

**RĪGAS TEHNISKĀ UNIVERSITĀTE**

**Edžus ŽEIRIS**

**E-PAKALPOJUMU SISTĒMU ARHITEKTŪRA**

**Promocijas darba kopsavilkums**

**Rīga 2009**

**RĪGAS TEHNISKĀ UNIVERSITĀTE**  
Datorzinātnes un informācijas tehnoloģijas fakultāte  
Datorvadības, automātikas un datortehnikas institūts

**Edžus ŽEIRIS**

Automātikas un datortehnikas doktora programmas doktorants

**E-PAKALPOJUMU SISTĒMU ARHITEKTŪRA**

**Promocijas darba kopsavilkums**

Zinātniskais vadītājs  
Dr. sc. ing., as. prof.  
**M.ZIEMA**

**RTU Izdevniecība**

**Rīga 2009**

UDK 004.78+004.2(043.2)

Že 250 e

E. Žeiris. E-pakalpojumu sistēmu arhitektūra.  
Promocijas darba kopsavilkums. – R.: RTU, 2009. –  
28 lpp.

Iespiests saskaņā ar DAD institūta 2009. gada 2.  
oktobra lēmumu, protokols Nr. 89



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

**ISBN 978-9934-10-002-4**

**PROMOCIJAS DARBS**  
**IZVIRZĪTS RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ**  
**INŽENIERZINĀTŅU DOKTORA GRĀDA IEGŪŠANAI**

Promocijas darbs inženierzinātņu doktora grāda iegūšanai tiek publiski aizstāvēts 2010.g. 12. aprīlī plkst. 14:30 Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultātē, Meža iela 1/3, 202. auditorijā.

**OFICIĀLIE OPONENTI**

Profesors, Dr.sc.ing. Uldis Sukovskis  
Rīgas Tehniskā universitāte, Latvija

Profesors, Dr.sc.comp. Jānis Bičevskis  
Latvijas Universitāte, Latvija

Asoc. Profesors, Dr.sc.tech. Vytautas Pilkauskas  
Kaunas Tehnoloģiskā universitāte, Lietuva

**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.

Edžus Žeiris..... (paraksts)

Datums: .....

Promocijas darbs ir uzrakstīts latviešu valodā, satur ievadu, četras nodaļas, nobeigumu, literatūras sarakstu, trīs pielikumus, 63 attēlus, 24 tabulas, kopā 160 lappuses. Literatūras sarakstā ir 116 nosaukumi.

# VISPĀRĒJS DARBA RAKSTUROJUMS

**Problēmas aktualitāte.** Pēdējos gados dinamiska informācijas tehnoloģiju sistēmu attīstība un nepieciešamība pēc resursu optimizācijas ir veicinājusi arī e-pakalpojumu attīstību. Visās attīstītākajās pasaules valstīs noris strauja publiskā sektora pakalpojumu elektronizācija. Eiropas savienība ir noteikusi dalībvalstīs elektronizējamās pakalpojumu sfēras, kas veicina Eiropas līmeņa pakalpojumu veidošanos. E-pakalpojumu parasti iedala pēc to elektronizācijas līmeņiem, ko ir noteikusi Eiropas savienība. Latvijā ir realizēti vairāki dažādu elektronizācijas līmeņu e-pakalpojumi.

Pirmā un otrā elektronizācijas līmeņa e-pakalpojumi, tādi kā pakalpojumu sniedzēju, pašvaldības vai valsts iestāžu portāli, informācijas publicēšana un sagatavju lejupielādēšana, ir plaši realizēti un pieejami plašam klientu lokam. Trešā, ceturtnā un piektā elektronizācijas līmeņa e-pakalpojumu izstrāde un attīstība šobrīd attīstās visstraujāk. Latvijā it īpaši tas ir novērojams publiskajā sektorā. Vajadzība pēc e-pakalpojumiem aktualizējas, jo ir nepieciešams optimizēt valsts pārvaldes iestāžu resursus. Viens no veidiem, kā to panākt, ir pakalpojumu elektronizācija un personu pieprasījumu elektroniska apstrāde. Pieprasījums pēc e-pakalpojumiem strauji palielinās, tāpēc ir nepieciešams pielietot efektīvas e-pakalpojumu izstrādes metodes, kas nodrošina kvalitatīvu e-pakalpojumu attīstīšanu.

Izstrādājot e-pakalpojumus, ir būtiski izveidot e-pakalpojumu sistēmas arhitektūru tādu, lai e-pakalpojums apmierinātu visus tam izvirzītos kvalitātes kritērijus, kas veicinātu plašāku tā izmantojamību. Pie tam, elektronizējot pakalpojumus, tie ir jāturpina piedāvāt arī ierastajā veidā, jo ne visiem pakalpojumu saņēmējiem ir iespējams lietot e-pakalpojumu, tāpēc ir nepieciešams samazināt pakalpojuma elektroniskās versijas izstrādes izmaksas. Tas nozīmē, ka ir nepieciešams pielietot efektīvas projektēšanas metodes e-pakalpojumu sistēmu izveidei, kas nodrošinātu pieņemamas e-pakalpojumu sistēmas arhitektūras izveidi. Esošās projektēšanas un arhitektūras izvēles metodes ne vienmēr ir piemērotas šīs problēmas risināšanai. Galvenokārt to trūkumi ir saistīti ar nespēju izslēgt sistēmas projektētāja subjektīvo viedokli, ievērot visus sistēmai izvirzītos kvalitātes kritērijus un izvēlēties piemērotas drošības sistēmu. Pieņemamas e-pakalpojumu sistēmas arhitektūras izvēles metodikas izveide, ļauj efektīvāk veikt e-pakalpojumu izstrādi, dodot iespēju daudz straujāk attīstīt esošos un veidot jaunus e-pakalpojumus. Tādā veidā tiek ekonomēti resursi jaunu e-pakalpojumu izstrādei [28] [69] [83].

**Pētījuma mērķis un uzdevumi.** Promocijas darba mērķis ir izstrādāt e-pakalpojumu sistēmu arhitektūras izvēles metodiku, kas nodrošina pieņemamas e-pakalpojumu sistēmas arhitektūras izvēli, un e – pakalpojumu sistēmai piemērotas drošības sistēmas izveidi.

Lai sasniegtu izvirzīto mērķi, jāizpilda šādi uzdevumi:

1. izstrādāt e-pakalpojumu sistēmas arhitektūras izvēles metodiku, kura balstās uz formalizētu pieeju un kura ļauj izmantot matemātiskās optimizācijas metodes;
2. izstrādāt drošības sistēmas risinājumu efektīvai un integrētai datu pārraidei e-pakalpojumu sistēmām;
3. verificēt izstrādāto metodiku un drošības risinājumu ar praktisku piemēru palīdzību.

**Pētījuma priekšmets** ir pieņemamas e-pakalpojumu sistēmas arhitektūras izvēles metode, kas atrod kompromisu starp visiem sistēmai izvirzītajiem kvalitātes prasību kritērijiem. Pētījums ietver tīmekļa pakalpojuma grafā skaitlisku kvalitātes kritēriju novērtēšanas iespējas lai varētu veikt uz daudzkritēriālu optimizāciju balstītu kompromisa atrašanu, kā arī e-pakalpojumu drošības sistēmas izvēli.

**Pētījuma hipotēzes:**

1. Lai novērstu sistēmas arhitekta subjektīvo faktoru, iespējamās kļūdas un nepazaudētu kādu no potenciāli pieņemamiem e-pakalpojumu sistēmas arhitektūras risinājumiem ir iespējams

izveidot e-pakalpojumu sistēmas arhitektūras izvēles metodiku, kura balstās uz formālu daudzkritēriālu optimizāciju, risinājumu meklējot Pareto kompromisu kopas veidā.

2. Lai izvairītos no šķietami labākā arhitektūras risinājuma izvēles, kas var izrādīties tikai lokāls minimums, e-pakalpojumu sistēmas arhitektūras projektēšanā ir iespējams izmantot tīmekļa pakalpju projektēšanu, kas tiek balstīta uz e-pakalpojuma algoritma grafa segmentēšanu.
3. E-pakalpojumu sistēmu arhitektūras projektēšanai ir iespējams izveidot datorprogrammu, kas realizē e-pakalpojumu sistēmas projektēšanas metodi un ļauj sasniegt kvalitatīvu rezultātu pietiekoši ātrā laikā. Tādā veidā ir iespējams ātri reaģēt uz biznesa vides mainīgajiem nosacījumiem.
4. E-pakalpojumu sistēmu drošībai ir iespējams izveidot vienotu, uzticamu drošības sistēmu, kas uzlabo sistēmas pieejamību un ātrdarbību.

**Pētījuma metodes.** Šajā darbā ir pielietotas daudzkritēriālās optimizācijas metodes, grafu teorijas, algebras elementi, kā arī datorsistēmu projektēšanas metodes. Iegūto rezultātu verificācija ir veikta ar praktiskiem risinājumiem un eksperimentiem.

**Zinātniskie jauninājumi.** Darba zinātniskais jauninājums ir e-pakalpojumu sistēmas arhitektūras izvēles metodika, kas balstās uz tīmekļa pakalpju projektēšanu, izmantojot daudzkritēriālu optimizāciju.

Zinātniskais jauninājums sevī ietver:

1. Pieņemamas e-pakalpojumu sistēmas arhitektūras izvēles metodiku, kas balstās uz daudzkritēriālu optimizāciju, kompromisu kopas atrašanu un vairāku projektēšanas metožu apvienošanu;
2. E-pakalpojuma algoritma formālu pierakstu algoritma grafa veidā optimizācijas algoritmu pielietošanas mērķiem;
3. Datu pārraides drošības risinājums e-pakalpojumu sistēmās, izmantojot datu glabātuvī, serveru pudurus un vairāku drošības risinājumu apvienošanu.

**Praktiskā vērtība.** Darba izstrādes laikā ir radīti šādi praktiski pielietojami produkti:

1. E-pakalpojumu sistēmu arhitektūras izvēles metodika, kuru ir iespējams pielietot ne tikai e-pakalpojumu sistēmu projektēšanā, bet arī citos servisu orientētās arhitektūras (SOA) datorsistēmu risinājumos;
2. Drošības sistēma *Authority*, kuru ir iespējams pielietot visos SOA datorsistēmu risinājumos;
3. Tīmekļa pakalpju projektēšanas programnodrošinājums

**Darba aprobācija.** Darba rezultāti ir publicēti šādos izdevumos:

1. Zeiris E., Zieme M. E-Service Architecture Selection Based on Multi-criteria Optimization// Product-Focused Software Process Improvement 8th International Conference, PROFES 2007, Rīga, Latvia, July 2007 Proceedings. - Berlin Heidelberg: Springer-Verlag, 2007. - 345.-357. lpp.
2. Zieme M., Žeiris E. Pašvaldību e - pakalpojumi// Latvijas i – sabiedrības tehnoloģiju ekspozīcija LatSTE. - Rīga: LU, 2004. - 157.-162. lpp.
3. Žeiris E., Zieme M., Amanis I. Tīkla servisu drošības sistēma / Patents Nr. LV 13720 - Latvija, publicēts 20.08.2008.
4. Žeiris E., Zieme M. Drošības risinājumi e-pakalpojumu sistēmās// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 24. sējums. - Rīga: RTU, 2005. - 84.-91. lpp.
5. Žeiris E., Zieme M. Elektronisko pakalpojumu sistēmu arhitektūras izvēle// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 32. sējums. - Rīga: RTU, 2007. - 99.-107. lpp.
6. Žeiris E., Zieme M. Elektronisko pakalpojumu sistēmu arhitektūras novērtēšanas kritēriji un arhitektūras izvēle// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 27. - Rīga: RTU, 2006. - 91.-98. lpp.

7. Žeiris E., Ziema M. E-pakalpojumu izveides problēmas// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 19. sējums. - Rīga: RTU, 2004. - 48.-53. lpp.
8. Žeiris E., Ziema M. Grafu transformāciju un optimizācijas izmantošana WEB servisu projektēšanā// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 35. sējums. - Rīga: RTU, 2008. - 110.-117. lpp.

Ziņojumi konferencēs:

1. Zeiris E. Local Government E – Services, Baltic IT&T, Electronic Signatures Laboratory, April 18-19 2007. Riga, Latvia
2. Zeiris E., Ziema M. Municipal e – service systems, Baltic IT&T, Global Cities Dialogue Workshop: E – government in city and municipality administration 2005.04.06
3. Zeiris E., Ziema M. Security solutions for client management and e – service systems. eGovernment in the Hanse Passage Regions - requirements, activities and experiences Riga, October 27, 28.2005
4. Zeiris E., Ziema M. E – Service Architecture Selection Based on Multi-criteria Optimization. 8th International Conference, PROFES 2007, Riga, Latvia, 2-4 July 2007
5. Žeiris E. E-paraksta pielietojums pašvaldību darbā, Microsoft Latvija e-paraksta konference, Rīga, 2007.01.10
6. Žeiris E., Ziema M. Drošības risinājumi e–pakalpojumu sistēmās, RTU 46. starptautiskā zinātniskā konference, 2005.10.14
7. Žeiris E., Ziema M. E–pakalpojuma arhitektūras izvēles metožu salīdzinājums. LU 66. starptautiskā zinātniskā konference, 2008.02.08
8. Žeiris E., Ziema M. E–pakalpojumu sistēmu arhitektūra, RTU 45. starptautiskā zinātniskā konference, 2004.10.15
9. Žeiris E., Ziema M. Elektronisko pakalpojumu sistēmu arhitektūras novērtēšanas kritēriji, RTU 47. starptautiskā zinātniskā konference, 2006.10.13
10. Žeiris E., Ziema M. Elektronisko pakalpojumu sistēmu arhitektūras izvēle, RTU 48. starptautiskā zinātniskā konference, 2007.10.12
11. Žeiris E., Ziema M. Elektronisko pakalpojumu sistēmu arhitektūras projektēšana, RTU 50. starptautiskā zinātniskā konference, 2009.10.15
12. Žeiris E., Ziema M. Grafu transformāciju un optimizācijas izmantošana WEB servisu projektēšanā, RTU 49. starptautiskā zinātniskā konference, 2008.10.14
13. Žeiris E., Ziema M. Pašvaldību e–pakalpojumi, LIKTA 10. konference, Rīga, 2008.11.21

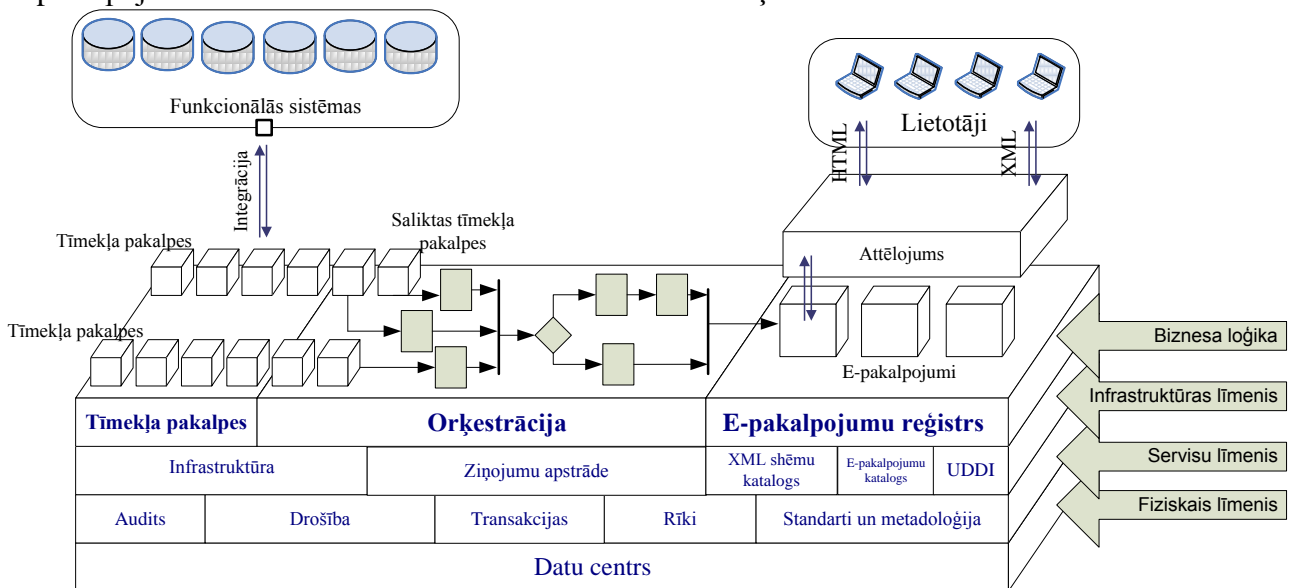
Darba rezultāti ir iekļauti Latvijas Zinātnes Padomes granta Nr. 05.1661 2004. – 2008. gads “Multivides balstītu e–pakalpojumu sistēmas arhitektūras izvēle” un Latvijas Zinātnes ministrijas pētījumu projekts Nr. R7083 2006. gads “Elektronisko pakalpojumu sistēmu arhitektūras izvēles metodikas izstrāde” atskaitēs.

**Darba struktūra un apjoms.** Darbs sastāv no ievada, četrām nodaļām, nobeiguma un trīs pielikumiem. Darbā ir 24 tabulas un 63 attēli. Literatūras saraksts satur 116 nosaukumus. Darba kopējais apjoms ir 160 lappuses. Promocijas darba pamatteksts ir izklāstīts uz 153 lappusēm.

# DARBA SATURA IZKLĀSTS

**Pirmajā nodaļā** ir aprakstīts un izskaidrots datorsistēmu arhitektūras jēdziens. Detalizēti ir aplūkota e-pakalpojumu sistēmas arhitektūra un tās izpratne promocijas darba ietvaros, kā arī ir aprakstītas e – pakalpojumu sistēmas arhitektūras izveides problēmas. Nodaļā ir dots ieskats sistēmas kvalitātes definīcijā, apskatītas divas biežāk izmantotās projektēšanas metodes, kā arī piecas plašāk zināmās sistēmu projektēšanas metodes, kuras ir vērstas uz datorsistēmu projektēšanu, ņemot vērā tām izvirzītās kvalitātes kritēriju prasības. Lai varētu pielietot šajā nodaļā aprakstītās sistēmu projektēšanas metodes, ir nepieciešams novērtēt sistēmas kvalitāti jau sākotnējā tās projektēšanas fāzē, tāpēc šajā nodaļā ir aprakstītas četras sistēmas kvalitātes novērtēšanas metodes, kuras ir iespējams pielietot, lai noteiktu sistēmas kvalitātes kritēriju relatīvās vai absolūtās vērtības. Datorsistēmu drošības prasības tiek aplūkotas atsevišķi kā speciāls gadījums sistēmas kvalitātes prasībām, jo e-pakalpojumu drošība būtiski ietekmē visu e-pakalpojumu. Šajā nodaļā ir aplūkotas arī tīmekļa pakalpju projektēšanas problēmas.

E-pakalpojumu sistēmas arhitektūra sastāv no vairākām daļām.



**1. att. E-pakalpojumu sistēmu arhitektūra**

1. att. ir parādīts, kā pakalpojumā iesaistītās sistēmas un sistēmu komponentes tiek saistītas vienotā e-pakalpojumu sistēmas arhitektūrā. Katram datu objektam, kuru nepieciešams iesaistīt e-pakalpojuma realizācijā, jāizstrādā XML shēmu kopums. Datu izsaukums no attiecīgās funkcionālās sistēmas tiek realizēts ar tīmekļa pakalpju palīdzību. Veicot tīmekļa pakalpju izsaukumus, tiek nosūtīti arī metadati, kas apraksta pieprasījumu. Ar metadatiem tiek nodota arī informācija, kas nepieciešama, lai aizpildītu auditācijas pierakstus. Tīmekļa pakalpes var iedalīt divās grupās: vienkāršās un sarežģītās. Sarežģītās tīmekļa pakalpes faktiski ir vairāku vienkāršo tīmekļa pakalpju loģisks apvienojums, kas izriet no procesu integrācijas prasībām, tajās var tikt apvienotas viena vai vairākas neatkarīgu funkcionālo sistēmu vienkāršās pakalpes. Sarežģīto tīmekļa pakalpju izpildi var nodrošināt, izmantojot BPEL procesoru. Kā e-pakalpojuma tīmekļa pakalpju orķestrācijas (integrācijas) vide ir jāizmanto BPEL procesors. Portāli, vienas pieturas aģentūras aplikācijas u.c. nodrošina e-pakalpojumu piegādi lietotājiem. E-pakalpojumu ievadformas, pieturas punkti, informācija par maksājumiem un izpildes rezultāti tiek nodoti ar HTML vai ar XML lappusēm, kuras portālā var izmantot, lai implementētu pakalpojumu, pielietojot XSLT transformāciju. Tīmekļa pakalpju un e-pakalpojumu turētājiem, t.i., iestāžu speciālistiem un sistēmu administratoriem, kas ir atbildīgi par tīmekļa pakalpju un e-pakalpojumu uzturēšanu un attīstīšanu, jābūt iespējai savstarpēji komunicēt par dažādiem jautājumiem, kas saistīti ar tīmekļa pakalpju un e-pakalpojumu izpildi un attīstību, kā arī jāizpilda asinhroni e-pakalpojumi. Šim nolūkam ir paredzēta ziņojumu sistēma. Ziņojumu sistēma nodrošina darbu ar teksta ziņojumiem un darba uzdevumiem.



Aplikācija ir savstarpēji integrēta ar E-pakalpojumu reģistru, no kura iegūst datus par XML shēmām, tīmekļa pakalpēm un e-pakalpojumiem. Ziņojumu aplikācija ir orķestrāciju klients, jo no tām tiek saņemti ziņojumi par tīmekļa pakalpojumu un e-pakalpojumu izpildi. Dati par visām XML shēmām, tīmekļa pakalpēm un e-pakalpojumiem tiek reģistrēti vienotā reģistrā „E-pakalpojumu reģistrs”. Reģistrā tiek uzturētas visas shēmu, servisu un e-pakalpojumu versijas, lai šī informācija būtu pieejama jebkuram, kas nodarbojas ar e-pakalpojumu izstrādāšanu un attīstīšanu.

E-pakalpojumu sistēmas arhitektūrā galvenās problēmas ir saistītas ar e-pakalpojumu datu standartizāciju, auditu, tīmekļa pakalpojumu kataloga izveidi, asinhronu e-pakalpojumu izpildi, drošību un e-pakalpojumu orķestrācijas izveidi, kas ir saistīta ar tīmekļa pakalpojumu projektēšanu.

E-pakalpojumu sistēmai var tikt izvirzīti vairāki kvalitātes kritēriji, kuri ir jāapmierina e-pakalpojumu sistēmas arhitektūrai: izstrādes izmaksas, uzturēšanas izmaksas, integritāte, mērogojamība, drošība u.c. Kvalitātes kritēriji parasti ir savstarpēji pretrunīgi. Sistēmas arhitektūras kvalitātes noteikšanai var izmantot ISO standartu – „ISO 9126 informācijas tehnoloģijas-programmatūras produktu novērtēšana – kvalitātes atribūti un vadlīnijas to izmantošanai”. Lai varētu ievērot sistēmai izvirzītās kvalitātes prasības, ir nepieciešams izveidot sistēmas arhitektūru, kas apmierina tai izvirzītās prasības. Šādos gadījumos vienkārši funkcionāli bāzēta vai objektu orientēta projektēšanas pieeja ir nepietiekama. Ir nepieciešams pielietot metodes, kas risina arhitektūras izveides kvalitātes problēmas [40] [58] [102].

Darbā aplūkotas sekojošas arhitektūras projektēšanas metodes, kas risina datorsistēmas arhitektūras kvalitātes jautājumus.

*Atribūtu virzīta projektēšanas metode - ADD [8] [9].* Kā ieejas parametri ADD metodei ir kvalitātes atribūtu apraksti un darbinieku zināšanas par sistēmu, tās arhitektūru un to, kā sasniegt kvalitātes atribūtus, lai izveidotu sistēmas arhitektūru. ADD metodi var aplūkot arī kā paplašinājumu plaši izplatītajam RUP (Ration Unified Process) [84] izstrādes procesam. ADD risinājums balstās uz sistēmas arhitektūras definēšanu sistēmas dekompozīcijas procesā gan sistēmai, gan tās kvalitātes atribūtiem. Projektēšanas gaitā tiek veikts rekursīvs dekompozīcijas process, kur katrā solī tiek izvēlēti arhitektūras elementi tādā veidā, lai apmierinātu kvalitātes rādītājus, un tad tiek piemēlēta funkcionalitāte, kas ir jāiekļauj izvēlētajā modulī.

*Programmatūras arhitektūras analīzes metode - SAAM [48] [102].* Kā ieejas parametri SAAM projektēšanas metodei ir programmatūras arhitektūras apraksts vai arī vairāki iespējamie arhitektūras risinājumi, kurus ir nepieciešams analizēt, un kvalitātes prasības, kuras sistēmai ir nepieciešams sasniegt. Arhitektūras risinājumu apraksti šai metodei ir nepieciešami tādā detalizācijas pakāpē, lai tie būtu saprotami personālam, kas ir iesaistīts arhitektūras analīzē. Nav nepieciešams specializēts detalizācijas līmenis. Ņemot vērā, ka šī metode tiek balstīta uz lietošanas gadījumu analīzi, tad sistēmai izvirzītās kvalitātes prasības ir jānosaka lietošanas scenāriju veidā. SAAM metodes sarežģītākā daļa ir kvalitātes prasību noteikšana lietošanas scenāriju veidā. Izstrādātie kvalitātes scenāriji tiek lietoti kā sistēmas etalons, ar kuru salīdzināt izstrādātās arhitektūras atbilstību un nepieciešamās arhitektūras izmaiņas, lai sasniegtu izvirzītās prasības.

*Jan Bosch arhitektūras projektēšanas metode [12] [13].* Šo arhitektūras projektēšanas metodi var aplūkot kā procesu, kur kā ieejas parametri ir sistēmas prasību specifikācija, un rezultātā ir piedāvātā sistēmas arhitektūra. Projektēšanas process sākas ar sistēmas arhitektūras definēšanu, kas tiek balstīta uz sistēmas specifikācijā noteiktajām prasībām, tā rezultātā rodas pirmā sistēmas arhitektūras versija. Izveidotā sistēmas arhitektūra tiek novērtēta atbilstoši sistēmas specifikācijā noteiktajām kvalitātes prasībām. Katrs kvalitātes rādītājs tiek noteikts kā skaitliska vērtība, kura tiek izrēķināta, izmantojot kādu no kvalitātes atribūtu novērtēšanas metodēm. Iegūtās arhitektūras kvalitātes vērtības tiek salīdzinātas ar prasībās noteiktajām. Ja tās atbilst, tad tiek uzskatīts, ka izvēlētajā sistēmas arhitektūra ir pieņemama, ja nē, tad sistēmas arhitektūra tiek transformēta un tiek atkārtoti novērtēta. Tā šo procesu atkārtoti, kamēr tiek sasniegti kvalitātes rādītāji. Ja kvalitātes rādītājus nav iespējams sasniegt un sistēmas arhitekts vairs neredz iespējas transformēt un uzlabot sistēmas arhitektūru, tad ir nepieciešams mainīt sistēmas specifikāciju.

*Arhitektūras kompromisu analīzes metode - ATAM [49] [102].* Ja ir nepieciešams aplūkot vairākus kvalitātes rādītājus vienlaicīgi, tad ir jāmeklē kompromiss starp savstarpēji pretrunīgiem kvalitātes kritērijiem. Tas ir tādēļ, ka kvalitātes rādītāji ir savstarpēji pretēji saistīti. Ja kvalitātes novērtējums pēc viena no rādītājiem ir maksimāli labs, tad pēc cita tajā pašā laikā var būt slikts. Šādos gadījumos arhitektūras projektēšanā ir jāizvēlas līdzsvars starp konfliktējošiem rādītājiem, lai sasniegtu kopējo sistēmas arhitektūras kvalitāti. ATAM ir strukturēta metode, kuru pielieto jau sākotnējā sistēmas arhitektūras izstrādē. ATAM sistēmas projektēšanā tiek iesaistīts personāls ar dažādiem skatījumiem uz sistēmu un kvalitātes interesēm, kas var risināt dažādas ar kvalitāti saistītas problēmas. Metodes pamatā ir komandas darbs, kura rezultātā ar sistemātisku pieeju tiek sasniegts rezultāts.

*Iespējamo draudu analīzes projektēšanas metode - HASARD [101] [102].* Šī projektēšanas metode ir balstīta uz drošības risinājumu prasību sasniegšanu. To pielieto sistēmu projektēšanā dažādās sfērās, kas ir saistītas ar augstām drošības prasībām. HASARD projektēšanas metode ir modeļu bāzēta projektēšanas metode. Tajā tiek veidoti kvalitātes modeļi, par pamatu ņemot arhitektūras modeļus.

Visas aplūkotās metodes risina jautājumu par sistēmas kvalitāti, bet tām ir viens trūkums - sistēmas arhitekta vai personas, kas veic sistēmas arhitektūras projektēšanu, subjektīvais viedoklis, kā arī iespēja, ka tiks palaists garām pats labākais risinājums, vai arī tas vispār netiks aplūkots. Var tikt izvēlēts šķietami labākais variants, kas var izrādīties tikai lokālais minimums, jo netiek aplūkoti pilnīgi visi iespējamie varianti. Turklāt, pilnīgi visu iespējamo variantu aplūkošana piedāvātajās metodēs ir praktiski neiespējama, jo izstrādājamo, iespējamo sistēmas arhitektūru skaits varētu būt pietiekoši liels, lai to spētu apstrādāt sistēmas arhitekts.

Lai novērtētu sistēmas arhitektūras kvalitāti atbilstoši izvirzītajiem kvalitātes kritērijiem, ir nepieciešams pielietot kādu no sistēmas novērtēšanas metodēm jau sistēmas projektēšanas fāzē. Sistēmas kvalitātes novērtēšanai tās projektēšanas stadijā var izmantot kādu no četrām kvalitātes novērtēšanas metodēm: uz scenārijiem bāzētu, simulācijas, matemātiskās modelēšanas vai argumentētu novērtēšanu [13] [47].

Sistēmas drošības projektēšanai un kvalitātes novērtējumam ir jāpievērš īpaša uzmanība. Jebkurai datorsistēmai ir nepieciešams analizēt iespējamus riskus, kas saistīti ar veidojamo sistēmu, kā arī ar datorsistēmas drošību. Datorsistēmu izstrādē drošības projektēšanai bieži tiek izmantots Microsoft piedāvātais iespējamo draudu modelēšanas process, kas ir pietiekami vienkāršs un viegli apgūstams, tāpēc arī plaši pielietots. Lai ievērotu datorsistēmas drošības kvalitāti, nepieciešams par to rūpēties gan sistēmas projektēšanas laikā, gan arī visā turpmākajā programmatūras izstrādes dzīves cikla laikā. Papildus izvēlētajai sistēmas arhitektūrai, kas nodrošina nepieciešamo drošības kvalitāti, programmatūras izstrādē ir nepieciešams izmantot pārbaudītus drošības standartus, kā arī veikt izstrādājamai sistēmai regulārus drošības auditus un novērtēt izstrādātās sistēmas drošību gan pirms, gan arī pēc sistēmas izstrādes [36] [70] [71] [91] [114] [116].

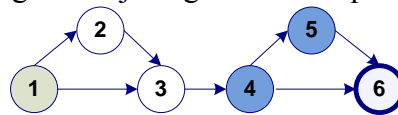
**Otrajā nodaļā** tiek piedāvāta autora izstrādāta e-pakalpojumu sistēmu arhitektūras izvēles metode.

Projektējot e-pakalpojumus, ir nepieciešams reducēt e-pakalpojumu izstrādei patērētos resursus, tajā pašā laikā sasniedzot vēlamo kvalitāti izvēlētajai e-pakalpojuma sistēmas arhitektūrai. Tas nozīmē, ka ir nepieciešams aplūkot sistēmas arhitektūras projektēšanas metodi, kas būtu daļēji automatizējama un aplūkotu visus iespējamus arhitektūras risinājumus, lai izvairītos no lokāla minimuma izvēles, kā arī tiktu ņemti vērā visi sistēmas arhitektūrai izvirzītie kvalitātes kritēriji. E-pakalpojumu sistēmas arhitektūras izveidē liela nozīme ir e-pakalpojumā iesaistīto tīmekļa pakalpju un to savstarpējās sasaistes (orķestrācijas) projektēšanai, lai izvēlētos pieņemamu e-pakalpojumu sistēmas arhitektūru. Lai varētu veikt tīmekļa pakalpju projektēšanas metodes daļēju automatizāciju, kas ir par pamatu pieņemamas e-pakalpojumu sistēmas arhitektūras izvēlē, ir nepieciešams formalizēt tīmekļa pakalpju un to orķestrācijas pierakstu. Darbā tiek piedāvāta e-pakalpojumu sistēmas arhitektūras izvēle, kas tiek balstīta uz tīmekļa pakalpju projektēšanu.

E-pakalpojumu izstrāde tiek sākota ar prasību definēšanu un sistēmanalīzi, kuras laikā tiek aprakstīts e-pakalpojumu procesa modelis, kas tiek ņemts par pamatu e-pakalpojumu algoritma grafa izstrādē un tīmekļa pakalpju projektēšanā.

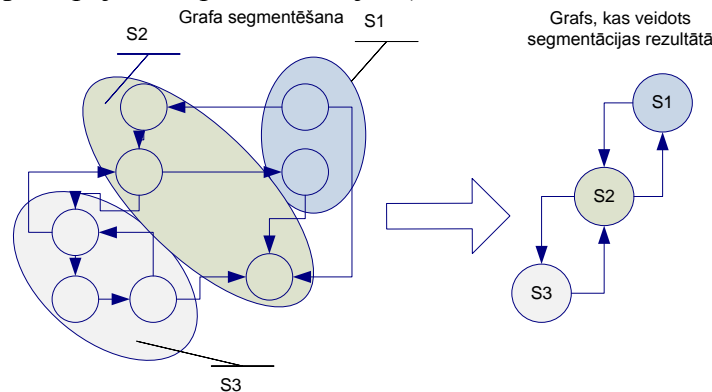
E-pakalpojuma algoritma grafs ir definēts, kā orientēts grafs  $G = (S, L)$ , kur  $S = \{s_1, s_2, s_3, \dots, s_n\}$  ir galīga kopa – grafa virsotnes, kas pēc savas būtības ir e-pakalpojuma algoritma izpildāmas darbības, un  $L \subset S \times S$ . Loks  $l = (s_j, s_k)$  grafā nozīmē to, ka e-pakalpojuma algoritma izpildē pēc darbības  $s_j$  izpildes seko darbības  $s_j$  izpilde. Loki e-pakalpojuma algoritma grafā norāda informācijas plūsmu (2. att.). Veidojot e-pakalpojuma algoritma grafu, lai to varētu izmantot par pamatu e-pakalpojumu sistēmas arhitektūras izveidē, ir jāņem vērā daži ierobežojumi, kas saistīti ar SOA:

1. Katrai algoritma grafa virsotnei  $s_i$  ir jārealizē sevī noslēgta izpildāma darbība. Tas nozīmē, ka virsotne darbojas vienā transakcijā un nav saistīta ar citās virsotnēs realizētajiem algoritmiem. Šis nosacījums ir saistīts ar augstu kohēziju un minimālu atkarību.
2. Katrai virsotnei ir jā satur vismaz viena izpildāma darbība. Praksē tas ir saistīts ar algoritmu implementējošajām metodēm. Turpmākiem mērķiem virsotnes darbības implementējošās metodes tiks apzīmētas ar  $M_{s_i} = \{m_{s_i}^0, m_{s_i}^1, m_{s_i}^2, \dots, m_{s_i}^g\}$  un katras metodes  $m_{s_i}^j$  koda rindiņu skaitu ar  $O(m_{s_i}^j)$
3. Darbības, kas atkārtojas algoritma izpildes laikā nevar tikt veidotas grafā kā dažādas virsotnes  $\forall s_i, s_j, s_i \neq s_j, s_i \in S, s_j \in S, M_{s_i} \cap M_{s_j} = \emptyset$ . Tas ir nepieciešams lai nodrošinātu augstu kohēziju un jau sākotnēji izslēgtu nevajadzīgu variantu apstrādi.



2. att. E-pakalpojuma algoritma grafa piemērs

E-pakalpojuma sistēmas arhitektūras izvēles pamatā ir tīmekļa pakalpju izvēle. Tas nozīmē, ka e-pakalpojuma algoritms ir jārealizē kā tīmekļa pakalpe, lai varētu izveidot e-pakalpojuma sistēmas arhitektūru. Algoritma grafu ir iespējams vairākos veidos realizēt kā tīmekļa pakalpju grafu. Sākot no tā, ka katru algoritma virsotni realizē kā atsevišķu tīmekļa pakalpi, un beidzot ar to, ka visas algoritma virsotnes tiek realizētas kā viena tīmekļa pakalpe. Tas nozīmē, ka dažādi algoritma grafa segmenti var tikt realizēti kā tīmekļa pakalpes, tādā veidā mainot e-pakalpojuma arhitektūru un ietekmējot dažādus e-pakalpojuma izpildes rādītājus (3. att.).



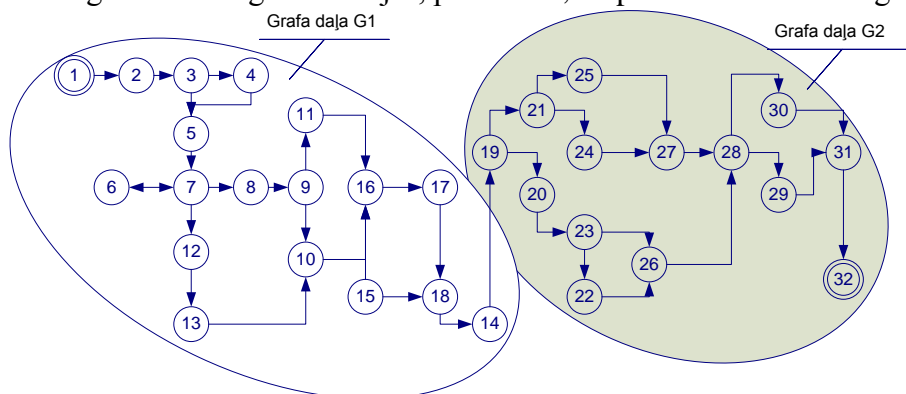
3. att. Algoritma grafa segmentēšana

Lai varētu veikt grafa segmentēšanu un formāli to aprakstīt, grafs  $G$  tiek aplūkots kā attēlojums  $\Gamma$  virsotņu kopā  $S$ , katrai virsotnei piekārtojot apakškopu ( $s_i \subset S$ ), kas pēc savas būtības ir no virsotnēs  $s_i$  sasniedzamās virsotnes.  $\Gamma s_i$  ir no virsotnes  $s_i$  izejošie loki un  $\Gamma^{-1} s_i$  ir virsotnēs  $s_i$  ieejošie loki. Šādam e-pakalpojumu algoritma grafam eksistē vismaz viena virsotne  $s_0 (s_0 \in S)$ , kurā neieiet neviens loks  $\Gamma^{-1} s_0 = \emptyset$ , kuru sauc par e-pakalpojuma algoritma sākumu un tāpat arī eksistē vismaz viena virsotne  $s_r (s_r \in S)$ , no kuras neiziet neviens loks  $\Gamma s_r = \emptyset$ , kuru sauc par e-pakalpojuma algoritma rezultātu. Lai atrastu visus iespējamus tīmekļa pakalpju grafus, no kuriem

izvēlēties tādu, kuru realizēt e-pakalpojumu sistēmas arhitektūrā, sākotnēji par pirmo iespējamo tīmekļa pakalpju grafu pieņem doto algoritma grafu  $G$ . Visi pārējie tīmekļa pakalpju grafi tiek iegūti segmentējot sākotnējo tīmekļa pakalpju grafu, kas sakrīt ar algoritmu grafu. Par cik ir iespējamas vairākas segmentācijas, tiek definēta kopa  $X = \{G'\}$ , kas satur visus iespējamus grafus, kas ir rekursīvi atvasināti no sākotnējā tīmekļa pakalpju grafa  $G$ . Grafa pārveidojumi tiek veikti, apvienojot virsotnes. Divu virsotņu  $s_i$  un  $s_j$  apvienojums  $s'$  ir kā abu virsotņu izejošo un ieejošo loku apvienojums.  $\Gamma s' = \Gamma s_i \cup \Gamma s_j$  un  $\Gamma^{-1} s' = \Gamma^{-1} s_i \cup \Gamma^{-1} s_j$ . Metožu kopa jaunizveidotajā virsotnē veidojas sekojoši:  $M_{s'} = M_{s_i} \cup M_{s_j}$ . Praksē, veidojot tīmekļa pakalpju grafu, pastāv dažādi ierobežojumi, piemēram, vienā e-pakalpojumā ir nepieciešams integrēt dažādu sistēmu funkcionalitāti, vai arī nepieciešams izmantot jau eksistējošas tīmekļa pakalpes. Tāpēc sākotnējam tīmekļa pakalpju grafam par kuru tiek pieņemts algoritma grafs, var uzlikt ierobežojumus, jeb, citiem vārdiem sakot, virsotņu pazīmes  $I(s_i)$ . Dotajā piemērā (2. att.) virsotnes „4” un „5” ir veidotas citā sistēmā, tāpēc tās tiek iekrāsotas citā krāsā. Tātad  $I(s_1) = I(s_2) = I(s_3) = I(s_6) = 0$  un  $I(s_4) = I(s_5) = 1$ . Šādā gadījumā virsotņu apvienojums  $s'$  ir iespējams tikai tad, ja  $I(s_i) = I(s_j)$ .

Ņemot vērā to, ka algoritma grafam, kas sākotnēji tiek pieņemts par tīmekļa pakalpju grafu, ir piekārtotas pazīmes, virsotņu apvienojums  $s'$  ir iespējams tikai tad, ja  $I(s_i) = I(s_j)$ .

Visu iespējamo algoritma grafa segmentu iegūšana ir NP-pilna problēma, kur visus iespējamus tīmekļa pakalpju grafus var iegūt tikai rekursīva algoritma izpildes rezultātā. Ņemot vērā, ka šāda segmentēšana ir darbietilpīga, šādā veidā reālā laikā ir iespējams segmentēt algoritma grafu, kura virsotņu skaits nepārsniedz 20. Ja ir nepieciešams apstrādāt lielāku e-pakalpojuma algoritma grafu ar vairāk nekā 20 virsotnēm, un izveidot visus iespējamus tīmekļa pakalpju grafus, to segmentējot, tad šo uzdevumu ir nepieciešams sadalīt vairākos mazākos uzdevumos. Algoritma grafā ir jāatrod tādi algoritma grafa  $G = (S, L)$  segmenti  $G_1$  un  $G_2$ , kas ir loģiski atdalāmi, un doto algoritma grafu ir jāsadala divos vai vairākos mazākos algoritma grafos, kuriem jau var pielietot pilnu segmentāciju. Citiem vārdiem sakot  $G_1 = (S_1, L_1)$ ,  $S_1 = \{s_1 \dots s_j\}$  un  $G_2 = (S_2, L_2)$ ,  $S_2 = \{s_k \dots s_n\}$ , kur  $S_1 \cup S_2 = S$ ,  $S_1 \cap S_2 = \emptyset$ ,  $\Gamma^{-1} s_k \subseteq \Gamma s_j$  un  $\forall_{s' \in S_1, s'' \in S_2} \Gamma s' \cap \Gamma s'' = \emptyset$ . Liela algoritma grafa sadalīšana mazākos parādīta 4. att., kur katru no norādītajiem segmentiem ir iespējams risināt kā atsevišķu tīmekļa pakalpju projektēšanas uzdevumu. Grafu sadalīšana ir attiecināma uz skaitļošanas uzdevumiem, kur nepieciešams grafu sadalīt divās (vai vairākās) lielās daļās, minimizējot šķautņu skaitu, kas šķērso šķēlumu. Lai atrisinātu šo problēmu, iespējams izmantot standarta algoritmus no grafu teorijas, piemēram, uz plūsmām balstītie algoritmi [6]



4. att. Liela e-pakalpojuma algoritma grafa sākotnējie segmenti

Ja grafa sadalīšanas uzdevuma rezultātā nav iespējams atrast tādas grafa daļas, kur tās savā starpā saista tikai viena šķautne, tad algoritma grafā atrod grafa daļas ar minimālu šķautņu sasaisti, un algoritma grafā, pārveidojot to uz tīmekļa pakalpju grafu, ievieš papildus divas virsotnes. Vienu virsotni grafa daļai  $G_1$  kā beigu stāvokli un vienu virsotni grafa daļai  $G_2$  kā sākuma virsotni.

Algoritma grafa segmentācijas rezultātā iegūto tīmekļa pakalpju grafi ir atšķirīgi. Tie atšķiras ne tikai pēc sava izskata, bet tiem ir dažādi kvalitātes rādītāji. Aplūkosim dažus no iespējamajiem kvalitātes rādītājiem, kas raksturo tīmekļa pakalpju grafu kvalitāti arhitektūras terminoloģijā:

izstrādes izmaksas, ātrdarbība, uzturēšanas sarežģītība (izmaksas), tīmekļa pakalpju atkārtota izmantojamība un integritāte. Minētie kvalitātes rādītāji nav visi, kas var raksturot tīmekļa pakalpju grafu. Šī darba ietvaros ir aplūkoti šie pieci rādītāji un piedāvātas autora izstrādātas kvalitātes kritēriju novērtēšanas formulas, jo vienmēr ir iespējams pievienot un šādā pašā kontekstā aplūkot citus kvalitātes rādītājus kopā ar šiem vai arī atsevišķi. Dotie kvalitātes rādītāji var kalpot kā tīmekļa pakalpju grafa novērtējums, ja tiem piekārto kādas skaitliskas vērtības. Izstrādājot e-pakalpojumu gan pasūtītājam, gan izpildītājam ir būtiski zināt, cik maksās pakalpojuma elektronizācijas izstrāde, cik maksās uzturēšana, cik labi un stabili strādās e-pakalpojums.

*Izstrādes izmaksas.* Lai aprēķinātu tīmekļa pakalpju grafa izstrādes izmaksas  $C$  (Costs), par pamatu tiek ņemts koda rindiņu skaits LOC (Lines of Code), kas nepieciešams, lai realizētu tīmekļa pakalpju grafā ietverto algoritmu. Lai izrēķinātu katras tīmekļa pakalpju grafa virsotnes izstrādes izmaksas  $C_{s_i}$ , pielieto sekojošu algoritmu:

$$C_{s_i} = \left( \sum_{m_{s_i} \in M_{s_i}} O(m_{s_i}) + W_c(s_i) \right)^p \quad (1)$$

$M_{s_i} = \{m_{s_i}^0, m_{s_i}^1, m_{s_i}^2, \dots, m_{s_i}^g\}$  ir grafa virsotnes (tīmekļa pakalpes) darbības implementējošās metodes.  $O(m_{s_i}^j)$  ir katras metodes  $m_{s_i}^j$  koda rindiņu skaits.  $W_c(s_i)$  ir tīmekļa pakalpju grafa virsotnes  $s_i$  tīmekļa pakalpes implementācijas koda rindiņu skaits, kas ir konstante katrai no tīmekļa pakalpju realizācijas vidēm. Pakāpe  $p$  raksturo attiecīgas realizācijas programmēšanas valodas implementācijas sarežģītību ( $p$  pēc savas būtības ir daļskaitlis). Visa e-pakalpojuma tīmekļa pakalpju grafa izstrādes izmaksas  $C$  tiek aprēķinātas sekojoši:

$$C = \sum_{s_i \in S} C_{s_i} \quad (2)$$

Ņemot vērā to, ka piemēros tiek izmantota C# programmēšanas valoda, tad aprēķiniem tiek izmantotas sekojošas konstantes  $p = \frac{1}{2}$  un  $W_c = 200$  [47] [100] [109] [110].

*Ātrdarbība.* E-pakalpojuma ātrdarbības  $T$  aprēķiniem katrai tīmekļa pakalpju grafa virsotnei piekārto vidējo tās izpildes laiku milisekundēs pie vidēja apjoma ieejas datiem (tas nozīmē, ka netiek izmantota maksimāli pieļaujamā datu plūsma)  $t_{s_i}$ . Lai katru no tīmekļa pakalpju grafa virsotnēs ietvertajām tīmekļa pakalpēm izpildītu aplikāciju serverī, kas nodrošina tīmekļa pakalpju darbību, ir nepieciešams papildus laiks uz katru no tām  $t_{ws} = \log(t_{s_i})$ . E-pakalpojuma kopējās ātrdarbības aprēķins  $T$ :

$$T = \sum_{s_i \in S} (t_s + t_{ws}) + t_{epak} + t_{data} \quad (3)$$

Izpildes laikam pieskaita papildus laiku  $t_{epak} = \log(n)$ , kas nepieciešams, lai darbinātu e-pakalpojumu kādā no e-pakalpojumu izpildes vidēm (BPEL procesors), kur  $n = |S|$  tīmekļa pakalpju (virsotņu) skaits grafā, un laiku, kas nepieciešams, lai pārsūtītu datus no viena e-pakalpojuma stāvokļa uz nākamo  $t_{data} = \sum_{l_i \in L} t_{l_i}$ . Katram no tīmekļa pakalpju grafa lokiem  $l_i \in L$  piekārto vidējo laiku milisekundēs  $t_{l_i}$ , kas nepieciešams, lai pārsūtītu datus pa šo loku [90] [100] [109] [110].

*Uzturēšanas izmaksas.* E-pakalpojuma uzturēšanas izmaksas  $E$  ir atkarīgas no tā, cik daudz loku ir e-pakalpojuma tīmekļa pakalpju grafā un cik izmaksā izmaiņas. Jo vairāk ir loku un vairāk ir metožu vienā tīmekļa pakalpē, jo grūtāk ir veikt izmaiņas, kā arī ir potenciāli vairāk iespēju rasties kļūdām, kas palielina uzturēšanas izmaksas. Uzturēšanas izmaksas  $E$  aprēķina visam tīmekļa pakalpju grafam.

$$E = k \times \frac{\sum_{m_{s_i} \in M_{s_i}} O(m_{s_i})}{n} \quad (4)$$

$k = |L|$  ir loku skaits grafā,  $n = |S|$  ir tīmekļa pakalpju (virsoņu) skaits grafā un  $M_{s_i} = \{m_{s_i}^0, m_{s_i}^1, m_{s_i}^2, \dots, m_{s_i}^g\}$  ir grafa virsošnes (tīmekļa pakalpes) darbības implementējošās metodes.  $O(m_{s_i}^j)$  ir un katras metodes  $m_{s_i}^j$  koda rindiņu skaits. [12] [100] [109] [110].

*Atkārtota izmantojamība.* Tīmekļa pakalpju izstrāde e-pakalpojumu izveidei ir darbietilpīgs process. Izmaksas ir pietiekoši lielas, un ne vienmēr e-pakalpojumam sākotnēji ir tāds lietotāju skaits, lai īstermiņa tas atmaksātos. Tīmekļa pakalpes ir nepieciešams veidot tā, lai tās būtu lietderīgas, skatoties īstermiņa un arī ilgtermiņā. Šim nolūkam ir nepieciešams pakalpes veidot tā, lai tās var maksimāli izmantot atkārtoti. Ir nepieciešams veidot pakalpes maksimāli neatkarīgas no citiem resursiem, lai tās varētu lietot dažādiem mērķiem, kur nepieciešama jau iepriekš izveidotā pakalpes funkcionalitāte. Runājot par tīmekļa pakalpēm, tiek pieņemts, ka tās atbilst SOA, tātad tās ir maksimāli neatkarīgas no citām, nesatur stāvokļa informāciju un var strādāt autonomi. Tas nozīmē, ka ir iespējams novērtēt izveidotā tīmekļa pakalpju grafa savstarpējo sasaisti un atkarību no citām tīmekļa pakalpēm. Jo mazāka tīmekļa pakalpju atkarība no citām tīmekļa pakalpēm, jo lielāka iespēja tās atkārtoti izmantot citā e-pakalpojumā. Lai aprēķinātu atkārtotu izmantojamību  $R$ , izmanto vidējās tīmekļa pakalpju atkarības aprēķinu no citām pakalpēm ARSD (Average Required Service Dependency) tīmekļa pakalpju grafā. Šim nolūkam izmanto sekojošu formulu:

$$R = \frac{1}{n} \times \sum_{i=1}^n R_i \quad (5)$$

$n = |S|$  ir kopējais tīmekļa pakalpju (virsoņu) skaits tīmekļa pakalpju grafā un  $R_i$  ir ar tīmekļa pakalpi  $i$  saistīto pakalpju skaits, jeb, citiem vārdiem sakot, ienākošo un izejošo loku skaits tīmekļa pakalpju grafa virsošnē  $R_i = |\Gamma s_i| + |\Gamma^{-1} s_i|$  [82] [100] [109] [110].

*Integritāte.* Tā kā tīmekļa pakalpju grafs no praktiskās implementācijas viedokļa var tikt sadalīts tīklā (katra no tīmekļa pakalpēm var tikt darbināta uz atsevišķa aplikāciju servera), kur katrs pakalpojuma solis ir jāizsauc attālināti, tas var ietekmēt pakalpojuma integritāti  $I$ , ja nav sasniedzama vai neizpildās kāda no tīmekļa pakalpēm. No šāda viedokļa e-pakalpojuma integritāte būs vislabākā, ja visu funkcionalitāti realizēs viena tīmekļa pakalpe un nenotiks datu pārsūtīšana starp vairākām (minimāls loku skaits). Integritāti  $I$  aprēķina pēc sekojošas formulas:

$$I = k \quad (6)$$

$k = |L|$  ir loku skaits grafā [100] [109] [110].

Segmentējot algoritma grafu, tiek iegūti tīmekļa pakalpju grafi, kurus var izmantot par pamatu e-pakalpojumu sistēmas arhitektūras izveidē. Izvēloties pieņemamu e-pakalpojumu sistēmas arhitektūru, ir jāņem vērā vairāki kritēriji, kuri bieži vien ir pretrunīgi. Tas nozīmē, ka ir nepieciešams atrast tīmekļa pakalpju grafu kopā  $X$  (kopa ir veidojusies, segmentējot algoritmu grafu), kas ir vislabākais (vai arī ne sliktāks) pēc visiem izvirzītajiem kritērijiem, uz kura pamata var izveidot pieņemamu e-pakalpojumu sistēmas arhitektūru. Būtiski ir aplūkot visus kritērijus vienlaicīgi, jo nevar pateikt, kurš no tiem ir svarīgāks, kā arī tie savstarpēji nav salīdzināmi. Šo tīmekļa pakalpju grafa atrašanas uzdevumu var reducēt kā daudzkriteriālas optimizācijas uzdevumu. Daudzkriteriāla optimizācijas rezultātā tiek meklēta Pareto kompromisu kopa, ko formāli var definēt sekojoši:

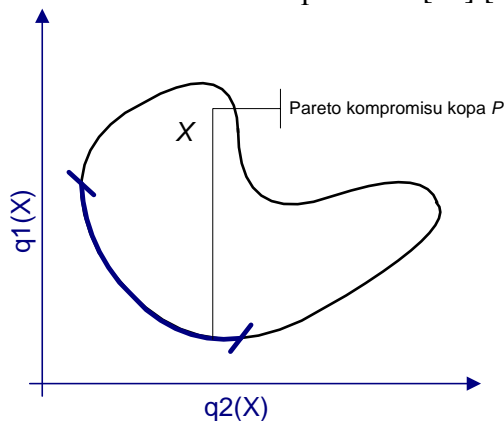
$$Q(X) \rightarrow \min_{X \in \Omega} \rightarrow P \quad (7)$$

$Q(X) = \{q_1(X), q_2(X), \dots, q_N(X)\}$  ir minimizējamie kritēriji;  $\Omega - X$  definīcijas apgabals. Risinājumu  $X_i \in \Omega$  var saukt par pareto optimālu tad un tikai tad, ja neeksistē  $X_j \in \Omega$  tāds, kad

$$q_i(X_j) \leq q_i(X_i) \quad (8)$$

visiem  $i = \{1, 2, \dots, N\}$ , kur vismaz viena ir stingra nevienādība. Citiem vārdiem sakot, jebkurai vērtībai  $X_i \in P$  kopā  $\Omega$  nav atrodama labāka vērtība par izvēlēto pēc katra no optimizējamajiem

kritērijiem. Grafiski attēlojot Pareto kompromisu kopu pēc diviem optimizējamajiem kritērijiem, tie ir visi risinājumi, kas atrodas tuvāk koordināšu sākumpunktam [67] [115].



5. att. Pareto kompromisu kopa

Lai varētu savstarpēji salīdzināt un vienlaicīgi aplūkot dažādus minimizējamus kritērijus, tos ir nepieciešams normalizēt, lai iegūtās vērtības būtu vienāda lieluma. Optimizējamo mērķa funkciju vērtību normalizēšanai katru aprēķināto vērtību daļa ar attiecīgā kritērija maksimālo aprēķināto, ne nulles vērtību. Šādā gadījumā vērtību apgabals katrai no mērķa funkcijām ir apgabalā  $[0,1]$ . Vērtību normalizēšanai tiek izmantota formula:

$$q_i(X_j) = \frac{q_i(X_j)}{\max_{X \in \Omega} q_i(X)} \quad (9)$$

Tīmekļa pakalpojuma grafā izvēles gadījumā Pareto kompromisu kopas pielietojums ir sekojošs. Ir dots e-pakalpojuma algoritma grafis. Sākotnēji tas tiek pieņemts par tīmekļa pakalpojuma grafu  $G$ . Uzdevums ir atrast tīmekļa pakalpojuma grafu kopu  $P = \{G'^*\}$ , kas ir definēta apgabalā  $\Omega$  (t.i. visas iespējamās algoritma grafā segmentu kombinācijas). Minimizē kritērijus  $Q$  (mūsu gadījumā  $q_1$  ir izstrādes izmaksas  $C$ , ko aprēķina, izmantojot formulu (2),  $q_2$  ir ātrdarbība  $T$ , ko aprēķina, izmantojot formulu (3),  $q_3$  ir uzturēšanas izmaksas  $E$ , ko aprēķina, izmantojot formulu (4),  $q_4$  ir atkārtota izmantojamība  $R$ , ko aprēķina, izmantojot formulu (5) un  $q_5$  ir integritāte  $I$ , ko aprēķina, izmantojot formulu (6)).

*E-pakalpojumu sistēmas arhitektūras projektēšanas metode* sastāv no sekojošiem soļiem [100] [109] [110]:

1. Sākotnēji ir nepieciešams izveidot e-pakalpojuma algoritma grafu  $G$ , kas apraksta projektējamo e-pakalpojumu;
2. E-pakalpojuma algoritma grafā ir nepieciešams noteikt iespējamus ierobežojumus virsotņu savstarpējā segmentēšanā. Ja e-pakalpojuma algoritma grafā virsotņu skaits pārsniedz 20 un nav neviena ierobežojuma, tad ir nepieciešams izdalīt e-pakalpojuma grafā segmentus un katru no segmentiem apstrādāt kā atsevišķu projektēšanas uzdevumu;
3. No e-pakalpojuma algoritma grafā rekursīvas segmentēšanas rezultātā tiek iegūti tīmekļa pakalpojuma grafi  $X = \{G'\}$ , kas ir par pamatu e-pakalpojumu sistēmas arhitektūras izveidē. Par sākotnējo tīmekļa pakalpojuma grafu tiek pieņemts e-pakalpojuma algoritma grafis;
4. Iegūtajiem tīmekļa pakalpojuma grafiem tiek aprēķinātas kvalitātes rādītāju skaitliskās vērtības, izmantojot aprēķina formulas;
5. No visiem tīmekļa pakalpojuma grafiem tiek iegūta Pareto kompromisu kopa  $P = \{G'^*\}$ ;
6. Atrastās kopas  $P$  tīmekļa pakalpojuma grafi var kalpot par pamatu pieņemamas e-pakalpojumu sistēmas arhitektūras izvēlei. Tīmekļa pakalpojuma grafus, kas ietilpst atrastajā Pareto kompromisu kopā, var detalizēti projektēt un implementēt e-pakalpojumu sistēmas arhitektūrā;
7. Pareto kompromisu kopa  $P$  vairumā gadījumu satur vairāk kā vienu risinājumu. Ja atrastā kopa ir pietiekoši maza, tad tīmekļa pakalpojuma grafu izvēli no šīs kopas var atstāt sistēmas

projektētāja ziņā. Savukārt, ja iespējamo risinājumu skaits ir liels, tad, lai varētu izvēlēties konkrētu tīmekļa pakalpju grafu, var tikt pielietota kritēriju samazināšana vai apvienošana ar kādu citu optimizācijas vai projektēšanas metodi, piemēram, svaru metodi vai Jan Bosch metodi.

Datorsistēmu projektēšanas metožu salīdzinājums (1. tabula 16. lpp.) ļauj sistēmu arhitektiem izvēlēties konkrētajā gadījumā pielietojamās metodes lietderību, kā arī atklāj katras projektēšanas metodes iespējamus plusus un mīnus.

Darbā aplūkoto datorsistēmu projektēšanas metožu salīdzinājums ir veikts pēc šādiem novērtēšanas kritērijiem:

1. Daudzkriteriāla – izvēloties projektēšanas metodi ir būtiski ņemt vērā, vai vienlaicīgi tiek aplūkoti visi sistēmai izvirzītie kvalitātes kritēriji;
2. Izslēgts eksperta subjektīvais viedoklis – daudzās projektēšanas metodēs risinājuma izvēle ir atkarīga no sistēmas arhitekta, kura subjektīvais viedoklis ietekmē topošo sistēmu;
3. Iespējama iepriekšēja novērtēšana – ņemot vērā, ka sistēmas izstrāde ir laikietilpīga, ir būtiski tās kvalitātes rādītājus novērtēt vēl pirms pašas sistēmas veidošanas;
4. Garantēts Pareto – ja risinājums tiek izvēlēts no visiem risinājumiem, kas atrodas Pareto kompromisu kopā, tad ir pārliecība par to, ka izvēlētais risinājums ir viens no iespējami labākajiem;
5. Konkrēts risinājums – būtiski ir pārliecināties par to, vai projektēšanas metode sniedz vienu vai vairākus risinājumus;
6. Risinājuma precizitāte: eksperts – risinājuma precizitāti nosaka sistēmas arhitekta pieredze, metodika – risinājuma precizitāti nosaka projektēšanas metodi aprakstošās metodikas ievērošana, formula – risinājuma precizitāti nosaka projektēšanā izmantojamo formulu precizitāte.

Kā redzams dotajā salīdzinājumā (1. tabula 16. lpp.), tad eksistē metodes, kas ir daudzkriteriālas, un ir iespējams jau pirms izstrādes novērtēt sistēmas arhitektūras kvalitātes rādītājus, bet šo metožu precizitāti nosaka tikai eksperta pieredze un metodikas ievērošana. Tas nozīmē, ka netiek izslēgta iespēja, ka tiks izvēlēts lokāls minimums. Trūkums šīm metodēm ir arī tas, ka netiek izslēgts eksperta subjektīvais viedoklis, jo vairumā gadījumu sistēmu projektēšanā ir iesaistīts tikai viens sistēmas arhitekts, kas pieņem lēmumu. Ņemot vērā to, ka datorsistēmu veiksmīgu attīstību nosaka to kvalitāte, tad projektēšanas metodes, kas nav daudzkriteriālas, vairumā risinājumu nedod vēlamu rezultātu. Šo metožu pielietojums var radīt zaudējumus gan datorsistēmu pasūtītājam, gan to izstrādātājam.

Esošās daudzkriteriālās projektēšanas metodes ir pieņemamas lielu un sarežģītu datorsistēmu projektēšanai, kuras nav iespējams formāli aprakstīt un kurām nav iespējams pielietot matemātiskas daudzkriteriālas optimizācijas metodes. Šādos gadījumos labi varētu darboties ATAM metode, kur ir iesaistīti vairāki eksperti sistēmas projektēšanā. Arī citas aplūkotās metodes varētu būt lietderīgas lielu datorsistēmu projektēšanā, ja tiek iesaistīti kompetenti attiecīgās nozares eksperti, kas var pieņemt lēmumus un novērtēt veidojamās datorsistēmas kvalitātes rādītājus.

Tomēr e-pakalpojumu sistēmu arhitektūras projektēšanā ir nepieciešama projektēšanas metode, kas sniegtu ātru un kvalitatīvu rezultātu neatkarīgi no projektēšanā iesaistīto ekspertu kvalifikācijas un skaita. Pateicoties tam, ka e-pakalpojuma algoritmu ir iespējams formalizēti pierakstīt algoritma grafa veidā, ir iespējams pielietot piedāvāto tīmekļa pakalpju projektēšanas metodi un pieņemamas e-pakalpojumu sistēmas arhitektūras izvēli, kas ir balstīta uz daudzkriteriālu optimizāciju. Šādos un līdzīgos gadījumos ir ieteicams lietot tieši daudzkriteriālas optimizācijas metodes, lai izvairītos no lokāla minimuma.

Ņemot vērā ka daudzkriteriālas optimizācijas rezultātā tiek iegūta Pareto kompromisu kopa, kas satur vairākus rezultātus, tad sarežģītākos gadījumos, kad sistēmas arhitektam nav iespējams veikt izvēli no Pareto kompromisu kopas, var pielietot jebkuru citu projektēšanas, lēmumu pieņemšanas vai optimizācijas metodi, kas palīdz izvēlēties risinājumu, no Pareto kompromisu kopas. Šādā gadījumā tiek izslēgta varbūtība, ka tiks izvēlēts lokāls minimums.



## Arhitektūras projektēšanas metožu salīdzinājums

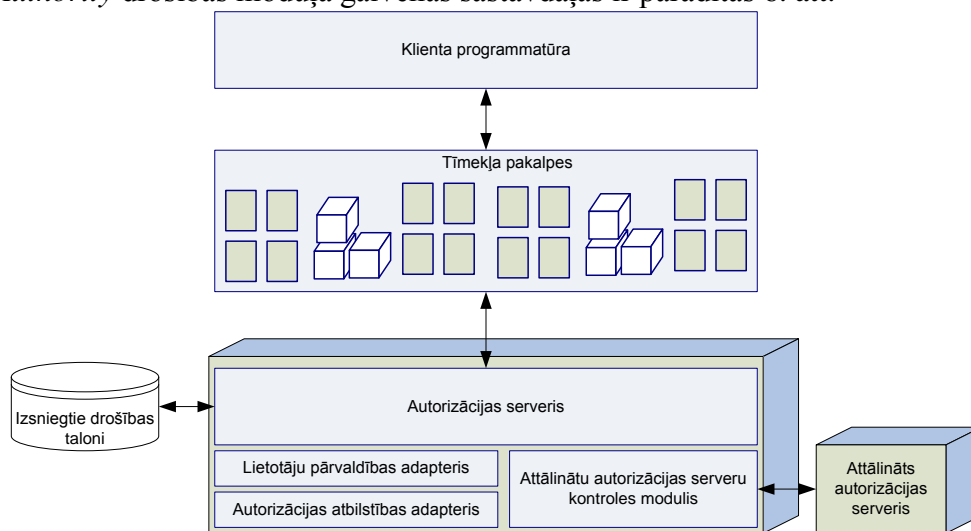
Metode	Daudzkriteriāla	Izslēgts eksperta subjektīvais viedoklis	Iespējama iepriekšēja novērtēšana	Garantēts Pareto	Konkrēts risinājums	Risinājuma precizitāte
Funkcionāli bāzēta	Nē	Nē	Nē	Nē	Jā	Eksperts; Metodika
Objektu orientēta	Nē	Nē	Nē	Nē	Jā	Eksperts; Metodika
ADD	Jā	Nē	Jā	Nē	Jā	Eksperts; Metodika
SAAM	Jā	Nē	Jā	Nē	Jā	Eksperts; Metodika
Jan Bosch	Jā	Nē	Jā	Nē	Jā	Formulas; Eksperts; Metodika
ATAM	Jā	Daļēji, jo ir iesaistīti vairāki eksperti.	Jā	Nē	Jā	Eksperti; Metodika
HASARD	Daļēji, jo metode vairāk ir orientēta uz drošības prasību sasniegšanu.	Nē	Jā	Nē	Jā	Eksperts; Metodika
Daudzkriteriāla, Pareto kompromisu kopas atrašana	Jā	Jā	Jā	Jā	Nē	Formulas
Kombinēta daudzkriteriāla	Jā	Daļēji, jo risinājuma izvēlei no Pareto kompromisu kopas ir nepieciešama papildus analīze.	Jā	Jā	Jā	Formulas; Eksperti; Metodika

**Trešajā nodaļā** ir aprakstīts e-pakalpojumu sistēmas drošības risinājums.

E-pakalpojumu drošība ir saistīta ar e-pakalpojumu pieejamību, konfidencialitāti un integritāti. Tā ir atkarīga no kopējās e-pakalpojumu sistēmas arhitektūras. Atkārtotas, vienotas, uzticamas drošības sistēmas pielietošana visos e-pakalpojumos ļauj ietaupīt resursus katrā nākamajā e-pakalpojuma projektēšanā, realizācijā un ekspluatācijā, jo to nav nepieciešams izstrādāt katram e-pakalpojumam individuāli.

Jebkuras drošības sistēmas pielietojums datorsistēmām rada tām papildus slodzi datu apstrādei, galvenokārt, ietekmējot kopējo datorsistēmas ātrdarbību. Runājot par SOA un e-pakalpojumu sistēmām, ir nepieciešams aplūkot arī atbilstošās drošības sistēmas darbību datortīklā, kur katra no e-pakalpojuma darbību nodrošinošām tīmekļa pakalpēm atrodas attālināti. Papildus tam ir nepieciešams risināt problēmas, kas saistītas ar dažādu datorsistēmu un ar tām saistīto autorizācijas datu apvienošanu vienotā e-pakalpojumā. Pielietojot jebkuru drošības sistēmu, ir jārisina arī e-pakalpojumu pieejamības jautājums, jo drošības sistēmas darbība nedrīkst samazināt e-pakalpojumu pieejamību.

Lai realizētu drošību e-pakalpojumos, ar autora līdzdalību ir izveidots un patentēts [107] drošības sistēmas modulis – *Authority*. Šis drošības modulis ir izveidots pietiekami universāls, un to var izmantot ne tikai e-pakalpojumu realizācijā, bet arī jebkurā SOA uz tīmekļa pakalpēm bāzētā aplikācijā. *Authority* drošības moduļa galvenās sastāvdaļas ir parādītas 6. att.



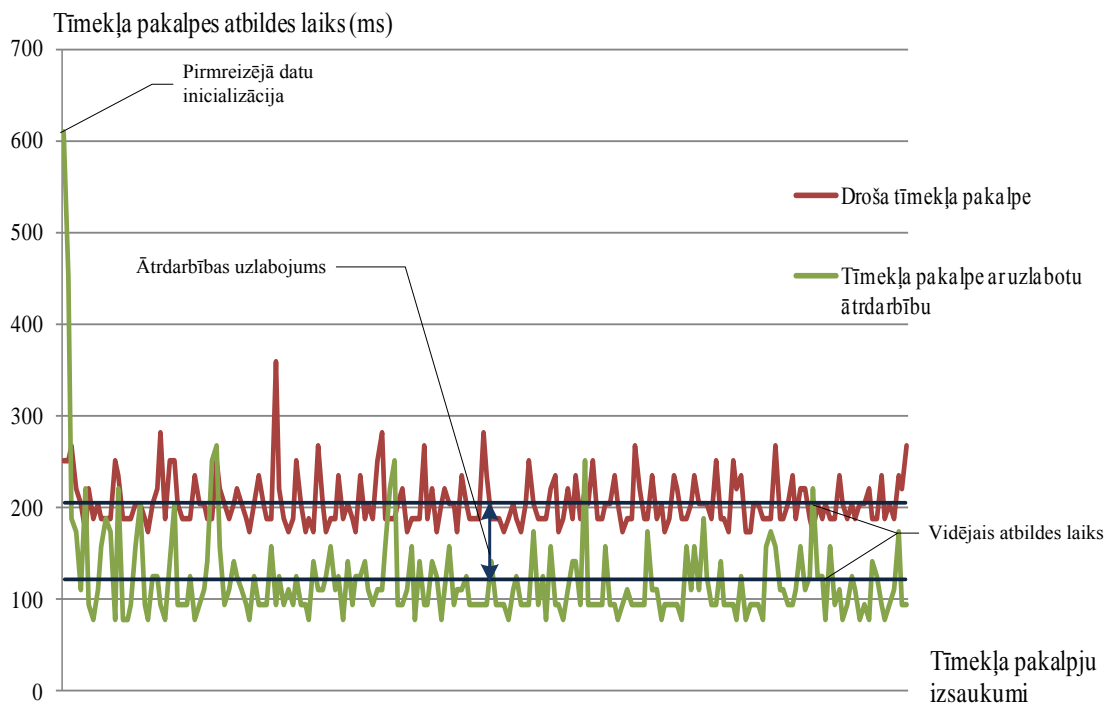
**6. att. Drošības sistēmas *Authority* sastāvdaļas**

Drošības sistēmas *Authority* darbība ir sekojoša. Lietotājs autorizējas darbam datorsistēmas klienta programmatūrā un iegūst drošības talonu. Klienta programmatūra izsauc tīmekļa pakalpes, lai iegūtu nepieciešamo informāciju. Informācijas pieprasījumam tiek pievienots drošības talons, ar kuru nosūtāmie dati tiek elektroniski parakstīti un nepieciešamības gadījumā arī šifrēti. Ja drošības talons netiek pievienots, tad informācijas pieprasījuma apstrāde nenotiek. Tīmekļa pakalpe pārbauda drošības talonu sev piesaistītajā autorizācijas serverī. Autorizācijas serveris pārbauda drošības talona derīgumu un tā statusu izsniegto drošības talonu datu bāzē. Ja drošības talons datu bāzē nav atrodams un tas ir izsniegts citā attālinātā autorizācijas serverī, tad drošības talons tiek nosūtīts pārbaudei uz attālinātu autorizācijas serveru kontroles moduli. Attālinātu autorizācijas serveru kontroles modulis pieprasa drošības talona izsniedzējam - attālinātam autorizācijas serverim, pārbaudīt drošības talona derīgumu. Ja drošības talons ir derīgs, tad attālinātu autorizācijas serveru kontroles modulis izsniedz jaunu drošības talonu lokālā autorizācijas serverī, pierēģistrē to izsniegto drošības talonu datu bāzē un nodod autorizācijas serverim. Autorizācijas serveris nosūta drošības talonu autorizācijas atbilstības adapterim, lai papildinātu jaunizveidoto drošības talonu ar autorizācijas datiem, kas nepieciešami, lai darbinātu izsaukto tīmekļa pakalpi. Autorizācijas atbilstības adaptera uzdevums ir sagatavot autorizācijas datus atbilstoši izsauktās tīmekļa pakalpes prasībām. Ja autorizācijas serveris saņem pārbaudei drošības talonu, kas satur citādus autorizācijas datus nekā nepieciešams klienta programmatūras izsauktajai tīmekļa pakalpei,

tad tas nosūta drošības talonu uz autorizācijas atbilstības adapteri, lai sagatavotu autorizācijas datus tādus, kā nepieciešami tīmekļa pakalpei. Ja nepieciešams, tad dati tiek iegūti no lietotāju pārvaldības moduļa. Papildinātais drošības talons tiek atgriezts tīmekļa pakalpei. Tīmekļa pakalpe kontrolē autorizācijas datus un atgriež datorsistēmas klienta programmatūrai tīmekļa pakalpei prasīto informāciju.

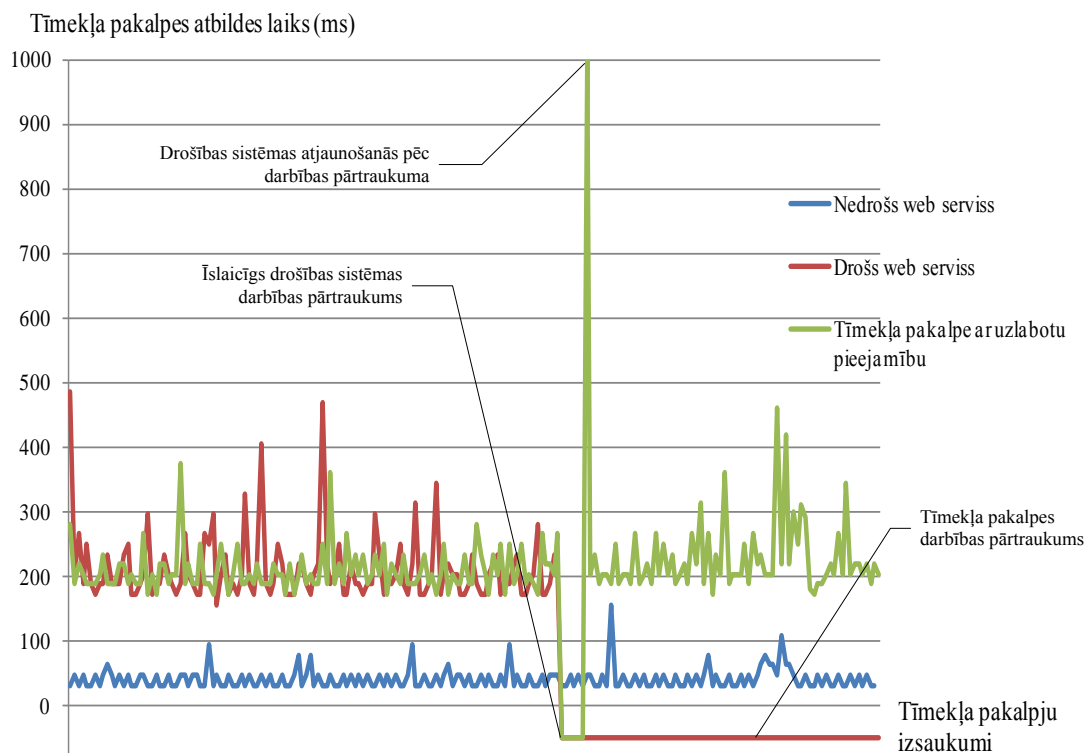
Pateicoties tam, ka autorizācijas serverim ir pievienots autorizācijas atbilstības adapteris, kas nodrošina autorizācijas datu atbilstības sagatavošanu, ir iespējams no vienas datorsistēmas klienta programmatūras izsaukt dažādu e-pakalpojumu tīmekļa pakalpes.

Attālinātu autorizācijas serveru kontroles modulis nodrošina sasaisti ar citiem attālinātiem autorizācijas serveriem. Tādā veidā ir iespējams uzlabot datorsistēmu ātrdarbību un pieejamību, jo tiek veidoti lokāli drošības taloni un nav jāvēršas caur tīklu pie attālināta autorizācijas servera. Ātrdarbības uzlabojums, lokalizējot drošības talonus, parādīts 7. att.



**7. att. Drošības talonu lokalizācijas ietekme uz sistēmas ātrdarbību**

Izsniegto drošības talonu datu bāze ļauj veidot autorizācijas serveru pudurus ar kopēju drošības talonu glabātuvī, tādā veidā palielinot kopējo drošības sistēmas pieejamību un ātrdarbību, kas ļauj palielināt autorizācijas servera pieejamību. 8. att. ir atspoguļots tīmekļa pakalpojumu izsaukumu salīdzinājums, ja tiek izmantota datu bāze un ja tā netiek izmantota. Gadījumos, ja ir īslaicīgs autorizācijas servera darbības pārtraukums, tad pēc darbības atjaunošanās datu bāzes pielietojuma gadījumā drošības talona dati tiek ielasīti servera operatīvajā atmiņā, un tīmekļa pakalpojumu izsaukumi var tikt veikti ar esošo drošības talonu.

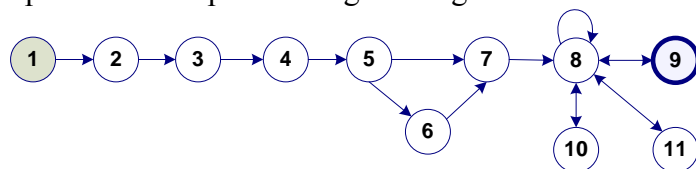


8. att. Drošības sistēmas pieejamības uzlabojums

**Ceturtajā nodaļā** ir aprakstīti divi e-pakalpojumu piemēri, kuru projektēšana ir balstīta uz pieņemamas arhitektūras izvēli un tīmekļa pakalpju projektēšanu, izmantojot daudzkritēriālu optimizāciju, kā arī drošības sistēmas *Authority* pielietojumu. Aprakstītie piemēri ir balstīti uz pašvaldību informācijas sistēmas un pašvaldību e-pakalpojumu pamata. E-pakalpojumu sistēmu arhitektūras izvēlei tiek izmantota programmatūra, kura realizē algoritma grafa segmentēšanu un tīmekļa pakalpju grafu novērtēšanu, kā arī veic Pareto kompromisu kopas atrašanu un papildus svaru metodes pielietojumu. Nodaļā ir aprakstīta arī minētas sistēmas arhitektūras izvēles programmatūras sistēmas struktūra un darbības algoritmi.

Piemēros pieņemamas e-pakalpojumu sistēmas izvēles novērtēšanai ir izvirzīti pieci kvalitātes kritēriji, kuru skaitlisko vērtību aprēķināšanai tiek izmantotas otrajā nodaļā dotās kvalitātes kritēriju aprēķina formulas: izstrādes izmaksas, ātrdarbība, uzturēšanas izmaksas, atkārtotas izmantojamības iespējas un integritāte.

Pirmā e-pakalpojuma piemēra pamatā ir darba plūsma, kuras rezultātā tiek iegūta izziņa par zemes gabala perspektīvo izmantošanu. Piemērā ir veikta e-pakalpojuma sistēmas arhitektūras projektēšana, ņemot par pamatu 9. att. parādīto algoritma grafu.



9. att. Izziņas par zemes gabala perspektīvo izmantošanu algoritma grafs

Dotajā piemērā, izrēķinot visas iespējamās virsotņu segmentu kombinācijas, tiek iegūti **1015** tīmekļa pakalpju grafi. Katram no iegūtajiem tīmekļa pakalpju grafiem tiek aprēķinātas kvalitātes kritēriju novērtējuma vērtības, kuras tiek ņemtas par pamatu Pareto kompromisu kopas aprēķinam, kas parādīta sekojošā tabulā (2. tabula).

2. tabula

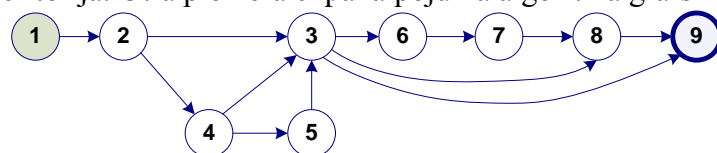
Izziņas par zemes gabala perspektīvo izmantošanu tīmekļa pakalpju grafu Pareto kompromisu kopa

Nr.	Grafa apzīmējums	Izstrādes izmaksas	Ātrdarbība	Uzturēšanas izmaksas	Atkārtotas izmantojamības iespējas	Integritāte
1.	(1,2,3,4,7,8),5,(6,10,11),9	0.90581	0.80969	1.00000	0.89286	0.46667

2.	(1,2,3,4),5,(6,10,11),(7,8),9	0.91986	0.81521	0.80100	0.71429	0.46667
3.	(1,2,3,4),5,6,(7,8),9,(10,11)	0.93369	0.83918	0.66904	0.67460	0.53333
4.	1,(2,3,4),5,6,(7,8),9,(10,11)	0.94732	0.86393	0.57498	0.64626	0.60000
5.	1,2,(3,4),5,6,(7,8),9,(10,11)	0.96076	0.88853	0.50462	0.62500	0.66667
6.	1,2,3,4,5,6,(7,8),9,(10,11)	0.97402	0.91246	0.45005	0.60847	0.73333
7.	1,2,3,4,5,6,7,8,9,(10,11)	0.98709	0.95604	0.40725	0.61905	0.86667
8.	1,2,3,4,5,6,7,8,9,10,11	1.00000	1.00000	0.37249	0.64935	1.00000

Aplūkojot tīmekļa pakalpju grafus, kas ir iegūti daudzkritēriālas optimizācijas rezultātā (2. tabula) ir redzams, ka ir iegūti tīmekļa pakalpju grafi, ko var iedalīt trīs lielās daļās. Pirmkārt, Pareto kompromisu kopā ietilpst tīmekļa pakalpju grafs, kurā ir apvienotas maksimāli visas iespējamās virsotnes vienkopus (Nr. 1.). Otrkārt, ietilpst tīmekļa pakalpju grafs, kas satur maksimāli daudz virsotnes un atbilst algoritma grafam (Nr. 8.). Treškārt, ietilpst tīmekļa pakalpju grafi, kur ir apvienotas dažas algoritma grafā virsotnes. Analizējot iegūto Pareto kompromisu kopu, projektētājam ir jāpieņem lēmums par labu kādam no tīmekļa pakalpju grafiem pieņemamas e-pakalpojumu sistēmas arhitektūras realizāciju.

Otrā e-pakalpojuma piemēra pamatā ir darba plūsma, kuras rezultātā tiek saskaņota reklāma vai izkārtnes pašvaldības teritorijā. Otrā piemēra e-pakalpojuma algoritma grafs ir parādīts 10. att.



10. att. Reklāmas un izkārtnes saskaņošanas algoritma grafs

Izrēķinot visas iespējamās virsotņu segmentu kombinācijas otrajā piemērā dotajam e-pakalpojumu algoritmu grafam, tiek iegūti **877** tīmekļa pakalpju grafi. Tāpat kā pirmajā piemērā katram no iegūtajiem tīmekļa pakalpju grafiem tiek aprēķinātas novērtējuma vērtības un aprēķināta Pareto kompromisu kopa, kas parādīta sekojošā tabulā (3. tabula).

3. tabula

Reklāmas un izkārtnes saskaņošanas tīmekļa pakalpju grafu Pareto kompromisu kopa

Nr.	Grafa apzīmējums	Izstrādes izmaksas	Ātrdarbība	Uzturēšanas izmaksas	Atkārtotas izmantojamības iespējas	Integritāte
1.	1,2,3,4,5,6,7,8,9	1.00000	1.00000	0.33605	0.60386	1.00000
2.	1,2,(3,4),5,6,7,8,9	0.99281	0.89371	0.37719	0.57065	0.83333
3.	1,2,(3,5),4,6,7,8,9	0.99281	0.89371	0.37719	0.57065	0.83333
4.	1,2,3,(4,5),6,7,8,9	0.99281	0.89371	0.37719	0.57065	0.83333
5.	1,2,(3,4,5),6,7,8,9	0.98556	0.78729	0.43023	0.52795	0.66667
6.	1,2,(3,4,5,8),6,7,9	0.97826	0.71213	0.50111	0.47101	0.50000
7.	1,2,(3,4,5,7,8),6,9	0.97091	0.70053	0.60079	0.47826	0.41667
8.	1,2,(3,4,5,6,7,8),9	0.96350	0.66708	0.75018	0.38043	0.25000
9.	(1,3,4,5,6,7,8),2,9	0.95604	0.70769	1.00000	0.50725	0.25000

Analizējot abos piemēros iegūtos rezultātus un salīdzinot tos ar jau realizētiem e-pakalpojumiem, ir redzams, ka projektēšanas metode ir pielietojama arī praksē, un tai nav redzamu trūkumu, kas izslēgtu kādus būtiskus risinājumus (izmantojamajās projektēšanas metodēs atšķirība ir resursos, kas nepieciešami pieņemamas sistēmas arhitektūras radīšanai).

Drošības sistēmas *Authority* pielietojums ir aplūkots, balstoties uz lietotāju autorizācijas mehānismu pašvaldības portālā, izmantojot mobilā e-paraksta autorizāciju ar Mabilo ID. Piemērā minētais drošības sistēmas pielietojums parāda, ka drošības sistēmu *Authority* ir iespējams pielietot arī pietiekami sarežģītos risinājumos. Par pamatu šādai pielietošanai ir piedāvātās drošības sistēmas modularitāte, kas dod iespēju paplašināt esošo tās funkcionalitāti.

**Pielikumos** ir dots pieņemamas sistēmas arhitektūras izvēles programmatūras lietošanas apraksts, kā arī ceturtajā nodaļā aprakstīto e-pakalpojumu piemēru algoritma grafu pieraksts UML stāvokļu diagrammas XMI formātā, kas var kalpot kā ieejas parametri minētās programmatūras darbībai.

## DARBA REZULTĀTI

E-pakalpojumu sistēmu projektēšanā ir jārisina problēmas, kas saistītas ar sistēmai izvirzītajiem kvalitātes kritērijiem. Parasti e-pakalpojumu sistēmai tiek izvirzīti vairāki savstarpēji pretrunīgi kvalitātes kritēriji. Šādos gadījumos ir nepieciešams meklēt kompromisa risinājumu, izvēloties pieņemamu e-pakalpojumu sistēmas arhitektūru. Šobrīd ir zināmas vairākas projektēšanas metodes, kas sistēmu projektēšanā ņem vērā tai izvirzītos kvalitātes prasību kritērijus. Esošajām metodēm ir vairāki trūkumi. Pirmkārt, tās neaplūko visus iespējamās sistēmas arhitektūras variantus. Tas nozīmē, ka ne vienmēr tiks izvēlēts labākais iespējamais risinājums. Otrkārt, visās aplūkojamajās metodēs netiek izslēgts sistēmas arhitekta subjektīvais viedoklis. Šo problēmu risināšanai ir iespējams izmantot tīmekļa pakalpojumu projektēšanas metodi, kas balstīta uz daudzkritēriju optimizāciju, kura atrod kompromisa rezultātu, samazinot subjektīvo faktoru. Darbā ir piedāvāta šāda e-pakalpojumu sistēmu arhitektūras izvēles metode.

E-pakalpojumiem ir nepieciešams kontrolēt e-pakalpojuma autorizētu un integrētu darbību visos e-pakalpojuma izpildes posmos. Ieviešot e-pakalpojumu drošības sistēmu, ir jārisina e-pakalpojumu pieejamības un ātrdarbības problēmas. Šim nolūkam ar autora līdzdalību ir izstrādāta un darbā aprakstīta e-pakalpojumu drošības sistēma *Authority*, kurā ir iespējams veidot dažādus autentifikācijas līmeņus, kā arī konfigurēt pašu sistēmu, lai palielinātu e-pakalpojumu pieejamību un integritāti.

Promocijas darba gaitā ir atrisināti vairāki uzdevumi:

1. Izstrādāta e-pakalpojumu sistēmas arhitektūras izvēles metodika, kura balstās uz formalizētu pieeju un kura ļauj izmantot matemātiskās optimizācijas metodes. Izstrādātā e-pakalpojumu sistēmas arhitektūras izvēles metodika salīdzināta ar vairākām projektēšanas metodēm: ADD, SAAM, Jan Bosch, ATAM, HASARD;
2. Izstrādāts uzlabots drošības sistēmas risinājums, kas paaugstina sistēmas pieejamību un ātrdarbību. Veiktie uzlabojumi eksperimentāli salīdzināti ar Microsoft korporācijas piedāvāto drošības risinājumu;
3. Izstrādātā metodika un drošības risinājums verificēts ar praktisku piemēru palīdzību;

Promocijas darbam izvirzītais mērķis ir sasniegts, kā arī ir iezīmētie iespējamie nākotnes pētījumu virzieni.

## TĀLĀKO PĒTĪJUMU VIRZĪBA

Piedāvāto e-pakalpojumu sistēmu projektēšanas metodiku ir iespējams pielietot arī citām sistēmām, kas nav algoritmiski sarežģītas (tādas, lai ir iespējams uzbūvēt algoritma grafu) un ir balstītas uz SOA un tīmekļa pakalpēm. Savukārt, piedāvāto drošības sistēmu ir iespējams izmantot pilnīgi visās (arī sarežģītās) sistēmās, kas ir balstītas uz SOA un tīmekļa pakalpēm, tajā skaitā savstarpēji integrētās datorsistēmās.

Nākotnē piedāvāto projektēšanas metodi iespējams attīstīt, to izmantojot dažādās automatizētās e-pakalpojumu sistēmās, kur e-pakalpojuma izpildē iesaistīto tīmekļa pakalpojumu izvēle notiek izmantojot mākslīgo intelektu, lai pārliecinātos par izpildāmā e-pakalpojuma kvalitāti [51] [64]. Papildus ir iespējams attīstīt arī izmantojamo tīmekļa pakalpojumu grafu izvēles metodiku, pielietojot arī citas daudzkritēriālas lēmumu pieņemšanas metodes (piemēram Fuzzy loģiku, Ekspertu sistēmas u.c.) [1] [34] [44] [105]. Iespējams analizēt citu sistēmu projektēšanas metožu apvienošanu ar daudzkritēriālu optimizāciju [2] [62]. Nākotnē iespējams pētīt arī e-pakalpojumu aģentu bāzētu sistēmu izveidi, kas tiek balstītas uz daudzkritēriālu optimizāciju, kā arī drošības sistēmas iesaisti tajās [76].

## IZMANTOTĀS LITERATŪRAS SARAKSTS

1. Aalst W., Hirnschall A., Verbeek H. An Alternative Way to Analyze Workflow Graphs// Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02), volume 2348 of Lecture Notes in Computer Science. - Berlin: Springer-Verlag, 2002. - 535.-552. lpp.
2. Aiello M., Papazoglou M., Yang J., Carman M., Pistore M., Serafini L., Traverso P. A Request Language for Web-Services Based on Planning and Constraint Satisfaction// Tehnologies for E-Services Third International Workshop, TES 2002 Hong Kong, China, August 2002 Proceedings. - Berlin: Springer-Verlag, 2002. - 76.-85. lpp.
3. Alberts C., Dorofee A., Stevens J., Woody C. Introduction to the OCTAVE Approach. - Carnegie Mellon University , 2003.
4. Ambroszkiewicz S. enTish: An Approach to Service Composition// Tehnologies for E-Services 4th International Workshop, TES 2003 Berlin, Germany, September 2003 Proceedings. - Berlin: Springer-Verlag, 2003. - 168.-178. lpp.
5. Anglijas GovTalk / Internets. - <http://www.cabinetoffice.gov.uk/govtalk>
6. Arora S., Rao S., Vazirani U. Geometry, Flows, and Graph-Partitioning Algorithms// Communications of the ACM. - New York: ACM, 2008. - Nr. Vol.51 No. 10 - 96.-105. lpp.
7. Australian/New Zealand Standard AS/NZS 4360:2004 Risk management. - , 2004.
8. Bass L., Clements P., Kazman R. Software Architecture in Practice 2nd ed. - Boston: Addison-Wesley , 2003.
9. Bass L., Klein M., Bachmann F. Quality Attribute Design Primitives and the Attribute Driven Design Method// Lecture Notes In Computer Science. Revised Papers from the 4th International Workshop on Software Product-Family Engineering. - London: Springerferlag, 2001. - Nr. 2290 - 169.-186. lpp.
10. Benjamin A., Dustdar Schmit, Dustdar Schahram Systematic Design of Web Service Transactions// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 23.-33. lpp.
11. Biswas D., Vidyasankar K. Monitoring for Hierarchial Web Services Compositions// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 98.-112. lpp.
12. Bosch J., Bengtsson P. Assessing optimal software architecture maintainability// Fifth European Conference on Software Maintenance and Reengineering. - 2001. - 168.-175. lpp.
13. Bosch J. Design and Use of Software Architectures. - Harlow: Addison-Wesley , 2000.
14. Bradford L., Milliner S., Dumas M. Varying Resource Consumption to Achieve Scalable Web Services// Tehnologies for E-Services 4th International Workshop, TES 2003 Berlin, Germany, September 2003 Proceedings. - Berlin: Springer-Verlag, 2003. - 179.-190. lpp.
15. Cape Clear Software Inc. Principles of SOA Design. - Cape Clear , 2004.
16. Cardoso J., Sheth A., Miller J. Workflow Quality of Service// Proceedings of the IFIP TC5/WG5.12 International Conference on Enterprise Integration and Modeling Technique: Enterprise Inter- and Intra-Organizational Integration: Building International Consensus. - Deventer: Kluwer, 2002. - Nr. Vol. 236 - 303. - 311. lpp.
17. Cardoso J. Control-flow Complexity Measurement of Processes and Weyuker's Properties// Proceedings. 2005 IEEE International Conference Transactions on Engineering, Computing and Technology V8 October 2005. - 2005. - 213.-218. lpp.
18. Carnegie Mellon Software Engineering Institute: Software Architecture for Software-Intensive Systems / Internets. - <http://www.sei.cmu.edu/architecture/definitions.html>



19. Castellanos M., Casati F., Sayal M., Dayal U. Challenges in Business Process Analysis and Optimization// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 1.-10. lpp.
20. Chatterjee S., Webber J. Developing Enterprise Web Services An Architect's Guide. - New Jersey: Pearson Education , 2004.
21. Christophides V., Hull R., Karvounarakis G., Kumar A., Tong G., Xiong M. Beyond Discrete E-Services: Composing Session-Oriented Services in Telecommunications// Tehnologies for E-Services Second International Workshop, TES 2001 Rome, Italy, September 2001 Proceedings. - Berlin: Springer-Verlag, 2001. - 58.-73. lpp.
22. Cross Platform Network Authentication and Authorization Model / Patents Nr. US 7,178,163 B2 - ASV, publicēts 13.02.2007.
23. Dambītis J. Modernā grafu teorija. - Rīga: Datorzinību Centrs , 2002.
24. Dānijas OIO / Internets. - <http://www.oio.dk>
25. Erl T. Service-Oriented Architecture A Field Guide to Integrating XML and Web Services. - New Jersey: Prentice Hall , 2004.
26. Estonian X-road / Internets. - <http://www.eesti.ee/eng/xtee/>
27. European Commission Directorate General for Information Society and Media The User Challenge, Benchmarking the supply of online public services. - EU , 2007.
28. Framework, IDA working document European Interoperability Framework for PAN-European eGovernment Services Version 4.2. - , 2004.
29. Gasevic D., Djuric D., Devedzic V. Model Driven Architecture and Ontology Development. - Berlin: Springer-Verlag , 2006.
30. Grefen P., Ludwig H., Angelov S. A Three-Level Framework for Process and Data Management of Complex E-Services// International journal of cooperative information systems. - 2003. - Nr. 12(2003), No. 3 - 487.-531. lpp.
31. Gross J., Yellen J. Hanbook of Graph Theory. - New York: CRC Press , 2004.
32. Hacid M., Leger A., Rey C., Toumani F. Dynamic discovery of e-services// Proceedings of the 18th French conference on advanced databases. - Paris: 2002.
33. Hausmann J., Heckel R., Lohmann M. Model-based development of Web service descriptions enabling a precise matching concept// International Journal of Web Services Research. - 2005. - Nr. Vol.2, No.2
34. Hellman M. Fuzzy Logic Introduction. - Whitepaper , 2001.
35. Hofmeister C., Nord R., Dilip S. Applied Software Architecture. - Massachusetts: Addison-Wesley , 2000.
36. Institute for Security and Open Methodologies Open-Source Security Testing Methodology Manual 2.1.1. - ISECOM , 2005.
37. Integrēta valsts informācijas sistēma: E-pakalpojumu standarts / Internets. - <https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry41.aspx>
38. Integrēta valsts informācijas sistēma: IS servisu izstrādes standarts / Internets. - <https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry10.aspx>
39. Integrēta valsts informācijas sistēma: Metadatu un e-pakalpojumu identifikācijas standarts / Internets. - <https://ivis.eps.gov.lv/IVISPortal/files/folders/standarti/entry38.aspx>
40. International Organization for Standardization: Software engineering - Product quality / Internets. - [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=22749](http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749)
41. Īrijas PSB / Internets. - <http://www.citizensinformation.ie>

42. Ivanovs M., Ņikiforova O. Elektroniskās komercijas sistēmu objektorientēta projektēšana// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. Lietišķās datorsistēmas 4. izdevums. - Rīga: 2003.
43. Jaeger M., Rojec-Goldmann G. SENECA - Simulation of Algorithms for the Selection of Web Services for Compositions// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 84.-97. lpp.
44. Jantzen J. Tutorial On Fuzzy Logic. - Technical University of Denmark: Dept. of Automation , 1998.
45. Kafeza E, Chiu D., Kafeza I. View-Based Contracts in an E-Service Cross-Organizational Workflow Environment// Tehnologies for E-Services Second International Workshop, TES 2001 Rome, Italy, September 2001 Proceedings. - Berlin: Springer-Verlag, 2001. - 74.-88. lpp.
46. Kakakota R, Robinson M. e-Business 2.0 Roadmap for Success. - Boston: Addison-Wesley , 2001.
47. Kan S. Metrics and Models in Software Quality Engineering 2nd ed. - Boston: Addison-Wesley , 2003.
48. Kazman R., Bass L., Webb M., Abowd G. SAAM: A Method for Analyzing the Properties of Software Architectures// International Conference on Software Engineering. Proceedings of the 16th international conference on Software engineering. - Los Alamitos: IEEE Computer Society Press, 1994. - 81.-90. lpp.
49. Kazman R., Klein M., Clements P. ATAM: Method for Architecture Evaluation. - Pittsburgh: Carnegie Mellon Software Engineering Institute CMU/SEI-2000-TR-004 ESC-TR-2000-004 , 2000.
50. Khalaf R., Leymann F. On Web Services Aggregation// Tehnologies for E-Services 4th International Workshop, TES 2003 Berlin, Germany, September 2003 Proceedings. - Berlin: Springer-Verlag, 2003. - 1.-13. lpp.
51. Kokash N. Web Service Discovery with Implicit QoS Filtering// Proceedings of the IBM PhD Student Symposium, in conjunction with the ICSOC. - Amsterdam: 2005. - 61.-66. lpp.
52. Kriķis D., Zariņš P., Ziobrovskis V. Diferencēti uzdevumi matemātikā. - Rīga: Zvaigzne ABC , 1994.
53. Latvijas Republikas Īpašu uzdevumu ministra elektroniskās pārvaldes lietās sekretariāts Integrētās valsts informācijas sistēmas (e-pakalpojumu infrastruktūras) koncepcija. - Rīga: , 2005.
54. Latvijas Republikas Tieslietu ministrija Latvijas e – pārvaldes koncepcija. - , 2002.
55. Latvijas valsts portāls: E-pakalpojumi / Internets. - <https://www.latvija.lv/LV/LDV/Default.aspx>
56. Likumprojekts Elektronisko pakalpojumu likums. - IUMEPLS , 2008.
57. Likums Par pašvaldībām. - Latvijas Saeima
58. Losavio F. Quality Models to Design Software Architecture// Journal of Object Technology. - 2002. - Nr. Vol. 1, No. 4
59. Lundberg L., Bosch J., Haggander D., Bengtsson P. Quality Attributes in Software Architecture Design// Proceedings of the IASTED 3rd International Conference on Software Engineering and Applications. - 1999. - 353.-362. lpp.
60. Martin-Diaz O., Ruiz-Cortes A., Benavides D., Duran A., Toro M. A Quality-Aware Approach to Web Services Procurement// Tehnologies for E-Services 4th International Workshop, TES 2003 Berlin, Germany, September 2003 Proceedings. - Berlin: Springer-Verlag, 2003. - 42.-53. lpp.
61. Mattern T., Woods D. Enterprise SOA: Designing IT for Business Innovation. - Beijing: O'Reilly , 2006.

62. Mecella M., Presicce F., Pernici B. Modeling E-service Orchestration through Petri Nets// Tehnologies for E-Services Third International Workshop, TES 2002 Hong Kong, China, August 2002 Proceedings. - Berlin: Springer-Verlag, 2002. - 38.-47. lpp.
63. Mell P., Scarfone K., Romanosky S. Common Vulnerability Scoring System (v2). - CVSS , 2007.
64. Michael C., Rojec-Goldmann J, Rojec-Goldmann G. SENECA - Simulation of Algorithms for the Selection od Web Services for Compositions// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 84.-97. lpp.
65. Microsoft Corporation Connected Government Framework. Architecture and Design Blueprint. - , 2005.
66. Microsoft Corporation Research Directions on Microsoft. Biztalk Server 2004 Drives Microsoft integration Strategy. - Microsoft Corporation , 2004.
67. Miettinen K. Nonlinear Multiobjective Optimization. - Boston: Kluwer Academic Publishers , 1998.
68. Mohan C. Dynamic E-business: Trends in Web Services// Tehnologies for E-Services Third International Workshop, TES 2002 Hong Kong, China, August 2002 Proceedings. - Berlin: Springer-Verlag, 2002. - 1.-5. lpp.
69. Morison J. e-Government: a new architecture of government and a new challenge for learning and teaching public law. - Whitepaper , 2003.
70. National Institute of Standards and Technology Technology Administration U.S. Department of Commerce An Introduction to Computer Security: The NIST Handbook Special Publication 800-12. - National Institute of Standards and Technology
71. National Institute of Standards and Technology Technology Administration U.S. Department of Commerce Security Metrics Guide for Information Technology Systems. - Gaithersburg: National Institute of Standards and Technology , 2003.
72. Oberleitner J, Rosenberg F., Dustdar S. A Lightweight Model-Driven Orchestration Engine for e-Services// Tehnologies for E-Services 6th International Workshop, TES 2005 Trondheim, Norway, September 2005 Revised Selected Papers. - Berlin: Springer-Verlag, 2006. - 48.-57. lpp.
73. Object Management Group Unified Modeling Language Specification Version 1.5. - OMG , 2003.
74. Orriens B., Yang J., Papazoglou M. A Framework for Business Rule Driven Service Composition// Tehnologies for E-Services 4th International Workshop, TES 2003 Berlin, Germany, September 2003 Proceedings. - Berlin: Springer-Verlag, 2003. - 14.-27. lpp.
75. Papazoglou M., Yang J. Design Methodology for Web Services and Business Processes// Tehnologies for E-Services Third International Workshop, TES 2002 Hong Kong, China, August 2002 Proceedings. - Berlin: Springer-Verlag, 2002. - 54.-64. lpp.
76. Papazoglou M. Agent – Oriented Technology in Support of E – Business// Communications of the ATM. - New York: ACM, 2001. - Nr. Volume 44, Issue 4 - 71.-77. lpp.
77. Papazoglou M. Service – Oriented Computing: Concepts, Characteristics and Directions// Keynote for the 4th International Conference on Web Information Systems Engineering, December 10-12. - 2003. - 3.-12. lpp.
78. Piccinelli G., Emmerich W., Williams S., Stearns M. A Model-Driven Architecture for Electronic Service Management Systems// Service-Oriented Computing - ICSOC 2003. - Berlin: Springer-Verlag, 2003. - 241.-255. lpp.
79. Pilioura T., Tsalgatidou A. E-Services: Current Technology and Open Issues// Tehnologies for E-Services Second International Workshop, TES 2001 Rome, Italy, September 2001 Proceedings. - Berlin: Springer-Verlag, 2001. - 1.-15. lpp.

80. Pirjanian P., Mataric M. Multiple Objective vs. Fuzzy Behavior Coordination// Lecture Notes in Computer Science on Fuzzy Logic Techniques for Autonomous Vehicle Navigation. - 1999.
81. Puschmann T., Alt R. Enterprise Application Integration Systems and Architecture – the Case of the Robert Bosch Group// Journal of Enterprise Information Management. - 2004. - Nr. 2004.02.17 - 105.-116. lpp.
82. Qian K., Liu J., Tsui F. Decoupling Metrics for Services Composition// Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR 2006. 5th IEEE/ACIS International Conference on Component-Based Software Engineering. - Honolulu: 2006. - 44.-47. lpp.
83. Ramboll Management Top of the Web. User Satisfaction and Usage Survey of eGovernment services. - Copenhagen: European Commission , 2004.
84. Rational Software Rational Unified Process Best Practices for Software Development Teams. - Lexington: Rational Software , 1998.
85. Reģionālās attīstības un pašvaldību lietu ministrija: E-pārvalde / Internets. - <http://www.rapl.gov.lv/pub/index.php?id=1804>
86. Rīgas dome: Rīgas pašvaldības e-pakalpojumi / Internets. - [http://www.riga.lv/LV/Channels/Municipal\\_services/Advanced\\_Services/default.htm](http://www.riga.lv/LV/Channels/Municipal_services/Advanced_Services/default.htm)
87. Rīgas domes Attīstības departaments: Izkārtņu un reklāmas izvietošana / Internets. - <http://www.rdpad.lv/services/advertisement/>
88. Rīgas domes Informācijas tehnoloģiju centrs E-pilsētas informācijas sistēmu arhitektūra. - Rīga: , 2003.
89. Seltzsam S., Borzsonyi S., Kemper A. Security for Distributed E-Service Composition// Tehnologies for E-Services Second International Workshop, TES 2001 Rome, Italy, September 2001 Proceedings. - Berlin: Springer-Verlag, 2001. - 147.-162. lpp.
90. Sion R., Tatemura J. Dynamic Stochastic Models for Workflow Response Optimization// IEEE International Conference on Web Services (Industry Track) IEEE ICWS. - 2005. - 664.-672. lpp.
91. The Open Web Application Security Project A Guide to Building Secure Web Applications and Web Services 2nd ed. - OWASP , 2005.
92. Triantaphyllou E. Multi-Criteria Decision Making Methods: A Comparative Study. - Dordrecht: Kluwer academic publishers , 2000.
93. Valsts informācijas sistēmu savietotājs / Internets. - <https://ivis.eps.gov.lv/ivisportal/>
94. Weijters A., Aalst W. Rediscovering Workflow Models from Event-Based Data// Integrated Computer-Aided Engineering. - Amsterdam: IOS Press , 2003. - Nr. Volume 10 , Issue 2 - 151.-162. lpp.
95. Weske M. Business Process Management. Concepts, Languages, Architectures. - Berlin: Springer-Verlag , 2007.
96. Wiehler G. Web Services and Service Oriented Architectures. - Siemens , 2004.
97. Yang S., Lam H., Su S. Trust-Based Security Model and Enforcement Mechanism for Web Service Technology// Tehnologies for E-Services Third International Workshop, TES 2002 Hong Kong, China, August 2002 Proceedings. - Berlin: Springer-Verlag, 2002. - 151.-160. lpp.
98. Yu Tao., Lin Kwei-Jay Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints// International Conference on Service-Oriented Computing. - Berlin: Springer-Verlag, 2005. - 130.-143. lpp.
99. Yunyao Li, Jagadish H. Compatibility Determination in Web Services // ICEC eGovernment Services Workshop. - 2003.

100. Zeiris E., Zieme M. E-Service Architecture Selection Based on Multi-criteria Optimization// Product-Focused Software Process Improvement 8th International Conference, PROFES 2007, Riga, Latvia, July 2007 Proceedings. - Berlin Heidelberg: Springer-Verlag, 2007. - 345.-357. lpp.
101. Zhang Q., Wu J., Zhu H. Tool Support to Model-based Quality Analysis of Software Architecture// Computer Software and Applications Conference, 2006. COMPSAC 06. 30th Annual International. - Chicago: 2006. - Nr. 1 - 121.-128. lpp.
102. Zhu H. Software Design Methodology From Principles to Architectural Styles. - Amsterdam: Elsevier Butterworth Heinemann , 2005.
103. Zieme M., Rudmanis R. E-pakalpojumi un sistēmu integrācija// E-Pasaule. - Rīga: Data Media Group, 2005. - Nr. 05.2005 - 28.-31. lpp.
104. Zieme M., Žeiris E. Pašvaldību e - pakalpojumi// Latvijas i – sabiedrības tehnoloģiju ekspozīcija LatSTE. - Rīga: LU, 2004. - 157.-162. lpp.
105. Zimmermann H. Fuzzy Sets, Decision Making, and Expert Systems. - Boston Dordrecht Lancaster: Kluwer Academic Publishers , 1993.
106. Zimmermann O., Tomlinson M., Peuser S. Perspectives on Web Services. Applying SOAP, WSDL and UDDI to Real-World Projects 2nd corrected printing. - Berlin: Springer-Verlag , 2005.
107. Žeiris E, Zieme M., Amanis I. Tīkla servisu drošības sistēma / Patents Nr. LV 13720 - Latvija, publicēts 20.08.2008.
108. Žeiris E., Zieme M. Drošības risinājumi e-pakalpojumu sistēmās// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 24. sējums. - Rīga: RTU, 2005. - 84.-91. lpp.
109. Žeiris E., Zieme M. Elektronisko pakalpojumu sistēmu arhitektūras izvēle// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 32. sējums. - Rīga: RTU, 2007. - 99.-107. lpp.
110. Žeiris E., Zieme M. Elektronisko pakalpojumu sistēmu arhitektūras novērtēšanas kritēriji un arhitektūras izvēle// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 27. - Rīga: RTU, 2006. - 91.-98. lpp.
111. Žeiris E., Zieme M. E-pakalpojumu izveides problēmas// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 19. sējums. - Rīga: RTU, 2004. - 48.-53. lpp.
112. Žeiris E., Zieme M. Grafu transformāciju un optimizācijas izmantošana WEB servisu projektēšanā// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 35. sējums. - Rīga: RTU, 2008. - 110.-117. lpp.
113. Žeiris E., Zieme M. ZZ Dats elektronisko pakalpojumu laboratorijā - viss notiek!!! Sakaru pasaule. - Rīga: Sakaru pasaule, 2007. - Nr. 2(46) - 70.-71. lpp.
114. Зегжда Д., Ивашко А. Основы безопасности информационных систем. - Москва: Горячая линия-Телеком , 2000.
115. Зieme M., Растрингин Л. Парето-зависимые множества при решении многокритериальных задач на графе и их применение для синтеза специализированных вычислительных систем// Архитектура и проектирование вычислительных систем. Проблемы адаптации и моделирования. - Рига: 1986. - 128.-137. lpp.
116. Романец Ю., Тимофеев П., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. - Москва: Радио и связь , 2001.