RĪGAS TEHNISKĀ UNIVERSITĀTE

Jānis Kampars

PIEPRASĪJUMA DATU INTEGRĀCIJAS RISINĀJUMI ATTĀLIEM DATU AVOTIEM

Promocijas darba kopsavilkums

Rīga 2011

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūts

Jānis Kampars
Doktora studiju programmas „Vadības informācijas tehnoloģija" doktorants

PIEPRASĪJUMA DATU INTEGRĀCIJAS RISINĀJUMI ATTĀLIEM DATU AVOTIEM

Promocijas darba kopsavilkums

Zinātniskais vadītājs
Dr. sc. ing., profesors
J. GRABIS

Rīga 2011

UDK 004.78(043.2)
Ka 354 p

Kampars J. Pieprasījuma datu integrācijas risinājumi attāliem datu avotiem. Promocijas darba kopsavilkums.-R.:RTU, 2011.-32 lpp.

Iespiests saskaņā ar RTU Informācijas tehnoloģijas institūta 2011. gada 6. jūlija Padomes sēdes lēmumu, protokols Nr.11-07.

Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā „Atbalsts RTU doktora studiju īstenošanai”.

ISBN 978-9934-10-237-0

PROMOCIJAS DARBS
IZVIRZĪTS DOKTORA GRĀDA IEGŪŠANAI
RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ

Promocijas darbs inženierzinātņu doktora grāda iegūšanai tiek publiski aizstāvēts 2011.g. 14. decembrī plkst. 14:30 Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultātē, Meža ielā 1/3, 202. auditorijā.

OFICIĀLIE RECENZENTI:

Profesors, Dr.habil.sc.ing. Leonīds Novickis
Rīgas Tehniskā universitāte, Latvija

Profesors, Dr.habil.sc.ing. Pēteris Rivža
Latvijas Lauksaimniecības universitāte, Latvija

Profesors, Dr.sc.ing. Enn Õunapuu
Tallinas Tehnoloģiju universitāte, Igaunija

APSTIPRINĀJUMS

Apstiprinu, ka esmu izstrādājis doto promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē inženierzinātņu doktora grāda iegūšanai. Promocijas darbs nav iesniegts nevienā citā universitātē zinātniskā grāda iegūšanai.

Jānis Kampars …….(Paraksts)

Datums: ……

Promocijas darbs ir uzrakstīts latviešu valodā, satur ievadu, 5 nodaļas, secinājumus, literatūras sarakstu, 5 pielikumus, 26 tabulas, 75 attēlus, kopā 165 lappuses. Bibliogrāfiskajā sarakstā ir 174 nosaukumi.

SATURS

1. VISPĀRĒJS DARBA RAKSTUROJUMS

Promocijas darbā tiek pētīti risinājumi attālu, heterogēnu avotu pieprasījuma datu integrācijai. Nodaļa ietver tēmas aktualitāti, darba mērķi un uzdevumus, kā arī izmantotās pētījumu metodes. Tajā tiek aprakstīti arī galvenie rezultāti, zinātniskā novitāte un praktiskais pielietojums, aprobācija un sniegts ieskats darba struktūrā.

1.1. Tēmas aktualitāte

Uzņēmumi ikdienā pieņem dažāda veida lēmums. Lēmumpieņemšanai ir nepieciešami dati, kas parasti ir atrodami dažādos datu avotos. Datu kombinēšana no dažādiem avotiem un vienota saprotama apvienoto datu skata iegūšana tiek saukta par datu integrāciju [4]. Šajā darbā tiek pētīta biznesa intelekta datu integrācija [9], kurā tiek veikta datu savākšana un pārveidošana analīzei piemērotā formā. Datu pieejamība ir būtisks faktors lēmumpieņemšanai [8, 12, 20, 25].

Tradicionālā datu integrācijas scenārijā dati tiek glabāti lokālajās sistēmās. Šajā gadījumā tipiskie avoti ir uzņēmuma kontrolētas datubāžu vadības sistēmas, izplātās datnes un izklājlapas, savukārt datu izgūšanu no datu avotiem un konsolidēšanu veic ETL (*Extract, Transform, Load*) sistēmas.

Datu glabāšanas tehnoloģiju relatīvi zemās izmaksas un intertīkla augstā pieejamība ir ļāvusi dažādām organizācijām gadu laikā uzkrāt milzīgus datu apjomos un padarīt tos pieejamus citiem [26], līdz ar to pastāv lielas ārējo datu avotu izvēles iespējas. Tradicionāli ārējie dati tiek pārkopēti pagaidu centralizētā bāzē, tomēr šāda pieeja ir sarežģīti realizējama [27].

Alternatīva pieeja ir „Dati kā pakalpojums" (*Data as a Service* jeb DaaS) [5, 23, 24], kurā datu avoti ir attālas tīmekļa pakalpes, kas nodrošina datu izgūšanas un apstrādes funkcionalitāti. Izmantojot tīmekļa pakalpes, ir iespējams izgūt tieši nepieciešamos datus tad, kad tas ir nepieciešams, un nav nepieciešama ārējo avotu datu pilnas kopijas glabāšanas lokāli. Pakalpes saskarnes piedāvātās metodes darbā tiek sauktas par datu izgūšanas operācijām, jo to piedāvātās metodes atgriež datus. DaaS pieejas rezultātā iegūtajam datu integrācijas risinājumam ir dalīta arhitektūra, bet pats risinājums ir uzskatāms par dalītu sistēmu.

DaaS bāzēta datu integrācijas risinājuma izstrāde ir sarežģīta, jo:

- tradicionālie datu integrācijas un lietojumprogrammatūras integrācijas risinājumi nav piemēroti DaaS bāzētai datu integrācijai [3, 11];
- eksistē liela protokolu un standartu dažādība [14, 15, 16, 21] tīmekļa pakalpju jomā;
- pārsvarā dati tiek pārsūtīti tekstuālā, daļēji strukturētā formātā, tomēr var tikt izmantoti arī bināri dati [6, 16, 28];
- datu avotu (tīmekļa pakalpju) interfeisi ir dinamiski mainīgi [24];
- ir nepieciešams veikt slodzes līdzsvarošanu un pēckļūdu atkopšanu;
- ir jāņem vērā pakalpes kvalitātes rādītāji [2, 18, 22], tomēr pakalpes uzturētāja sniegtā informācija var neatbilst reālajai situācijai [10];
- ir problemātiski veikt datu avotu automatizētu meklēšanu pēc funkcionālajām un nefunkcionālajām prasībām [17, 18, 22, 27];
- ir neskaidri pakalpju un to datu izmantošanas legālie aspekti [23].

1.2. Darba mērķis un uzdevumi

Promocijas darba mērķis ir izstrādāt pieprasījuma datu integrācijas risinājumus, kas nodrošinātu datu izgūšanu no attāliem, heterogēniem datu avotiem un transformētu datus nepieciešamajā formā. Mērķa sasniegšanai tiek definēti uzdevumi:

1. Identificēt heterogēnu, attālu avotu datu integrācijas problēmas.
2. Sintezēt vispārīgu heterogēnu, attālu avotu datu integrācijas modeli.
3. Definēt prasības heterogēnu, attālu avotu datu integrācijas arhitektūrai.
4. Balstoties uz definētajām prasībām un modeli, izstrādāt attālu avotu datu integrācijas arhitektūru.
5. Izstrādāt arhitektūras prototipu.
6. Novērtēt arhitektūrā izmantotos risinājumus.
7. Izpētīt izstrādātās arhitektūras izmantošanas iespējas.

Pētījuma objekts ir datu integrācijas risinājumi attāliem, heterogēniem datu avotiem, bet **pētījuma priekšmets** – attālu avotu pieprasījuma datu integrācijas sistēmas arhitektūra, tās izstrāde un izmantošana.

1.3. Pētījumu metodes

Darbā tiek veikta literatūras analīze, lai pētītu risinājumus datu integrācijas jomā un identificētu prasības attālu avotu datu integrācijas sistēmai. Datu integrācijas sistēmas augsta līmeņa abstrakcija tiek definēta izmantojot kopu teoriju, bet detalizēts arhitektūras modelis tiek aprakstīts ar UML (*Unified Modeling Language*) [1] diagrammu palīdzību. Izmantojot sintēzi, modelī tiek apvienoti dažādi elementi – datu integrācijas process, slodzes līdzsvarošana, vēlā saistīšana, pakalpes kvalitātes rādītāju kontrole, pēckļūdu atkopšana. Datu apstrāde tiek balstīta uz XML un ar to saistītajām specifikācijām. Izmantojot programmatūras inženieriju, tiek izstrādāts uz .NET ietvaru bāzēts arhitektūras prototips. Lai eksperimentāli novērtētu izstrādātos risinājumus, tiek izmantota eksperimentu plānošana [7]. Regresijas analīze tiek izmantota eksperimentālo rezultātu novērtēšanā.

1.4. Galvenais rezultāts un zinātniskā novitāte

Promocijas darba galvenais rezultāts ir attālu avotu pieprasījuma datu integrācijas sistēmas modeļa un arhitektūras izstrāde. Tā svarīgākais zinātniskais ieguldījums ir:

1. Jaunas attālu avotu datu integrācijas metodes izstrāde, kas balstās uz abstrakcijas izmantošanu tīmekļa pakalpju metožu līmenī un datu integrācijas procesa nodalīšanu no tīmekļa pakalpju piekļuves loģikas.
2. Uz nefunkcionālām un funkcionālām prasībām balstīta adaptīva tīmekļa pakalpju izvēles un slodzes līdzsvarošanas algoritma izstrāde.
3. Algoritma, kas nodrošina attālu avotu datu integrācijas procesu veidojošo uzdevumu pareizu un savlaicīgu izpildi, izstrāde.
4. Izstrādāto risinājumu efektivitātes novērtējums.

Ir izstrādāts individuālo datu integrācijas uzdevumu paralelizācijas un savstarpējo atkarību noteikšanas algoritms, vēlās saistīšanas un slodzes līdzsvarošanas algoritms, pēckļūdu atkopšanas loģika. Darbā definētajā arhitektūrā ir panākta pilnīga datu integrācijas procesa loģikas nodalīšana no tīmekļa pakalpju piekļuves loģikas.

Ir pierādīta darbā definētā datu integrācijas uzdevumu paralelizācijas algoritma efektivitāte, apstiprināts pieņēmums par slodzes līdzsvarošanas nozīmīgumu un definēts, kādos gadījumos izmantot attālus avotus ir efektīvāk nekā lokālus.

1.5. Darba praktiskais pielietojums

Lai pierādītu darbā definētās arhitektūras implementējamību, tika izstrādāts sistēmas prototips. Darba izstrādes laikā iegūti rezultāti ir tikuši izmantoti kurjerpasta sūtījumu plānošanas problēmas risināšanā uzņēmumā „Latvijas Pasts" (pētījumi tika veikti projekta „Globālās pozicionēšanas tehnoloģiju integrācija uzņēmuma informācijas sistēmā: Izmantošana Latvijas transporta plānošanas problēmu risināšanā" ietvaros). Izstrādātais sistēmas prototips ir ticis modificēts uzņēmuma „BalticTaxi" vajadzībām, lai risinātu pasažieru pārvadājumu plānošanas problēmu uzņēmumā. Prototips tika izmantots arī objekta izvietošanas problēmas risināšanā.

Darbā izstrādātā arhitektūra ir izmantojama, lai nodrošinātu nepieciešamo datu izgūšanu no attāliem, heterogēniem datu avotiem jeb tīmekļa pakalpēm, datu transformēšanu un agregēšanu.

1.6. Aprobācija

Promocijas darbā veikto pētījumu rezultāti ir atspoguļoti astoņās publikācijās:

1. Bonders M., Grabis J., Kampars J. Web Service Selection: Beyond Quality of Service// 9th Conference on Databases and Information Systems (DB&IS). - Latvia, Riga: University of Latvia, 2010. - p.125-137.
2. Grabis J., Bonders M., Kampars J. Combining Functional and Nonfunctional Attributes for Cost Driven Web Service Selection Frontiers in Artificial Intelligence and Applications// Databases and Information Systems: Selected Papers from the Ninth International Baltic Conference, Db&is 2010. - 2011. - p.227-239.
3. Grabis J., Kampars J., Bonders M. A Methodology for Integration of Spatial Data in Enterprise Applications// International Business Information Management Conference (7th IBIMA). - Brescia, Italy: IBIMA, 2006. - p.169-175.
4. Kampars J. Globālās pozicionēšanas datu integrēšana uzņēmuma informācijas sistēmā// a/s DATI Exigen Group informācijas tehnoloģijas speciālistu un Latvijas universitāšu datorzinātņu studentu XI konference. - Rīga: DATI Exigen Group, 2006. - p.51-56.

5. Kampars J. New Generation Enterprise Geographic Information Systems// 7.starptautiskā konference "Vide.Tehnoloģija.Resursi". - Rēzekne: Rēzeknes Augstskola, 2009. - p.235-240.
6. Kampars J., Grabis J. Development of adapter for connecting GPS/GIS and ERP systems// RTU zinātniskie raksti, Datorzinātne. - 2006. - p.51-56.
7. Kampars J., Grabis J. Enterprise Application Integration problems, approaches and standards// RTU zinātniskie raksti, Datorzinātne. - 2007. - p.77-83.
8. Kampars J., Grabis J. Spatial Data Integration Approach with Application in Facility Location// 16th International Conference on Information and Software Technologies. - Kaunas, Lithuania: Kaunas University of Technology, 2010. - p.117-125.

Pētījumos iegūtie rezultāti tika prezentēti sešās konferencēs:

1. RTU 47. starptautiskās zinātniskā konference, 2006. gada 13. oktobris, Rīga, Latvija.
2. International Business Information Management Conference (7th IBIMA), 2006. gada 14-16 Decembris, Brescia, Itālija.
3. RTU 48. starptautiskās zinātniskā konference, 2007. gada 12. oktobris, Rīga, Latvija.
4. Starptautiskā zinātniski praktiskā konference "Vide. Tehnoloģija. Resursi" , 2009.gada 25-27 jūnijs Rēzekne, Latvija.
5. RTU 50. starptautiskās zinātniskās konference, 2009. gada 16. oktobrī Rīga, Latvija.
6. 16th International Conference on Information and Software Technologies, 2010. gada 21.-23. aprīlis, Kauņa, Lietuva.

Promocijas darba zinātniskie un praktiskie rezultāti tika izmantoti 2006. gada pētnieciskajā projektā „Globālās pozicionēšanas tehnoloģiju integrācija uzņēmuma informācijas sistēmā: Izmantošana Latvijas transporta plānošanas problēmu risināšanā".

1.7. Darba struktūra un apjoms

Promocijas darbs satur ievadu, 5 nodaļas, secinājumus, 5 pielikumus un literatūras sarakstu.

Ievadā tiek pamatota promocijas darbā pētītās problēmas aktualitāte, formulēts darba mērķis un definēti uzdevumi. Tiek aprakstītas pētījumu metodes, darba zinātniskais jaunieguvums, praktiskā nozīme un aprobācija.

Pirmajā nodaļā tiek identificētas galvenās problēmas, kas sarežģī attālu avotu datu integrāciju. Tiek pretnostatīts tradicionāls un uz DaaS pieeju balstīts datu integrācijas scenārijs. Ir analizētas pieejas un risinājumi attālu avotu datu integrācijas jomā.

Otrajā nodaļā, balstoties uz pirmajā nodaļā identificētajām problēmām, tiek definēts datu integrācijas sistēmas modelis. Iegūtais modelis ir no platformas un konkrētām tehnoloģijām neatkarīgs.

Trešajā nodaļā tiek definētas prasības tehniskajai arhitektūrai un definēta modelim atbilstoša arhitektūra. Tiek izvērsti aprakstītas arhitektūras komponentes, to mijiedarbība un darbības princips.

Ceturtajā nodaļā tiek salīdzināta DaaS bāzēta pieeja ar lokālu datu glabāšanu. Lai novērtētu arhitektūrā iekļauto risinājumu efektivitāti, tiek izstrādāts arhitektūras prototips. Tas tiek salīdzināts ar komerciālu ETL sistēmu un speciāli izstrādātu secīgu datu integrācijas risinājumu. Tiek novērtēta slodzes līdzsvarošanas ietekme uz datu integrācijas laiku.

Piektajā nodaļā ir aplūkoti divi izmantošanas gadījumi. Darbā definētā arhitektūra tiek izmantota, lai risinātu pasažieru pārvadājumu plānošanas un objekta izvietošanas problēmas.

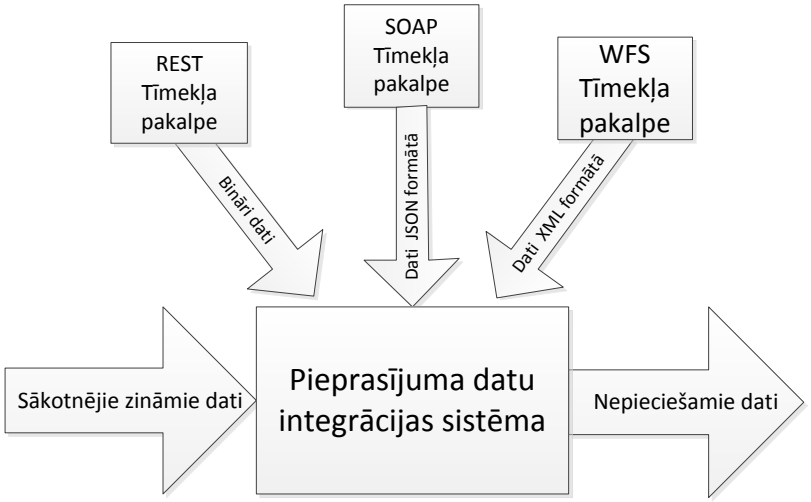
Darba noslēgumā tiek apkopoti promocijas darba izstrādes laikā iegūtie rezultāti, secinājumi un definēti tālākie pētījumu virzieni.

Promocijas darbam ir 5 pielikumi:

1. pielikums - datubāzes modelis;
2. pielikums - lietotāja saskarne pasažieru pārvadājumu plānošanas izmantošanas gadījumam;
3. pielikums - datu integrācijas procesa loģika pasažieru pārvadājumu plānošanas izmantošanas gadījumam;
4. pielikums - lietotāja saskarne objekta izvietošanas izmantošanas gadījumam;
5. pielikums - datu integrācijas procesa loģika objekta izvietošanas izmantošanas gadījumam.

2. Promocijas darba nodaļu īss izklāsts

Nodaļā ir dots promocijas darba satura izklāsts, galveno vērību pievēršot pētījuma rezultātiem un secinājumiem. Promocijas darbā ir pētīta heterogēnu, attālu avotu datu integrācijas problēma, kas shematiski ir parādīta 2.1. attēlā.



2.1. att. Attālu avotu datu integrācijas problēma

Uzsākot datu integrācijas problēmas risināšanu, lēmumpieņēmējs ir definējis nepieciešamos datus, ir pieejami sākotnēji zināmie dati un attāli, heterogēni datu avoti. Pieprasījumu datu integrācijas sistēma veic datu izgūšanu, apvienošanu un transformēšanu, rezultātā iegūstot nepieciešamos datus. Datu integrācijas laikā ir jāņem vērā avotu heterogenitāte jeb atšķirīgie izmantotie komunikācijas protokoli, datu formāti, modeļi un nodrošinātā funkcionalitāte. Lai gan datu avoti ir heterogēni no tehniskā viedokļa, tie var būt funkcionāli vienādi un nodrošināt vienu un to pašu informāciju.

2.1. Esošās situācijas analīze un galveno problēmu identificēšana

Veicot literatūras analīzi, promocijas darba 1. nodaļā tiek identificētas DaaS pieejas atšķirības no tradicionāli izmantotās lokālās datu glabāšanas. Nodaļas mērķis ir identificēt būtiskākās problēmas, kas kavē veiksmīgu DaaS bāzētu datu integrācijas risinājumu izstrādi.

Svarīgākās DaaS bāzētu datu integrācijas risinājumu atšķirības no lokālu datu integrācijas risinājumiem apkopotas 2.1. tabulā.

Datu izgūšana no DaaS bāzētiem avotiem ir komplicētāka nekā no lokālām sistēmām. Galvenās problēmas ir datu avotu heterogenitāte un mainība [24], atbilstošu risinājumu un metodoloģiskā atbalsta trūkums.

Salīdzinoši maz pētījumi ir veikti dalītu datu integrācijas jomā, apskatot gadījumus, kuros datu avoti ir attālas, autonomas, heterogēnas tīmekļa pakalpes. Pie tam daži pētījumi balstās uz pieņēmumu, ka ir iespējams piemērot tīmekļa pakalpes datu integrācijas risinājuma vajadzībām, kas vairumā gadījumu, izmantojot ārējos datu avotus, nav iespējams. Netiek

pievērsta pietiekama uzmanība individuālo datu integrācijas uzdevumu maksimāli savlaicīgai izpildei, slodzes līdzsvarošanai, kopējā datu integrācijas laika minimizēšanai, nefunkcionālo prasību ievērošanai un atkārtotas izmantošanas veicināšanai.

2.1. tabula

Lokālu datu izmantošanas pretnostatījums DaaS pieejai

Lokālie dati	**DaaS**
iespējams piemērot datu avotu uzņēmuma vajadzībām	nav iespējams piemērot datu avotu uzņēmuma vajadzībām
relatīvi statiski datu avoti	dinamiski mainīgi datu avoti [24]
pārsvarā strukturēti dati	pārsvarā daļēji strukturēti dati
datu pārsūtīšana lokālajā tīklā, intertīkla pieslēguma nozīme ir sekundāra	datu pārsūtīšana intertīklā, svarīgs intertīkla pieslēguma ātrums un stabilitāte
risks saskarties ar kļūdu datu integrācijas procesa laikā vērtējams kā zems	risks saskarties ar kļūdu datu integrācijas procesa laikā vērtējams kā augsts
licencēšanas nosacījumi ir skaidri zināmi	neskaidri licencēšanas nosacījumi [23]
potenciālie datu avoti, to datu modeļi ir labi zināmi	avoti var būt iepriekš nezināmi, to meklēšana problemātiska [17, 18, 22, 27], metadati var būt nepilnīgi [10]
maz ticama izvēles iespēja starp vairākiem avotiem	liela vairāku alternatīvu datu avotu pieejamības varbūtība, jāņem vērā kvalitātes rādītāji [2, 18, 22]
veiktspējas palielināšana tiek realizēta datu avotā, var izmantot praktiski jebkādus slodzes līdzsvarošanas algoritmus	veiktspējas palielināšana tiek realizēta datu integrācijas risinājumā, var izmantot tikai centralizētus slodzes līdzsvarošanas risinājumus
ETL sistēmas veiksmīgi risina avotu heterogenitātes problēmas, pieejami arī dažādi adapteri	tradicionālie datu integrācijas un lietojumprogrammatūras integrācijas risinājumi nav piemēroti [11, 27]

2.2. Attālu avotu pieprasījuma datu integrācijas sistēmas modelis

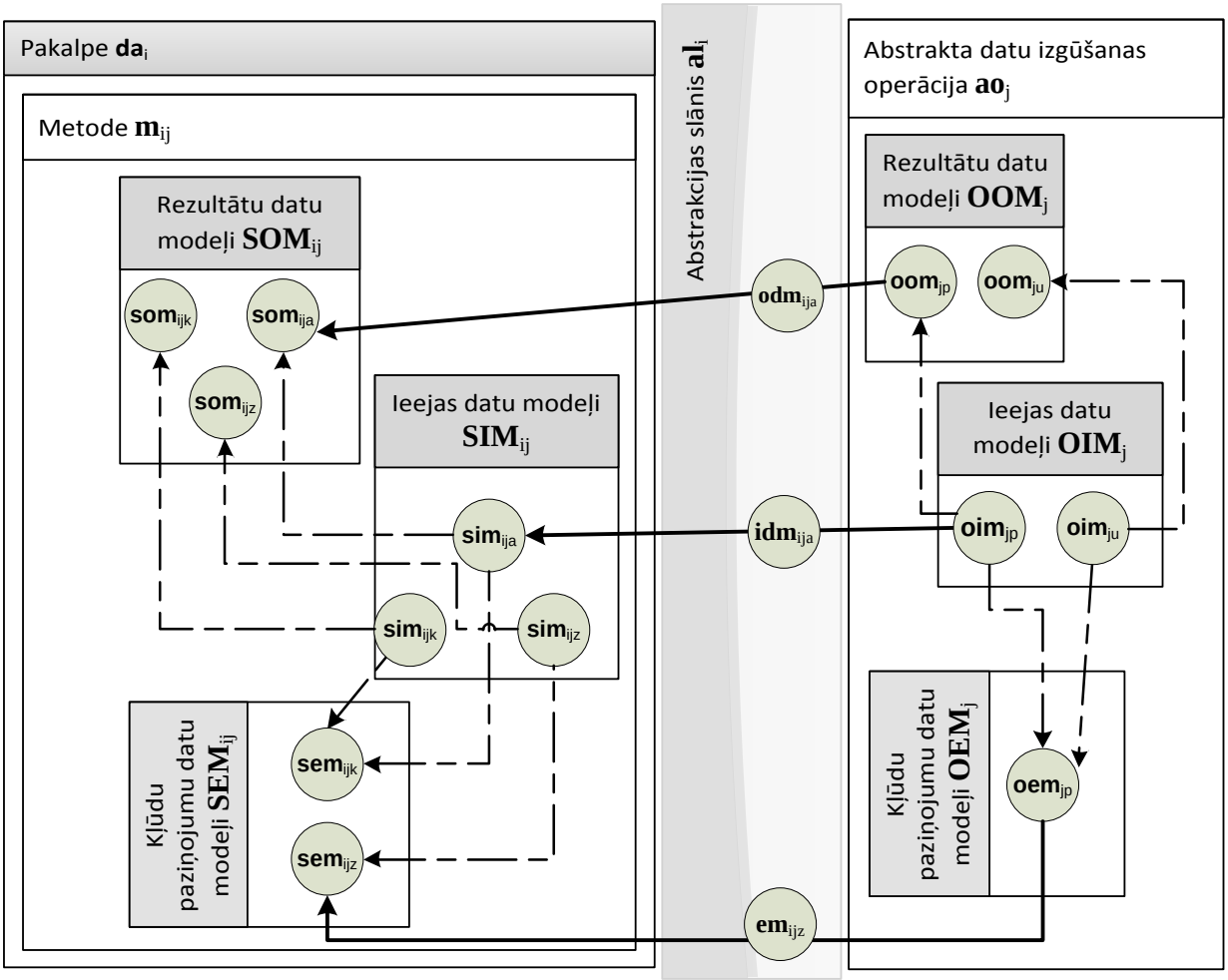
Lai risinātu identificētās pieprasījuma datu integrācijas problēmas, 2. nodaļā ir izstrādāts datu integrācijas sistēmas modelis. Tas vispārīgā veidā parāda, kā no heterogēniem, attāliem datu avotiem iegūt nepieciešamos datus. Modeļa vispārīgā forma ir:

IM={**DA**, **AO**, **AL**, **ID**, **OD**, **P**, **Q**, **LB**}, kur

DA ir datu avoti jeb attālas tīmekļa pakalpes, **AO** ir abstraktas datu izgūšanas operācijas, **AL** ir abstrakcijas slāņi datu avotiem, **ID** ir sākotnējie dati, **OD** ir nepieciešamie dati, **P** ir datu

integrācijas process, **Q** ir pakalpju un to datu kvalitātes rādītāji, **LB** ir vēlā saistīšana (*late binding*) un slodzes līdzsvarošana.

Datu integrācijas sistēmas modeļa mērķis ir atrisināt datu integrācijas uzdevumu, minimizējot kopējo datu integrācijas laiku un vienkāršojot integrācijas sistēmas uzturēšanu un modificēšanu. Izstrādājot datu integrācijas sistēmas modeli, tiek ņemts vērā tas, ka būtiski ir nodalīt datu integrācijas procesa loģiku no datu avotu piekļuves loģikas. Šim nolūkam tiek izmantota abstrakcijas pieeja – tiek definētas abstraktas datu izgūšanas operācijas (**ao**) un to atbilstošie ieejas (**oim**), rezultējošo (**oom**) un kļūdu paziņojumu (**oem**) datu modeļi. Tīmekļa pakalpes (**da**) metodes (**m**) tiek kartētas uz noteiktām abstraktām datu izgūšanas operācijām abstrakcijas slānī (**al**). Abstrakcijas slāņa darbības princips ir parādīts 2.2. attēlā.



2.2. att. Abstrakcijas slāņa darbības princips

Piemērā redzams, ka abstrakcijas slānis nodrošina tīmekļa pakalpes metodes atsevišķu ieejas (**sim**), rezultējošo (**som**) un kļūdu paziņojumu (**sem**) datu modeļu kartēšanu uz atbilstošajiem abstraktās datu izgūšanas operācijas datu modeļiem.

Dati tiek izgūti no dažādiem avotiem, un individuālo datu integrācijas uzdevumu atkarības un korektu izpildes secību definē datu integrācijas process **P**. Tas satur darbību

kopumu, kas jāizpilda, lai no sākotnējiem datiem iteratīvi iegūtu nepieciešamos datus, šo darbību korektu izpildes secību un savstarpējās atkarības, kā arī pagaidu datu kopu. Procesa sākumā pagaidu datu kopā tiek pārkopēti sākotnējie dati, kas tiek izmantoti nepieciešamo datu izgūšanai. Procesa beigās pagaidu datu kopa sakrīt ar nepieciešamajiem datiem.
Darbā ir definētas piecu veidu darbības, kas tiek izmantotas datu integrācijas procesa loģikas realizācijai:

- abstraktu datu izgūšanas operāciju izpilde;
- transformāciju veikšana;
- ciklu veidošana;
- nosacījuma izteiksmju pārbaude;
- datu agregācija.

Datu integrācijas procesa laikā notiek abstraktai datu izgūšanas operācijai atbilstošas fiziskas pakalpes meklēšana un tās metodes izpilde caur pakalpes abstrakcijas slāni. Meklēšanas procesa laikā ir jāņem vērā abstraktās datu izgūšanas operācijas ieejas datu modelis, jo ne visas pakalpju metodes, kas ir kartētas uz nepieciešamo abstrakto datu izgūšanas operāciju, atbalsta visus tās ieejas datu modeļus. Ja ir vairākas alternatīvas tīmekļa pakalpes, ir jāņem vērā lietotāja nefunkcionālās prasības, ko veido minimālie pieļaujamie tīmekļa pakalpes kvalitātes rādītāji, kvalitātes rādītāju svari un slodzes līdzsvarošanai izmantojamo pakalpju maksimālais skaits vienas datu izgūšanas operācijas ietvaros. Veicot izsaukumus tīmekļa pakalpēm, tiek atjaunoti to kvalitātes dati.

Iegūtais modelis ir augsta līmeņa abstrakcija un nav piesaistīts noteiktai platformai, programmēšanas valodai vai tehnoloģijai. Modelī tiek definētas būtiskākās datu integrācijas risinājuma sastāvdaļas un to struktūra, kas nodrošina efektīvu attālu avotu datu integrāciju.

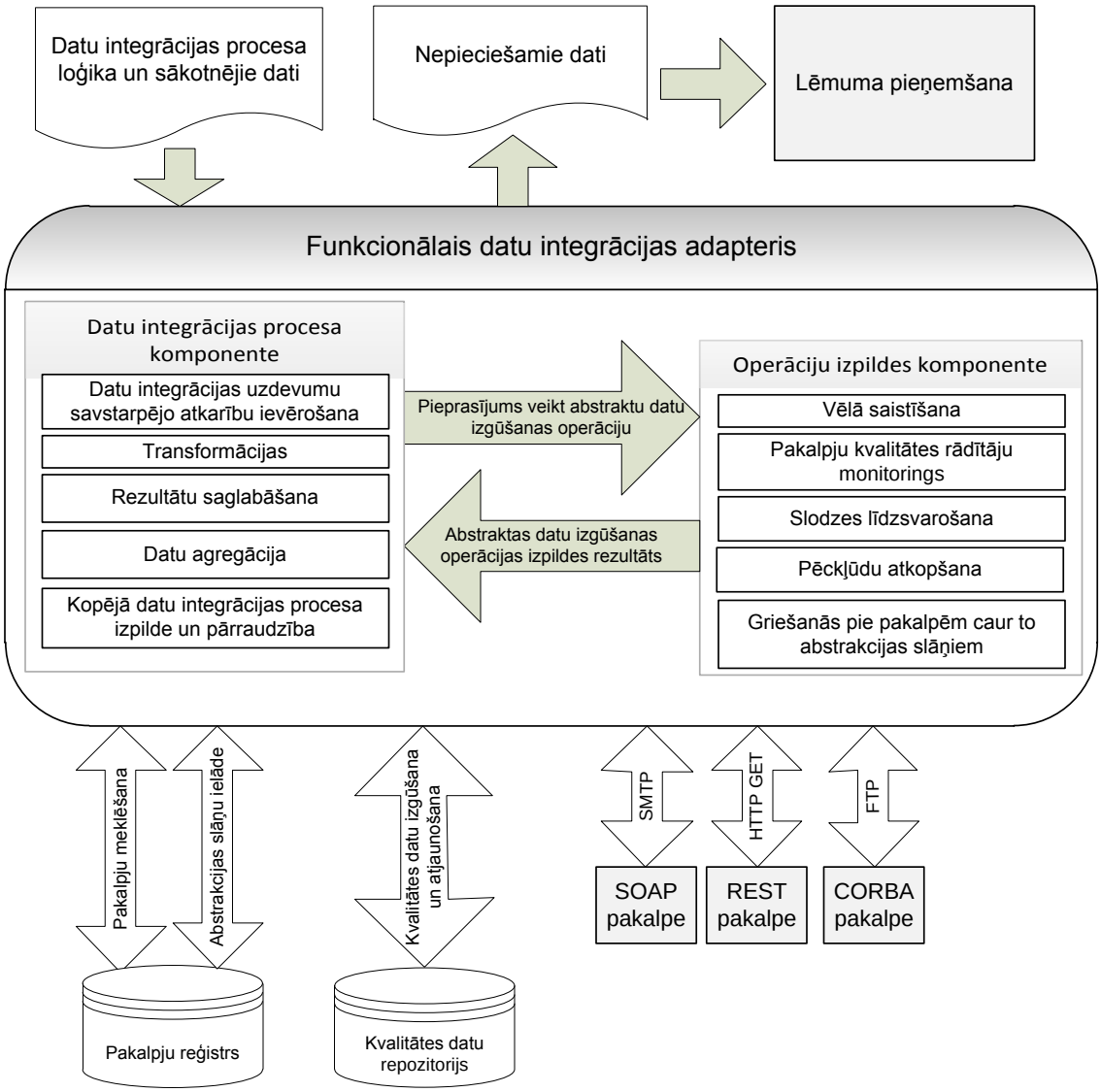
2.3. Attālu avotu pieprasījuma datu integrācijas sistēmas arhitektūra

Promocijas darba 3. nodaļā ir definēta attālu avotu pieprasījuma datu integrācijas arhitektūra. Tā definē attālu avotu datu integrācijas sistēmas komponentes un saites to starpā, ietver izstrādātos attālu avotu datu integrācijas risinājumus un kalpo par pamatu datu integrācijas sistēmas implementācijai.

Prasības arhitektūrai ir izvirzītas, balstoties uz tendencēm dalītu sistēmu izstrādē, pieejām starpuzņēmumu lietojumprogrammu integrācijā un literatūras apskatā identificētajām biežāk sastopamajām heterogēnu, attālu datu avotu integrācijas problēmām:

- datu avotu heterogenitātes iekapsulēšana datu integrācijas risinājumā;
- dažādu datu formātu, piekļuves protokolu atbalsts;
- datu integrācijas procesa definēšana izpildāma biznesa procesa veidā;
- slodzes līdzsvarošana un pēckļūdu atkopšana;
- atkārtotas izmantošanas veicināšana un vēlā saistīšana;
- funkcionālo un nefunkcionālo prasību ievērošana.

Arhitektūra parādīta 2.3. attēlā, un to veido pakalpju reģistrs (SR), kvalitātes datu repozitorijs (QR) un funkcionālais datu integrācijas adapteris (FDIA).



2.3. att. Vispārīgā attālu avotu pieprasījuma datu integrācijas arhitektūra

SR glabā informāciju par abstraktajām datu izgūšanas operācijām (to ieejas, izejas datu un kļūdu paziņojumu datu modeļiem) un pakalpju abstrakcijas slāņus, kas nodrošina pakalpju metožu kartēšanu uz abstraktām datu izgūšanas operācijām. Datu modeļi ir definēti XML shēmu veidā.

Pakalpju un to nodrošināto datu kvalitātes rādītāji tiek glabāti QR. Datu integrācijas procesa laikā tiek veikts pakalpju kvalitātes rādītāju vērtību monitorings un papildināta QR glabātā informācija. QR ļauj veikt pakalpju izvēli atbilstoši nefunkcionālajām prasībām un efektīvu slodzes līdzsvarošanu.

Centrālais arhitektūras elements ir FDIA, kas saņem XML formātā glabātus sākotnējos datus un datu integrācijas procesa loģiku, bet rezultātā atgriež nepieciešamos datus. Procesa loģika spēj attēlot sarežģītas individuālo datu integrācijas uzdevumu savstarpējās atkarības. Datu integrācijas procesa loģika satur:

- izmantojamo tīmekļa pakalpju metožu minimālās pieļaujamās kvalitātes rādītāju vērtības;
- kvalitātes rādītāju svarus;
- datu integrācijas procesu veidojošo uzdevumu kopumu;
- šo uzdevumu korektu izpildes secību un savstarpējās atkarības;
- slodzes līdzsvarošanai izmantojamo maksimālo pakalpju skaitu.

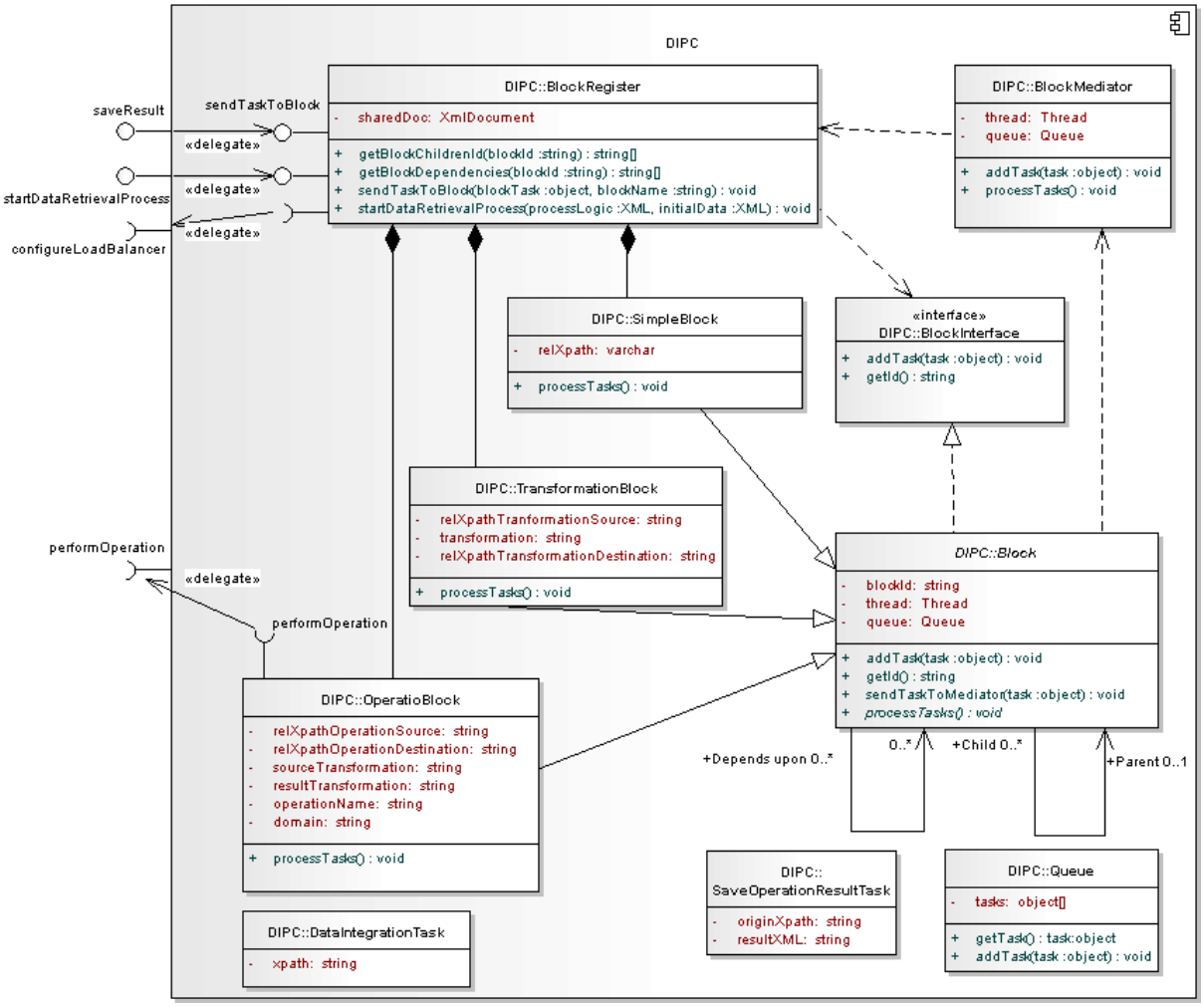
Datu integrācijas procesa sākumā tiek izveidota sākotnējo datu kopija jeb pagaidu XML dokuments, kurā esošie dati tiek izmantoti kā ieejas dati abstraktām datu izgūšanas operācijām. Pagaidu XML dokuments tiek iteratīvi papildināts, līdz tas satur visus nepieciešamos datus.

FDIA ir sīkāk sadalīts divās komponentēs - Datu integrācijas procesa komponentē (DIPC) un Operāciju izpildes komponentē (OEC). DIPC struktūra ir parādīta 2.4. attēlā. Promocijas darbā ir izstrādātas arī citu komponenšu klašu diagrammas, un tās ir aprakstītas darba 3. nodaļā.

DIPC pārrauga kopējo datu integrācijas procesu, transformē datus atbilstoši abstraktas datu izgūšanas operācijas prasībām, realizē ciklus, nosacījuma izteiksmju pārbaudi un datu agregāciju. DIPC komponentē datu integrācijas process tiek definēts izmantojot blokus, to konfigurācijas parametrus un saites starp tiem (pēctecība vai atkarība). Kopējais datu integrācijas process veidojas, blokiem apmainoties savā starpā ar datu integrācijas uzdevumiem.

Bloku starpnieka klase (BM) kontrolē individuālo datu integrācijas uzdevumu savstarpējās atkarības. Bloki izveidotos datu integrācijas uzdevumus nosūta BM, kas pārbauda vai bloka pēcteči drīkst sākt konkrētā uzdevuma apstrādi. Bloka pēctecis saņem datu integrācijas uzdevumu tikai tad, kad izpildās bloku atkarībās definētie apstrādes ierobežojumi. Ir definēti iespējamie datu integrācijas uzdevumu stāvokļi – dīkstāve, izveidots, nosūtīts, tiek

apstrādāts, pabeigts, kļūda. Ir izstrādāti algoritmi, kas nodrošina uzdevumu savlaicīgu un pareizu izpildi, vēlo saistīšanu un slodzes līdzsvarošanu, definēta pēckļūdu atkopšanas loģika.



2.4. att. DIPC struktūra

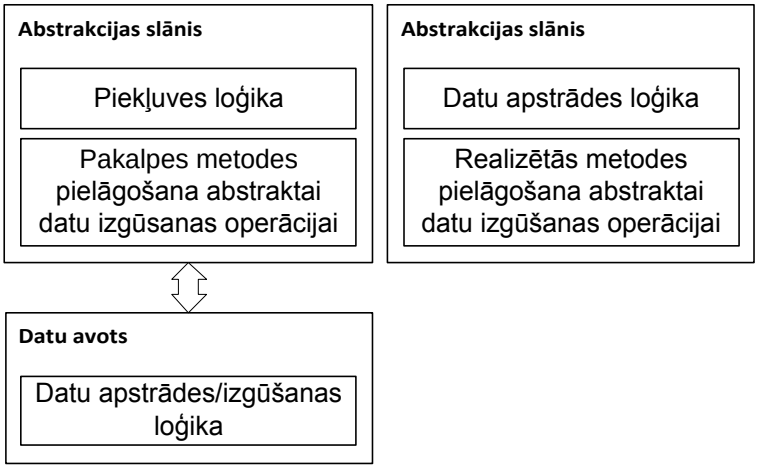
Arhitektūrā ir trīs bloku veidi:

- vienkāršais bloks (SB) - tiek izmantots ciklu veidošanai un nosacījuma izteiksmju pārbaudei;
- transformāciju bloks (TB) – tiek izmantots pagaidu XML dokumenta transformēšanai;
- operacionālais bloks (OB) – izmanto abstraktu datu izgūšanas operāciju izpildei.

Izpildot abstraktu datu izgūšanas operāciju, OB asinhroni griežas pie OEC, kas veic abstraktai datu izgūšanas operācijai atbilstošu fizisku pakalpju meklēšanu SR un to metožu izpildi caur abstrakcijas slāni. Šī procesa laikā tiek realizēta pēckļūdu atkopšana, slodzes līdzsvarošana un tīmekļa pakalpju metodēm atbilstošo kvalitātes datu atjaunošana.

Atsevišķos gadījumos ir lietderīgi virtualizēt datu avotu abstrakcijas slānī (2.5. attēls). Šajā gadījumā datu avotu var uzskatīt par virtuālu, jo no arhitektūras viedokļa nav atšķirību no fizisku pakalpju izmantošanas, tomēr datu apstrādes loģika tiek realizēta abstrakcijas slānī un nenotiek griešanās pie ārēja datu avota.

FDIA funkcionalitāte ir padarīta pieejama, izmantojot lietojumprogrammas saskarni, tādā veidā ļaujot automatizēt datu integrācijas procesu un tā sasaisti ar citām sistēmām.



2.5. att. Abstrakcijas slānī realizēts virtuāls datu avots

Izstrādātā arhitektūra ļauj minimizēt datu integrācijai nepieciešamo laiku, tajā pašā laikā vienkāršojot uzturēšanu, datu integrācijas procesa loģikas modificēšanu un jaunu tīmekļa pakalpju pievienošanu.

2.4. Arhitektūrā ietverto risinājumu novērtējums

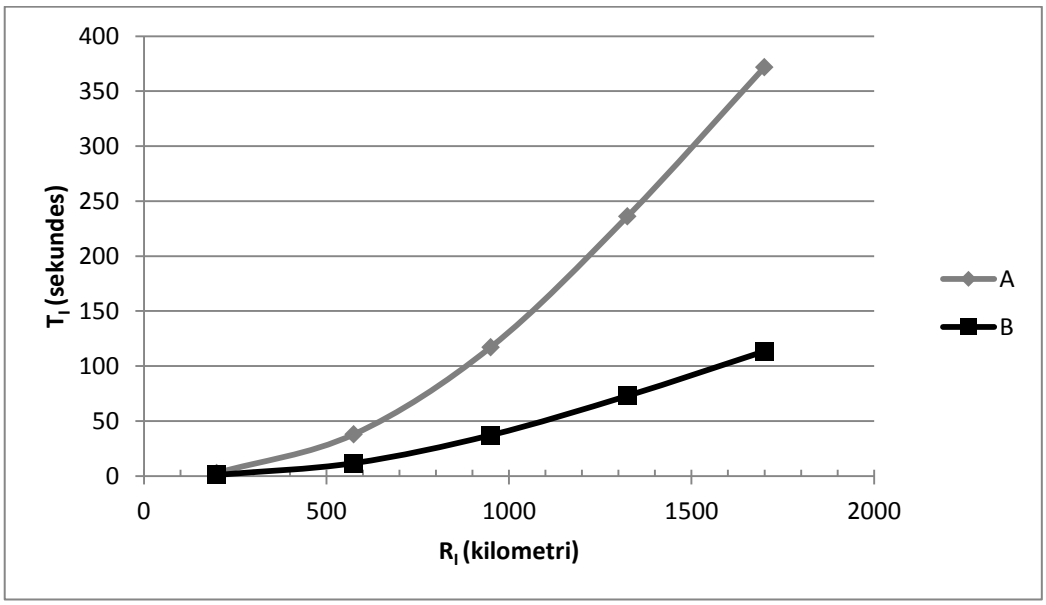
Promocijas darba 4. nodaļā ir eksperimentāli novērtēta DaaS pieejas efektivitāte un darbā izstrādātā arhitektūra. Ir salīdzināti lokāli datu glabāšanas risinājumi ar attālu tīmekļa pakalpi, novērtēta darbā definētā datu integrācijas uzdevumu paralelizācijas pieejas efektivitāte un noteikta slodzes līdzsvarošanas ietekme uz kopējo datu integrācijas laiku.

Eksperimentāli ir novērtēti divi lokāli risinājumi (A un B) ar atšķirīgu veiktspēju un viens attāls datu avots (C). Lokālajiem risinājumiem kopējo datu integrācijas laiku (T_K) veido datu ielādes (T_I) un izgūšanas laiks (T_S), savukārt attālajam risinājumam T_K=T_S, jo datu ielāde netiek aplūkota. Iegūtie rezultāti ir aprakstīti ar regresijas modeļu palīdzību.

Izmantotais datu integrācijas scenārijs ir iedzīvotāju skaita aprēķins noteiktā rādiusā ap izvēlētu centrālo koordināti. Attālais risinājums nodrošina piekļuvi datiem caur WFS tīmekļa pakalpi, savukārt lokālie risinājumi glabā datus Microsoft SQL Server 2008 datubāzē.

Lokālajos risinājumos tiek glabāti tikai konkrētajam uzdevumam nepieciešami dati, tādējādi paātrinot datu ielādes un izgūšanas laiku. Eksperimentu laikā tiek novērtēta T_I un T_S atkarība no datu ielādes rādiusa $R_I \in \{200km, 575km, 950km, 1325km, 1700km\}$ un datu izgūšanas rādiusa $R_S \in \{10km, 60km, 110km, 155km, 200km\}$.

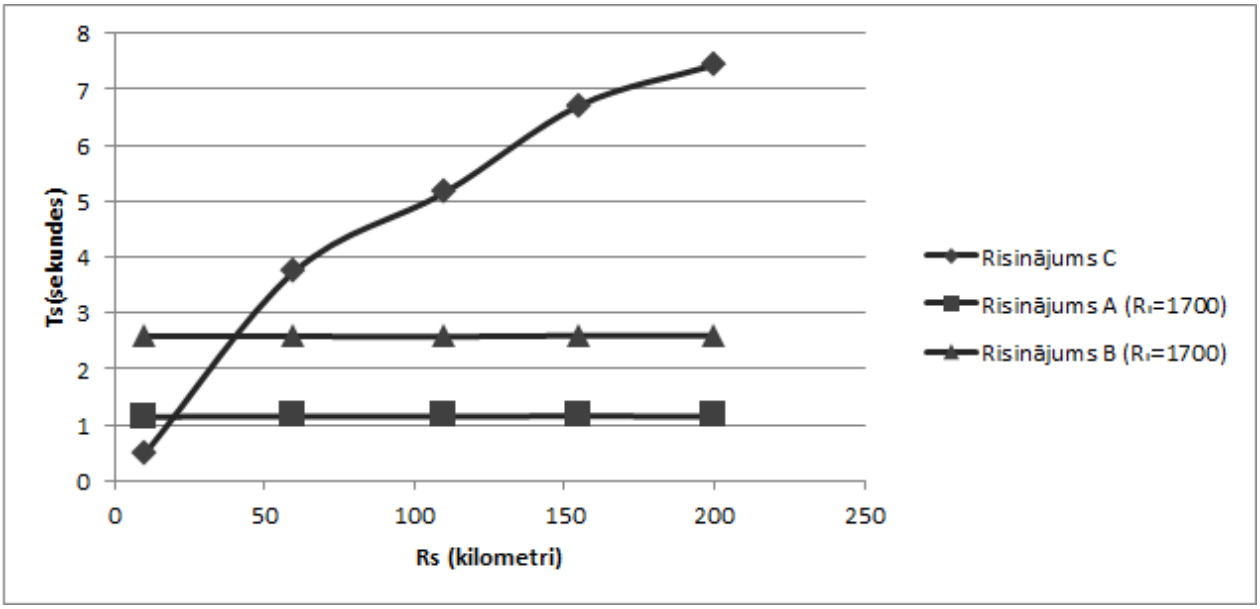
Grafiski T_I atkarība no R_I lokālajiem risinājumiem ir parādīta 2.6. attēlā. Ielādes ilgums ir cieši saistīts ar apgabala izmēru un izmantotā datora jaudu. Ielādējot mazu datu apjomu, datora jauda nav kritiski svarīga, tomēr, palielinot datu apjomu, ir ievērojamas atšķirības.



2.6. att. T_I atkarībā no R_I

Veidojot regresijas modeļus, kas apraksta datu izgūšanas laiku pie atšķirīgām R_S un R_I vērtībām, tika secināts, ka R_S ietekme uz datu izgūšanas laiku ir mazāka par eksperimentālo kļūdu. Daudz nozīmīgāks faktors ir pārmeklējamo datu apjoms, ko nosaka R_I. Attālā risinājuma gadījumā situācija ir diametrāli pretēja, jo, palielinot R_S vērtību, ir novērojams straujš T_K pieaugums.

Lokālajiem risinājumiem ir tendence sniegt atbildi īsākā laikā. Neskatoties uz to, ka eksperimentos izmantotais attālais risinājums nav optimizēts konkrētajam datu izgūšanas scenārijam, tas ir efektīvāks gadījumos, kad pārmeklējamo datu apjoms ir relatīvi liels, salīdzinot ar izgūstamo datu apjomu. Lokālā risinājuma gadījumā ir jārēķinās ar atbilstošas infrastruktūras izveidi, uzturēšanu un periodisku datu atjaunošanu. Novērotās T_S vērtību atšķirības attālajā un lokālajos risinājumos (R_I=1700 km) grafiski parādītas 2.7. attēlā.



2.7. att. T_S risinājumiem A, B un C

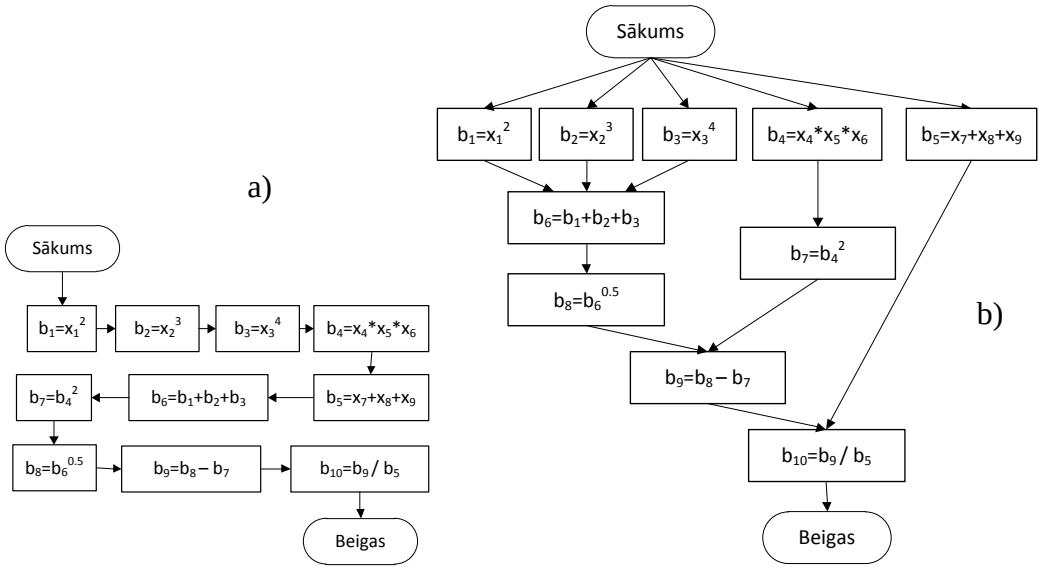
Lai novērtētu, cik lielā mērā T_K ietekmē datu integrācijas uzdevumu paralēla izpilde un cik efektīva ir arhitektūrā īstenotā uzdevumu paralelizācijas loģika, arhitektūras prototips tiek salīdzināts ar secīgu datu integrācijas risinājumu un komerciālu ETL sistēmu (Microsoft SQL Server 2008 Integration Services). Prototipam un ETL sistēmai eksperimentu laikā tika mainīts vienai pakalpei maksimālais vienlaicīgi veicamo pieprasījumu skaits L_V ($L_V=\infty$ un $L_V=1$).

Eksperimentos aplūkotais piemērs ir matemātiskas izteiksmes (2.1) atrisinājums. Tiek novērtētas T_K izmaiņas atkarība no atrisināmo izteiksmju skaita $N_A \in \{1,3,5\}$.

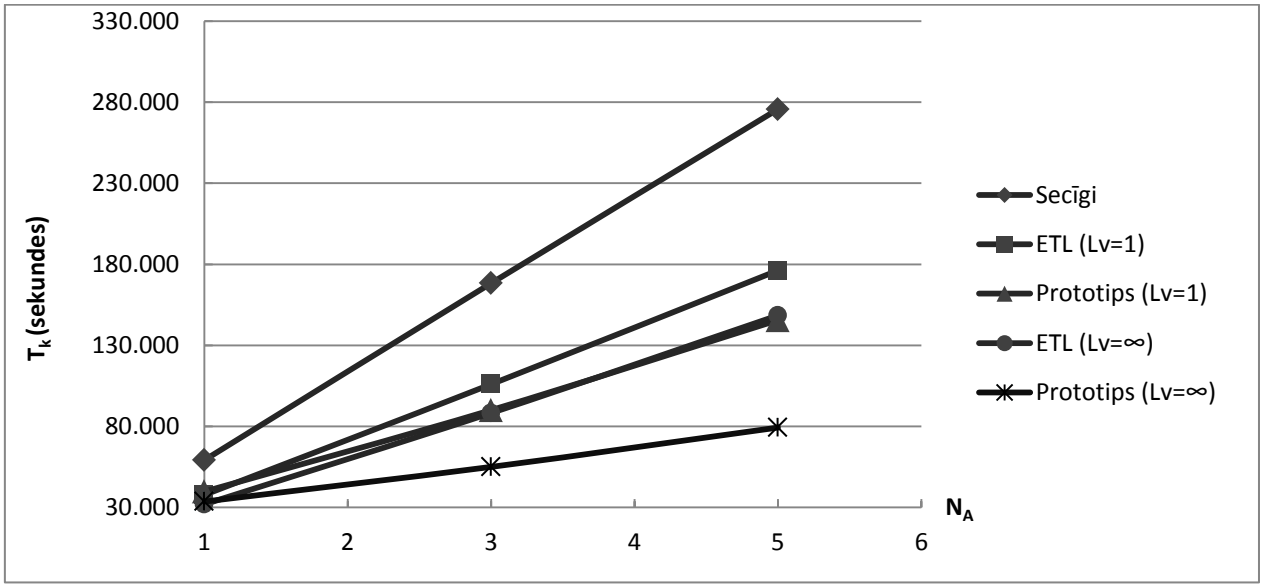
$$Y = \frac{\sqrt{x_1^2+x_2^3+x_3^4}-(x_4*x_5*x_6)^2}{x_7+x_8+x_9} \qquad (2.1)$$

Datu avoti ir piecas REST bāzētas tīmekļa pakalpes, kas veic matemātisko operāciju izpildi. Izteiksmes aprēķinam nepieciešamo darbību izpilde secīgajā un paralēlajā datu integrācijas scenārijā ir parādīta 2.8. attēlā. T_K vērtības izmaiņas atkarībā no N_A katrā no apskatītajiem gadījumiem ir grafiski parādītas 2.9. attēlā.

Palielinot N_A, arvien uzkrītošāks ir prototipā realizētā uzdevumu paralelizācijas algoritma pārākums pār ETL sistēmā īstenoto loģiku. Maksimālā vienlaicīgi izdarāmo pieprasījuma skaita limita noņemšana ļauj būtiski samazināt prototipā realizēto datu integrācijas procesa laiku, savukārt ETL sistēmā ieguvums nav tik nozīmīgs. Eksperimenti pierāda, ka datu integrācijas uzdevumu paralelizācija ir ļoti nozīmīga, jo secīgais datu integrācijas risinājums ir pārliecinoši sliktākais.



2.8. att. Matemātisko operāciju izpildes secība a) secīgi b) paralēli

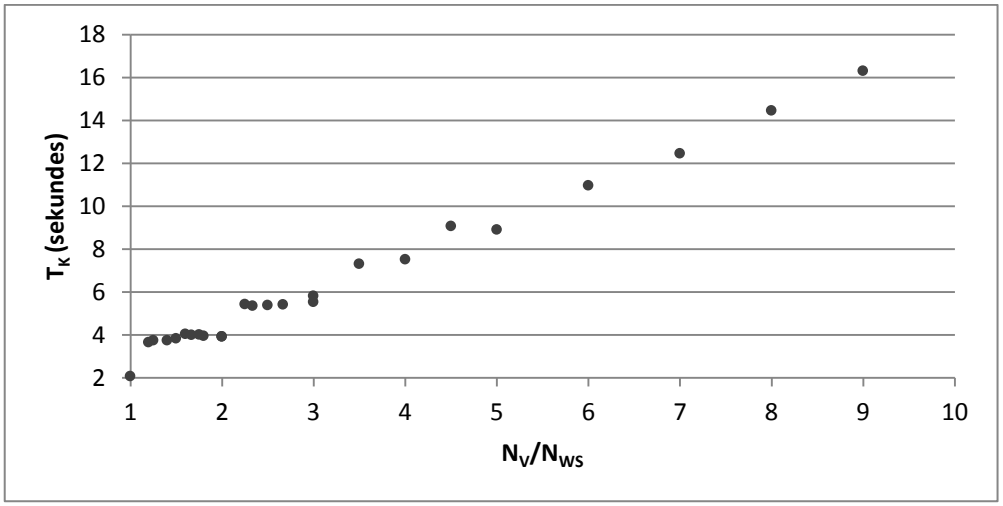


2.9. att. T_K vērtības izmaiņas atkarībā no N_A

Ir novērtēta vienlaicīgi izpildāmo pieprasījumu $N_V \in \{5,6,7,8,9\}$ un alternatīvo tīmekļa pakalpju skaita $N_{WS} \in \{1,2,3,4,5\}$ ietekme uz datu integrācijas laiku T_K. Iegūtie rezultāti ļauj izdarīt secinājumus par slodzes līdzsvarošanas nozīmi.

Izvēlētais uzdevums ir pārbaudīt, vai skaitlis 1'000'000 ir Fibonači skaitlis. Tika izvēlēti pieci identiskas konfigurācijas datori, uz kuriem tika izvietotas REST pakalpes, kas pārbauda, vai ieejas datos norādītais skaitlis ir Fibonači skaitlis.

Kamēr nepieaug vienai pakalpei paralēli izpildāmo pieprasījumu skaits, kopējais datu integrācijas laiks praktiski nemainās. Maksimālais vienai pakalpei vienlaicīgi izpildīto pieprasījumu skaits, kas tiešā veidā ietekmē T_K, tiek aprēķināts kā uz augšu noapaļota N_V/N_{WS} vērtība. T_K atkarība no N_V/N_{WS} ir parādīta 2.10. attēlā.



2.10. att. T_K atkarība no N_V/N_{WS}

Eksperimentos iegūtie rezultāti apstiprina pieņēmumu, ka liels alternatīvu datu avotu skaits ļauj ievērojami samazināt datu integrācijas laiku. Slodzes līdzsvarošana ļauj nodrošināt to, ka datu integrācijas laiks nav lineāri saistīts ar izgūstamo datu apjomu.

2.5. Izmantošana

Šajā nodaļā ir aprakstīti divi darbā definētās arhitektūras izmantošanas piemēri. Pirmajā no tiem arhitektūra ir izmantota, lai risinātu pasažieru pārvadājumu plānošanas problēmu uzņēmumā „BalticTaxi". Ir aplūkots taksometra pasūtīšanas process, kurā, saņemot klienta zvanu, operatoram no pieejamajām automašīnām ir jāizvēlas piemērotākā, jāpaziņo plānotais taksometra ierašanās laiks pie klienta, izmaksas un ierašanās laiks klienta galamērķī. Sākotnējie dati ir klienta atrašanās vieta un vēlamais galamērķis, katrs no kuriem var tikt definēti gan kā adrese, gan kā divu ielu krustojums, gan arī kā noteikta objekta nosaukums.

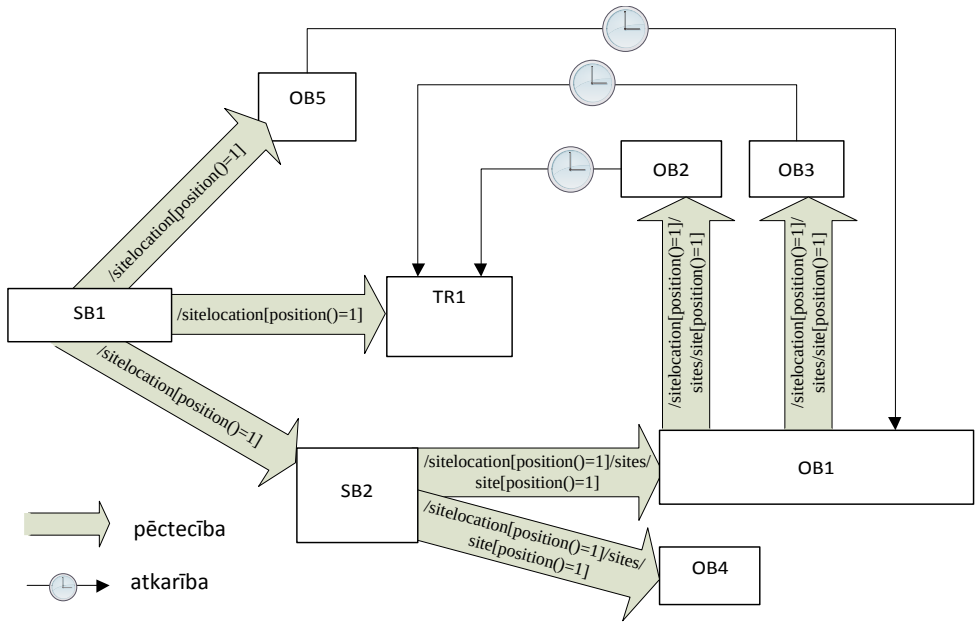
Taksometra izvēle ir realizēta kā virtuāls datu avots. Ir izmantoti pieci attāli datu avoti – Bing, Yahoo, Google, MapQuest, CloudMade, kas nodrošina ģeokodēšanu un maršrutēšanu. Informāciju par pieejamajiem taksometriem tiek iegūta no uzņēmuma iekšējās sistēmas. Ir definētas piecas abstraktas datu izgūšanas operācijas un to datu modeļi, izveidoti avotu abstrakcijas slāņi. Maršrutēšanas abstraktās datu izgūšanas operācijas ieejas datu modelis ir dots 2.11. attēlā.

Veicot maršrutēšanas operācijas izpildi, notiek griešanās pie kāda no četriem alternatīvajiem datu avotiem, iepriekš pārveidojot datus atbilstoši tā ieejas datu modelim. Definētā datu integrācijas procesa loģika ir redzama 2.12. attēlā. Lai ģenerētu sākotnējo datu

failu, vizualizētu rezultātus un vieglāk sekotu līdzi datu integrācijas procesam, ir izstrādāta lietotāja saskarne (2.13. attēlā).

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="getRoute">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="from">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="latitude" type="xs:decimal" />
       <xs:element name="longitude" type="xs:decimal" />
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="to">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="latitude" type="xs:decimal" />
       <xs:element name="longitude" type="xs:decimal" />
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element></xs:schema>
```

2.11.att. Maršrutēšanas abstraktās datu izgūšanas operācijas ieejas datu modelis

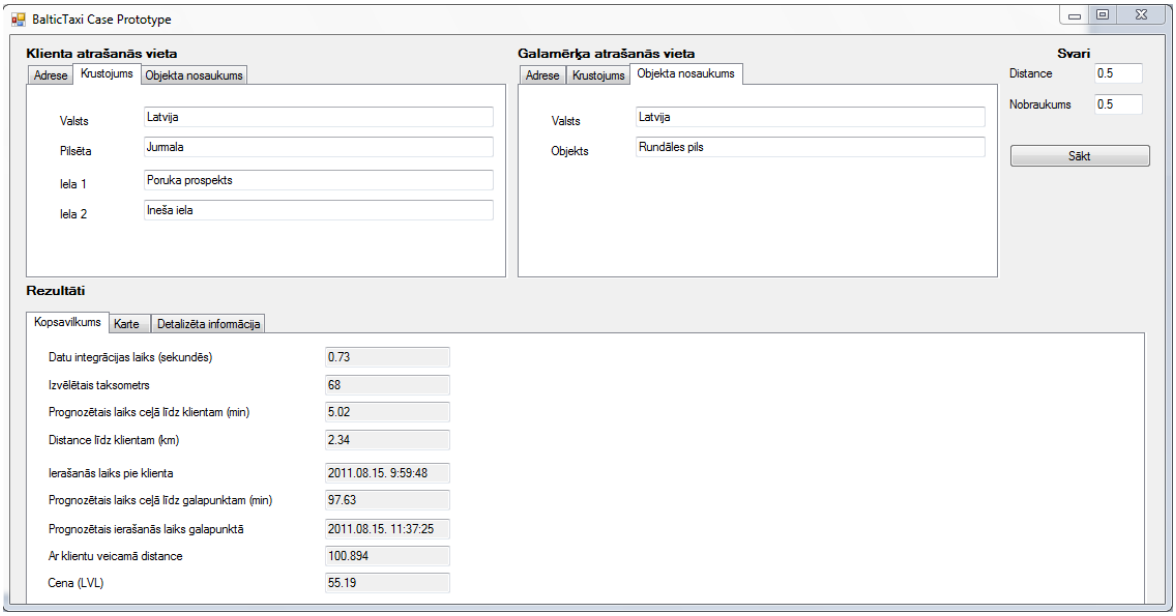


2.12. att. Datu integrācijas procesa loģika „BalticTaxi" gadījumā

Viena no risinājuma priekšrocībām ir ārēju bezmaksas telpisko datu apstrādes tīmekļa pakalpju izmantošanu, kas ļauj uzņēmumam samazināt izmaksas. Neviena telpisko datu apstrādes tīmekļa pakalpe pilnībā neatbilst lietotāja prasībām (adreses, krustojuma, objekta nosaukuma ģeokodēšana un maršrutēšana).

Tas gan nekādā veidā nesarežģī datu integrācijas procesa realizāciju, jo darbā definētā arhitektūra spēj ērtā veidā apvienot visu tīmekļa pakalpju nodrošināto funkcionalitāti. Vairāku alternatīvu tīmekļa pakalpju izmantošana sniedz būtiskas priekšrocības:

- tiek paātrināts kopējais datu integrācijas laiks, kas konkrētajā gadījumā svārstās dažu sekunžu robežās;
- tiek samazināta varbūtība, ka tīmekļa pakalpes kļūdas dēļ apstāsies datu integrācijas process;
- viena tīmekļa pakalpe novērš citas pakalpes datu kvalitātes problēmas.



2.13. att. „BalticTaxi" izmantošanas gadījumā izstrādātā lietotāja saskarne

Minētās priekšrocības novērtēja arī „BalticTaxi" pārstāvji, un tika saņemtas pozitīvas atsauksmes. Šobrīd uzņēmumam tiek izstrādāta jauna informācijas sistēma, un tiek apsvērta iespēja adaptēt uzņēmuma vajadzībām darbā izstrādāto datu integrācijas sistēmu.

Otrais izmantošanas piemērs ir izstrādāts, balstoties uz literatūrā [19] aprakstītām objekta izvietošanas problēmām. Ir aplūkota ātrās ēdināšanas iestādes izvietošanas problēma atkarībā no iedzīvotāju un konkurentu skaita, nekustamā īpašuma cenas potenciālajā izvietošanas vietā. Sākotnējos datus veido potenciālo ātro ēdināšanas iestāžu izvietojuma adreses, rādiuss, kurā tiek aplūkots potenciālo klientu un konkurentu skaits, izvietojamo ātrās ēdināšanas iestāšu skaits un minimālais pieļaujamais attālums starp divām atvērtām ātrās ēdināšanas iestādēm. Lai risinātu objekta izvietošanas problēmu, ir definēts vairākmērķu matemātiskās programmēšanas modelis [13]. Modeļa risināšanai ir izmantota komerciāla optimizācijas programmatūra.

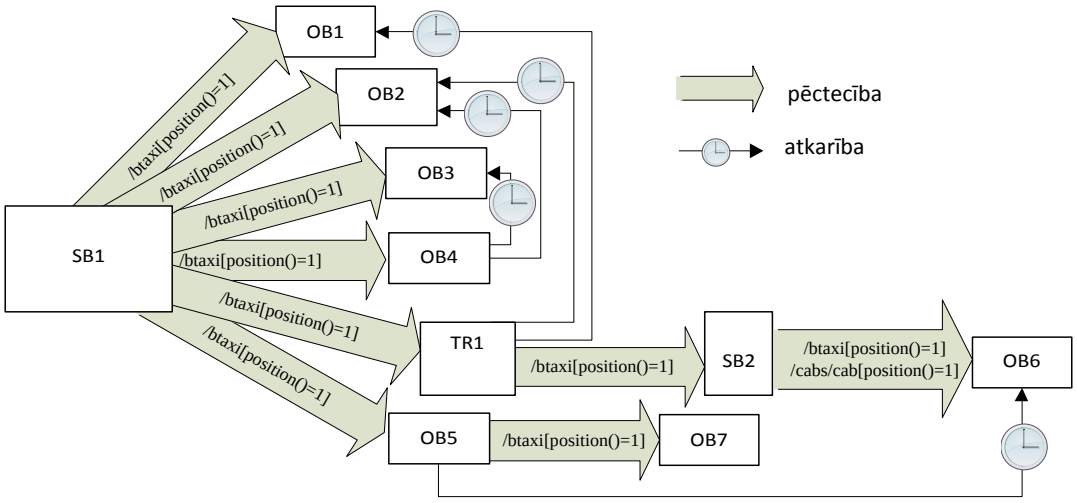
Lēmumpieņemšanas problēmas atrisināšanai nepieciešamie dati ir iedzīvotāju un konkurentu skaits noteiktā rādiusā ap katru no potenciālajām izvietošanas vietām, nekustamā īpašuma cena un radiālās distances starp visām izvietošanas vietām.

Ģeokodēšanai tika izmantotas REST tīmekļa pakalpes, ko nodrošina Google, Bing, Yahoo, Cloudmade. Apdzīvotības dati tika iegūti no Sedac WFS pakalpes, savukārt Zillow un MapQuest tīmekļa pakalpes nodrošina attiecīgi nekustamā īpašuma cenas iegūšanu un konkurējošo uzņēmumu skaita aprēķinu. Tika izveidots virtuāls datu avots, kas realizē radiālo distanču matricas aprēķinu starp potenciālajām ātrās ēdināšanas iestāžu izvietošanas vietām. Ir definētas piecas abstraktas datu izgūšanas operācijas. Ģeokodēšanas abstraktās datu izgūšanas operācijas ieejas datu modelis ir dots 2.14. attēlā. Veicot ģeokodēšanu, notiek griešanās pie kāda no četriem alternatīvajiem datu avotiem, iepriekš pārveidojot datus atbilstoši tā ieejas datu modelim.

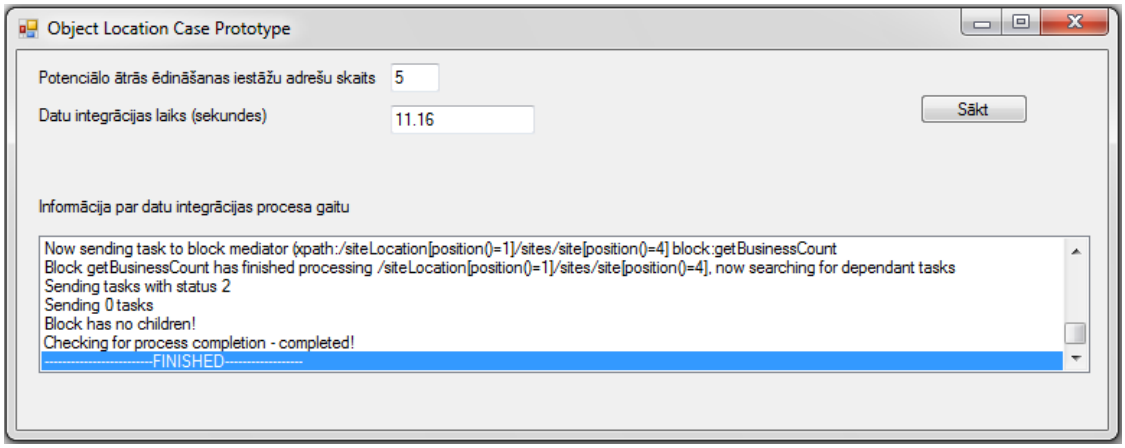
Ir definēta datu integrācijas procesa loģika (2.15. attēls) un izstrādātā lietotāja saskarne (2.16. attēls), kas ļauj sekot līdzi datu integrācijas procesa progresam un novērtēt datu integrācijai nepieciešamo laiku.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="geocode">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="location">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="street" />
       <xs:element name="house" />
       <xs:element name="city" />
           <xs:element name="zipcode" type="xs:unsignedShort" minOccurs="0"/>
       <xs:element name="country" />
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

2.14. Ģeokodēšanas abstraktās datu izgūšanas operācijas ieejas datu modelis

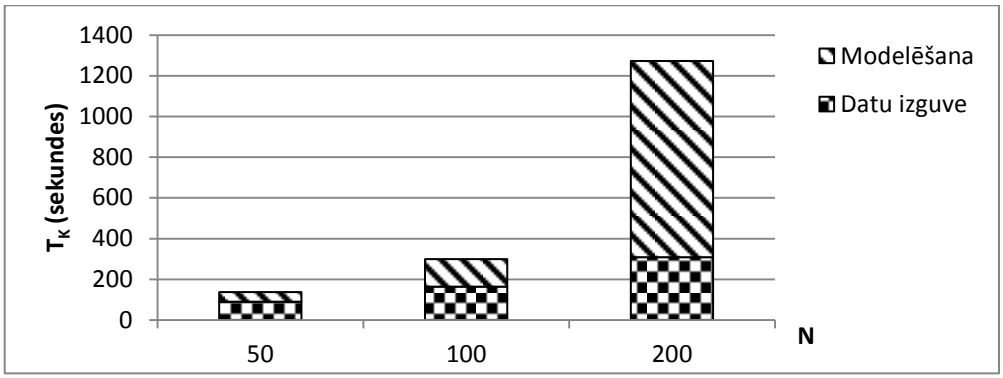


2.15. att. Datu integrācijas procesa loģika objekta izvietošanas gadījumā



2.16. att. Objekta izvietošanas gadījumā izstrādātā lietotāja saskarne

Eksperimentu laikā tika variēts potenciālo ātrās ēdināšanas iestāžu skaits $N \in \{50,100,200\}$ un aplūkotas datu integrācijas laika T_K un modelēšanas laika T_M izmaiņas (2.17. attēls).

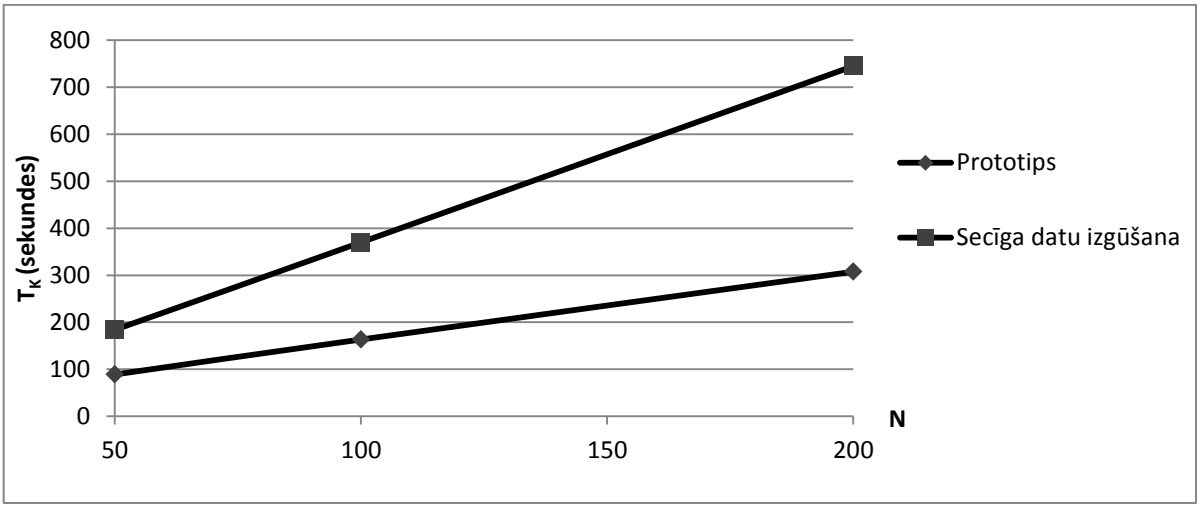


2.17. att. Datu integrācijas laiks salīdzinot ar modelēšanas laiku

Datu integrācijas laika proporcija sarūk, pieaugot N. Tas izskaidrojams ar to, ka objekta atrašanās vietas izvēle ir NP-grūta problēma, savukārt datu integrācijas laiks pieaug lineāri. Nelineāri pieaug tikai distanču matricas aprēķina laiks, tomēr pret kopējo datu integrācijas laiku tas ir proporcionāli mazs, jo tiek veikts lokāli un operācija ir jāizpilda tikai vienreiz katra datu integrācijas procesa ietvaros.

Tā kā datu kvalitātes repozitorijā (QR) tika fiksēti datu avotu atbildes laiki visiem pieprasījumiem, tos summējot, ir iespējams iegūt potenciālo datu integrācijas laiku secīgās izpildes gadījumā. 2.18. attēlā tas ir salīdzināts ar prototipā novērotajām T_K vērtībām.

Datu integrācijas uzdevumu paralelizācija un slodzes līdzsvarošana ir ļāvusi ievērojami samazināt kopējo datu integrācijas laiku. Lai vēl vairāk paātrinātu datu integrācijas procesu, ir nepieciešams nodrošināt vairākas alternatīvas pakalpes, kas veiktu uzņēmumu meklēšanu un apdzīvotības datu iegūšanu, jo tieši šo operāciju izpilde ir vislaikietilpīgākā.



2.18. att. T_K vērtības otrajā izmantošanas gadījumā pie atšķirīgām N vērtībām

3. Promocijas darba rezultāti un secinājumi

Promocijas darba galvenais teorētiskais rezultāts ir attālu avotu pieprasījuma datu integrācijas arhitektūras definēšana, kas ietver gan no platformas neatkarīgu modeli, gan arī izvērstu arhitektūras komponenšu darbības aprakstu un nepieciešamos algoritmus.

Arhitektūras prototips ir ticis izmantots, lai risinātu teorētisku objekta izvietošanas problēmu un praktisku pasažieru pārvadājumu plānošanas problēmu uzņēmumā „BalticTaxi". No uzņēmuma ir saņemtas pozitīvas atsauksmes, jo, lai gan datu integrācijas laiks svārstās dažu sekunžu robežās, datu integrācijas sistēmas aparatūras prasības ir zemas un nav nepieciešams iegādāties vai īrēt ģeogrāfiskās informācijas sistēmas.

Izstrādātā pieprasījuma datu integrācijas arhitektūra risina biznesa intelekta datu integrācijas problēmu, izmantojot DaaS pieeju. 2. nodaļā definēto attālu avotu pieprasījuma datu integrācijas sistēmas modeli var implementēt dažādos veidos. 3. nodaļā izstrādātā arhitektūra un tās implementācija ir tikai viens no tiem. Slodzes līdzsvarošana un datu integrācijas uzdevumu paralelizācija ļauj ievērojami samazināt datu integrācijai nepieciešamo laiku. Uz darbā definētās arhitektūras bāzētie risinājumi ir viegli modificējami, jo tīmekļa pakalpju piekļuves loģika ir pilnībā nodalīta no datu integrācijas procesa loģikas.

Darbā iegūtie teorētiskie rezultāti ir:

1. Identificētas būtiskākās heterogēnu, attālu avotu datu integrācijas problēmas, kas ir datu avotu meklēšana, nepilnīgi metadati, piekļuves protokolu un datu formātu dažādība, dinamiski mainīgi datu avoti, daļēji strukturēts datu formāts,

drošums, nefunkcionālo prasību nozīmīgums, nepieciešamība veikt slodzes līdzsvarošanu, neskaidra licencēšanas politika. Ir analizēti saistītie pētījumi attālu avotu datu integrācijas jomā un secināts, ka netiek pievērsta pietiekama uzmanība individuālo datu integrācijas uzdevumu maksimāli savlaicīgai izpildei, slodzes līdzsvarošanai, kopējā datu integrācijas laika minimizēšanai, nefunkcionālo prasību ievērošanai un atkārtotas izmantošanas veicināšanai.

2. Ir definēts datu integrācijas sistēmas arhitektūras modelis, kas balstās uz abstrakcijas pieeju. Ir definētas abstraktas datu izgūšanas operācijas, kam, izmantojot abstrakcijas slāņus, tiek piesaistītas tīmekļa pakalpju metodes. Datu integrācijas process ir definēts pilnībā atsevišķi no datu avotu piekļuves loģikas, un to veido datu integrācijai nepieciešamo darbību kopums (tajā skaitā abstraktu datu izgūšanas operāciju izpilde) un šo darbību korekta izpildes secība.
3. Ir definētas arhitektūras komponentes, to darbības princips un savstarpējā mijiedarbība. Arhitektūru veido pakalpju reģistrs (tīmekļa pakalpju abstrakcijas slāņi, abstraktas datu izgūšanas operācijas un to datu modeļi), kvalitātes datu repozitorijs (caur abstrakcijas slāni izpildītas abstraktas datu izgūšanas operācijas atbilstošie kvalitātes dati, kas piesaistīti tīmekļa pakalpei) un funkcionālais datu integrācijas adapteris (FDIA). FDIA sīkāk tiek iedalīts datu integrācijas procesa komponentē, kas nodrošina kopējā datu integrācijas procesa loģikas realizāciju, un operāciju izpildes komponentē, kas inicializē abstraktu datu izgūšanas operāciju izpildi, meklējot tām atbilstošas tīmekļa pakalpju metodes, veic slodzes līdzsvarošanu un pēckļūdu atkopšanu.
4. Izstrādāts datu integrācijas uzdevumu paralelizācijas un to savstarpējo atkarību noteikšanas algoritms, kas balstās uz XPath adresāciju un definētiem datu integrācijas uzdevumu iespējamajiem stāvokļiem.
5. Ir definēti datu integrācijas procesā izmantojamie bloku veidi – vienkāršais, operacionālais un transformāciju bloks, un to iespējamās saites – atkarība un pēctecība.
6. Definēti datu integrācijas uzdevumu iespējamie statusi – dīkstāve, izveidots, nosūtīts, tiek apstrādāts, pabeigts, kļūda.
7. Izstrādāts uz nefunkcionālām un funkcionālām prasībām balstīts adaptīvs tīmekļa pakalpju izvēles un slodzes līdzsvarošanas algoritms. Algoritma pirmajā solī tiek atrastas tīmekļa pakalpes, kas atbilst funkcionālajām

prasībām, tiek atlasītas pakalpes, kuru kvalitātes rādītāji nav zemāki par minimālajiem pieļaujamajiem, tālāk tiek aprēķināts pakalpju piemērotības rādītājs un izvēlēts noteikts skaits labāko pakalpju. Datu izgūšanas operācija tiek izpildīta pakalpē, kam prognozēts ātrākais atbildes laiks.

Darba izstrādes laikā tika iegūti šādi praktiskie rezultāti:

1. Izmantojot .NET ietvaru, ir realizēts arhitektūras prototips. Datu integrācijas uzdevumu paralelizācija tiek veikta izmantojot vairākpavedienošanu.
2. Ir veikts praktisks DaaS avotu un lokālu datu glabāšanas risinājumu izmantošanas salīdzinājums un noteikti katras pieejas trūkumi un priekšrocības. Lokālā risinājuma gadījumā trūkums ir nepieciešamība veikt datu ielādi un atjaunošanu, nepieciešamās infrastruktūras izveide, bet priekšrocība – tendence sniegt ātrāku atbildes laiku. DaaS bāzēta risinājuma trūkums ir tāds, ka risinājums var nebūt ideāli piemērots konkrētajam datu izgūšanas scenārijam un dati tiek pārsūtīti caur intertīklu, līdz ar to tas prasa vairāk laika. Galvenā priekšrocība ir tāda, ka nav jāizveido nepieciešamā infrastruktūra, jāveic datu ielāde un atjaunošana. Gadījumos, kad pārmeklējamo datu apjoms ir relatīvi liels pret atgriežamās informācijas apjomu, attālais datu avots var būt ātrāks par lokāliem risinājumiem.
3. Eksperimentāli ir apstiprināta datu integrācija uzdevumu paralelizācijas nepieciešamība un arhitektūrā iekļautā algoritma efektivitāte.
4. Ir praktiski notestētas vairākas publiski pieejamas tīmekļa pakalpes un savākti to atbilstošie kvalitātes dati, kas liecina par ievērojamām atbildes laika svārstībām un nepieciešamību īstenot pēckļūdu atkopšanu.
5. Ir aprakstīti divi izmantošanas gadījumi – teorētiska objekta izvietošanas problēma un praktiska pasažieru pārvadājumu plānošanas problēma uzņēmumā „BalticTaxi”. No uzņēmuma ir saņemtas pozitīvas atsauksmes, tiek apsvērta darbā izstrādātās sistēmas tālāka pielāgošana un iekļaušana jaunajā uzņēmuma informācijas sistēmā.

Turpmāko pētījumu virzieni ir:

1. Definētās arhitektūras bloku klāsta papildināšana un jaunu bloku konfigurācijas parametru definēšana.

2. Algoritma izstrāde, kas ļautu atjaunot datu integrācijas procesu gadījumā, ja tas ir ticis pārtraukts manuāli vai kļūdas rezultātā.
3. Datu integrācijas procesa vizualizācija un pilnvērtīgas lietotāja saskarnes izstrāde, kas ļautu ērtākā veidā definēt datu integrācijas procesa loģiku un sekot līdzi procesa izpildei.
4. Komplicētāku slodzes līdzsvarošanas algoritmu implementācija.
5. Arhitektūras komponenšu decentralizācijas iespēju izvērtēšana.
6. Jaunu izmantošanas gadījumu definēšana.

Literatūra

1. Arlow J., Neustadt I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design - USA: Addison-Wesley Professional, 2005. - p.624.
2. Batini C., Cappiello C., Francalanci C., Maurino A. Methodologies for data quality assessment and improvement// ACM Computing Surveys. - 2009. - Vol.41 - No.3 - p.1-52.
3. Bhide M., Agarwal M. K., Bar-Or A., Padmanabhan S., Mittapalli S. K., Venkatachaliah G. XPEDIA: XML processing for data integration// Proceedings of the VLDB Endowment. - 2009. - Vol.2 - No.2 - p.1330-1341.
4. Casters M., Bouman R., Dongen J. Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration - Canada: Wiley, 2010. - p.720.
5. Dan A., Johnson R., Arsanjani A. Information as a Service: Modeling and Realization// International Workshop on Systems Development in SOA Environments (SDSOA). - Minneapolis, Minnesota, USA: IEEE, 2007. - p.2-2.
6. Donglai Z., Coddington P., Wendelborn A. Binary Data Transfer Performance over High-Latency Networks Using Web Service Attachments// IEEE International Conference on e-Science and Grid Computing. - Bangalore, India: IEEE, 2007. - p.261-269.
7. Eriksson L., Johansson E., Kettaneh-Wold N., Wikström C., Wold S. Design of Experiments, Principles and Applications - Sweden: Umetrics Academy, 2008. - p.329.

8. Frada Burstein C. W. H. Handbook on Decision Support Systems 1: Basic Themes - Germany: Springer, 2008. - p.854.
9. Giordano A. D. Data Integration Blueprint and Modeling: Techniques for a Scalable and Sustainable Architecture - USA: IBM Press, 2011. - p.416.
10. Grosu D., Chronopoulos A. T. A Truthful Mechanism for Fair Load Balancing in Distributed Systems// Network Computing and Applications (NCA). - Cambridge, MA, USA: IEEE, 2003. - p.289-296.
11. Guohua Y., Jingting W. The Design and Implementation of XML Semi-structured Data Extraction and Loading into the Data Warehouse// International Forum on Information Technology and Applications (IFITA). - Guangzhou, China: IEEE, 2010 -p.30-33.
12. Hillier F. S. Introduction to Operations Research - USA: McGraw Hill Higher Education, 2000. - p.1220.
13. Kampars J., Grabis J. Spatial Data Integration Approach with Application in Facility Location// 16th International Conference on Information and Software Technologies. - Kaunas, Lithuania: Kaunas University of Technology, 2010. - p.117-125.
14. Kopecký J., Gomadam K., Vitvar T. hRESTS: An HTML Microformat for Describing RESTful Web Services// IEEE/WIC/ACM International Conference. - Sydney, Australia: IEEE, 2008. - p.619-625.
15. Lathem J., Gomadam K., Sheth A. P. SA-REST and (S)mashups : Adding Semantics to RESTful Services// International Conference on Semantic Computing (ICSC). - Irvine, CA, USA: IEEE, 2007. - p.469-476.
16. Lu W., Chiu K., Gannon D. Building a Generic SOAP Framework over Binary XML// 15th IEEE International Symposium on High Performance Distributed Computing. - Paris, France: IEEE, 2006. - p.195-204.
17. Maximilien E. M., Singh M. P. A framework and ontology for dynamic web, services selection// IEEE Internet Computing. - 2004. - Vol.8 - No.5 - p.84-93.
18. Mou Y.-j., Cao J., Zhang S.-s., Zhang J.-h. Interactive Web service choice-making based on extended QoS model// The Fifth International Conference on Computer and Information Technology (CIT). - Shanghai, China: IEEE, 2005. - p.1130-1134.
19. Owen S. H., Daskin M. S. Strategic facility location: A review// European Journal of Operational Research. - 1998. - Vol.111 - No.3 - p.423-447.
20. Paršutins S., Borisovs A. Data Mining Driven Decision Support// Polish Journal of Environmental Studies. - 2009. - Vol.18 - No.4 - p.8-11.

21. Pautasso C., Zimmermann O., Leymann F. RESTful web services vs. "Big" web services: Making the right architectural decision// 17th International Conference on World Wide Web 2008. - Beijing, China: ACM, 2008. - p.805-814.
22. Ran S. A model for web services discovery with QoS// ACM SIGecom Exchanges. - 2003. - Vol.4 - No.1 - p.1-10.
23. Truong H. L., Dustdar S. On analyzing and specifying concerns for data as a service// 2009 IEEE Asia-Pacific Services Computing Conference (APSCC). - Biopolis, Singapore: IEEE, 2009. - p.87-94.
24. Truong H. L., Dustdar S. On evaluating and publishing data concerns for data as a service// IEEE Asia-Pacific Services Computing Conference. - Hangzhou, China: IEEE, 2010. - p.363-370.
25. Turban E., Sharda R., Delen D. Decision Support and Business Intelligence Systems (9th Edition) - USA: Prentice Hall, 2010. - p.780.
26. Vercellis C. Business Intelligence: Data Mining and Optimization for Decision Making - United Kingdom: Wiley, 2009. - p.436.
27. Wang J., Yu A., Zhang X., Qu L. A dynamic data integration model based on SOA// Second ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM). - Sanya, China: IEEE, 2009 -p.196-199
28. Werner C., Buschmann C. Compressing SOAP messages by using differential encoding// IEEE International Conference on Web Services. - San Diego, CA, USA: IEEE, 2004. - p.540-547.