

Nefunkcinių reikalavimų kintamumas paslaugų architektūros stiliaus sistemose

Vieslav Lapin

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Didlaukio g. 47, LT-08303 Vilnius
vieslav.lapin@mif.stud.vu.lt

Santrauka. Šiame straipsnyje aptariamas kintamų nefunkcinių reikalavimų specifikuojimas ir modeliavimas kuriant paslaugų architektūros stiliaus programų sistemas. Paslaugų sistemų kūrime šiai problemai spręsti dar nėra skiriama pakankamai dėmesio. Pasiūlytas kintamų savybių specifikuojimo metodas remiasi modeliais grindžiama sistemų kūrimo paradigma ir naudoja produktų šeimos kūrimo idėjas. Paslaugos variantai yra nustatomi analizuojant paslaugos nefunkcines savybes ir sudaromas individualizuotas paslaugos teikimo procesas, kurio modelis gali būti generuojamas automatizuotai.

Raktiniai žodžiai: paslaugomis grindžiamos sistemos, nefunkcinės savybės, adaptuojamumas, kintamumo specifikuojimas.

1 Įvadas

Programų sistemos turėtų būti kuriamos taip, kad būtų nesunku jas modifikuoti, kai to reikalauja besikeičiantys vartotojo poreikiai. Sistemų adaptuojamumą užtikrina paslaugomis grindžiamos sistemos, kurios yra išskirstytos ir sudarytos iš silpnai sukibusių, savarankiškų ir nuo realizavimo platformos nepriklausomų paslaugų [1]. Jos gali būti sukomponuojamos vykdymo metu, kas įgalina pakartotinį paslaugų panaudojimą ir adaptuojamumą.

Tačiau netgi fundamentiniai paslaugų architektūros stiliaus sistemų principai nenumato darbo su kintamybėmis kaip esminės šio tipo sistemų savybės [1]. Todėl turi būti išspręstos šios problemos:

- Siekiant bet kuriuo metu gauti sistemos paslaugas, turi būti užtikrinta galimybė vieną paslaugą pakeisti kita to paties funkcionalumo, bet galbūt kitos kokybės paslauga.
- Siekiant užtikrinti paslaugos kokybę ir optimizuoti sistemos veikimą, turi būti galimybė perkonfigūruoti visą sistemą, t. y. turi būti galimybė paslaugos realizaciją pakeisti kita ar net pasinaudoti kito teikėjo paslauga.

- Siekiant užtikrinti tinkamą sistemos veikimą įvairiuose kontekstuose, apimant skirtingas aplinkas ir naudotojų ypatumus, turi būti galimybė pakoreguoti sistemos veikimą. Tam reikia turėti skirtingas savybes įgyvendinančius skirtingus tos pačios paslaugos egzempliorius.
- Paslaugos paprastai nėra kuriamos atsižvelgiant į galimybę jomis naudotis tam tikro jų kintamumo ribose ar lanksčiai pritaikomas konkrečiam atvejui. Siekiant paslaugų kintamumo, kartu turi būti kuriami ir atitinkami įgyvendinimui reikiami artefaktai (tokie kaip specifikacijos ir modeliai), kurie sudarytų prielaidas kintamumui užtikrinti [2].
- Jei paslaugų kintamumas užtikrinamas nesistemiškai ir sistema adaptuojama nekontroliuojamu būdu, kyla sistemos dalių interoperabilumo problema [3].

Kuriant paslaugomis grindžiamas sistemas atsiranda savita darbo su šio tipo sistemų kintamumu specifika. Pirma, nefunkcinės charakteristikos, tokios kaip prižiūrimumas, apsauga, patikimumas ir našumas, pasižymi didesniu įvairumu nei tai yra kitokio tipo sistemose [4]. Antra, paslaugų nefunkcinės charakteristikos yra apibrėžiamos paslaugų lygmens susitarimuose (toliau SLA), tai yra dėl paslaugos kokybės turi susitarti paslaugos teikėjas ir gavėjas. Ilgą kūrimo istoriją turinčiose programų sistemose nefunkcinių savybių kintamumo užtikrinimas yra neišspręsta problema [5].

Šiame straipsnyje aptariamas paslaugomis grindžiamų sistemų nefunkcinių savybių kintamumo specifikavimo metodas, užtikrinantis galimų alternatyvų nagrinėjimą ir paslaugos individualizuoto varianto konfigūravimą. Antrame skyriuje apžvelgiami susiję darbai, nagrinėjantys paslaugų stiliaus sistemų kintamųjų specifikavimą ir modeliavimą. Trečiame skyriuje aptariamas paslaugų stiliaus sistemų nefunkcinių savybių kintamųjų specifikavimo ir modeliavimo metodas. Paskutiniame skyriuje pateikiamos išvados.

2 Susiję darbai

Darbas su kintamomis sistemos savybėmis vykdomas visose programų sistemų kūrimo stadijose [6]. Tačiau literatūroje pagrindinis dėmesys skiriamas architektūrinio lygmens sprendimams ir kintamųjų tipams, akcentuojant nefunkcinių reikalavimų kintamumą. Kintamumui užtikrinti plačiausiai taikomas produktų linijų inžinerijoje naudojamas sprendimo būdas [7] [8]. Darbe [9] daroma išvada, kad produktų linijų inžinerijos sprendimai nepakankami nefunkcinių savybių kintamumui užtikrinti paslaugų stiliaus ar-

chitektūros sistemose. Antras mažiau taikytas sprendimo būdas – šablonų naudojimas kintamybės modeliuoti ir variacijų taškams aprašyti [5]. Paslaugų kokybės atributų kintamumo modeliavimas tik pradėdamas tyrinėti [10].

Kintamumas modeliuojamas pasirenkant procesų variantų klases iš kelių veiklų šakų. Kiekvienas šakojimosi taškas atitinka variacinį tašką arba sprendimo tašką. Variacija produktų šeimose nusako kaip šeimos nariai skiriasi vienas nuo kito [11]. Variacijos taškas yra sprendimo (išsišakojimo) taškas kartu su pasirinkimais. Kiekvienas pasirinkimas, išsišakojantis iš variacinio taško (funkcijos ar kokybės produktų linijų kontekste), yra apibrėžiamas kaip *variantas* [12]. Produktų linijų kūrime nėra išreikštinių skirtumų tarp variacinių ir sprendimo taškų.

Frankova ir kiti siūlo verslo procesų su fiksuotais kokybės matais išvedimą iš ankstyvųjų reikalavimų [13] [14]. Projektuojant paslaugų sistemų verslo procesus svarbu užtikrinti ne tik tam tikrus darbų srautus, bet ir tinkamą paslaugos vykdymo kokybę, t. y. nefunkcines savybes, fiksuojamas SLA. BP&SLA metodika formuoja ankstyvuosius reikalavimus ir aprašo naudotojo valdomos transformacijos ir samprotavimo priemones, ko rezultatas yra verslo procesai, aprašyti Secure BPEL kalba bei SLA kontraktų aibė. Nefunkcinės savybės yra derinamos tarp užsakovo ir paslaugos tiekėjo specifikuojant SLA.

Rosa ir kiti išskiria šiuos specifikuojamų verslo procesų kintamumo mechanizmus: mazgų konfigūravimą, elementų anotavimą, veiklos specializavimą ir fragmentų adaptavimą [15]. Šiame metode variacijos taškas yra adaptavimo proceso modelio mazgas, kuriam priskiriami skirtingi nustatymai. Veiklos, įvykiai, sietuvai (angl. *gateway*), resursai ir objektai žymimi kaip konfigūruojami mazgai. Adaptavimas yra pasiekiamas pasirenkant vieną nustatymą vienam konfigūruojamam mazgui. Elementų anotavimas suteikia valdymo srautų mazgams grafines anotacijas su dalykinės srities ypatybėmis. Specializuojant veiklas abstrakčių veiklų variaciniams taškams priskiriami vienas arba daugiau variantų. Fragmentų adaptavime ribojamas arba išplečiamas proceso modelis naudojant keitimo operacijas, pvz.: trinti, įdėti, perkelti.

DiVA metodas apibrėžia dinaminio kintamumo adaptuojamose paslaugomis grindžiamose sistemose valdymo bei kompleksiško kontrolės būdą [16]. Todėl nekontroliuojamas konfigūracijų skaičiaus augimas yra suvaldomas, modeliujant kintamumo dimensijas ir variantus, o ne visą pa-

slaugos konfigūraciją. Adaptacijos logika jame yra aprašoma aukštame abstrakcijos lygmenyje, tokiu būdu atskiriama adaptacija nuo pačios sistemos. Aspektų modeliai validuojami ir verifikuojami ankstyvuose sistemos kūrimo gyvavimo ciklo etapuose, todėl galimos problemos atsiskleidžia projektavimo, o ne vykdymo metu, kai jų sprendimas taptų sudėtingas ir brangus. Produktų šeimos kintamumo specifikavimo ir modeliavimo metodika DiVA skirta modeliuoti, analizuoti, vertinti ir komponuoti kintančias prisitaikančių sistemų funkcinį savybių konfigūracijas paslaugos gyvavimo cikle [16]. Ši metodika neapima paslaugas realizuojančio proceso automatizuoto specifikavimo, nefunkcinį savybių specifikavimo ir modeliavimo.

3 Paslaugų ir procesų kintamų nefunkcinį savybių specifikavimas

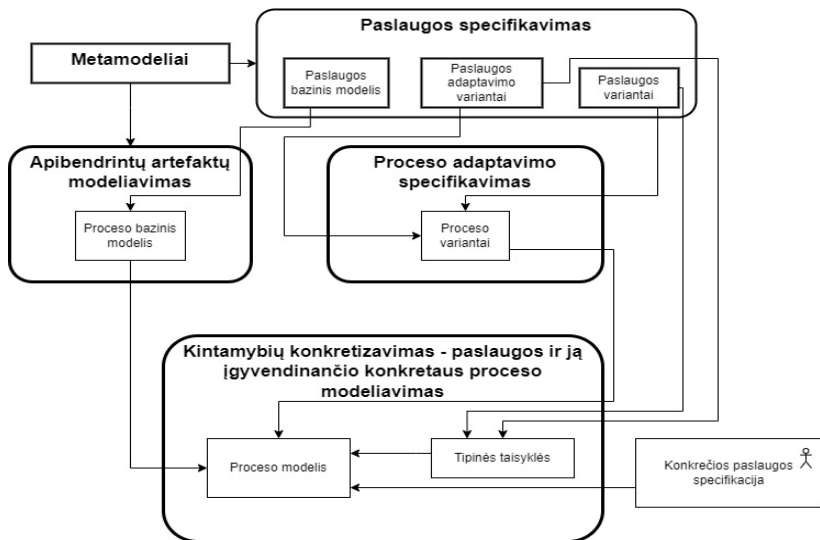
Paprastai yra kuriamos ne atskiros paslaugos, o paslaugų šeimos, skirtos individualizuoti paslaugų teikimą, kai paslaugos adaptuojamos specifiniam naudojimo kontekstui. Naudojimo kontekstas apima naudotojus, jų tikslus ir veiklas, veikimo aplinką ir technologijas [17]. Verslo procesus įtakoja suteikiamos paslaugos ir naudojimo konteksto kitimas, kas savo ruožtu sąlygoja funkcinį ir nefunkcinį reikalavimų kitimą.

Šiame skyriuje aptariamas kintamų savybių specifikavimo metodas, pateikiami paslaugos ir proceso metamodeliai, aprašomas rezultatų gavimo procesas, pateikiama jo konkretizacija ir pagrindžiantys argumentai.

3.1 Metodo bendroji charakteristika

Kintamų savybių specifikavimo metodas įgyvendinamas naudojantis modeliais grindžiama sistemų kūrimo paradigma ir produktų šeimos kūrimo idėjomis. DiVA metodika yra modifikuojama ir išplečiama, užtikrinant kintančių procesų nefunkcinį savybių kintamumo specifikavimą ir modeliavimą.

Paslaugas įgyvendinančiose sistemose skirtingi vartotojai gali gauti skirtingas to paties tipo paslaugas. Tam tikslui identifikuojamos kintamos paslaugų ir jas įgyvendinančių procesų dalys, kurios adaptuojamos konkrečiam atvejui. Siūlomame metode paslaugos variantai yra išskiriami iš naudojimo scenarijų, kuriuose akcentuojamos paslaugos nefunkcinės savybės. Metodas apima paslaugos specifikavimą, apibendrintų artefaktų modeliavimą, proceso adaptavimo specifikavimą ir kintamųjų konkretizavimą (1 pav.). Stačiakampiai su apvalintais kampais žymi procesus, stačiakampiai – artefaktus.



1 pav. Kintamybių specifikavimo koncepcinis modelis.

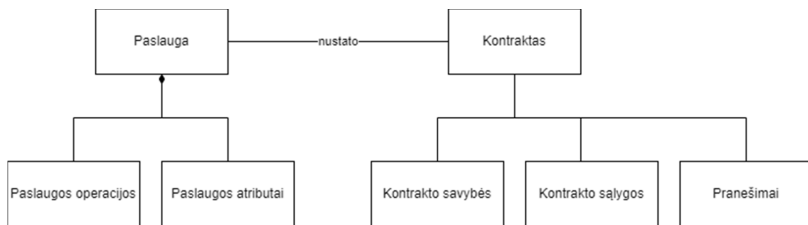
3.2 Paslaugos ir proceso metamodeliai

Paslauga suprantama kaip savybių ir funkcionalumo visuma (atributai ir operacijos); paslauga teikiama pagal kontrakte nustatytas sąlygas (2 pav.). Operacijos yra veiksmai susiję su teikiamais paslaugos variantais. Paslaugos atributai apibrėžia esybes ir jų savybes, kuriomis manipuliuoja operacijos. Paslaugos kontrakte apibrėžiami paslaugos teikimo kokybės lygmenys ir teikiamų paslaugos variantų sąlygos.

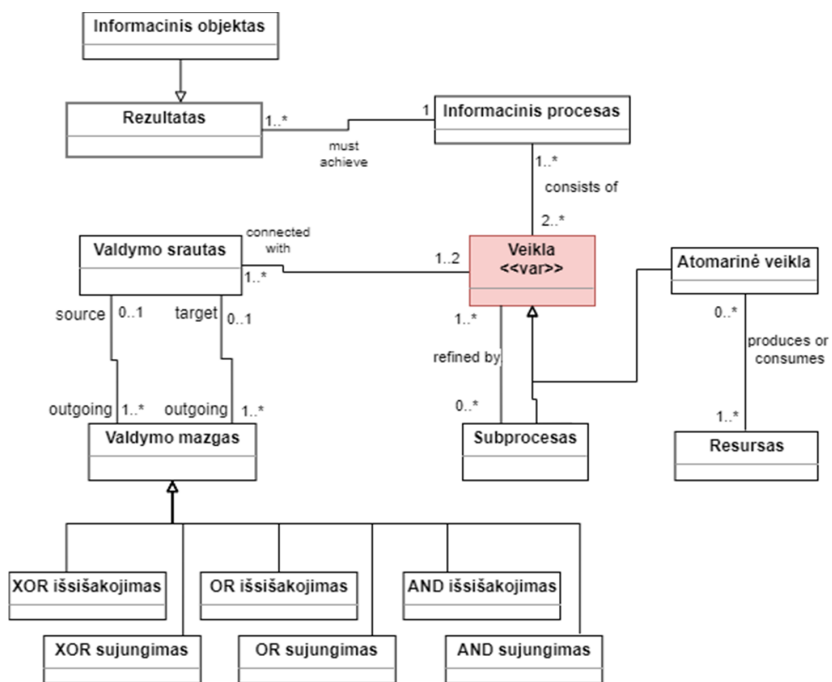
Projektuojant paslaugą su kintamybėmis, paslaugą realizuojantys verslo procesai yra sudaryti iš kintamų ir nekintamų veiklų (3 pav.). Siūlomą verslo proceso metamodelį (dalinai adaptuojamas iš [18]) sudaro šie elementai:

- 1) informacinis procesas,
- 2) veiklos (kintamos ir nekintamos),
- 3) atominė veikla,
- 4) subprocesas.

Informacinis procesas yra sudaromas iš veiklų. Kiekviena veikla turi teikti rezultatą, kuris yra informacinis objektas teikiantis reikalingą informaciją veiklą inicijavusiam naudotojui. Veikla atitinka atominę veiklą arba



2 pav. Paslaugos metamodelis.



3 pav. Verslo proceso metamodelis (kintamas elementas pažymėtas raudonai)

subprocesą. Atominė veikla yra nedaloma proceso dalis. Subprocesas, savo ruožtu, yra sudarytas iš kitų veiklų, dažnai atominių, rečiau dar iš kitų subprocesų. Iš paslaugomis grindžiamos sistemos perspektyvos, atominėje veikloje yra apdorojami informaciniai išteklių. Veiklų eiliškumą nurodo valdymo srautas. Valdymo srautas yra valdomas operatorių, vadinamų valdymo mazgais. Valdymo mazgai gali būti šių rūšių:

1. XOR išsišakojimas – iš skirtingų šakų pasirenkamas vienas pagal nustatytą sąlygą.
2. XOR sujungimas – pasirinkta šaka baigiasi ir grįžtama į bendrą proceso srautą.
3. OR išsišakojimas – iš skirtingų šakų pasirenkama viena arba daugiau pagal nustatytą sąlygą.
4. OR sujungimas – pasirinktos šakos baigia vykdomą ir visi srautai susilieja atgal į vieną srautą.
5. AND išsišakojimas – proceso šakos vykdomos lygiagrečiai.
6. AND sujungimas – proceso šakos baigia vykdomą ir visi srautai susilieja atgal į vieną srautą.

3.3 Kintamumo mechanizmas

Atsižvelgiant į literatūros analizėje nagrinėtą medžiagą ir ją apibendrinant, nustatyti atliekamo tyrimo kontekste nagrinėtini kintamumo įgyvendinimo mechanizmai, kuriems įgyvendinti anotuotos veiklos gali būti:

1. variacinis taškas,
2. numatytas variantas,
3. variantas,
4. nulinė veikla,
5. neprivaloma veikla.

Variacinis taškas nusako veiklos elementą, kuris specifikuojamas arba projektavimo metu dar neturi konkrečios suplanuotos veiklos, bet galimi keli variantai iš kurių vienas turi būti pasirinktas. Jie pasirenkami atsižvelgiant į pasirinkimo metu susiklosčiusią situaciją. Galimai toks pasirinkimas yra atliekamas paslaugos vykdymo metu.

Numatytas variantas yra veikla, kuri turėtų būti įkelta į variacinio taško vietą ir inicijuojama bendru atveju, jeigu iniciatorius nedaro išreikštinio sprendimo, kurį variantą naudoti.

Variantas yra veikla, kurią naudotojas gali pasirinkti iš galimų alternatyvų, kurios bus vykdomas tame variaciniame taške. Kitaip negu numatyto varianto atveju, varianto veiklą naudotojas turi pasirinkti išreikštinu būdu.

Nulinė veikla yra panašus į variacinį tašką, tik gali būti praleistas, tai yra nepasirenkamas joks iš apibrėžtų variantų, jeigu konkrečioje situacijoje tam nėra poreikio.

Neprivaloma veikla yra veiklos realizacija ir nėra variacinis taškas. Ji gali įvykti arba ne.

Apibrėžti kintamumo mechanizmai leidžia specifikuoti proceso kintamumus slepiant konkretizacijos detales, kurios yra apibrėžiamos žemesniame realizacijos lygmenyje. Tokiu būdu naujo varianto atsiradimas neturi poveikio bazinio proceso specifikacijai.

3.4 Paslaugos specifikavimas

Paslaugos bazinis modelis nusako operacijas, nekintančias paslaugos teikimo procese. Tai pagrindinė paslaugos dalis, kuri būtina visuose jos variantuose.

Paslaugos adaptavimo variantai yra susiję su paslaugos kintamumu ir dinamiu adaptavimu. Adaptavimo variantai apibrėžiami remiantis naudojimo konteksto scenarijais suderintais su suinteresuotų šalių tikslais. Naudojimo konteksto scenarijuose specifikuojami paslaugos bazinis modelis, adaptavimo variantai ir paslaugos variantai. Adaptavimo variantai nusako paslaugų nefunkcines savybes.

Aukščiausio lygio sistemos ypatybes atitinka aukščiausio lygmens suinteresuotųjų šalių tikslus, kurie realizuojami kaip klasikinio pavidalo paslaugos. Skirtingi iš naudojimo konteksto išplaukiantys tikslai leidžia suformuluoti adaptavimo kategorijas, kurios savo ruožtu nusako potencialius variacijos taškus ir **paslaugos variantus**. Papildomai, bet koks svarbus kontekstas ir apribojimai yra irgi išgaunami, kad galima būtų įsitikinti, jog visas adaptacijas galima teisingai pritaikyti dinamiškai be vykdymo metu sukeltų klaidų ar konfliktų.

Paslaugos specifikacijoje išvardijamos minėtos operacijos ir atributai (1 lentelė).

Paslauga apibrėžia kontraktą, kuris išvardina nefunkcines paslaugos savybes ir aprašo jos teikimo kokybės matus. Vartotojas pasirenka atitinkantį jo poreikius paslaugos teikimo kokybės lygmenį. Paslaugų lygmenų kriterijai yra apibrėžiami, analizuojant naudojimo konteksto scenarijus (2 lentelė).

1 lentelė. Paslaugos specifikacija

Paslaugos operacijos ir atributai	Apibrėžimas
Paslaugos pavadinimas	
Paslaugos aprašas	Trumpas paslaugos aprašas
Bazinis paslaugos variantas	Aprašomi bendri paslaugų komponentai
Paslaugos variantai	Aprašomi kintami paslaugos komponentai
Paslaugos prieinamumas	Numatytas procentais paslaugos prieinamumas
Paslaugos savininkas	Atsakingo asmens kontaktai
Paslaugos naudotojai	Paslaugos tikslinės naudotojų grupės
Paslaugos veikimo kokybės matai	Vienas ar daugiau bazinio lygmens veikimo matų
Pagrindiniai tikslų rodikliai	Bendra paslaugos nauda klientui

2 lentelė. Paslaugos kontrakto aprašas

Paslaugų adaptavimo variantai	Paslaugos lygmuo 1	Paslaugos lygmuo 2	...	Paslaugos lygmuo N
Nefunkcinė charakteristika 1	Kriterijus ₁₁	Kriterijus ₂₁	...	Kriterijus _{N1}
Nefunkcinis charakteristika 2	Kriterijus ₁₂	Kriterijus ₂₂	...	Kriterijus _{N2}

3.5 Apibendrintų artefaktų modeliavimas

Šiame etape, iš ankstesnės paslaugos specifikavimo dalies sudaryto paslaugos bazinio modelio, sudaromas proceso bazinis modelis. Proceso baziniame modelyje yra sukomponuojami paslaugos specifikacijoje apibrėžti paslaugos proceso bazinės veiklos. Procese nustatomos bazinės veiklos ir jų eiliškumas.

3.6 Proceso adaptavimo specifikavimas

Šiame etape yra dekomponuojami paslaugos specifikacijoje išvardinti adaptavimo variantai, kurie susiejami su paslaugos variantais. Rezultate suformuojami proceso variantai. Dekomponuojant adaptavimo variantus, yra analizuojamos nefunkcinės savybės (NF) ir identifikuojamos charakteristikos, įtakančios paslaugos varianto pasirinkimą. Charakteristikoms yra priskiriamos vertės, gaunamos analizuojant paslaugos naudojimo kontekstą (3 lentelė).

3 lentelė. Adaptavimo kriterijų (nefunkcinių savybių) dekompozicija

Paslaugos variantai	NF ₁			NF ₂			NF _M		
	Ch ₁₁	...	Ch _{1N}	Ch ₂₁	...	Ch _{2N}	Ch _{M1}	...	Ch _{MN}
Bazinis	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas2	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1

3.7 Kintamybių konkretizavimas

Konkretizuojant kintamybes modeliuojamas konkretaus paslaugos varianto pasirinkimo procesas. Pasirinkimas yra apibrėžiamas tipinėse taisyklėse, kuriose susiejami adaptavimo variantai su paslaugos variantais. Proceso modelis nusako konkretaus paslaugos varianto proceso realizaciją. Tipinės taisyklės yra aprašomos pasitelkiant šabloną, kuriame yra keturi elementai:

- **pavadinimas** – identifikuoja šabloną,
- **taikymo sąlyga** – aprašoma problema ir sąlygos, kada šablonas gali būti pritaikomas,
- **sprendinys** – aprašomas problemos sprendinys – elementai, jų ryšiai bei sąveika. Sprendinys yra apibendrintas ir turi būti konkretizuojamas konkrečiu atveju,
- **rezultatas** – aprašomas tipinio sprendinio taikymo rezultatas, kuris gali būti nusakytas po-sąlygomis.

Tipinės taisyklės specifikuojamos atsižvelgiant į visas nefunkcines savybes (4 lentelė) arba į konkrečią vieną nefunkcinę savybę (5 lentelė).

4 lentelė. Variacinio taško, atsižvelgiančio į visas nefunkcines charakteristikas, tipinė taisyklė

Pavadinimas	VariacinisTaškas1
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkre-tinta veikla ir reikia užtikrinti visas paslaugos nefunkcines savybes.
Sprendinys	Pagal nustatytus leidžiamus visų adaptavimo variantų (nefunkcinių charakteristikų) kriterijų reikšmių intervalus, atrenkami proceso veiklos variantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos variantais (<<Variant>>, vienas arba daugiau).

5 lentelė. Variacinio taško, atsižvelgiant į vieną nefunkcinę charakteristiką, tipinė taisyklė

Pavadinimas	VariacinisTaškas2
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkretinta veikla ir reikia užtikrinti vieną paslaugos nefunkcinę savybę.
Sprendinys	Pagal nustatytus leidžiamus vieno adaptavimo varianto (nefunkcinę charakteristiką) kriterijų reikšmių intervalus, atrenkami proceso veiklos variantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos variantais (<<Variant>>), vienas arba daugiau).

Proceso modelis nusako konkrečios paslaugos varianto realizaciją. Jeigu sąlygas atitinka keli variantai, galutinį pasirinkimą padaro vartotojas. Proceso modelis yra kuriamas atsižvelgiant į proceso bazinį modelį, proceso variantus, tipines taisykles ir konkrečios paslaugos specifikaciją, formuojamą realiuoju laiku iš konkretaus naudojimo konteksto duomenų.

Apibendrinant pabrėšime, kas pasiūlytas nefunkcinių savybių susiejimas su paslaugų šeimos kūrimu. Tai realizuojama šiuo būdu:

1. Analizuojant paslaugos naudojimo kontekstą yra kuriama paslaugos specifikacija. Jos pagrindu išskiriamas paslaugos bazinis modelis, naudojimo kontekstą atitinkantys adaptavimo variantai ir paslaugos variantai.
2. Modeliuojant apibendrintus artefaktus proceso baziniame modelyje identifikuojamos pagrindinės nekintamos ir kintamos veiklos.
3. Proceso adaptavimo specifikavimo fazėje identifikuojami proceso variacinių taškų variantai, bei siejami su nefunkcinių savybių charakteristikomis, priskiriant įverčius.
4. Konkretizuojant kintamybes, tipinės taisyklės aprašo konkretų pasirinkimą konkrečiame naudojimo kontekste.
5. Galiausiai, naudojant proceso bazinį modelį, proceso variantus ir jų adaptavimo variantus, tipines taisykles, formuojamas reikiamas proceso modelis. Jeigu jame išlieka keli pasirinkimai, galutinį sprendimą priima vartotojas.

Metodo charakterizuojamas eile ypatumų. Pirma, kreipiant pagrindinį dėmesį į variantų modeliavimą, nuo skirtingų kontekstų ir nuo paslaugos savybių priklausantis variantų skaičiaus padidėjimas gali būti valdomas kontroliuojamu būdu. Antra, naudojant modeliais grindžiamą sistemų kūrimo paradigmą, konceptai ir adaptavimo logika gali būti specifikuojama aukš-

tesniame abstrakcijos lygmenyje, taip užtikrinant turinių atskyrimą koncepcinių, kintamų ir konkrečių turinių. Trečia, taikant produktų šeimos idėjas, adaptuotas turinys gali būti generuojamas automatizuotai, kas apima konkretaus artefakto gavimą iš apibendrinto modelio.

4 Išvados

Šiame darbe sukurtas paslaugomis grindžiamų sistemų kintamumo specifikavimo metodas, kuriame paslaugos variantai yra nustatomi analizuojant paslaugos nefunkcines savybes ir sudaromas individualizuotas paslaugos teikimo procesas. Atlikus literatūros analizę nustatyta, kad išnagrinėtose metodikose modeliuojami tik funkcinių savybių kintami turiniai ir specifiuojami paslaugomis grindžiamų sistemų procesai, apimant nefunkcinius reikalavimus, bet nekalbama apie kintamus turinius.

Sukūrus paslaugomis grindžiamų sistemų paslaugų variantų kintamumo specifikavimo metodą, nustatyta, kad nefunkcines savybes nusakančių charakteristikų pagrindu ir naudojant tipines taisykles proceso variantams pasirinkti. Taikant produktų šeimos idėjas ir tipines taisykles, konkrečiai paslaugai adaptuotas turinys, t. y., konkretaus proceso modelis, gali būti generuojamas automatizuotai. Taip pat nustatyta, kas kreipiant pagrindinį dėmesį į kintamųjų modeliavimą, nuo skirtingų kontekstų ir nuo skirtingų paslaugos savybių priklausantis variantų skaičiaus padidėjimas gali būti valdomas kontroliuojamu būdu.

Literatūra

- [1] Erl, T. (2005). *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Upper Saddle River, NJ. Prentice Hall.
- [2] Medeiros, F. M., de Almeida, E. S. & de Lemos Mei, S. R. (2009). Towards an approach for service-oriented product line architectures. *Workshop on Service-oriented Architectures and Software Product Lines*. San Francisco, CA.
- [3] Jiang, J., Ruokonen, A. & Systa, T. (2005). Pattern-based variability management in web service development. *Third European Conference on Web Services, IEEE*, 12 p.
- [4] Gu, Q. & Lago, P. (2009). Exploring service-oriented system engineering challenges: a systematic literature review. *Service Oriented Computing Applications*, 3(3), 171-188.
- [5] Galster, M. & Avgeriou, P. (2013). *Variability in Web Services, Systems and Software Variability Management*. R. Capilla, J. Bosch & K. Kang, Mont., Springer Verlag, p. 269–278.
- [6] Galster, M., Weyns, D., Tofan, D. & Michalik, B. (2014). Variability in Software Systems – a Systematic Literature Review. *IEEE Transactions On Software Engineering*, 40(3), 282–306.

- [7] Narendra, N.C. & Ponnalagu, K. (2010). Towards a Variability Model for SOA-based Solutions, IEEE International Conference on Services Computing, Miami, USA, 562–569.
- [8] Sun, C., Rossing, R., Sinnema, M., Bulanov, P. & Aiello, M. (2010). Modeling and Managing the Variability of Web-Service-Based Systems. *Journal of Systems & Software*, 83(3), 502–516.
- [9] Mahdavi-Hezavehi, S., Galster, M. & Avgeriou, P. (2013). Variability in Quality Attributes of Service-Based Software Systems: A Systematic Literature Review, *Information and Software Technology*, 55(2), 320–343.
- [10] Alebrahim, A., Faßbender, S., Filipczyk, M. & Goedi, M. (2016). Variability for Qualities in Software Architecture. *SIGSOFT Software Engineering Notes*, 41(1), 32–35.
- [11] Weiss, D.M. & Lai, C.T.R. (1999). *Software product-line engineering: a family-based software development process*. Addison-Wesley.
- [12] Halmans, G. & Pohl, K. (2004). Communicating the variability of a software product family to customers. *Inform – Forsch und Entwicklung*, 18, 113–131.
- [13] Frankova, G., Yautsiukhin, A. & Seguran, M. (2007). From early requirements to business processes with service level agreements. University of Trento.
- [14] Frankova, G., Séguran, M., Gilcher, F., Trabelsi, S., Dörflinger, J. & Aiello, M. (2011). Deriving business processes with service level agreements from early requirements. *Journal of Systems and Software*, 84(8), 1351–1363.
- [15] Rosa, M. L., Aalst, W. M. V. D., Dumas, M. & Milani, F.P. (2017). Business process variability modeling: A survey. *ACM Computing Surveys (CSUR)*, 50 (1), 1–45.
- [16] Greenwood, P., Chitchyan, R., Ayed, D., Girard-Reydet, V., Fleurey, F., Dehlen, V. & Solberg, A. (2011). Modelling service requirements variability: The DiVA way. *Service Engineering*, 55–84.
- [17] ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, 2018.
- [18] List, B. & Korherr, B. (2006). An evaluation of conceptual business process modelling languages. *Proceedings of the 2006 ACM Symposium on Applied Computing*.