SELECTION OF E-SERVICE SYSTEMS ARCHITECTURE

ВЫБОР АРХИТЕКТУРЫ СИСТЕМЫ ЭЛЕКТРОННЫХ УСЛУГ.

ELEKTRONISKO PAKALPOJUMU SISTĒMU ARHITEKTŪRAS IZVĒLE

E. Žeiris, M. Ziema

Atslēgas vārdi: e-pakalpojums, e-pakalpojumu sistēmu arhitektūra, SOA, arhitektūras izvēle, e-pakalpojumu piemēri, arhitektūras izvēles metožu salīdzinājumi.

1. E-pakalpojumu sistēmas arhitektūras izvēles problēma

E-pakalpojumi tradicionāli tiek veidoti, elektronizējot jau esošus pakalpojumus, kas pastāv jebkurā klientu apkalpošanas procesā. Pakalpojuma elektronizācija ir pietiekami sarežģīts process. Veicot pakalpojumu elektronizāciju, ir jāņem vērā vairāki būtiski faktori, kuri galvenokārt ietekmē sniegtā pakalpojuma kvalitāti un kvalitātes rādītājus tādus kā ātrdarbība, uzturamība, integritāte, atkārtota izmantojamība utt. (QoS – Quality of Service[1]) un līdz ar to arī klientu apmierinātību un elektronizācijas lietderību pakalpojuma sniedzējam un saņēmējam.

Elektronizācijas process sastāv no vairākiem soļiem, starp kuriem noteikti ir sekojoši:

- Elektronizācijas plāna izstrāde;
- E-pakalpojuma prasību specifikācijas izstrāde;
- E-pakalpojuma projektējuma izstrāde;
- E-pakalpojuma izstrāde.

Šajā rakstā galveno uzmanību pievērsīsim e-pakalpojuma projektējuma vienai no galvenajām sastāvdaļām – e-pakalpojuma sistēmas arhitektūras izvēlei. Piemērotas e-pakalpojuma sistēmas arhitektūras izvēle ir svarīga, jo tā tiešā veidā ietekmē sniegtā pakalpojuma kvalitāti (QoS). Vienu un to pašu pakalpojumu ir iespējams veidot, izmantojot dažādas arhitektūras.

Piemēram, iepērkoties internetā, no sagaidāmā rezultāta viedokļa nav svarīgi vai izvēlēto preču ielikšana pirkumu grozā notiek vienā vai vairākos loģiskos soļos (autorizācijas pārbaude, preču skaits noliktavā, groza papildināšana, rēķina papildināšana, utt.). Toties, skatoties no sniegtās pakalpojuma kvalitātes viedokļa, klientam ir svarīgi vairāki kvalitātes faktori tādi kā ātrdarbība, precizitāte, utt., tas nozīmē, ka ir jānodrošina precīza pakalpojuma soļu savstarpējā sadarbība, kas nodrošina integritāti un, protams, arī pietiekami mazs algoritma izpildes laiks. Savukārt skatoties no pakalpojuma sniedzēja viedokļa, svarīgi ir tādi pakalpojuma rādītāji, kā izstrādes izmaksas, uzturēšanas izmaksas, utt. Veidojot e-pakalpojuma sistēmas arhitektūru, ir jāņem vērā šie augstākminētie e-pakalpojuma novērtējuma faktori.

2. Arhitektūra un tās izvēle

Projektējot pakalpojumu, vispirms ir nepieciešams identificēt visus pakalpojuma izpildes soļus, kas tiek noteikti pakalpojuma elektronizācijas un prasību specifikācijas fāzē. Nākamais uzdevums ir noteikt, kā izveidot Web servisus no identificētajiem e-pakalpojuma izpildes

soļiem. Web servisu izveides izvēle ir būtiska, jo tā tiešā veidā ietekmē pakalpojuma sistēmas arhitektūru.

E-pakalpojumi izpildās e-pakalpojumu vidē, kuras arhitektūra ir pietiekami sarežģīta. Neskatoties uz konkrētu e-pakalpojuma izpildes vidi un tās arhitektūru, katra konkrēta e-pakalpojuma būtisks rādītājs ir e-pakalpojumā iesaistītie web servisi un to savstarpējā orķestrācija, kuru nodrošina e-pakalpojuma izpildes vide.

Turpmāk šajā rakstā ar e-pakalpojuma sistēmas arhitektūru sapratīsim e-pakalpojumā iesaistītos web servisus un to savstarpējo sadarbību, kas pēc savas būtības ir web servisu orķestrācija e-pakalpojumu izpildes vidē[2].

E-pakalpojuma arhitektūras izvēlei pielietosim sekojošu metodiku. Vispirms definēsim orientētu e-pakalpojuma algoritma grafu $G = (S, L)$, kur $S = \{s_1, s_2, ...s_n\}$ ir galīga kopa - grafa virsotnes, kas pēc savas būtības ir e-pakalpojuma algoritma izpildāmas darbības, un $L \subset S \times S$. Loks $l_i = (s_j, s_k)$ grafā nozīmē to, ka e-pakalpojuma algoritma izpildē pēc darbības s_j izpildes seko darbības s_k izpilde. Savukārt Web servisu grafs, kas apraksta e-pakalpojuma arhitektūru, tiek iegūts, segmentējot algoritmu grafu. Tas nozīmē, ka vienā Web servisā tiek iekļautas vairākas darbības. Par cik ir iespējamas vairākas segmentācijas, definēsim kopu $X = \{G'\}$, kas satur visus iespējamos grafus, kas ir rekursīvi atvasināti no algoritma grafa G, ko pieņem par sākotnējo Web servisu grafu[2][3]. Būtiski ir izvēlēties konkrētu risinājumu no vairākiem iespējamajiem kopā *X*, jo izvēlētā pakalpojuma arhitektūra ietekmē vairākus e-pakalpojuma raksturlielumus, kas savukārt atsaucas uz sniegtā pakalpojuma kvalitāti, kā arī pakalpojuma lietotāju apmierinātību. Izvēloties kādu no iespējamajiem risinājumiem, būtiski ir rast kompromisu, jo e-pakalpojuma raksturlielumi bieži vien ir pretrunīgi.

Aplūkosim e-pakalpojuma arhitektūras izvēli, kas balstīta uz sekojošiem novērtēšanas raksturlielumiem:

- Izstrādes izmaksas;
- Ātrdarbība;
- Uzturēšanas izmaksas;
- Atkārtota izmantojamība;
- Integritāte.

Jāpiebilst, ka šie arhitektūras novērtēšanas kritēriji nav vienīgie iespējamie. To skaits un veids var būt atkarīgs gan no e-pakalpojuma, gan pasūtītāja, gan izstrādātāja, gan citām ārējām prasībām. Lai varētu veikt e-pakalpojuma arhitektūras novērtēšanu, ir nepieciešams skaitliski novērtēt katru no izvirzītajiem arhitektūras novērtēšanas kritērijiem. Ir vairākas metodes, kā katram no šiem kritērijiem piekārtot skaitliskas vērtības. No formulu precizitātes, protams, ir atkarīga iegūtā rezultāta precizitāte un ticamība[3][4].

Šajā rakstā izmantosim sekojošas e-pakalpojuma arhitektūras nevērtēšanas raksturlielumu aprēķina formulas:

Izstrādes izmaksas

Web servisu grafa virsotnes izstrādes izmaksas $$C_{s_i} = \left(\sum_{m_{s_i} \in M_{s_i}} \mathrm{O}(m_{s_i}) + W_C(s_i) \right)^p, \quad (1)$$

kur $W_C(s_i)$ ir Web servisu grafa virsotnes s_i Web servisa implementācijas koda rindiņu skaits un $O(m_s)$ virsotnes s_i implementējošās metodes rindiņu skaits. Pakāpe *p* raksturo attiecīgas realizācijas programmēšanas valodas implementācijas sarežģītību (*p* pēc savas būtības ir daļskaitlis).

E-pakalpojuma izstrādes izmaksas $$C = \sum_{s_i \in S} C_{s_i} \quad (2)$$

Ātrdarbība

E-pakalpojuma ātrdarbība $T = \sum_{s_i \in S} (t_{s_i} + t_{ws}) + t_{epak} + t_{data}$, (3)

kur t_{s_i} virsotnes vidējais izpildes laiks, $t_{ws} = \log(t_{s_i})$ papildus laiks, kas nepieciešams, lai algoritmu realizētu kā Web servisu, $t_{epak} = \log(n)$ laiks, kas nepieciešams, lai darbinātu e-pakalpojumu kādā no e-pakalpojumu izpildes vidēm, kur $n = |S|$ (Web servisu skaits grafā), un $t_{data} = \sum_{l_i \in L} t_{l_i}$ laiks, kas nepieciešams, lai pārsūtītu datus no viena e-pakalpojuma stāvokļa uz nākamo.

Uzturēšanas izmaksas

E-pakalpojuma uzturēšanas izmaksas $E = k + \frac{\sum_{s_i \in S} \sum_{m_{s_i} \in M_{s_i}} O(m_{s_i})}{n}$, (4)

kur $k = |L|$ loku skaits grafā.

Atkārtota izmantojamība

Atkārtota izmantojamība $R = \frac{1}{n} * \sum_{i=1}^{n} R_i$, (5)

kur $n = |S|$ kopējais virsotņu skaits web servisu grafā un R_i virsotņu skaits, kas saistīts virsotni *i*.

Integritāte

Integritāte $I = k$, (6)

kur *k* ir loku skaits web servisu grafā.

Arhitektūras izvēle var tikt balstīta uz vairākām pieejām:

- Daudzkriteriāla optimizācija;
- Svaru pielietošanas metode;
- Cits.

Šajā rakstā aplūkosim un salīdzināsim pirmās divas pieejas - daudzkriteriāla optimizācija un svaru pielietošanas metodi[2][3][5][6][7] .

3. Arhitektūras izvēles metodes

Viena no uzdevuma risināšanas iespējām ir daudzkriteriāla optimizācija, kas reducējas uz pareto kompromisu kopas P atrašanu[8]:

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

kur $Q(X) = \{q_1(X), q_2(X) ... q_N(X)\}$ - minimizējamie kritēriji; Ω - X definīcijas apgabals. Risinājumu $X_{i^*} \in \Omega$ sauksim par pareto optimālu $X_{i^*} \in P$ tad un tikai tad, ja neeksistē $X_j \in \Omega$ tāds, ka

$$q_i(X_j) \le q_i(X_{i^*}) \quad (8)$$

visiem $i = \{1,2,...N\}$, kur vismaz viena ir stingra nevienādība. Citiem vārdiem sakot, jebkurai vērtībai $X_{i^*} \in P$ kopā Ω nav atrodama labāka vērtība par izvēlēto.

Šādā gadījumā optimāla e-pakalpojuma arhitektūra ir kāds no grafiem, kas atrodas kopā $P = \{G'^*\}$. Kompromisu kopa P vairumā gadījumu satur vairāk kā vienu risinājumu, tāpēc šādas pieejas galvenais trūkums ir tāds, ka ir nepieciešams pielietot papildus līdzekļus, lai varētu izvēlēties konkrētu arhitektūras risinājumu (var tikt pielietota kritēriju samazināšana, vai apvienošana ar kādu citu metodi). Savukārt šīs pieejas galvenie ieguvumi ir tādi, ka optimizējot netiek zaudēts neviens no risinājumiem.

Kā otru no arhitektūras izvēles risināšanas pieejām var minēt svaru pielietošanas metodi. Šis ir veids kā vairākas optimizējamās mērķa funkcijas tiek reducētas uz vienu optimizējamo mērķa funkciju. Metode tiek balstīta uz svaru w_i pielietošanu, kas ir reāli skaitļi tādi, ka $w_i \geq 0$ visiem $i = \{1,2,...N\}$. Pielietotajiem svariem ir jābūt normalizētiem , kas nozīmē $\sum_{i=1}^{N} w_i = 1$. Šādā gadījumā daudzkriteriāla optimizācijas metode tiek novesta uz sekojošu svaru izvēles problēmu[5]:

$$\min(\sum_{i=1}^{N} w_i q_i(X)) \tag{9}$$

Piedāvātās metodes iespējams apvienot, kā arī pielietot citas metodes, piemēram, ekspertu novērtējumu vai Fuzzy loģiku, kuras arī savstarpēji var kombinēt, lai atrastu optimālu arhitektūru pēc vairākām metodēm, tādā veidā samazinot iespējamās kļūdas arhitektūras izvēlē[7].

4. Piemēri

E-pakalpojuma arhitektūras grafa definēšanai pakalpojuma projektēšanas fāzē var tikt izmantota modelēšanas valodas UML (Unified Modeling Language) stāvokļu pārejas diagramma. UML izmantošana projektēšanas fāzē ir plaši pielietota prakse. Projektēšanā parasti tiek izmantotas dažādas (aktivitāšu, klašu, komponenšu, lietojumu situāciju un citas) UML pieejamās diagrammas un apraksti, tāpēc stāvokļu pārejas diagrammas pielietojums kā e-pakalpojuma web servisu un to savstarpējo saišu apraksts ir praktiski pielietojams[9].

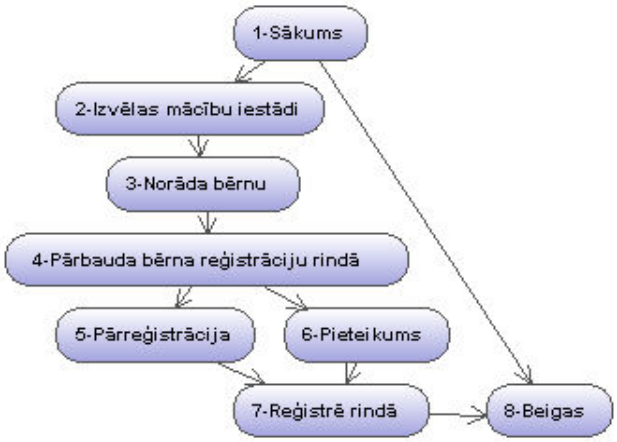
Lai varētu veikt automatizētu e-pakalpojuma arhitektūras izvēli, sākotnēji ir jāapraksta e-pakalpojuma algoritma grafs UML stāvokļu pārejas diagrammā un to jāeksportē uz metadatu apmaiņas XMI (XML Metadata Interchange) formātu. Šāda pieeja nodrošina to, ka sākotnējo algoritma grafu var apstrādāt automatizētās sistēmās, kas aprēķina iespējamos web servisu grafus un izvēlas iespējamos risinājumus, balstoties uz novērtēšanas kritērijiem.

E-pakalpojumu arhitektūras novērtēšanas metodes un to salīdzinājumu parādīsim uz sekojošu divu praktisku piemēru pamata „Pieteikšanās pašvaldības bērnudārzu rindā" un „Reklāmas/izkārtnes saskaņošana pašvaldībā". Šie piemēri tiek balstīti uz reālu e-pakalpojumu izstrādi, ko finansē Rīgas pilsētas pašvaldība.

Pieteikšanās pašvaldības bērnudārzu rindā

Sakarā ar aizvien pieaugošajām bērnudārzu rindām un rindu uzskaites nepieciešamību, rodas arī nepieciešamība pēc šāda e-pakalpojuma. Pakalpojums pieejams autorizētiem lietotājiem, tas nozīmē, ka pakalpojuma izpildes pirmajā solī, pakalpojuma lietotājs tiek autorizēts. Lietotājam ir iespējams norādīt mācību iestādi un bērnu. Bērna norāde ir saistīta ar datu atlasi Pilsonības un migrācijas lietu pārvaldes Iedzīvotāju reģistra datu bāzi, tāpēc šim piemēram un arhitektūras izvēlei tiek uzlikts ierobežojums, ka nav iespējams izvēlēties tādu arhitektūru kur pilnīgi visu funkcionalitāti nodrošina viens Web serviss. Kad lietotājs norāda bērnu, tad tiek pārbaudīti dati bērnudārzu rindu reģistrā, lai noskaidrotu vai šī ir bērna pārreģistrācija vai arī pirmreizējā reģistrācija bērnudārzā. Atkarībā no šī tiek piedāvāta vai nu pieteikuma forma vai arī pārreģistrācijas forma. Pēc pieteikuma formas aizpildīšanas, izvēlētais bērns tiek reģistrēts rindā. Beigās lietotājam tiek attēlots reģistrācijas rezultāts[10].

Pakalpojuma izpildes algoritma grafs, kas pierakstīts UML formāta stāvokļu pārejas diagrammā parādīts 1. attēlā. Ērtības labad, šo piemēru turpmāk tekstā sauksim kā G_1.



1. attēls. Pieteikšanās pašvaldības bērnudārzu rindā.

Reklāmas/izkārtnes saskaņošana pašvaldībā

Pašvaldību teritorijā izvietoto reklāmu/izkārtni nepieciešams saskaņot ar pašvaldību. To nosaka vairāki normatīvie dokumenti. Arī šis pakalpojums tāpat kā iepriekš aprakstītais ir pieejams tikai autorizētiem lietotājiem. Vispirms pakalpojuma pieprasījums tiek reģistrēts klientu apkalpošanas sistēmā, kas nodrošina papildus funkcionalitāti, pakalpojuma sniegšanas kvalitātes uzlabošanas jomā. Pēc reģistrācijas klientu apkalpošanas sistēmā pakalpojuma pieprasījums(reklāmas/izkārtnes skice un to aprakstošie dati) tiek nodoti izvērtēšanai valsts valodas inspektoram, kas sniedz savu rezolūciju, pēc tam var tikt noteikti citi saskaņojumi, var nodot izvērtēšanai māksliniekam vai arī var tikt izbeigts pakalpojums. Pēc citiem saskaņojumiem, tostarp arī kultūras pieminekļu saskaņojumu saņemšanas dati tiek nodoti māksliniekam uz saskaņošanu, kas var sniegt uzreiz savu rezolūciju vai arī palūgt vēl nodaļas vadītāja rezolūciju. Pakalpojuma rezultāts ir saskaņota (noraidīta) reklāmas/izkārtnes skice. Pakalpojuma izpildes algoritma grafs, kas pierakstīts UML formāta stāvokļu pārejas diagrammā parādīts 2. attēlā. Šo piemēru turpmāk tekstā sauksim kā G_2.

2. attēls. Reklāmas/izkārtnes saskaņošana pašvaldībā.

Arhitektūras izvēle

Izrēķināsim visas iespējamās kombinācijas, izmantojot rekursīvu algoritmu. Šādā gadījumā kopējais kopas $X_1 = \{G_1'\}$ apjoms ir 877 atvasinātie grafi, kas iegūti segmentējot grafu G_1 visās iespējamajās kombinācijās un kopas $X_2 = \{G_2'\}$ apjoms ir 4121 atvasinātie grafi. Visus web servisu grafus novērtēsim, izmantojot formulas (2), (3), (4), (5) un (6).

Lai uzskatamāk varētu parādīt rezultātus, visi mērķa funkciju rezultāti ir nonormalizēti, dalot rezultātu ar maksimālo iespējamo vērtību piemēra robežas[5]. Šādā gadījumā optimizējamo funkciju vērtības tiek iegūtas robežās [0,1].

Veiksim daudzkriteriālu optimizāciju datu kopām $X_1 = \{G_1'\}$ un $X_2 = \{G_2'\}$, izmantojot formulas (7) un (9). Daudzkriteriālās optimizācijas rezultāti ir attēloti tabulās 1, 2, 3 un 4. Tabulas kolonnā *Grafs* ir grafu raksturojošs nosaukums, kur iekavās ir norādīts grafa segments, kas apvienojams vienā web servisā. Pārējās kolonnās ir attiecīgi rezultāti, kas iegūti, izmantojot formulas (2), (3), (4), (5) un (6).

Tabula 1. *Pieteikšanās bērnudārzu rindā* web servisu grafa daudzkriteriāla optimizācija - $P_1 = \{G_1' *\}$ risinājumi:

Nr.	*Grafs*	*C*	*T*	*E*	*R*	*I*
1.	(1,2,4,5,6,7),3,8	0.7845	0.7375	0.67	0.614	0.3333
2.	(1,2,4,5,6,7,8),3	0.7338	0.6955	1	0.6579	0.2222
3.	(1,5,6,7),2,3,4,8	0.8771	0.8162	0.4084	0.5789	0.5556
4.	(1,5,6,7,8),2,3,4	0.8321	0.7775	0.506	0.5921	0.4444
5.	(1,2,5,6,7),3,4,8	0.8321	0.7775	0.506	0.5921	0.4444
6.	(1,2,5,6,7,8),3,4	0.7845	0.7375	0.67	0.614	0.3333
7.	(1,4,5,6,7),2,3,8	0.8321	0.7775	0.506	0.5921	0.4444
8.	(1,4,5,6,7,8),2,3	0.7845	0.7375	0.67	0.614	0.3333
9.	1,2,3,4,(5,6),7,8	0.9608	0.9	0.2985	0.5639	0.7778
10.	1,2,3,4,(5,6,7),8	0.9199	0.8631	0.344	0.5702	0.6667
11.	1,2,3,(4,5,6),7,8	0.9199	0.8631	0.344	0.5702	0.6667
12.	1,2,3,4,5,6,7,8	1	0.9824	0.2669	0.625	1

Tabula 2. *Reklāmas/izkārtnes saskaņošanas* web servisu grafa daudzkriteriāla optimizācija - $P_2 = \{G_2' *\}$ risinājumi:

Nr.	*Grafs*	*C*	*T*	*E*	*R*	*I*
1.	1,2,3,4,5(6,7),8	0.9428	0.8896	0.1879	0.517	0.8182
2.	1,2,3,4,(5,6),7,8	0.9428	0.8896	0.1879	0.517	0.8182
3.	1,2,3,4,(5,7,6),8	0.8819	0.782	0.2017	0.4762	0.6364
4.	1,2,3,(4,6),5,7,8	0.9428	0.8896	0.1879	0.517	0.8182
5.	1,2,3,5,(4,6,7),8	0.8819	0.782	0.2017	0.4762	0.6364
6.	1,2,3,(4,5,6),7,8	0.8819	0.782	0.2017	0.4762	0.6364
7.	1,2,3,(4,5,6,7),8	0.8165	0.6727	0.225	0.419	0.4545
8.	1,2,3,(4,5),6,7,8	0.9428	0.8896	0.1879	0.517	0.8182
9.	1,2,(3,6),4,5,7,8	0.9428	0.8896	0.1879	0.517	0.8182
10.	1,2,(3,6,7),4,5,8	0.8819	0.782	0.2017	0.4762	0.6364
11.	1,2,(3,5,6),4,7,8	0.8819	0.782	0.2017	0.4762	0.6364
12.	1,2,(3,5,6,7),4,8	0.8165	0.6727	0.225	0.419	0.4545
13.	1,2,(3,4,6),5,7,8	0.8819	0.782	0.2017	0.4762	0.6364
14.	1,2,(3,4,6,7),5,8	0.8165	0.6727	0.225	0.419	0.4545
15.	1,2,(3,4,5,6),7,8	0.8165	0.6727	0.225	0.419	0.4545
16.	1,2,(3,4,5,6,7),8	0.7454	0.5623	0.265	0.3333	0.2727
17.	1,2,(3,4,5),6,7,8	0.8819	0.782	0.2017	0.4762	0.6364
18.	1,2 (3,4),5,6,7,8	0.9428	0.8896	0.1879	0.517	0.8182
19.	1,(2,3,4,5,6,7),8	0.6667	0.5088	0.3433	0.3175	0.1818
20.	1,2,(3,4,5,6,7,8)	0.6667	0.5088	0.3433	0.3175	0.1818
21.	(1,2,3,4,5,6,7),8	0.5774	0.4471	0.505	0.2857	0.0909
22.	(1,2,3,4,5,6,7,8)	0.4714	0.3888	1	0.1905	0
23.	1,2,3,4,5,6,7,8	1	0.9938	0.18	0.5476	1

Tabula 3. *Pieteikšanās bērnudārzu rindā* web servisu grafa optimizācija ar svaru metodi, kur $w_i = 0.2$:

Nr.	*Grafs*	*C*	*T*	*E*	*R*	*I*
1.	(1,2,4,5,6,7),3,8	0.7845	0.7375	0.67	0.614	0.3333
2.	(1,2,5,6,7,8),3,4	0.7845	0.7375	0.67	0.614	0.3333
3.	(1,4,5,6,7,8),2,3	0.7845	0.7375	0.67	0.614	0.3333

Tabula 4. *Reklāmas/izkārtnes saskaņošanas* web servisu grafa optimizācija ar svaru metodi, kur $w_i = 0.2$:

Nr.	*Grafs*	*C*	*T*	*E*	*R*	*I*
1.	(1,2,3,4,5,6,7),8	0.5774	0.4471	0.505	0.2857	0.0909

Kā redzam, tad abos šajos piemēros svaru metodes risinājums ir apakškopa Pareto optimumu kopai. Tas nozīmē, ka šajos piemēros minētajos gadījumos, pielietojot svaru metodi, nerodas jauni risinājumi. Dotajos piemēros svaru metodes pielietojums palīdz izvēlēties e-pakalpojumu arhitektūras risinājumu. E-pakalpojuma arhitektūra var tikt izvēlēta kā jebkurš no atrastajiem risinājumiem.

5. Kopsavilkums

Veidojot e-pakalpojumu, ir jārisina veidojamā pakalpojuma kvalitātes problēma (QoS). Izveidot e-pakalpojumu un sasniegt gaidāmo rezultātu ir iespējams ar dažādām e-pakalpojuma sistēmas arhitektūrām. Būtiski ir tas, ka dažādas e-pakalpojumu arhitektūras, sniedz dažādus kvalitātes rādītājus. Izvirzot nosacījumus e-pakalpojumu kvalitātei, ir iespējams veikt e-pakalpojumu arhitektūras optimizāciju, kā daudzkriteriālu optimizāciju. Šajā rakstā ir sniegts ieskats divās daudzkriteriālās optimizācijas pieejās – Pareto optimuma iegūšana un svaru metodes pielietojums. Rezultāti parādīti ar divi praktisku piemēru palīdzību. Rakstā minētajos piemēros pielietotās metodes papildina viena otru. Svaru pielietošanas metode šajos piemēros nesniedz jaunus risinājumus ārpus Pareto optimuma kopas. Svaru pielietošanas metodes trūkums ir subjektīvisms, kas rodas pie svaru izvēles problēmas. Lai pielietotu svaru metodi ir nepieciešams noteikt prioritātes optimizējamajiem kritērijiem un veikt eksperta novērtējumu, lai varētu izvēlēties pielietojamos svarus. Šajā rakstā minētajos piemēros tika pielietoti vienlīdzīgi svari ($w_i = 0.2$), jo netika izvirzītas prioritātes nevienam no kritērijiem. Praksē svaru metodi var pielietot jau izvēlētas Pareto kopas ietvaros, lai netiktu zaudēti risinājumi. Šo metodi var pielietot, lai veiktu izvēli starp risinājumiem kas atrodas Pareto kopā. Pielietojot svaru metodi vienu pašu, ir iespējams zaudēt būtiskus risinājumus. [5][6][7]

6. Literatūra

1. Tao Yu, Kwei-Jay Lin. Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. International Conference on Service-Oriented Computing 2005 – p. 130-143

2. E. Zeiris, M. Ziema. E-Service Architecture Selection Based on Multi-criteria Optimization. 8th International Conference, PROFES 2007, Riga, Latvia, July 2007 Proceedings. Springer-Verlag Berlin Heidelberg 2007. p.345-357

3. E. Žeiris, M. Ziema. Elektronisko pakalpojumu sistēmu arhitektūras novērtēšanas kritēriji un arhitektūrās izvēle. Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 27. sējums. Rīga 2006.g. 91 – 98 lpp

4. 15. Stephen H. Kan. Metrics and Models in Software Quality Engineering. Second Edition. Addison Westley 2003

5. Kaisa M. Miettinen. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers. 1998

6. Martin Hellmann. Fuzzy Logic Introduction. 2001. p.7. http://www.fpk.tu-berlin.de/~anderl/epsilon/fuzzyintro4.pdf

7. János Fülöp. Introduction to Decision Making Methods. http://academic.evergreen.edu/projects/bdei/documents/decisionmakingmethods.pdf

8. М. Г. Зиема, Л. А. Растригин. 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 – с. 128-137.

9. UML http://en.wikipedia.org/wiki/State_diagram

10. E. Žeiris, M. Ziema. ZZ Dats elektronisko pakalpojumu laboratorijā — viss notiek!. Sakaru pasaule. #2(46) 2007.g. 70-71lpp.

Edzus Zeiris, M. Sc. Comp.
Riga Technical University
Faculty of Computer Science and Information Technology,
Institute of Computer Control, Automation and Computer Engineering
Address: Meza 1/3, 3rd floor. LV-1048, Riga, Latvia
E-Mail: edzus@zzdats.lv

Maris Ziema, Dr. Sc. Ing.
Riga Technical University
Faculty of Computer Science and Information Technology,
Institute of Computer Control, Automation and Computer Engineering
Address: Meza 1/3, 3rd floor. LV-1048, Riga, Latvia
E-Mail: maris@zzdats.lv

Žeiris E., Ziema M. Elektronisko pakalpojumu sistēmu arhitektūras izvēle.
E-pakalpojumu sistēmu arhitektūras izvēle ir svarīga problēma. Vienu un to pašu e-pakalpojumu ir iespējams realizēt, izmantojot vairākas alternatīvas sistēmas arhitektūras. Katrai no sistēmām ir dažādi izpildes rādītāji, kas ir svarīgi e-pakalpojumu saņēmējiem un sniedzējiem. Šajā darbā ir piedāvāts salīdzinājums divām arhitektūras izvēles metodēm. E-pakalpojuma arhitektūras izvēle tiek risināta kā daudzkriteriāla optimizācija. Risinājums tiek parādīts, izmantojot divus praktiskus e-pakalpojumu piemērus „Pieteikšanās bērnudārza rindā" un „Reklāmas/izkārtnes saskaņošana". Piemēros arhitektūras optimizācija tiek veikta pēc pieciem kritērijiem (izstrādes izmaksas, ātrdarbība, uzturēšanas izmaksas, atkārtota izmantojamība un integritāte). Piemēros optimizācija notiek ar divām pieejām – Pareto optimumu kopas iegūšana un svaru metodes pielietošana..

Zeiris E., Ziema M. Selection of E-Service Systems Architecture
Selection of e-service systems architecture is important problem. One and the same e-service can be designed by using different alternative systems architectures. Each of these systems has its own execution estimation that is important for service provider and also for service receiver. This article gives comparison for two architecture selection methods. Selection of e-service systems architecture is based on multi-criteria optimization. The solution example is based on two practical e-services examples "To sign up to kindergarten queue" and "Advertisement/Signboard conformation". Architecture optimization in examples uses five criteria (costs of production, time of execution, exploitation costs, reusability and integrity). Optimization in examples is done by two optimization approaches – calculating of Pareto optimum set and using of weighting method.

Жейрис Э., Зиема М. Выбор архитектуры системы электронных услуг.
Выбор архитектуры системы электронных услуг – это важная проблема. Одна и та же электронная услуга может быть реализована при использовании различных системных архитектур. У каждой из этих систем есть собственные показатели выполнения, которые важны как для предоставителя услуги, так и для её пользователя Статья сравнивает два подхода к выбору архитектуры услуг, основываясь на многокритериальной оптимизации. Решение описывается на примере двух реальных электронных услуг – «Запись в очередь в детский сад» и «Согласование о рекламах и вывесках». Оптимизация архитектуры проведена по пяти критериям (стоимость разработки, быстродействие, стоимость поддержки, повторное использование и целосность). В примерах оптимизация использует два подхода - нахождение оптимума множества Парето и использование свёрки критериев.