

TECHNOLOGIES OF COMPUTER
CONTROL

DATORVADĪBAS TEHNOLOĢIJAS

ARCHITECTURE DESIGN OF ELECTRONIC SERVICES

ELEKTRONISKO PAKALPOJUMU SISTĒMU ARHITEKTŪRAS PROJEKTĒŠANA

Edzus Zeiris, Mg. 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

Atslēgas vārdi: Datoru sistēmu arhitektūras projektēšana, e – pakalpojumi, e – pakalpojumu sistēmas arhitektūras projektēšana, arhitektūras izvēle, projektēšanas metožu salīdzinājums

Ievads

Arhitektūras projektēšana ir process, kas klasificē datorsistēmai izvirzītās prasības, strukturē tās un izveido vienotu sistēmas arhitektūru. E – pakalpojumu sistēmas arhitektūra ir e – pakalpojumu sistēmas struktūra vai struktūras, kas sevī ietver sistēmas komponentes, komponentu īpašības un to savstarpējās saites, kas nepieciešamas e – pakalpojuma darbībai. E – pakalpojumu sistēma tiek veidota servisu orientētā arhitektūrā (SOA), kur komponentes tiek veidotas kā web servisi. Klasiski sistēmu arhitektūras projektēšanā tiek pielietotas divas projektēšanas metodes – funkcionāli bāzēta un objektu orientēta[1].

Funkcionāli bāzētā projektēšanas metode tiek balstīta uz pieeju, kad sistēma kā viens veselums tiek dalīta tās galvenajās sastāvdaļās. Projektēšana no augšas uz leju. Funkcionāli bāzēta projektēšana parasti sastāv no četriem izpildes soļiem. Pirmais projektēšanas solis ir saistīts ar projektējamās sistēmas ietvara noteikšanu. Tajā ir jānosaka sistēmas nepieciešamās saskarnes un ārējās atkarības. Kā arī nepieciešamā sistēmas uzvedība, jeb citiem vārdiem sakot ir jāsaprot, kādi ir sistēmas uzdevumi. Otrajā projektēšanas solī ir jānosaka iespējamās sistēmas sastāvdaļas un to prototipi. T.i. galvenās sistēmas abstrakcijas un atkarības starp tām. Trešajā projektēšanas solī sistēma tiek sadalīta tās galvenajās komponentēs un noteiktas komponentu savstarpējās atkarības un ietekmes. Pēdējā, ceturtajā, projektēšanas solī tiek aprakstīti veidojamie sistēmas moduļi, instances un sastāvdaļas, par pamatu ņemot izveidotos prototipus, komponentes un saskarnes starp tām. *Objektu orientēta projektēšanas metode* parasti tiek veidota, balstoties uz mazākajām pieejamajiem sistēmas komponentēm un liekot tās kopā. To var saukt arī par projektēšanu no

lejas uz augšu. Objektu orientēta metode sākas ar vienumu modelēšanu, kas atrodas problēmu apgabalā, tad tie tiek apkopoti objektos, hierarhijās, mantojumos utt. Objektu orientētas projektēšanas rezultātus parasti pieraksta UML formātā. Galvenās objektu orientētas projektēšanas sastāvdaļas ir lietošanas gadījumu diagramma, kas parāda saites lietošanas scenārijos starp sistēmu vai citu semantisku vienumu un to ietekmējošo personālu. Klašu diagramma attēlo sistēmas statisko struktūru. Tajā tiek attēloti sistēmā ietvertie objekti, klases, tipi, to iekšējā struktūra un saites starp tiem. Klašu diagramma neattēlo informāciju, ko objekti satur, kad iestājas kāds notikums. Stāvokļu diagramma apraksta vienumu uzvedību izpildes laikā, kas ir atkarīga no citiem vienumiem, kā atbilde uz notikumiem tajos. Aktivitāšu diagramma ir stāvokļu diagrammas variācija, kas parāda stāvokļus vai saistītās aktivitātes sistēmas izpildes laikā, kā arī pārejas, kas tiek izsauktas vienas vai otras aktivitātes izpildes rezultātā. Sekvenču diagramma attēlo sadarbību starp sistēmā iesaistītajiem vienumiem kā ziņojumus starp tiem noteiktā secībā. Sadarbības diagramma attēlo sadarbību starp vienumu instancēm un to lomām, kā arī nepieciešamās lomu atkarības izpildes vidē. Komponentu diagramma attēlo sistēmā iekļauto komponentu savstarpējās atkarības, kas iekļauj arī komponentu klasifikāciju un artefaktus. Uzstādīšanas diagramma attēlo izpildes laika vienumus, sistēmas komponentes, procesus un objektus, kas izpilda tos.

Izstrādājot e – pakalpojumus, ir ļoti būtiski izveidot e – pakalpojumu sistēmas arhitektūru tādu, lai e – pakalpojums apmierinātu visus tam izvirzītos kvalitātes kritērijus, kas veicinātu plašāku tā izmantojamību. Vēl jo vairāk, 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, kas ir nepieciešams pielietot efektīvas projektēšanas metodes e – pakalpojumu sistēmu izveidei, kas nodrošinātu pieņemamas e – pakalpojumu sistēmas arhitektūras izveidi. Lai varētu ievērot sistēmai izvirzītās kvalitātes prasības, ir nepieciešams projektēt sistēmas arhitektūru atbilstoši tām. Šādos gadījumos vienkārši funkcionāli bāzēta vai objektu orientēta projektēšanas pieeja ir nepietiekama. Ir nepieciešams pielietot metodes, kas risina arhitektūras izveides kvalitātes problēmas.

Aplūkosim dažas no arhitektūras projektēšanas metodēm, kas risina sistēmas arhitektūras kvalitātes jautājumus

Projektēšanas metodes

E – pakalpojumu izstrādē problēmas izraisa biznesa procesu straujas izmaiņas, kas prasa atbilstošo informācijas tehnoloģiju sistēmu izmaiņas, tajā skaitā arī e – pakalpojumu izmaiņas. E – pakalpojumu sistēmu izstrādei šādā dinamiskā vidē ir nepieciešama ātra un kvalitatīva projektēšanas metode, kas ļautu pielāgoties biznesa procesiem un varētu apmierināt visas e – pakalpojumu sistēmai izvirzītās biznesa prasības. 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[2]. Lai varētu ievērot sistēmai izvirzītās kvalitātes prasības, ir nepieciešams izveidot sistēmas arhitektūru, kas apmierina tai izvirzītās prasības. Ir nepieciešams pielietot metodes, kas risina arhitektūras izveides kvalitātes problēmas. 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, matemātiskā modelēšanas, argumentētu novērtēšanu.

Aplūkosim projektēšanas metodes, kuras risina sistēmas arhitektūras kvalitātes jautājumus jau tās projektēšanas fāzē.

Atribūtu virzīta projektēšanas metode ADD[1]. 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) 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[3]. 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[4]. Šo arhitektūras projektēšanas metodi var aplūkot, kā procesu, kur kā ieejas parametri ir sistēmas prasību specifikācija un izejā ir piedāvātā sistēmas arhitektūra. Projektēšanas process sākas ar sistēmas arhitektūras definēšanu, kas tiek balstīta uz sistēmas specifikācijā noteiktajām prasībām, kura 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 atkārto, 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[3]. Ja ir nepieciešams aplūkot vairākus kvalitātes rādītājus vienlaicīgi, tad ir nepieciešams veidot kompromisu starp savstarpēji konfliktējošiem kvalitātes nosacījumiem. Tas ir tādēļ, ka kvalitātes rādītāji ir savstarpēji pretēji saistīti. Ja kvalitātes novērtējums pēc viena no rādītājiem ir maksimāli labs, tad cits tajā pašā laikā ir ļoti 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, tāpēc tie 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[3]. Šī projektēšanas metode ir vairāk balstīta uz drošības risinājumu prasību sasniegšanu. To pielieto sistēmu projektēšanā no dažādām sfērām, 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.

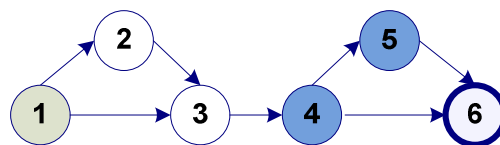
Visas šīs aplūkotās metodes risina jautājumu par sistēmas kvalitāti, bet tām ir viens liels trūkums. Sistēmas arhitekta vai personas, kas veic sistēmas arhitektūras projektēšanu, subjektīvais viedoklis, kā arī iespēja ka tiks palaists garām pats labākais risinājums, vai arī tas vispār netiks aplūkots. 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 sistēmas arhitekts apstrādāt pieņemamā laika periodā[5].

E – pakalpojumu projektēšana

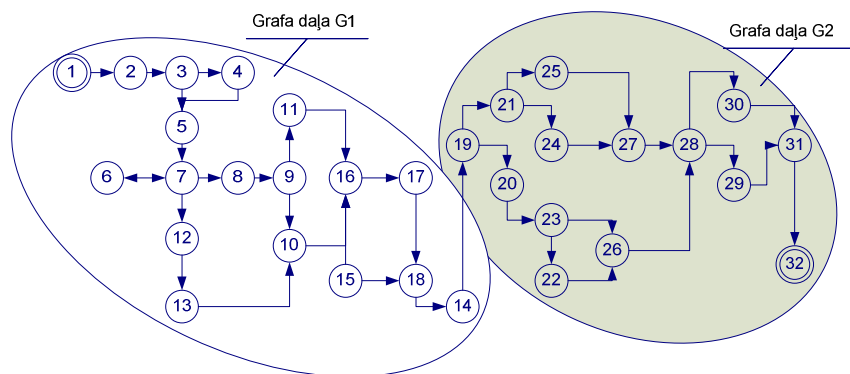
Ņemot vērā iepriekš minēto projektēšanas metožu nepilnības ir izstrādāta jauna projektēšanas metode, kas ir pielietojama e – pakalpojumu sistēmu arhitektūras projektēšanai. E – pakalpojumu sistēmas arhitektūras projektēšanas metode sastāv no sekojošiem soļiem:

1. Sākotnēji ir nepieciešams izveidot e – pakalpojuma algoritma grafu G , kas apraksta projektējamo e – pakalpojumu. E – pakalpojumu algoritma grafs ir 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_k izpilde. Loci e – pakalpojuma algoritma grafā norāda informācijas plūsmu[6].
- 2.



1. att. E – pakalpojuma algoritma grafa piemērs
Fig. 1. Example of e – service algorithm graph

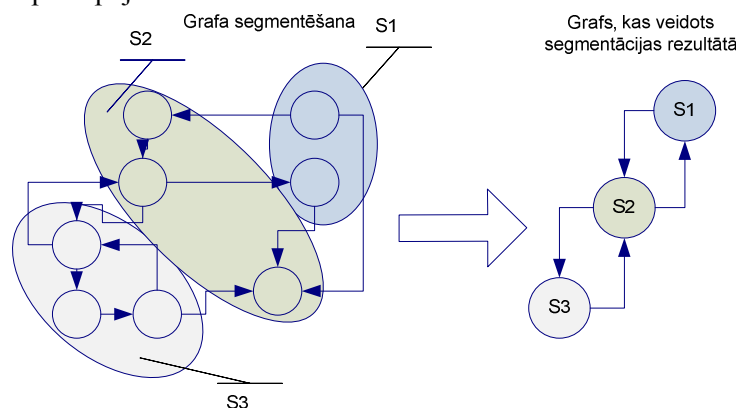
3. E – pakalpojuma algoritma grafā ir nepieciešams noteikt iespējamus ierobežojumus virsotņu savstarpējā segmentēšanā. Ja e – pakalpojuma algoritma grafa virsotņu skaits pārsniedz 20 un nav neviena ierobežojuma, tad ir nepieciešams izdalīt e – pakalpojuma grafā segmentus un katru no segmentiem apstrādāt kā atsevišķu projektēšanas uzdevumu[7];



2. att. Liela e – pakalpojuma algoritma grafa sākotnējie segmenti
Fig. 2. Initial segments of large e – service algorithm graph

4. No e – pakalpojuma algoritma grafa rekursīvas segmentēšanas rezultātā tiek iegūti web servisu grafi $X = \{G'\}$ kas ir par pamatu e – pakalpojumu sistēmas

arhitektūras izveidē. Par sākotnējo web servisu grafu tiek pieņemts e – pakalpojuma algoritma grafs[6,8,9];



3. att. Algoritma grafa segmentēšana
Fig. 3. Segmentation of algorithm graph

5. Iegūtajiem web servisu grafiem tiek aprēķinātas kvalitātes rādītāju skaitliskās vērtības, izmantojot aprēķina formulas [8,9,10];
6. No visiem web servisu grafiem tiek iegūta Pareto kompromisu kopa $P = \{G'^{*}\}$, izmantojot formulu (1). Pareto kompromisu kopās atrašanu formāli var definēt sekojoši:

$$Q(X) \rightarrow \min_{X \in \Omega} \rightarrow P \quad (1)$$

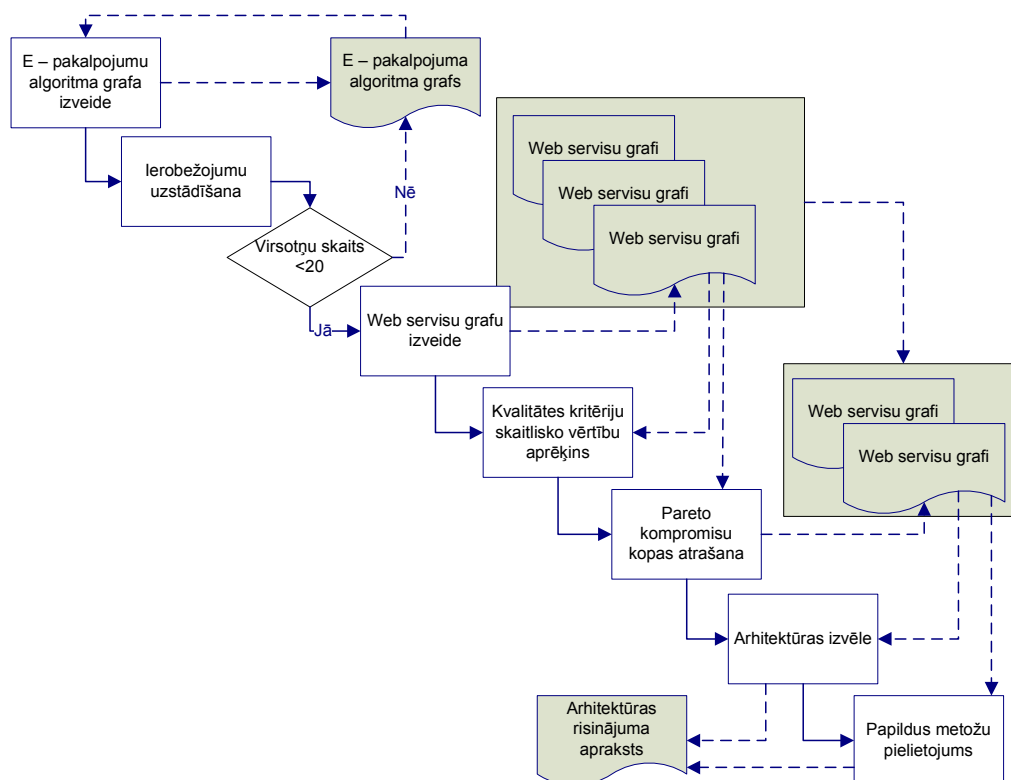
,kur $Q(X) = \{q_1(X), q_2(X), \dots, q_N(X)\}$ – minimizējamie kritēriji; Ω – X definīcijas apgabals. Risinājumu $X_i \in \Omega$ saucim par pareto optimālu tad un tikai tad, ja neeksistē $X_j \in \Omega$ tāds, ka

$$q_i(X_j) \leq q_i(X_i) \quad (2)$$

- visiem $i = \{1, 2, \dots, N\}$, kur vismaz viena ir stingra nevienādība. Citiem vārdiem sakot, jebkurai vērtībai $X_i \in P$ kopā Ω nav atrodama labāka vērtība par izvēlēto pēc katra no optimizējamajiem kritērijiem[9,11].
7. Atrastās kopas P web servisu grafi var kalpot par pamatu pieņemamas e – pakalpojumu sistēmas arhitektūras izvēlei. Web servisu grafus, kas ietilpst atrastajā Pareto kompromisu kopā, var detalizēti projektēt un implementēt e – pakalpojumu sistēmas arhitektūrā;
8. Pareto kompromisu kopa P vairumā gadījumu satur vairāk kā vienu risinājumu. Ja atrastā kopa ir pietiekoši maza, tad web servisu grafu izvēli no šīs kopas var atstāt sistēmas projektētāja pārziņā. Savukārt, ja iespējamo risinājumu skaits ir liels, tad, lai

varētu izvēlēties konkrētu web servisu grafu, var tikt pielietota kritēriju samazināšana, vai apvienošana ar kādu citu optimizācijas vai projektēšanas metodi, piemēram, svaru metodi vai Jan Bosch[6].

E – pakalpojumu sistēmas arhitektūras izvēles metodes projektēšanas dzīves cikls ir parādīts 4. att.



4. att. E – pakalpojumu projektēšanas metodes dzīves cikls

Fig. 4. Life circle of e – service design method

Projektēšanas metožu salīdzinājums

Datorsistēmu arhitektūras projektēšanas metožu salīdzinājums ļauj sistēmu arhitektiem izvēlēties konkrētajā gadījumā pielietojamās metodes lietderību, kā arī atklāj katras projektēšanas metodes iespējamus plusus un minusus. 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 ļoti laikietilpīga,

ir būtiski tās kvalitātes rādītājus novērtēt vēl pirms pašas sistēmas veidošanas;

1. 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;
4. Konkrēts risinājums – būtiski ir pārliecināties par to vai projektēšanas metode sniedz vienu vai vairākus risinājumus;
5. 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.

Projektēšanas metožu salīdzinājums pēc uzdotajiem kritērijiem ir parādīts 1. tabulā.

1. tabula
Table 1.

Arhitektūras projektēšanas metožu salīdzinājums
Comparison of Architecture Design methods

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

Kā redzams dotajā salīdzinājumā, tad eksistē metodes, kas ir daudzkriteriālas, un ir iespējams jau pirms izstrādes novērtēt sistēmas arhitektūras kvalitātes rādītājus, bet šo metožu precizitāti nosaka tikai un vienīgi eksperta pieredze un metodikas ievērošana. Tas nozīmē, ka netiek izslēgta iespēja, ka tiks izvēlēts lokāls minimums. Liels 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 pielietot matemātiskas daudzkriteriālas optimizācijas metodes. Šādos gadījumos ļoti 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.

E – pakalpojumu sistēmu arhitektūras projektēšanā ir nepieciešama projektēšanas metode, kas sniegtu ātru un kvalitatīvu rezultātu neatkarīgi no projektēšanā iesaistīto ekspertu kvalifikācijas. Pateicoties tam, ka e – pakalpojuma algoritmu ir iespējams formalizēti pierakstīt algoritma grafa veidā, tad ir iespējams pielietot piedāvāto web servisu 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 šķietami labākā risinājuma izvēles.

Ņemot vērā ka daudzkriteriālas optimizācijas rezultātā tiek iegūta Pareto kompromisu kopa, kas satur vairākus rezultātus, tad sarežģītākos ganījumos, 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, kas ietilpst Pareto kompromisu kopā. Šādā gadījumā tiek izslēgta varbūtība, ka tiks izvēlēts lokāls minimums.

Kopsavilkums

Lai apmierinātu visas e – pakalpojumam izvirzītās kvalitātes prasības ir būtiski izvēlēties tādu e – pakalpojumu sistēmas arhitektūras projektēšanas metodi, kas ir daudzkriteriāla, jo kvalitātes kritēriji ir savstarpēji pretrunīgi. Ja netiek izvēlēta daudzkriteriāla projektēšanas metode, tad ir iespējams, ka uzlabojot kādu no izvirzītajiem kvalitātes kritērijiem, tajā pašā laikā netiek ievērots kāds cits iespējami būtisks nosacījums. Izvēloties daudzkriteriālu projektēšanas metodi tiek rasts kompromiss starp e – pakalpojumam izvirzītajiem kvalitātes kritērijiem. Līdz šim zināmo projektēšanas metožu lielākais trūkums ir projektētāja subjektīvais viedoklis, kas ietekmē izveidojamās sistēmas kvalitatīvos rādītājus, kā arī iespēja, ka tiks izvēlēts lokāls minimums, tādējādi nemaz neapskatot pašu labāko iespējamo risinājumu. Piedāvātā projektēšanas metode ir pielietojama e – pakalpojumu sistēmu projektēšanai, kā arī to ir iespējams pielietot citām sistēmām, kas nav algoritmiski sarežģītas (tādas, lai ir iespējams uzbūvēt algoritma grafu) un ir balstītas uz servisu orientētu arhitektūru SOA, un web servisiem.



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

Literatūra

1. Bass L., Clements P., Kazman R. Software Architecture in Practice 2nd ed. - Boston: Addison-Wesley, 2003.
2. International Organization for Standardization: Software engineering - Product quality / Internets. - http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749
3. Zhu H. Software Design Methodology from Principles to Architectural Styles. - Amsterdam: Elsevier Butterworth Heinemann, 2005
4. Bosch J. Design and Use of Software Architectures. - Harlow: Addison-Wesley, 2000.

5. Žeiris E., Ziema M. E-pakalpojumu izveides problēmas// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 19. sējums. - Rīga: RTU, 2004. - 48.-53. lpp.
6. Žeiris E., Ziema M. Elektronisko pakalpojumu sistēmu arhitektūras izvēle// Rīgas Tehniskās universitātes zinātniskie raksti. Datorzinātne. 5. sērija 32. sējums. - Rīga: RTU, 2007. - 99.-107. lpp.
7. 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.
8. Žeiris E., Ziema 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.
9. Zeiris E., Ziema 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.
10. Kan S. Metrics and Models in Software Quality Engineering 2nd ed. - Boston: Addison-Wesley, 2003.
11. Miettinen K. Nonlinear Multiobjective Optimization. - Boston: Kluwer Academic Publishers, 1998.

Edzus Zeiris, Mg. 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 projektēšana

Sistēmas projektēšana ir process, kas klasificē sistēmai izvirzītās prasības, organizē tās un izveido vienotu sistēmas arhitektūru. E – pakalpojumu sistēmas arhitektūra ir e – pakalpojumu sistēmas struktūra vai struktūras, kas sevī ietver sistēmas komponentes, komponentu īpašības un to savstarpējās saites, kas nepieciešamas e – pakalpojuma darbībai. Klasiski sistēmu projektēšanā tiek pielietotas divas projektēšanas metodes – funkcionāli bāzēta un objektu orientēta. Izstrādājot e – pakalpojumus, ir ļoti būtiski izveidot e – pakalpojumu sistēmas arhitektūru tādu, lai e – pakalpojums apmierinātu visus tam izvirzītos kvalitātes kritērijus, kas veicinātu plašāku tā izmantojamību. Tas nozīmē, kas ir nepieciešams pielietot efektīvas projektēšanas metodes e – pakalpojumu sistēmu izveidei, kas nodrošinātu pieņemamas e – pakalpojumu sistēmas arhitektūras izveidi. Šādos gadījumos vienkārši funkcionāli bāzēta vai objektu orientēta projektēšanas pieeja ir nepietiekama. E – pakalpojumu projektēšanai var izmantot metodes, kuras risina sistēmas kvalitātes jautājumus jau sistēmas projektēšanas fāzē. Tādās, kā atribūtu virzīta projektēšanas metode ADD, Programmatūras arhitektūras analīzes metode SAAM, Jan Bosch arhitektūras projektēšanas metode, Arhitektūras kompromisu analīzes metode ATAM, Iespējamo draudu analīzes projektēšanas metode HASARD, u.c. Šīs metodes risina jautājumu par sistēmas kvalitāti, bet tām ir viens liels 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. Šādos gadījumos var pielietot e – pakalpojumu sistēmas projektēšanas metodi, kas tiek balstīta uz web servisu projektēšanu daudzkritēriālas optimizācijas ceļā. Piedāvātā metode sastāv no septiņiem galvenajiem projektēšanas soļiem. Sākumā e – pakalpojums tiek aprakstīts algoritma grafa veidā, kura virsotnes segmentējot visos iespējamajos veidos, tiek iegūti web servisu grafi. Algoritma grafa segmentācija nozīmē to, ka tiek aplūkoti visi iespējamie web servisu grafi, tādā veidā aplūkojot visus iespējamus web servisu risinājumus, kas ietekmē e – pakalpojumu sistēmas arhitektūras kvalitāti. Pirms segmentācijas ir jānosaka iespējamie ierobežojumi algoritma grafa segmentēšanai. Iegūtajiem web servisu grafiem, tiek aprēķinātas kvalitātes kritēriju skaitlisko novērtējumu vērtības, izmantojot aprēķina formulas. No visiem web servisu grafiem tiek iegūta Pareto kompromisu kopa. Atrastās Pareto kompromisu kopas web servisu grafi var kalpot par pamatu pieņemamas e – pakalpojumu sistēmas arhitektūras izveidei. Web servisu grafus, kas ietilpst atrastajā Pareto kompromisu kopā, var detalizēti

projektēt un implementēt e – pakalpojumu sistēmas arhitektūrā. Ja iespējamo risinājumu skaits ir liels, tad, lai varētu izvēlēties konkrētu web servisu grafu, var tikt pielietota kritēriju samazināšana, vai apvienošana ar kādu citu optimizācijas vai projektēšanas metodi. Aplūkotās projektēšanas metodes ir savstarpēji salīdzinātas, lai sistēmas arhitekts varētu izvēlēties risinājumam piemērojamo projektēšanas metodi.

Zeiris E., Ziema M. Architecture Design of Electronic Services

System engineering is the process of classifying the system requirements, organize them and establishing a common system architecture. E - service system architecture is the e - services system, structure or structures that include elements of the system components, characteristics and their interactions, which are required for e - service execution. For system design classically is used two design methods - a functionally-based and object-oriented. In development of e - services it is important to select e - service system architecture to meet all the set of quality criteria to promote the usefulness of e - service. This means that it is necessary to use effective methods of designing e - services systems that provide an acceptable e - services system architecture creation. In such cases, a simple functional-based or object-oriented design approach is inadequate. Methods that address quality system issues in the system design phase for e - service system architecture design can be used. Such as Attribute-Driven Design method ADD, the Software Architecture Analysis method SAAM, Jan Bosch software architecture design method, Architecture Tradeoff Analysis Method ATAM, Hazard Analysis of Software Architectural Designs HASARD, etc. These methods address the question of the quality, but they have one major drawback. Systems architect or a person carrying out the system architecture design, subjective opinion, as well as the possibility that he will be missed by far the best solution, or it will not be discussed. The solution can be selected, that is not the best in case and can be only local minimum because not all possible cases are discussed. In such cases, the e - services system design method can be used, which is based that web services are designed on the multi-criteria optimization. The proposed method consists of seven major design steps. In the beginning it is necessary to create the e-service algorithm graph that describes the e-service to be designed. In the e-service algorithm graph, it is necessary to determine the possible restrictions for inter-segmentation of vertexes. By recursively segmenting the e-service algorithm graph, the web service graphs are obtained that can be used as the basis for the development of the e-service system architecture. The e-service algorithm graph is assumed as the initial web service graph. The numerical values of quality metrics of the

obtained web service graphs are calculated using the calculation formulae. Then from all the web service graphs, the Pareto optimality set is obtained and the web service graphs of the obtained set can serve as the basis for selecting an acceptable e-service system architecture. The web service graphs that are contained in the obtained set can be designed in detail and implemented in the e-service system architecture. In most cases, the Pareto optimality set contains more than one solution. If the obtained set is sufficiently small, the selection of web service graphs from the set can be left to the system designer. However, if there are many possible solutions, then in order to find the particular web service graph, the criteria can be decreased or combined with another optimisation or design method, e.g. weighted method or Jan Bosch method. The described design methods are mutually comparable that the system architect can choose applicable design method option for e - service system architecture design.

Жейрис Э., Зиема М. Проектирование архитектуры систем электронных услуг

Проектирование системы – это процесс, в котором систематизируются требования, выдвинутые к системе, и разрабатывается архитектура системы. Архитектура системы e-услуг – эта структура или структуры, включающие в себя все компоненты системы, их особенности, а также внутренние связи, необходимые для работы e-услуги. Существуют два метода проектирования систем: функционально-ориентированное проектирование и объектно-ориентированное проектирование. Разрабатывая e-услуги, очень важно разработать архитектуру системы e-услуг таким образом, чтобы e-услуга соответствовала всем критериям качества, выдвинутых для нее, что, в свою очередь, будет способствовать её более широкому применению. Для того чтобы обеспечить создание приемлемой архитектуры системы e-услуг, необходимо применить эффективные методы проектирования. При этом недостаточно просто использовать функционально-ориентированный или объектно-ориентированный подход. Для проектирования услуг можно использовать методы, которые решают вопросы качества системы уже на стадии проектирования системы: ADD, SAAM, метод проектирования архитектуры Яна Босха, ATAM, HASARD и другие. Все эти методы решают вопросы качества системы, но у всех у них есть один большой недостаток: субъективное мнение системного архитектора или другого лица, проектирующего архитектуру системы, а также вероятность упустить из виду самое лучшее решение. Можно выбрать достаточно хороший вариант, однако тот окажется всего лишь локальным минимумом, так как не были

рассмотрены все возможные варианты. В таких случаях можно использовать метод проектирования системы е-услуг, основанный на проектировании веб-сервисов при помощи многокритериальной оптимизации. Предложенный метод включает в себя семь главных шагов проектирования. На первом шаге описывается граф алгоритма е-услуги, сегментируя вершины которого всеми возможными способами, создаются графы веб-сервисов. Сегментация графа алгоритма означает, что рассмотрены все возможные графы веб-сервисов, рассматривая, таким образом, все возможные решения для веб-сервисов, которые влияют на качество архитектуры системы. Перед сегментацией необходимо обозначить возможные ограничения. Для полученных графов веб-сервисов рассчитываются числовые значения критериев качества, используя формулы для расчётов. Из всех графов веб-сервисов вычисляется оптимум множества Парето. Графы веб-сервисов, вошедшие во множество Парето, могут служить основой для выбора архитектуры системы е-сервисов. Графы веб-сервисов, вошедшие в найденное множество, можно детально проектировать и реализовывать в архитектуре. Если число возможных вариантов велико, тогда для выбора конкретного графа веб-сервиса возможно уменьшение количества критериев или объединение с каким-либо другим методом проектирования или оптимизации. Сравнение рассмотренных методов проектирования представлено в виде таблицы, чтобы системный архитектор имел возможность выбрать наиболее подходящий метод для конкретного решения.