantage and disadvantage. Traditional database systems development methods are not dedicated for data warehouse development because data warehouse is different from operational management database systems. Currently, the data warehouse development process usually begins from analysis of data stored in business information systems. This development way can not fully meet business requirements for data warehouse. Data warehouse development process should begin from business process analysis. That approach allows meet business process requirements for data analysis and data warehouse.