

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ MAGISTRO STUDIJŲ PROGRAMA

**Kintamumo specifikavimas ir modeliavimas  
kuriant paslaugomis grindžiamas programų sistemas**

**Variability specification and modeling in service-based software systems**

Magistro baigiamasis darbas

Atliko: Vieslav Lapin (parašas)

Darbo vadovas: doc. dr. Audronė Lupeikienė (parašas)

Recenzentas: prof. dr. Romas Baronas (parašas)

Vilnius – 2022

## **Santrauka**

Verslo procesai nuolat kinta, todėl juos realizuojanti infrastruktūra turi būti lanksti. Paslaugų architektūros stiliaus sistemos turi įgyvendinti verslo procesų kintamumo palaikymą, užtikrinant paslaugos funkcionavimą skirtinguose kontekstuose bei jų pakartotinį panaudojimą. Darbe nagrinėjamos verslo lygmens poreikiams tenkinti paslaugomis grindžiamos sistemos, kurios turi būti pritaikomos tam tikram kitimų intervalui, kad palaikytų verslo procesų pokyčius, verslo funkcionalumo ir kokybės reikalavimų kintamumą. Esamose metodikose kintamumas specifikuojamas ir modeliuojamas funkcinių savybių pagrindu. Nefunkcinės savybės apibrėžiamos paslaugos kontrakte, kuriame yra nustatyti paslaugos kokybės parametrų lygmenys. Šios savybės nėra įtraukiamos į pačios paslaugos modeliavimą. Šiame darbe sukurtas paslaugomis grindžiamų sistemų nefunkcinių savybių kintamumo specifikavimo metodas, aprašantis paslaugos kokybinius atributus, įtakojančius paslaugos įgyvendinimo varianto pasirinkimą ir paslaugos individualizuoto varianto konfigūravimą. Metodo taikymas iliustruojamas pavyzdžiu, kuriame web paslaugų autentifikavimo paslaugos teikiami variantai konfigūruojami atsižvelgiant į nefunkcines panaudojamumo ir saugumo savybes.

Raktiniai žodžiai: paslaugų architektūros stiliaus sistemos, kintamumas, nefunkcinės savybės, specifikavimas, modeliavimas.

## **Summary**

Business processes are constantly changing, so the infrastructure that implements them must be flexible. Service-oriented architecture systems must support business process variability, ensuring adaptation for different contexts and reusability. The paper examines service-oriented systems aimed to meet the needs of the business level. They must be adapted to a certain range of changes to support variability of business processes, business functionality and quality requirements. In existing methodologies, variability specification and modeling are based on business service functional properties. Non-functional features are defined in the service contract, which sets the levels of service quality parameters. These features are not included in the modeling of the service itself. In this work, a method for specifying the variability of non-functional properties of service-oriented systems is developed. It describes the qualitative attributes of the service that influence the choice of the service implementation variant. Moreover, the developed method describes the configuration of the individualized variant of the service. The application of the method is illustrated by an example in which the variants provided by the authentication service of web services are configured considering the non-functional features such as usability and security.

**Keywords:** service-oriented systems, variability, non-functional features, specification, modeling.

## Turinys

Įvadas .....	6
1. Verslo procesas kaip inovacinis paslaugų sistemų elementas.....	10
1.1. Paslaugomis grindžiamų sistemų ypatumai .....	10
1.2. Paslaugomis grindžiamų sistemų tipai.....	13
1.3. Verslo procesas kaip tyrimo objektas .....	14
2. Verslo procesų kintamumo specifikavimo ir modeliavimo analizė .....	15
2.1. Bendrieji elementai kintamumui modeliuoti .....	15
2.2. Kintamumo įgyvendinimo būdai .....	16
2.3. Paslaugų sistemų reikalavimų specifikavimas.....	17
2.4. Paslaugų sistemų procesų specifikavimas BP&SLA metodika .....	19
2.5. Verslo procesų kintamumo modeliavimas.....	22
2.5.1 Mazgų konfigūravimas.....	22
2.5.2 Konfigūruojamos integruotų įvykiais grindžiamų procesų grandinės .....	23
2.5.3 Elementų anotavimas .....	25
2.5.4 Konfigūruojamų procesų modeliavimas .....	25
2.5.5 Veiklos specializacija.....	26
2.5.6 PESOA metodas.....	28
2.5.7 Fragmentų adaptavimas.....	29
2.5.8 Provop metodas .....	30
2.6. Adaptuojamų užduočių valdymas.....	31
2.6.1 CMMN notacija.....	32
2.6.2 CMMN taikymo pavyzdys.....	32
2.7. DiVA paslaugų kintamybių modeliavimas .....	35
2.8. Literatūros analizės apibendrinimas .....	37
3. Kintamų reikalavimų specifikavimo metodas.....	39
3.1. Kintamybių specifikavimo metodo bendra apžvalga.....	39
3.2. Metamodeliai .....	39
3.2.1 Paslaugos metamodelis .....	40
3.2.2 Proceso metamodelis.....	41
3.2.3 Kintamumo mechanizmai .....	42
3.3. Paslaugos specifikavimas .....	44
3.4. Apibendrintų artefaktų modeliavimas.....	45

3.5.	Proceso adaptavimo specifikavimas .....	47
3.6.	Kintamybių konkretizavimas .....	48
3.7.	Apibendrinimas .....	50
4.	Metodikos taikymo pavyzdys: autentifikacijos paslauga.....	51
4.1.	Paslaugos specifikacija .....	51
4.2.	Kontrakto specifikacija .....	52
4.2.1	Pirmasis naudojimo scenarijus – patogus.....	53
4.2.2	Antras naudojimo scenarijus – vidutiniškas (patogiai saugus) .....	53
4.2.3	Trečias naudojimo scenarijus – saugus .....	53
4.2.4	Kontrakto specifikacijos apibendrinimas .....	53
4.3.	Proceso bazinis modelis ir variantai .....	53
4.4.	Nefunkcinių savybių analizė.....	54
4.4.1	Panaudojamumo matai .....	55
4.4.2	Saugumo matai .....	56
4.5.	Web autentifikavimo konfigūracijos procesų variantų analizė.....	57
4.5.1	Tekstinis prisijungimas .....	57
4.5.2	Grafiniai slaptažodžiai.....	58
4.5.3	Telefonu grindžiamas prisijungimas .....	58
4.5.4	Biometrinė autentifikacija .....	58
4.5.5	Paslaugos variantų pasirinkimas .....	59
4.6.	Kintamybių konkretizavimas .....	59
	Rezultatai ir išvados .....	62
	Šaltiniai .....	63
	Priedai .....	68
	1 priedas. Darbo pagrindu sukurtas straipsnis.....	68

## Ivadas

### Nagrinėjamas objektas

Paslaugomis grindžiamos sistemos (angl. *service-based systems*) yra išskirstytos ir sudarytos iš silpnai sukibusių programinių paslaugų. Paslaugos yra savarankiški ir nuo realizavimo platformos nepriklausomi programiniai vienetai, kurie yra aprašomi, ieškomi, orkestruojami ir kuriami naudojant tarpusavyje sąveikaujančių sistemų kūrimo standartus [Erl05]. Paslaugomis grindžiamos sistemos gali būti sukomponuojamos netgi vykdymo metu. Taigi, paslaugų stiliaus architektūros (angl. *service-oriented architecture, toliau SOA*) sistemose akcentuojama silpna sankiba ir dinamiškas susiejimas, kas įgalina pakartotinį paslaugų panaudojimą (angl. *reuse*).

Paslaugos paprastai realizuoja verslui reikiamą funkcionalumą [Erl05], kurio procesai nuolat kinta. Paslaugomis grindžiamos sistemos yra lanksti infrastruktūra adaptuojamiems verslo procesams įgyvendinti. Taigi, paslaugų architektūros stiliaus sistemos turi įgyvendinti verslo procesų kintamumo palaikymą [ChK07], apimant skirtingą kokybę, pvz. skirtingo prioriteto naudotojams skirtingo lygmens našumo užtikrinimas, o taip pat ir funkcionavimą skirtinguose kontekstuose. Paslaugų pakartotinis panaudojimas būtų labai ribotas, jei šio tipo sistemos negalėtų adaptuotis prie kintančių verslo situacijų ir skirtingų kontekstų. Kitaip tariant, paslaugų kintamumas turi būti specifikuojamas ir įgyvendinamas išreikštiniu būdu.

Darbe nagrinėjamos verslo lygmens poreikiams tenkinti paslaugomis grindžiamos sistemos, kurios turi būti kuriamos kaip pritaikomos tam tikram kitimų intervalui, kad palaikytų verslo procesų pokyčius, verslo funkcionalumo ir kokybės reikalavimų kintamumą.

### Darbo aktualumas

Paslaugomis grindžiamos sistemos jau pagal savo apibrėžtį užtikrina tam tikrą adaptuojamumą. Tačiau netgi fundamentiniai paslaugų stiliaus architektūros sistemų principai nenumato darbo su kintamybėmis kaip esminės šio tipo sistemų savybės [Erl05]. Kita vertus, sukurti apibendrintas sistemas, kurios būtų pritaikomos bet kuriems verslo procesams palaikyti ir bet kurioms besikeičiančioms sąlygoms, yra nerealu. Taigi, šiame kontekste kintamumo užtikrinimas yra aktuali problema. Konkrečiai, priežastys, kurios sąlygoja aktualias spręstinas problemas, yra šios:

- Siekiant bet kuriuo metu gauti sistemos paslaugas, turi būti užtikrinta galimybė nefunkcionuojančią 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. jei sistema veikia neadekvačiai, turi būti galimybė paslaugos realizaciją pakeisti kita ar pasinaudoti kito teikėjo paslauga.

- Siekiant užtikrinti tinkamą sistemos veikimą įvairiuose kontekstuose, apimant skirtingas aplinkas ir naudotojų ypatumus, turi būti galimybė pakoreguoti sistemos veikimą atsižvelgiant į konkretaus konteksto ypatumus. Be to, 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 labai lanksčiai pritaikomas konkrečiam atvejui. Siekiant paslaugų kintamumo, kartu turi būti kuriami ir atitinkami paslaugomis grindžiamoms sistemoms įgyvendinti reikiami artefaktai (tokie kaip specifikacijos ir modeliai), kurie sudarytų prielaidas reikiamam kintamumui užtikrinti [MAL09].
- Jei paslaugų kintamumo užtikrinimo problema sprendžiama nesistemiškai ir sistema adaptuojama nekontroliuojamu būdu, kyla sistemos dalių interoperabilumo problema [JRS05].

### **Darbo naujumas**

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 [GL09], todėl jas gana sunku užtikrinti. Antra, reikiamos paslaugų nefunkcinės charakteristikos paprastai yra apibrėžiamos paslaugų lygmens susitarimuose (angl. *service level agreements*, toliau SLA), t.y. dėl paslaugos kokybės turi susitarti paslaugos teikėjas ir gavėjas. Šį kintamumo tipą yra sunku įgyvendinti. Net jau ilgesnė nagrinėjimų ir kūrimo istoriją turinčiose programų produktų linijose šio kintamumo užtikrinimas yra viena iš pagrindinių neišspręstų problemų [GA13]. Be to, paslaugų stiliaus architektūros sistemose identifikuojami kintamųjų tipai, kurių nėra kito tipo programų sistemose.

Darbas su kintamomis sistemos savybėmis vykdomas visuose programų sistemų kūrimo stadijose [GWT+14]. Tačiau literatūroje pagrindinis dėmesys yra skiriamas architektūrinio lygmens sprendimams ir šio lygmens kintamųjų tipams. Darbe [GWT+14] pateikta išsami literatūros, nagrinėjančios kintamumo užtikrinimo problemos sprendimą platesnėje srityje, t.y. programų sistemų inžinerijoje, analizė. Darbo išvadose konstatuojama, kad reikia plėsti tyrimus paslaugomis grindžiamose sistemose, atkreipiant dėmesį į nefunkcinių reikalavimų kintamumą.

Kintamumo užtikrinimo problema kuriant paslaugomis grindžiamas sistemas yra sprendžiama keliais būdais. Plačiausiai taikomas atitinkamai adaptuotas produktų linijų inžinerijoje (angl. *product line engineering*) naudojamas sprendimo būdas [NP10, SRS+10]. Darbe [MGA13] daroma išvada, kad produktų linijų inžinerijos sprendimai daro esminę įtaką, tačiau gali būti nepakankami nefunkcinių savybių kintamumui užtikrinti paslaugų stiliaus architektūros sistemose. Antras mažiau taikytas

sprendimo būdas – šablonų naudojimas kintamybėms modeliuoti ir variacijų taškams aprašyti [GA13]. Sprendžiant kintamumo užtikrinimo problemą naudojamos specializuotos kalbos, tokios kaip verslo procesų aprašymo kalbos BPEL plėtinys VxBPEL [KSS09] ar UML plėtinys [SRS+10]. Paslaugų kokybės atributų kintamumo modeliavimas tik pradedamas tyrinėti [AFF+16].

Apibendrinant galima teigti, kad paslaugomis grindžiamų sistemų nefunkcinių savybių kintamumo specifikuojimas yra nauja problema.

### **Tikslas ir uždaviniai**

Šio darbo tikslas – sukurti paslaugomis grindžiamų sistemų nefunkcinių savybių kintamumo specifikuojimo metodą, aprašantį paslaugos kokybinius atributus, įtakojančius paslaugos įgyvendinimo varianto pasirinkimą ir paslaugos individualizuoto varianto konfigūravimą.

Darbo uždaviniai:

1. Išanalizuoti paslaugomis grindžiamų sistemų kintamumus, nustatant reikalavimų inžinerijos ypatumus, kintamus verslo procesų ir paslaugų turinius bei jų specifikuojimo ir modeliavimo metodus.
2. Išanalizuoti pasiūlytus kokybės atributų (nefunkcinių savybių) kintamumo užtikrinimo problemos sprendinius paslaugomis grindžiamų sistemų reikalavimų inžinerijoje, siekiant nustatyti jų specifikuojimo galimybes.
3. Sukurti kokybės atributų kintamumo specifikuojimo ir modeliavimo metodą, užtikrinantį reikiamos kokybės sprendinius paslaugomis grindžiamų sistemų reikalavimų inžinerijoje.
4. Išnagrinėti reprezentatyvų paslaugų šeimos konfigūravimo pavyzdį, kurio pagrindu apibendrinti ir pademonstruoti nefunkcinių savybių ir jų kokybės atributų modeliavimo būdą, naudojamą konfigūruojant paslaugų šeimą.

### **Darbo atlikimo metodika**

Tyrimas vykdomas pagal induktyvinę-hipotetinę tyrimo strategiją [Sol82], kuri įgalina išplėsti mokslines žinias stebėjimų ir jų analizės pagrindu. Indukcinio-hipotetinio tyrimo strategijos charakteristikos yra šios:

- ji leidžia generuoti įvairius sprendinius, pradedant nuo esamos situacijos analizės;
- ji leidžia gauti grįžtamąjį ryšį ir įvertinti idėjas;
- ji naudinga, kai trūksta tinkamos teorijos ar metodinio palaikymo.

Dėl šių savybių strategija tinkama naujoms tyrimų sritims, kurių pavyzdys ir yra paslaugų architektūros stiliaus sistemos. Be to, nors literatūros nagrinėjama tema yra gana daug, tačiau tik nedi-



delė dalis yra relevantiška atliekamam tyrimui. Išspręsti paslaugų kintamų kokybės savybių specifikuavimo (reikiamuose lygmenyse) problemą grynai deduciniu būdu yra sunku, jei iš viso įmanoma. Šį tvirtinimą sustiprina faktas, kad nagrinėjamoje srityje dar nėra išsamios ir pagrįstos teorijos. Atsižvelgiant į tai, naujos žinios (darbo rezultatas) kuriamos nagrinėjant literatūrą, jas tikslinat ir vertinant praktinių pavyzdžių stebėjimo rezultate, ir apibendrinant gautas sumines žinias kad suformuoti teoretiškus rezultatus.

Tyrimas pradedamas nuo esamos literatūros analizės, kad apibrėžti pradinę tyrimo poziciją; be to, tai įgalina naujų idėjų suformavimą. Literatūros analizė taip pat naudojama, kad palyginti savo išvadas, išryškinti iš atvejo tyrimų kylančias idėjas ir geriau suprasti bei paaiškinti atvejo tyrimo rezultatus. Teorijos kūrimas grindžiamas indukcinė atvejo analize, bibliotekiniu tyrimu ir argumentavimo tyrimu (angl. *argumentative research*). Argumentavime taikomas Toulmino argumento modelis [Tou03].

## 1. Verslo procesas kaip inovacinis paslaugų sistemų elementas

Viena iš programų sistemų kūrimo problemų yra sukurti jas taip, kad būtų nesunku jas modifikuoti, kai to reikalauja besikeičiantys vartotojo poreikiai. Nuolatiniai sistemos pokyčiai yra neišvengiami šiandieniniame, greitai besikeičiančiame, pasaulyje. Paslaugų stiliaus sistemų kūrimas yra vienas šios problemos sprendimų būdų [MAL09].

Paslaugų paradigma skirta kurti lanksčias sistemas, kuriose paslaugos yra komponuojamos dinamiškai patenkinant verslo tikslus, atsižvelgiant į paslaugų vykdymo konteksto kintamumą [MGA13]. Paslaugų kūrimo metodikose pabrėžiama paslaugos viso gyvavimo ciklo svarba ir tos metodikos pateikia rekomendacijas ir priemones kaip nagrinėti reikalavimus, projektuoti ir įgyvendinti servisus. Paslaugų kompozicija yra paslaugų rinkinys, kuris yra vykdomas atsižvelgiant į jų apribojimus [RKS10]. Lankstumas vykdymo metu yra pagrindinė paslaugų kompozicijos charakteristika. Paslaugų kūrimas pasižymi specifinėmis charakteristikomis. Viena vertus, paslaugų kompozicija nėra būtina esant fiksuoto proceso vykdymo schemai. Kita vertus, tampa svarbios kai kurios proceso charakteristikos, susietos su globalia paslaugos kokybe ir pateikiamos sukomponuoto proceso. Proceso kokybė tampa paslaugos tiekėjo tikslu užtikrinant, kad paslaugos kompozicija tenkins pažadėtus kokybės apribojimus. Paslaugos kompozicija skiriasi nuo tradicinių darbo sekomis grindžiamų procesų tuo, kad paslaugų rinkinys, komponuojantis procesą, gali skirtis įvairių vykdymų metu.

### 1.1. Paslaugomis grindžiamų sistemų ypatumai

Paslaugų stiliaus architektūra (angl. *service-oriented architecture*) – tai organizacijos/įmonės lygmens sistemų ir jas palaikančių kompiuterinių sistemų kūrimo paradigma, kurios tikslas yra susieti poreikius ir jiems realizuoti reikiamus išteklius [Erl05]. Paslaugų paradigmos kontekste paslaugos suprantamos kaip būdai, kuriais naudotojų poreikiai siejami su paslaugų teikėjų galimybėmis. Ši paradigma akcentuoja ne sistemos sudedamųjų dalių sukūrimą, o organizavimą ir naudojimąsi funkcinių esybių galimybėmis spręsti problemas. Paslaugų sistema užtikrina bei valdo jų sąveiką.

Technologiniu požiūriu paslauga yra *mechanizmas*, skirtas pasiekti ir pasinaudoti galimybėmis [RMS06]. Todėl paslauga yra toks sistemos funkcionalumo įgyvendinimas, kad būtų prieinamas per aiškiai apibrėžtą sąsają.

Viena iš svarbiausių paslaugų paradigmos inovacijų yra turinių atskyrimo būdas [LMC06]. Tai atsispindi paslaugomis grindžiamų sistemų principuose [Erl05, Ars04]:

- paslaugos įgyvendina konkrečią veiklą, teikiančią reikiamą išėigą;
- naudotojams paslaugos yra juoda dėžė;
- paslauga gali būti komponuojama iš kitų paslaugų;

- paslaugos yra pakartotinai naudojamos;
- paslaugos turi formalius kontraktus;
- paslaugos inkapsuliuoja dalykinės srities logiką: vienintelė matoma iš išorės dalis yra paslaugos aprašas ir formalus kontraktas;
- paslaugos yra autonomiškos. Paslauga apima aiškias ribas turintį dalykinės srities logikos fragmentą, ir tik ji juo naudojasi bei modifikuoja. Autonomiškumas nusako, koku laipsniu paslauga valdo jai reikalingus išteklius;
- paslaugų sankiba yra silpna;
- paslaugos neturi būsenos, nes tai prieštarautų silpnos sankibos užtikrinimo reikalavimui. Būsenas turintys komponentai „iškeliami” iš paslaugų. Be to, projektuojant paslaugas, siekiama minimizuoti neišvengiamą jų būsenų kiekį ir buvimo tam tikroje būsenoje trukmę.
- naudojantis paslauga, nėra būtina žinoti jos fizinį adresą.

Paradigmos įvestas proceso ir paslaugos abstrakcijas išsamiau aptarsime paslaugų sistemų architektūros kontekste. Paslaugomis grindžiamos architektūros stiliaus sistemos paprastai apima visas automatizuojamos organizacijos veiklas, o architektūra yra sąlyginai sluoksninė, sudaryta iš šių horizontalių sluoksnių[Ars04] (1 pav.):

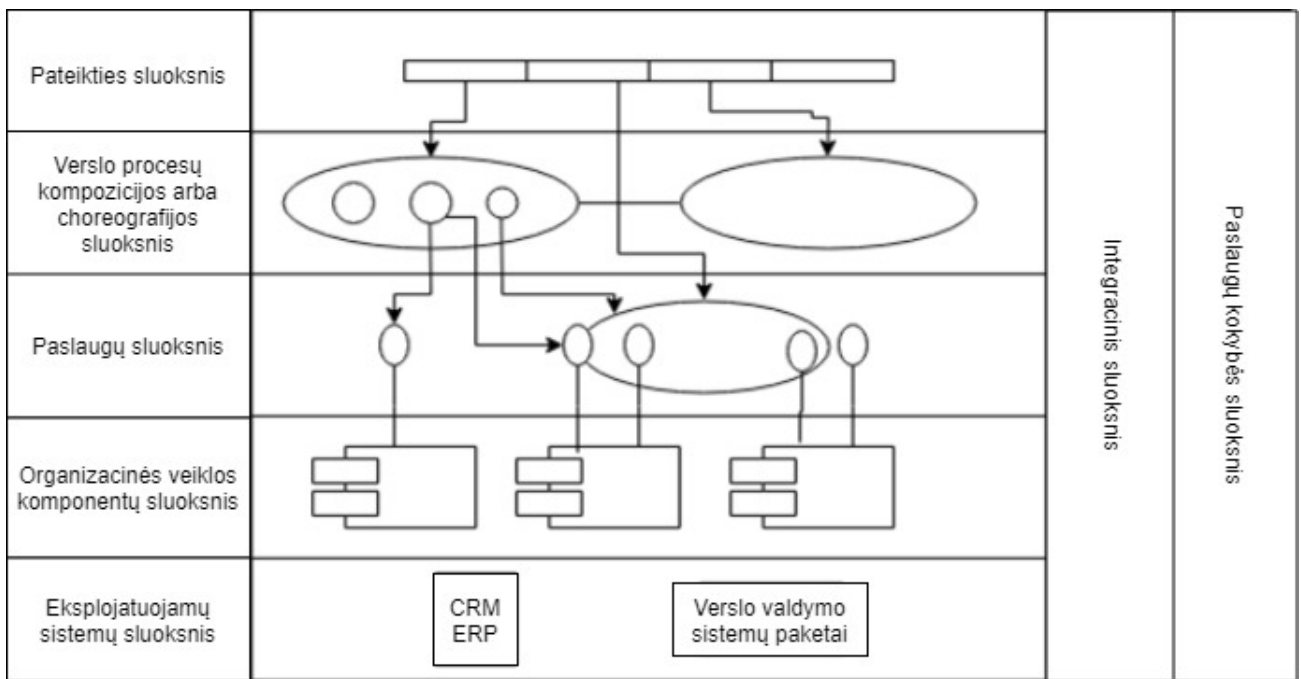
- 1) eksploatuojamų sistemų,
- 2) organizacijos veiklos komponentų,
- 3) servisų,
- 4) verslo procesų choreografijos,
- 5) pateikties.

Horizontalius sluoksnius kerta šie vertikalūs sluoksniai:

- 1) integracinis ir
- 2) paslaugų kokybės.

**Eksploatuojamų sistemų sluoksnis** apima egzistuojančias pagal užsakymą padarytas sistemas, taip pat ir liktines (angl. *legacy system*), kuriomis gali būti ir santykių su klientais ar verslo valdymo sistemų paketai. Kompozicinė sluoksniuota paslaugomis grindžiama architektūra leidžia jas panaudoti su maksimalia nauda ir integruoti jas naudojant paslaugų paradigmos technikas [Ars04].

**Organizacijos veiklos komponentų sluoksnyje** yra teikiamų paslaugų funkcionalumo realizavimo ir kokybės užtikrinimo komponentai. Šie komponentai yra valdomas organizacijos turtas, kuris yra finansuojamas verslo lygmenyje [Ars04].



1 pav. Servisinės architektūros sistemų sluoksniai (adaptuota iš [Ars04])

**Paslaugų sluoksnyje** yra nuspręstos teikti ir verslo finansuojamos paslaugos. Jos randamos arba statistiškai susiejamos, o vėliau išskviečiamos arba sujungiamos į vieną didesnę kompozicinę paslaugą. Šis paslaugų sluoksnis taip pat teikia mechanizmus paimti įmonės masto komponentus, verslo vieneto komponentus ir, kai kuriais atvejais, projekto komponentus bei pateikia jų interfeisus, aprašančius paslaugą. Tokiu būdu įmonės masto komponentai teikia paslaugų realizacijas vykdymo metu pasinaudojant interfeisų teikiamu funkcionalumu. Šiame sluoksnyje interfeisai eksportuojami kaip paslaugų aprašai. Šie aprašai gali būti tiek pavienių servisų, tiek jų kompozicijų [Ars04].

**Verslo procesų kompozicijos arba choreografijos sluoksnyje** pristatomos žemesnio sluoksnio servisų kompozicijos ar choreografijos. Servisai yra susieti į srautą, naudojant orkestravimą ar choreografiją. Jie veikia kaip viena programa. Tos programos atitinka konkrečius verslo uždavinius arba verslo procesus.

**Pateikties sluoksnis** paprastai yra už paslaugomis grindžiamų sistemų nagrinėjamos architektūros ribų. Tačiau apie jį reikėtų galvoti kaip apie sluoksnį, kurio nereikėtų užmiršti būsimuose sprendimuose. Paslaugomis grindžiamose sistemose vartotojo sąsaja yra atskirta nuo kitų komponentų, todėl reikalingas sprendimas, kaip ją susieti su paslauga arba jų kompozicija.

**Integracinis sluoksnis** užtikrina paslaugų integravimą, naudojant išmanius maršrutizatorius, tarpininkavimo protokolus, transformavimą ir kitas organizacijos paslaugų magistralės galimybes.

**Paslaugų kokybės sluoksnis** užtikrina funkcionalumą, skirtą stebėti, prižiūrėti, valdyti paslaugos kokybės charakteristikas (apsaugą, našumą, prieinamumą). Šis procesas veikia fone, naudoja *jausti ir reaguoti* mechanizmus ir įrankius, kurie stebi paslaugų architektūros programų savybes.

## 1.2. Paslaugomis grindžiamų sistemų tipai

Išskirstytos skirtingų teikėjų realizuojamų paslaugų sistemos yra labai neapibrėžtos, nepaisant pasirašomų kontraktų paslaugoms teikti. Skiriami du paslaugomis grindžiamų sistemų tipai:

- paslaugomis grindžiamos organizacijų sistemos,
- paslaugomis grindžiamos viešosios sistemos.

1 lentelėje yra pateikiami konceptualūs, technologiniai ir techniniai šių tipų paslaugomis grindžiamų sistemų skirtumai. Lentelė sudaryta darbe [LMC13] nurodytų skirtumų pagrindu, tačiau ji yra adaptuota šiame darbe nagrinėjamų problemų požiūriu.

1 lentelė. Įmonių bei viešomis paslaugomis grindžiamų sistemų skirtumai

<b>Paslaugomis grindžiamos viešosios sistemos</b>	<b>Paslaugomis grindžiamos organizacijų sistemos</b>
Atvira internete išskirstyta sistema. Kuriama stambinimo (angl. <i>bottom-up</i> ) metodu.	Sąlyginai uždara organizacijos viduje išskirstyta ir jos valdoma sistema.
Galima bet kokio verslo tipo paslauga. Nėra galimybių normalizuoti verslo paslaugas, t. y. gali egzistuoti kelios panašaus veikimo paslaugos.	Organizacijos veiklos normalizuotos paslaugos suderintos su organizacijos verslo funkcijomis.
Nėra galimybių apibrėžti globalių duomenų tipų.	Naudojami globalieji duomenų tipai.
Šios rūšies paslaugos nėra kuriamos realizuoti iš anksto žinomus verslo procesus.	Organizacijos verslo procesai koordinuoja organizacinės veiklos verslo paslaugų rinkinį.
Žinučių struktūra standartizuota, bet ne vieninga. Verslo paslaugų sąsajos yra standartizuotos, bet nėra aiškiai apibrėžtos ir nestabilios. Nėra galimybės naudoti globalius duomenų tipus sąsajose.	Paslaugų perduodamų žinučių struktūra yra vieninga. Organizacijos verslo paslaugų sąsajos yra aiškiai aprašomos, stabilios ir naudoja globalius duomenų tipus.
Susitarimas dėl paslaugų lygio derinamas tiekėjo ir tiesioginio vartotojo programos vykdymo metu.	Susitarimas dėl paslaugos lygio organizacijos vidaus sistemose vyksta projektavimo metu.

Neapibrėžtumas paslaugų sistemose gali būti valdomas ir minimizuojamas tik įmonių ar organizacijų sistemose, todėl toliau darbe nagrinėjamos tik tokio tipo sistemos.

### 1.3. Verslo procesas kaip tyrimo objektas

Verslo procesai yra mokslinių tyrimų objektas nuo 1950 metų, kai ir buvo įvestas šis terminas [BR10, Mil15]. Tuo metu bet kokia darbo veiklų seka buvo interpretuojama kaip verslo procesas. 1990-ųjų metų pradžioje susidomėjimas šia tematika vėl atsinaujino, kai amerikietiškos korporacijos pradėjo reorganizuoti procesus, kad įgautų konkurencinių pranašumų. Vienas verslo proceso apibrėžimas nusako, jog tai yra *struktūrizuotas, pamatuojamas rinkinys veiklų, suprojektuotų sukurti konkrečių išeigą konkrečiam klientui ar rinkai* [Mil15]. Kitas apibrėžimas nusako, jog tai yra *veiklų rinkinys, kurios priima vieną ar daugiau įeigos tipų ir kuria vertingą klientui išeigą*. Abu apibrėžimai yra skirti verslo lygmeniui, bet tie patys elementai yra naudojami ir bendresniuose apibrėžimuose. Pavyzdžiui, verslo procesas yra *tarpusavyje susijusių įvykių, veiklų ir sprendimo taškų rinkinys, apimantis aktorius ir objektus, ir skirtas gauti bent vienam klientui vertingą rezultatą* [Mil15]. Kitas bendro pobūdžio apibrėžimas nusako, jog verslo procesas yra *veiklų, derinančių organizacinę ir techninę aplinkas, rinkinys*. Šios veiklos įgyvendina verslo tikslą. Apibendrinant, visi apibrėžimai turi savyje *veiklų rinkinius*, kurie naudoja *įeigą* ir siekia gauti *nustatytą rezultatą* arba pasiekti *kliento verslo tikslą*.

Apibendrinant paslaugomis grindžiamų sistemų ypatumus reikia pabrėžti, kad šioje paradigmoje įvestos dvi aukšto lygmens abstrakcijos: paslauga ir verslo procesas [LMC13]. Šios abstrakcijos naudojamos įvairiuose įmonės sistemos lygiuose. *Paslaugos* yra dalykinių programų galimybių abstrakcijos, užtikrinančios reikiamą verslo funkcionalumą, kitaip sakant, dalykinės logikos vienetas. *Procesai* yra sistemos veiklos abstrakcija. Jie orkestruoja paslaugas. Todėl paslaugomis grindžiama sistema laikoma sąveikaujančių paslaugų, kurias koordinuoja verslo procesas, visuma. Paslaugų vartotojo požiūriu, verslo proceso specifikacija yra aprašas-programa, kurią gali vykdyti virtualioji mašina.

## 2. Verslo procesų kintamumo specifikavimo ir modeliavimo analizė

Šiame skyriuje pateikiama kritinė literatūros apžvalga. Tyrimas daromas siekiant atsakyti į šiuos klausimus:

1. Nustatyti paslaugų stiliaus sistemų reikalavimų inžinerijos savybes:
  - a. Kokios reikalavimų rūšys yra skiriamos?
  - b. Ką apima adaptuojamieji reikalavimai?
  - c. Kokius turinius turi apimti reikalavimų specifikacija?
2. Kokie kintami verslo proceso turiniai gali būti specifikuojami? Kaip?
3. Kokie kintami paslaugų turiniai gali būti specifikuojami? Kaip?
4. Kaip specifikuojamos verslo procesų ir paslaugų nefunkcinės savybės?

### 2.1. Bendrieji elementai kintamumui modeliuoti

Atsižvelgiant į proceso modelį arba į jų rinkinį, apimantį procesų variantų klases, būtinai atsiras taškai, kuriose daromas sprendimas pasirenkant iš kelių veiklų šakų. Pavyzdžiui, BPMN (angl. *Business Process Model and Notation*) notacijoje tokie pasirinkimai gali būti ARBA (angl. *OR*), IŠSKIRTINIO ARBA (angl. *XOR*) ar kitų tipų. Šie taškai vadinami išreikštinio šakojimosi taškais. Kitas šakojimosi tipas pasireiškia, kai yra renkama inicijuoti vieną ar kitą proceso modelį [Mil15].

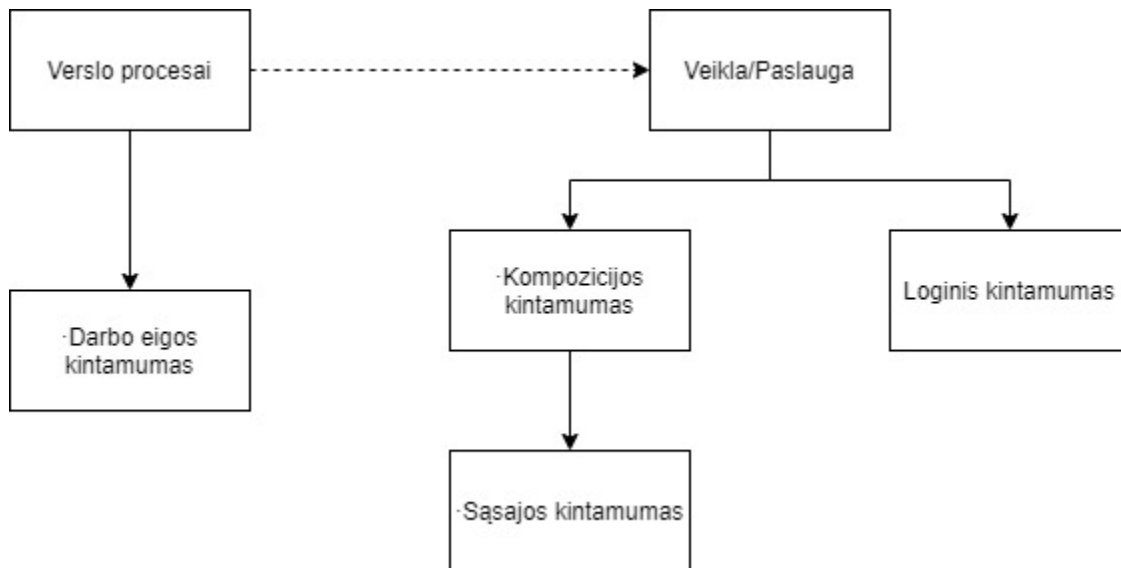
Kiekvienas šakojimosi taškas potencialiai atitinka arba variacinio tašką (angl. *variation point*) ar sprendimo tašką (angl. *decision point*). Terminas variacinis taškas yra žinomas programinių produktų linijos inžinerijoje (angl. *Software Product Line Engineering*, toliau SPLE) ir produktų linijos ypatybių diagramose (angl. *Feature Diagram*). Vienas variacijos apibrėžimas nusako, kad *variacija produktų šeimose nusako kaip šeimos nariai skiriasi vienas nuo kito* [WL99]. Remiantis šia prielaida, variacijos taškas yra sprendimo (išsišakojimo) taškas kartu su išplaukiančiais pasirinkimais. Kiekvienas pasirinkimas išsišakojantis iš variacinio taško (funkcijos ar kokybės produktų linijų kontekste) yra apibrėžiamas kaip *variantas* [HP04]. Produktų linijų srityje nėra išreikštinių skirtumų tarp variacinių ir sprendimo taškų.

Verslo procesų valdyme arba procesu patvirtinime svarbu išskirti variacinio taško ir sprendimo taško apibrėžimus. Variacinio ir sprendimo taško skirtumas yra nurodomas kaip kodo kompiliavimo ir kodo vykdymo skirtumas [RA07]. Jeigu pasirenkama programos kūrimo metu, tada identifikuojamas variacijos taškas. Jeigu pasirinkimas priklauso nuo aplinkybių, atsirandančių programos vykdymo metu [Mil15], tada tai yra sprendimo taškas.

## 2.2. Kintamumo įgyvendinimo būdai

Paslaugų kintamumas yra siejamas su skirtingais paslaugos reikalavimais klientams ir pačių paslaugų konteksto. Taigi, kintamumas yra identifikuojamas analizuojant paslaugos reikalavimus, o konkrečiai verslo procesus ir servisų vienetus (2 pav.). Verslo procesų ir servisų vienetų kintamumas yra keturių tipų:

- darbo eigos (angl. *workflow*) kintamumas,
- kompozicijos (angl. *composition*) kintamumas,
- sąsajos kintamumas,
- loginis kintamumas.



2 pav. Verslo procesų ir paslaugų vienetų kintamumo struktūra

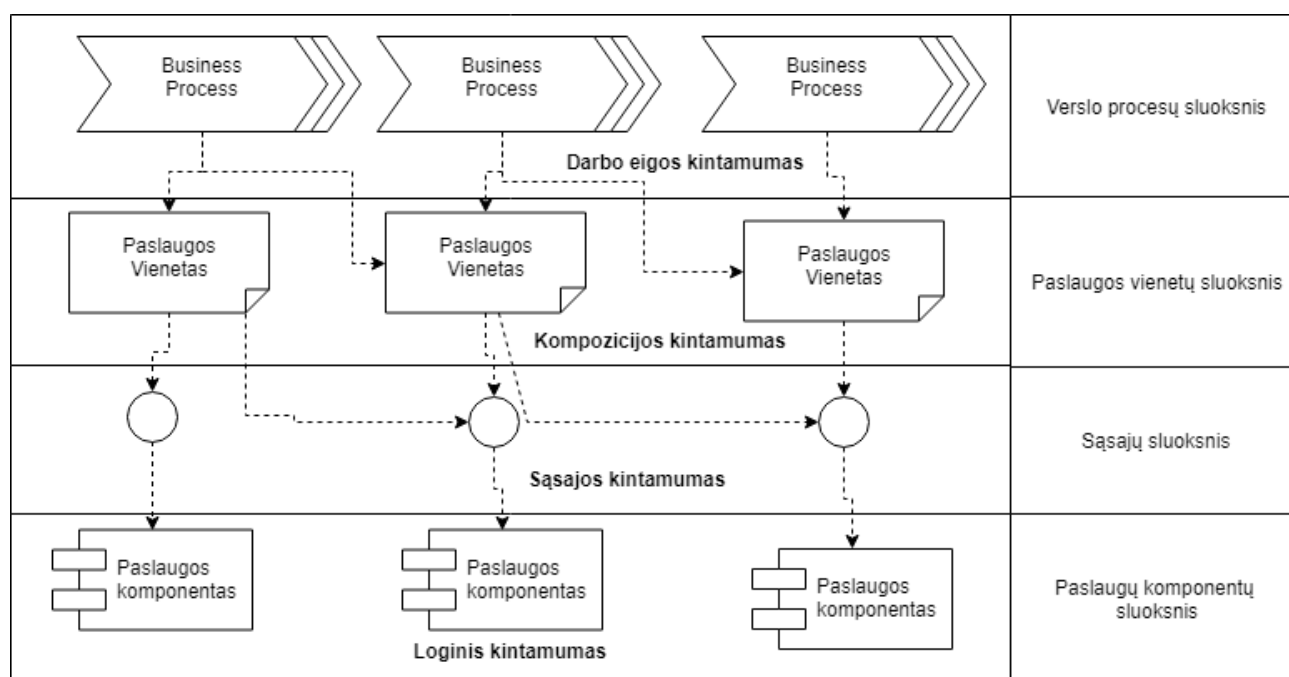
**Darbo eigos kintamumas.** Verslo procesas, vykdomas iš vartotojo pusės kaip veiklų seka, iš sistemos kūrėjo veikla realizuoja paslaugą, yra vadinamas darbo eiga (3 pav.). Darbo eigoje kai kurie servisų vienetai konkrečių naudotojų gali būti nenaudojami. Kai kurios sekos dalys gali būti vykdomos vienaip vienam naudotojui, kitam kitaip. Darbo eigos kintamumas yra apibrėžiamas kaip darbo eigos variacija konkrečiam verslo procesui. Kitaip tariant, veiklos/paslaugos yra vykdomos kitaip arba visiškai nevykdomos, priklausomai nuo individualaus naudotojo poreikių.

**Kompozicijos kintamumas.** Verslo procesas yra komponuojamas iš kelių veiklų/paslaugų, kad būtų patenkinti naudotojo reikalavimai. Vienai veiklai/paslaugai darbo eigoje gali būti priskirtas daugiau negu vienas interfeisas, realizuojantis paslaugą skirtinga realizacijos logika ir kokybės atributais (pvz. našumas ar saugumas). Kintamumas vyksta pasirenkant labiausiai tinkantį paslaugos interfeisą. Kitaip tariant, priklausomai nuo naudotojo reikalavimų, skirtingos paslaugų interfeisų realizacijos yra komponuojamos ir tai vadinama kompozicijos kintamumas.



**Sąsajos kintamumas.** Šis kintamumas vyksta veiklų/paslaugų sąsajose, kurios buvo publikuotos UDDI paslaugų registre (XML tipo registras verslo interneto paslaugoms kurtas). Prisimenant, paslaugos vienetas yra loginis vienetas, komponuojamas į skirtingus verslo procesus. Paskelbtos paslaugos servisų registre yra išpublikuotos ir prieinamos paslaugos, jų sąsajos tipiškai yra gaunamos iš realizuotų paslaugų komponentų ir sąsajų specifikacijų standartų.

**Loginis kintamumas.** Paslaugų komponentai apima operacijas, skirtas suteikti funkcionalumą veiklai/paslaugai. Verslo logika apibrėžiama kaip algoritmas ar loginė procedūra, naudojama operacijose. Paslaugos komponentas turi pateikti skirtingas realizacijos logikas, priklausomai nuo konkrečiai prašomų paslaugų.



3 pav. Servisų architektūros sluoksnių kintamumas

### 2.3. Paslaugų sistemų reikalavimų specifikavimas

Šiame skyriuje nagrinėjami pagrindiniai paslaugomis grindžiamos sistemos reikalavimų inžinerijos konceptai. Reikalavimai gali būti kategorizuojami skirtingais būdais skirtingose dimensijose.

Reikalavimai dažniausiai dalinami yra į dvi kategorijas: ankstyvieji ir vėlyvieji. Ankstyvųjų ir vėlyvųjų reikalavimų inžinerija buvo adaptuota į paslaugomis grindžiamas programų sistemas ir tai yra pristatyta šiose šaltiniuose [LS11, AG04, PPS+06, FSG+11, GMS05, RKS10, KBB+10].

Ankstyvieji reikalavimai yra formuluojami analizuojant būsimos sistemos organizacinį kontekstą, kuriame veiks kuriamoji sistema. Pradžioje su suinteresuotaisiais išaiškinami jų lūkesčiai, nustatoma, kodėl sistema yra kuriama [AG04]. Ankstyvųjų reikalavimų inžinerijos proceso rezultatas yra

ankstyvųjų reikalavimų modelis, kuriame suinteresuotieji modeliuojami kaip socialiniai aktoriai ir jų siekiamų intencijų, vykdomų užduočių ir skirtų išteklių priklausomybės [GMS05]. Pavyzdžiui, reikalavimų inžinerijos metodikoje *Tropos* suinteresuotųjų lūkesčių diagramoje modeliuojami suinteresuotųjų santykiai ir dekomponuojami suinteresuotųjų lūkesčiai bei tikslai [PPS+06]. Ankstyvųjų reikalavimų modelyje *Secure Tropos* šalia suinteresuotųjų lūkesčių ir tikslų aprašomi jų patikimumo bei saugumo santykiai [FSG+11]. Ankstyvųjų reikalavimų tikslas yra išvesti ir analizuoti funkcinės priklausomybes ir nefunkcinius reikalavimus. Bendru atveju, iš pradžių yra iš suinteresuotų šalių yra gaunami neformalūs reikalavimai, aukštesnio lygmens konceptai, poreikiai [BCD+09]. Ankstyvųjų reikalavimų kintamumą įtakoja besikeičiantys naudotojai ir jų poreikiai, taip pat kintantys prieinami resursai [GZW+17]. Suinteresuotųjų intencijų, fiksuojamų tikslų modeliuose, kitimas yra pagrindinis reikalavimų kintamumo šaltinis [BYY+18].

Vėlyvieji reikalavimai apibrėžia paslaugą arba sistemą jos operacinės aplinkos viduje, kartu su funkcionalumu ir kokybės atributais [GMS05]. Vėlyvųjų reikalavimų inžinerijos procesas susideda iš trijų fazių [FSG+11]:

1. Stambus verslo procesas ir darbų seka yra gaunama darbų sekos lygyje.
2. Iš ankstyvųjų reikalavimų modelio yra išvedamas saugus verslo procesas. Rezultatas yra verslo proceso hipergrafas.
3. Konstruojamas pagrindinis rezultatas: vykdomi saugūs verslo procesai, Secure BPEL forma, kartu su SLA kontraktais.

Paslaugų sistemos, kurios aprašomos dinaminiais reikalavimų specifikacijomis, yra atviros ir adaptuojamos. Todėl ankstyvųjų reikalavimų specifikacija turėtų būti nuolat atnaujinama, kad atitiktų visas variacijas. Paslaugomis grindžiamų programų sistemų reikalavimų inžinerija padengia statinius ir dinامينius reikalavimus. Tai reiškia skirtingo kintamumo laike reikalavimus [BBC+10], įtraukiant ir variacijas vykdymo metu. Taigi tokie klausimai, kaip „Ar plati turi būti pradinė specifikacija?“ arba „Kokių tipų reikalavimai turi būti specifikuojami ir koku lygiu jie galėtų kisti viso proceso gyvenimo cikle?“. Rečiau kinta paslaugų derinimo reikalavimai. Dažniau kinta tokie užsakovo reikalavimai, kaip kokybės atributai (paslaugos kokybė, vykdymo laikas) arba vykdomų operacijų lydintys koordinavimo, derinimo mechanizmų apribojimai (paslaugų kompozicijos) [BBC+10].

Iš to išplaukia terminas, pasiūlytas C. Rolland ir jos kolegų, – intencijomis grindžiama paslaugų specifikacija, kuris labiau orientuojasi į servisų tikslus negu į funkcinių vaizdą [RKS10]. Paslaugų reikalavimai verslo terminais yra transformuojami į paslaugų sistemos reikalavimus. Intencijomis grindžiamas paslaugos aprašas apima ketinamos paslaugos aprašą (paslaugos interfeisą, elgseną, paslaugų

kompoziciją susijusią su funkciniais aspektais, paslaugos kokybę), iš kurios vėliau išplaukia operacinis paslaugos aprašas (paslaugų sąveika, koordinavimas, funkcionalumas). Intencijomis grindžiama paslaugos aprašas leidžia nagrinėti kintamumą, pavyzdžiui alternatyvių paslaugų ar paslaugų kompozicijų specifikacijas, kas reikštų, kad tokių sistemų reikalavimų specifikacijos yra dinaminės.

Naujo tipo reikalavimai seka iš „savo“ (angl. *native*) paslaugomis grindžiamų sistemų funkcionalumo. Adaptuojamumas reiškia reakciją į aplinkos pokyčius ar naudotojo poreikius. Adaptacija paslaugose įgalina įvairias servisų kompozicijas ir servisų vykdymą. Adaptuojamieji reikalavimai apima servisų adaptavimo aplinkybes, prioritetus ir adaptacijos tikslus [BCD+09]. Verta paminėti, kad specifikuojant kintamumo reikalavimus, atsakoma į klausimą „Kokias kintamumo situacijas ir iki kokio lygio reiktų numatyti ir specifikuoti iš anksto?“. Stebėjimo reikalavimai yra susiję su situacijomis, kurios inicijuoja programinės įrangos adaptavimą, arba su situacijomis, kai reikia reaguoti į aptiktas problemas, klaidas. Stebėjimo reikalavimai kartais išvedami iš adaptuojamųjų reikalavimų [PKB14]. Funkcinių reikalavimų sąrašas nėra galutinis, nes paslaugomis grindžiamų sistemų infrastruktūra vis dar vystosi.

#### **2.4. Paslaugų sistemų procesų specifikavimas BP&SLA metodika**

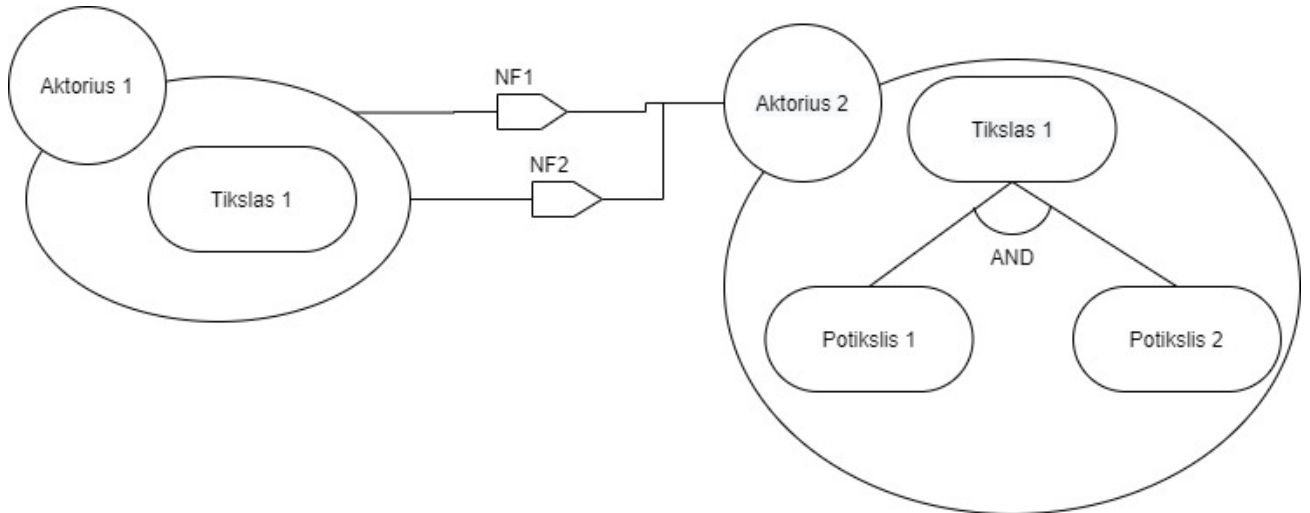
Autoriai Frankova ir kiti konceptualizuoja verslo procesų su fiksuotais kokybės matais išvedimą iš ankstyvųjų reikalavimų [FYS07, FSG+11]. Projektuojant paslaugų sistemų verslo procesus su silpna sankiba pasižyminčiais servais, svarbu užtikrinti ne tik tam tikrus darbų srautus, bet ir tinkamą paslaugos vykdymo kokybę, t. y. nefunkcines savybes, fiksuojamos 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 užsakovo ir serviso tiekėjo specifikuojant SLA, kuris apima aibę reikalavimų.

Metodika apima keturis etapus:

- 1) ankstyvųjų reikalavimų specifikavimą;
- 2) verslo procesų hipergrafo išvedimą;
- 3) verslo procesų hierarchijos išvedimą;
- 4) apribojimų nustatymas SLA išvedimui.

Pirmame etape naudotojas arba dalykinės srities ekspertas pateikia neformalius reikalavimus, o rezultatas yra ankstyvas reikalavimų modelis, t. y. koncepcinis verslo aktorių, jų verslo tikslų modelis

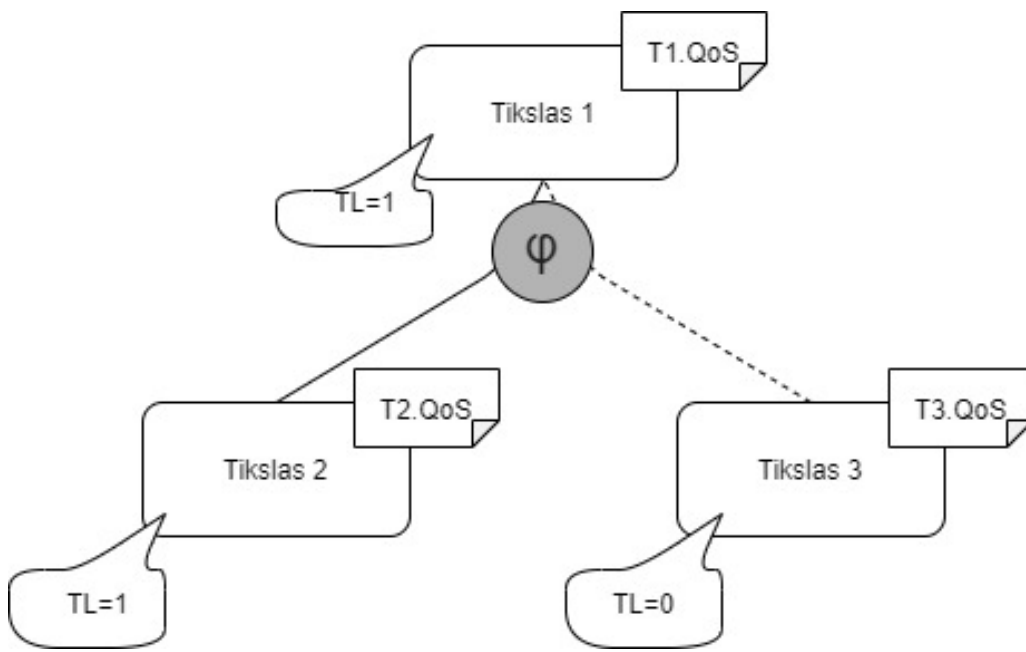
su kokybės matais, nusakančiais serviso nefunkcines savybes. Ankstyvųjų reikalavimų inžinerijos etape specifikuojamas būsimos paslaugos organizacinis kontekstas: identifikuojami aktoriai, jų tikslai ir potiksliai bei nefunkciniai savybės (4 pav.).



4 pav. Ankstyvųjų reikalavimų modelio notacija atspindinti būsimos paslaugos organizacinį kontekstą. Apskritimai žymi aktorius, ovalai – tikslus, lankai – dekompozicijos santykius AND/OR, penkiakampiai – reikalaujamas nefunkcines savybių perdavimas.

Antrame ir trečiame etape minėtas modelis yra transformuojamas į vykdomą verslo procesų aprašą. Transformacija nėra visiškai automatizuota, variacijos taškuose naudotojai gali būti įtraukiami į konkrečių sprendimų priėmimą.

Antras – verslo procesų hipergrafo išvedimo – etapas yra skirtas sukurti tarpinį modelį – verslo proceso hipergrafą, kuris apibrėžiamas kaip pora  $(B, H)$ , kur  $B$  yra verslo procesų rinkinys, o  $H$  yra hiperlankų rinkinys. Hiperlankas yra sutvarkyta pora  $(N, t)$  iš netuščios aibės  $N \subseteq B$  (įeities aibė) ir mazgo  $t \in N$  (tikslinis mazgas). Kiekvienas lankas yra asocijuojama su agregavimo funkcijų vektoriumi  $\varphi = [\varphi_1(N, t), \dots, \varphi_n(N, t)]$ , kurie išskaičiuoja tikslinį mazgą  $t$ , imant įeities aibę  $N$  kaip argumentą, įskaičiuojant QoS parametras. Jis yra kuriamas apeinant ankstyvųjų reikalavimų modelį ir įtraukiant, jei reikia, naudotoją. Hipergrafo kūrimas yra grindžiamas algoritmu, kuris ima ankstyvųjų reikalavimų modelį, aktorių su jo tikslais ir serviso kokybės (angl. *Quality of service*, toliau QoS) vektorių parametrus kaip įeigą. Kiekvienas verslo procesų hipergrafo mazgas yra verslo procesas, kuris atitinka aktoriaus tikslą iš ankstyvųjų reikalavimų modelio. Kiekvienas hiperlankas atitinka tikslo konkretizavimą arba tikslo vykdymo perdavimą (5 pav.).

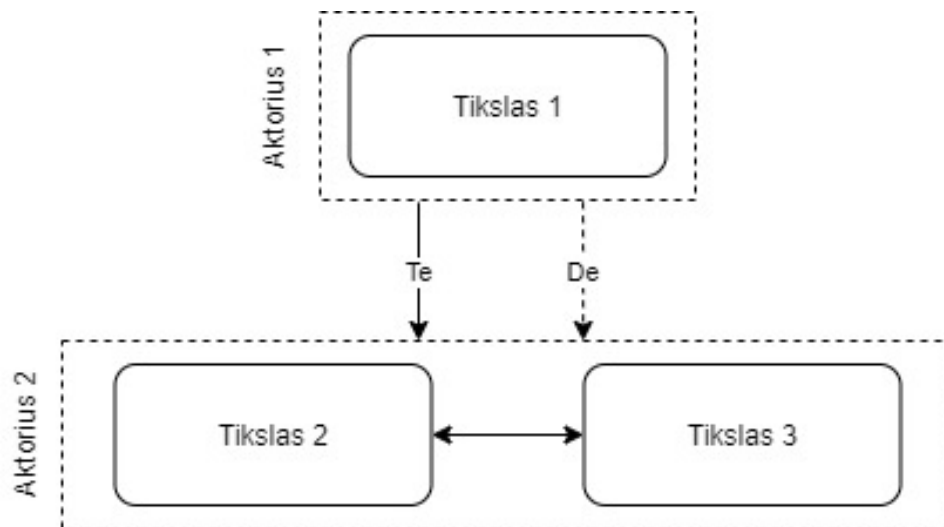


5 pav. Verslo procesų hipergrafas. Tikslai žymimi stačiakampiais su apvaliais kampais, tikslo konkretizavimo hiperlankas – vientisa linija, tikslo vykdymo perdavimo hiperlankas – punktyrinė linija, QoS parametras – komentaras, TL – patikimumo lygis

Trečiame – verslo procesų hierarchijos išvedimo – etape verslo procesų hierarchija yra kuriama su intencija gauti vykdomųjų verslo procesų rinkinį. Šie yra kuriami besilaikant *Secure BPEL* specifikacijos [FMS07], kur kiekvienas deleguotas verslo procesas turi atitinkama *SLA*, išvestą ketvirtoje fazėje. Tikslinant ankstyvuosius reikalavimus kuriamas tarpinis modelis, skirtas išvesti servisų kokybes savybes. Toliau kuriamas verslo procesų hipergrafas, iš kurio išvedama paruošta vykdymui verslo procesų hierarchija. Verslo procesų hierarchija yra išvedama tikslinant ankstyvąjį reikalavimų modelį, panašiai kaip hipergrafo kūrimo metu. Kuriant hierarchiją vykdomiesiems verslo procesams gauti reikia aiškiai nustatyti:

- 1) verslo procesus,
- 2) kuris aktorius vykdo kurį procesą,
- 3) vykdymo delegavimo ir pasitikėjimo priklausomybių tarp aktorių

Kiekvienas tikslas yra asocijuojamas su verslo procesu. Aktoriai vykdo priskirtus procesus ir perduoda vienas kitam delegavimo ir pasitikėjimo priklausomybes.



6 pav. Verslo procesų hierarchija. Tikslai/procesai yra žymimi stačiakampiais su apvaliais kampais, aktoriai – punktyriniais apskritimais, pasitikėjimo ir vykdymo delegavimo priklausomybės – atitinkamai vientisa ir punktyrine linija

Ketvirtam etape – apribojimų nustatymas *SLA* išvedimui – *SLA* yra nustatomi naudojant verslo procesų hipergrafą. Pagrindinė idėja yra pristatyti santykius tarp procesų kokybės ir veiklų kaip apribojimų rinkinį. Šiame etape verslo procesų hipergrafas yra toliau analizuojamas apibrėžiant apribojimus, kurie atvaizduoja įvairių procesų elementų nefunkcinių savybių santykius. Atsižvelgiant į apribojimus yra išvedami atitinkami *SLA*, kurių pasirašymas užtikrina konkrečias paslaugos kokybės savybes vykdant procesą. Metodikos galutinė išėiga yra įvykdomų verslo procesų hierarchija ir *SLA*, apibrėžiantys paslaugų kokybę.

## 2.5. Verslo procesų kintamumo modeliavimas

Rosa ir kiti išskiria šiuos specifikuojamų verslo procesų kintamumo mechanizmus [RAD+17]:

- 1) mazgų konfigūravimą (angl. *node configuration*),
- 2) elementų anotavimą (angl. *element annotation*),
- 3) veiklos specializavimą (angl. *activity specialization*),
- 4) fragmentų adaptavimą (angl. *fragment customization*).

### 2.5.1 Mazgų konfigūravimas

Šiame metode variacijos taškas yra adaptavimo proceso modelio mazgas, kuriam priskiriami skirtingi adaptavimo nustatymai. Veiklos (angl. *activity*), įvykiai (angl. *event*), sietuvai (angl. *gateway*), resursai ir objektai susiję su veiklomis, žymimi kaip konfigūruojami mazgai. Adaptavimas yra pasiekiamas pasirenkant vieną nustatymą vienam konfigūruojamam mazgui. Konfigūruojamas mazgas gali nesikeisti, kaip ir buvo originaliame modelyje, arba nustatymais apriboti mazgo elgseną.

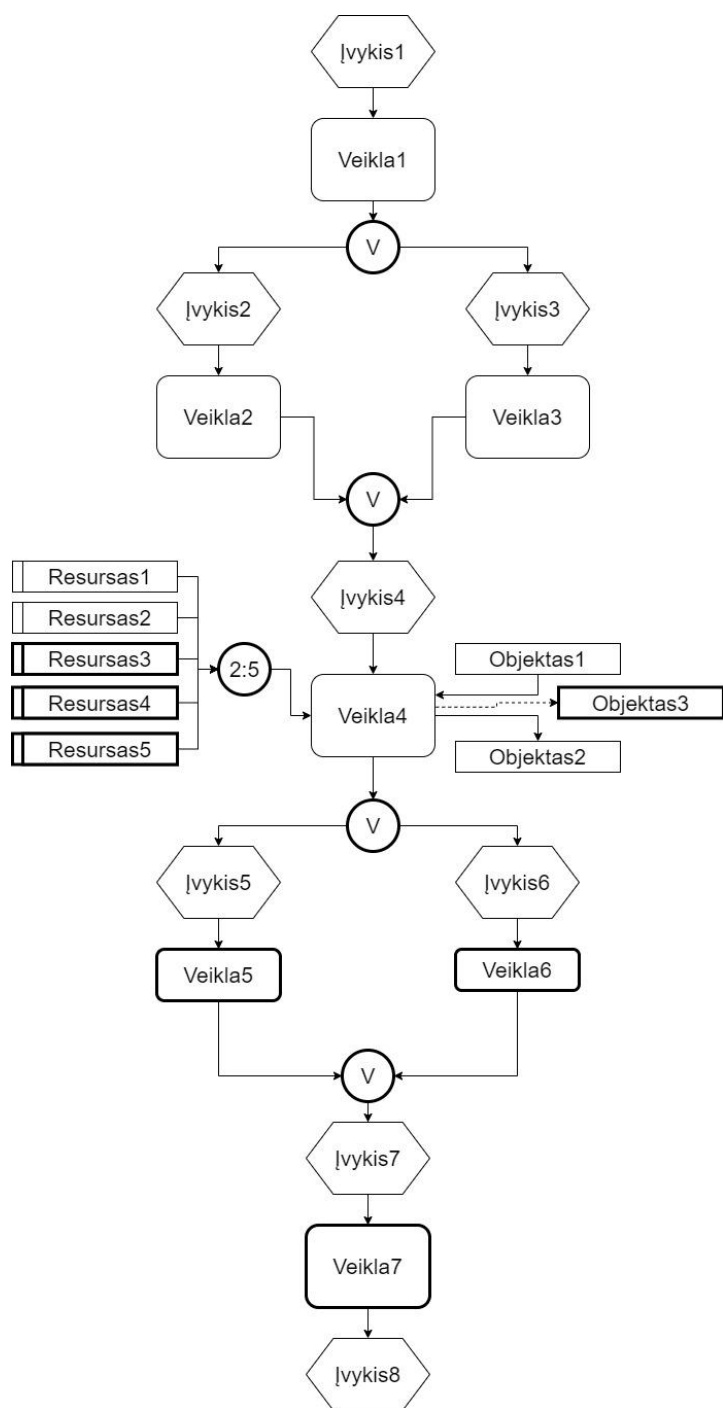
Konfigūruojamos veiklos, įvykiai, sietuvai, resursai ir objektai adaptuojami taikant jiems šias būsenas: *įjungta* (mazgas pasilieka adaptuotame modelyje) ar *išjungta* (mazgas nėra rodomas adaptuotame modelyje). Kitame skyriuje nagrinėjamas mazgo konfiguravimas naudojant įvykiais grindžiamos konfigūruojamos integruotos procesų grandinės (angl. *Configurable integrated Event-driven Process Chains*, toliau C-iEPCs) metodą.

### 2.5.2 Konfigūruojamos integruotų įvykiais grindžiamų procesų grandinės

Konfigūruojamų integruotų įvykiais grindžiamų procesų grandinių metodas (angl. *Configurable integrated Event-driven Process Chains*, toliau C-iEPCs) yra įvykiais grindžiamų procesų grandinės (EPC) kalbos plėtinys. iEPC išplečia EPC kalbą papildomai operuojant resursais ir prie veiklų priskirtais objektais. C-iEPC retai naudojama iEPC modifikacija, nes įveda konfigūruojamus objektus su skirtingais nustatymais. Konfigūruojamas objektas turi adaptavimo nustatymų rinkinį, kur kiekvienas nustatymas atitinka skirtingą adaptavimo procesą. Adaptavimas yra pasiekiamas apribojant C-iEPC elgseną taip, kad vienam adaptuojamam mazgui būtų priskiriamas vienas nustatymas. Tokiu būdu, C-iEPC yra transformuojamas į iEPC modelį, kai visi neaktualūs nustatymai yra pašalinami.

Šiame modelyje kintami artefaktai yra veiklos ir sietuvai. Grafiniuose modeliuose kintami artefaktai yra atvaizduojami kaip elementai su storesniu kraštu. Šiame modelyje įvykiai negali būti kintami (adaptuojami). Konfigūruojami sietuvai lieka tie patys arba labiau apriboti. Konfigūruojamas sietuvas gali būti disjunktyvus (OR operacija), kuris BPMN notacijoje atitinka įtraukiamą sietuvą (angl. *inclusive gateway*). Toks sietuvas paprastai neturi apribojimų. Labiau apribotas sietuvas atitinka griežtosios disjunkcijos operaciją (XOR). BPMN notacijoje jis atitinka išskirtinį sietuvą (angl. *exclusive gateway*), kur iš dviejų variantų pasirenkamas vienas. Kitas sietuvų apribojimas yra konjunkcijos operacija (AND), kuri BPMN notacijoje atitinka lygiagretų sietuvą (angl. *parallel gateway*). Papildomai, išeinančių srautų skaičius, jeigu sietuvas skaido vieną procesą į kelis srautus, arba įeinančių srautų skaičius, jeigu sietuvas jungia kelis srautus į vieną procesą, gali būti apribotas į bet kokį mažesnę skaičių, pavyzdžiui du srautai iš keturių. Srautams sumažėjus iki vieno, sietuvas pradingsta. Apribojant sietuvą iki vieno išeinančio srauto, kitas alternatyvus srautas yra trinamas. Neapribojus, modelis lieka nepakitęs. Apribojant sietuvą iš disjunktyvaus į lygiagretų, užtikrinama, kad abu srautai yra išsaugomi. Visais atvejais srautas pasirenkamas konfigūracijos nustatymuose projektavimo metu. Griežtosios disjunkcijos sietuvui sprendimas daromas vykdymo metu, kai procesas yra jau apibrėžtas.

Konfigūruojamos veiklos gali būti įjungtos ir išjungtos. Išjungimo atveju veikla yra slepiama adaptuotame modelyje. Taip pat, veikla gali būti nustatyta, kaip neprivaloma (angl. *optional*). Toks būdas leidžia turėti pasirinkimą, ar laikyti veiklą adaptuotame modelyje iki sistemos vykdymo.



7 pav. C-iEPC modelis. Veikla pažymėta stačiakampiu su apvaliais kampais, resursai – stačiakampis su stačiakampiu viduje, objektai – paprastas stačiakampis, sietuvai – apskritimai, režiniai sietuvai – apskritimai su skaičiais, kintami elementai – su storesniu kraštu.

Resursai (C-iEPC notacijoje – tai rolės) ir objektai yra konfigūruojami. C-iEPC atveju, kai objektas yra naudojamas kaip įeiga veikloje, jis pažymimas kaip suvartotas, kas reikštų, kad jis bus atlaisvintas, kai veikla yra baigiama. Tiek resursai, tiek objektai nustatomi kaip privalomi arba neprivalomi. Jie sujungiami su veiklomis naudojant loginius sietuvus, vadinamais režiniais sietuvais. Rėžiniai



sietuvai turi tris loginius disjunkcijos (OR), griežtos disjunkcijos (XOR), konjunkcijos (AND) tipus ir įgalina bet kokią kombinaciją susijusių resursų, pavyzdžiui, iš penkių galimų resursų minimaliai bent du. Rėžiniai sietuvai yra neprivalomi, kas reikštų, jog visi prijungti resursai irgi yra neprivalomi. Toks mechanizmas priskirti resursus ir objektus veikloms veda prie skirtingų adaptavimo galimybių. Jei resursas, objektas ar rėžinis sietuvas nustatyti kaip neprivalomi, tada juos galima adaptuoti kaip privalomus ir tada jie išsaugomi adaptuotame modelyje arba išjungiami. Jei jie yra privalomi, tai objektą, resursą ar sietuvą galima tik išjungti. Konfigūruojamiems veiklų įeigos objektams galima nustatyti būseną *naudojamas*, kas reikštų, kad jie nebus naikinami po to, kai veikla nustos juos naudoti.

### 2.5.3 Elementų anotavimas

Šie metodai jungia elementus su grafinėmis anotacijomis atspindinčiomis dalykinės srities ypatybes. Anotuojami gali būti valdymo srautų mazgai, tokie kaip veiklos, įvykiai ir sietuvai. Taip pat, anotuoti galima ir sekos srautus, resursus ir objektus. Modelio elementai su anotacijomis automatiškai tampa variaciniais taškais. Skirtingi anotavimo būdai atitinka skirtingus modelių elementus. Dalykinės srities savybės yra priskiriamos prie modelių elementų naudojant *dalykinės srities sąlygas* (angl. *domain conditions*), kurios yra dalykinės srities savybių loginės išraiškos, pavyzdžiui „mažas biudžetas = tiesa (angl. *true*)“.

Adaptavimas yra vykdomas pasirenkant dalykinės srities savybes. Tie pasirinkimai gali būti daromi tiesiogiai ar remiantis dalykinės srities modeliu. Vadovaujantis šiais pasirinkimais, dalykinės srities sąlygos yra vertinamos. Esant neigiamam atsakymui elementai yra trinami iš modelio.

### 2.5.4 Konfigūruojamų procesų modeliavimas

Modeliuojant konfigūruojamus procesus, adaptavimas yra pasiekiamas šalinant tuos modelio elementus, kurie nėra svarbūs einamajam verslo scenarijui. Dalykinės srities savybių rinkinys, vadinamas verslo charakteristikomis (angl. *business characteristics*), yra naudojamas galimų scenarijų nustatymui ir toliau vykdyti adaptavimą. Minėtos charakteristikos yra susietos su proceso modelio elementais naudojant dalykinės srities sąlygas, kurios yra charakteristikų loginės išraiškos. Ryšys grafiškai yra pateikiamas nurodant dalykinės srities sąlygas modelio elementams pilkuose stačiakampiuose.

Proceso modeliai yra gaunami naudojant išplėstą įvykio procesų grandinės notaciją (eEPC), EPC kalbos plėtinį, kuris įtraukia resursus ir objektus, panašiai į iEPC. Verslo charakteristikos priskiriamos prie veiklos, įvykio, resurso ir objekto elementų (8 pav.). Sietuvai tiesiogiai negali būti konfigūruojami. Atvirkščiai, modeliuotojas turi įtraukti galimus sietuvų variantus į adaptuojamą modelį.

Proceso modelio adaptavimas specifiniam scenarijui yra daromas žymint elementus kaip paslėptus, jei jiems dalykinės srities sąlyga yra klaidinga. Tada vykdoma transformacija šalinant paslėptus elementus. Taip pat trinami ir tie sietuvai, kurie tapo nebesvarbūs, kas reikalauja sujungti likusius nesujungtus mazgus. Transformuojant proceso modelį, kad jis būtų adaptuotas konkrečiam atvejui, reikia ištaisyti kai kurias problemas. Reikia pašalinti sietuvus, kurie turi tik vieną įeigos srautą ir vieną išeigos srautą, nors šis būdas negarantuoja struktūrinio ir elgesio korektiškumo galutiniuose modeliuose. Pavyzdžiui, jeigu galima trinti tiek įvykius, tiek veiklas, tada algoritmas neveiks ciklo atveju arba kai veikla tarp dviejų įvykių yra trinama. Struktūrinės problemos, kurių negalima ištaisyti, yra pristatomos modeliuotojui, kuris turi ištaisyti jas rankiniu būdu.

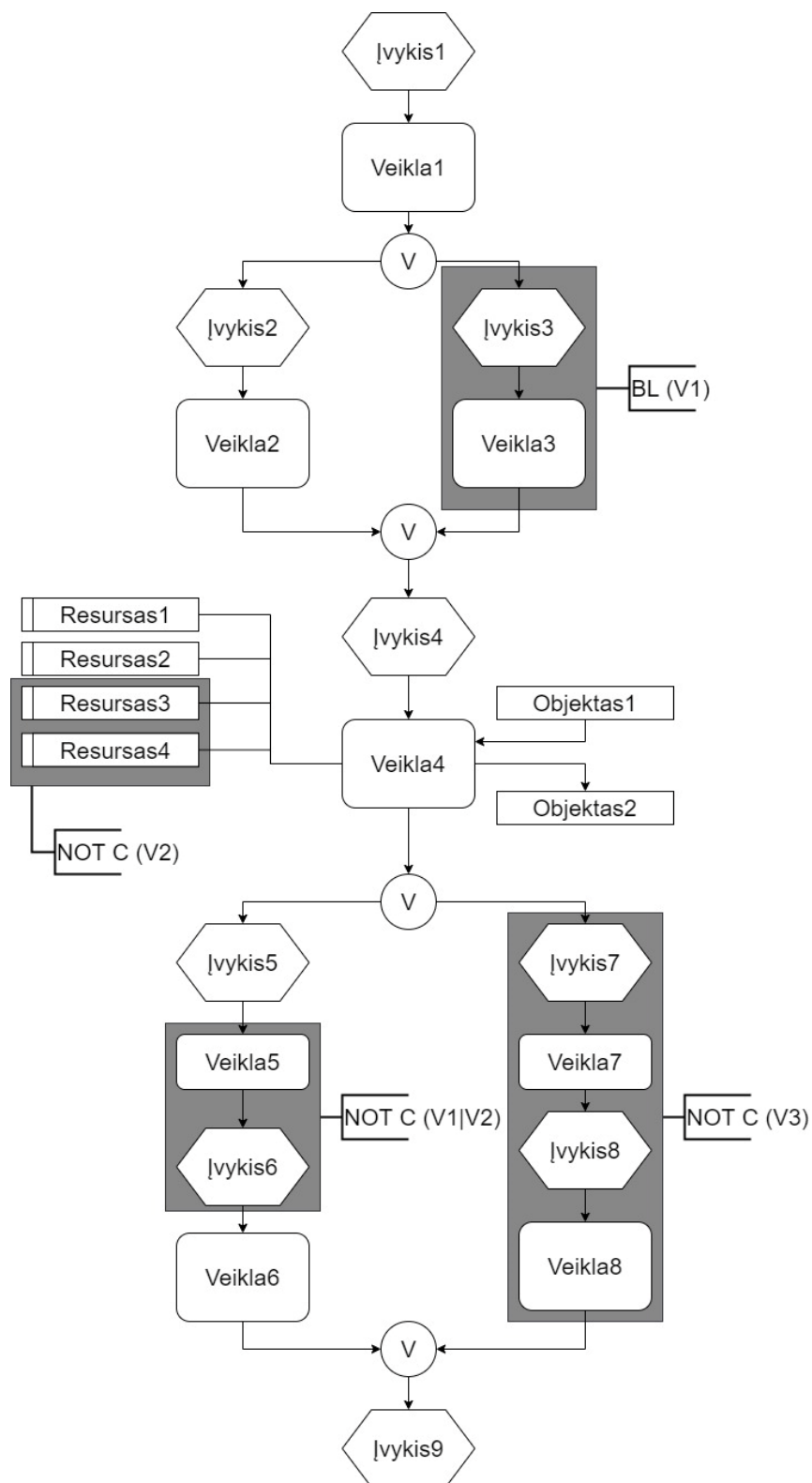
Verslo charakteristikos yra adaptuojamos meta-modelio lygmenyje pašalinant elementų tipus, nesvarbius specifiniam scenarijui. Modelyje galima paslėpti visus resursus vienu metu. Modelio adaptavimas atliekamas ne proceso lygmenyje, o įvertinant verslo charakteristikų rinkinį. Konfigūruojamų procesų modeliavimas nesiūlo rekomendacijų vartotojams, kaip priskirti charakteristikoms reikšmes.

#### 2.5.5 Veiklos specializacija

Šios grupės metodai adaptuojant procesus remiasi veiklos specializacija. Abstrakti veikla apibūdinama kaip variacinis taškas ir jam priskiriamas vienas arba daugiau variantų. Variantas yra abstrakčios veiklos specializacija, viena iš galimų abstrakčios veiklos realizacijų. Specialus variacijos taško tipas yra *neprivalomas* (angl. *optional*). Tai yra abstrakti veikla, kuri gali būti specializuota į tuščią veiklą.

Variantai priskiriami veiklų atributams, pavyzdžiui objektams arba resursams, kurie tampa variaciniais taškais. Įvykiai ir sietuvai negali būti adaptuojami. Adaptavimas yra grafiškai konkretizuojamas žymint veiklas, jų atributus kaip variacinius taškus ir sujungiant variantus su variaciniais taškais naudojant specializacijos lankus.

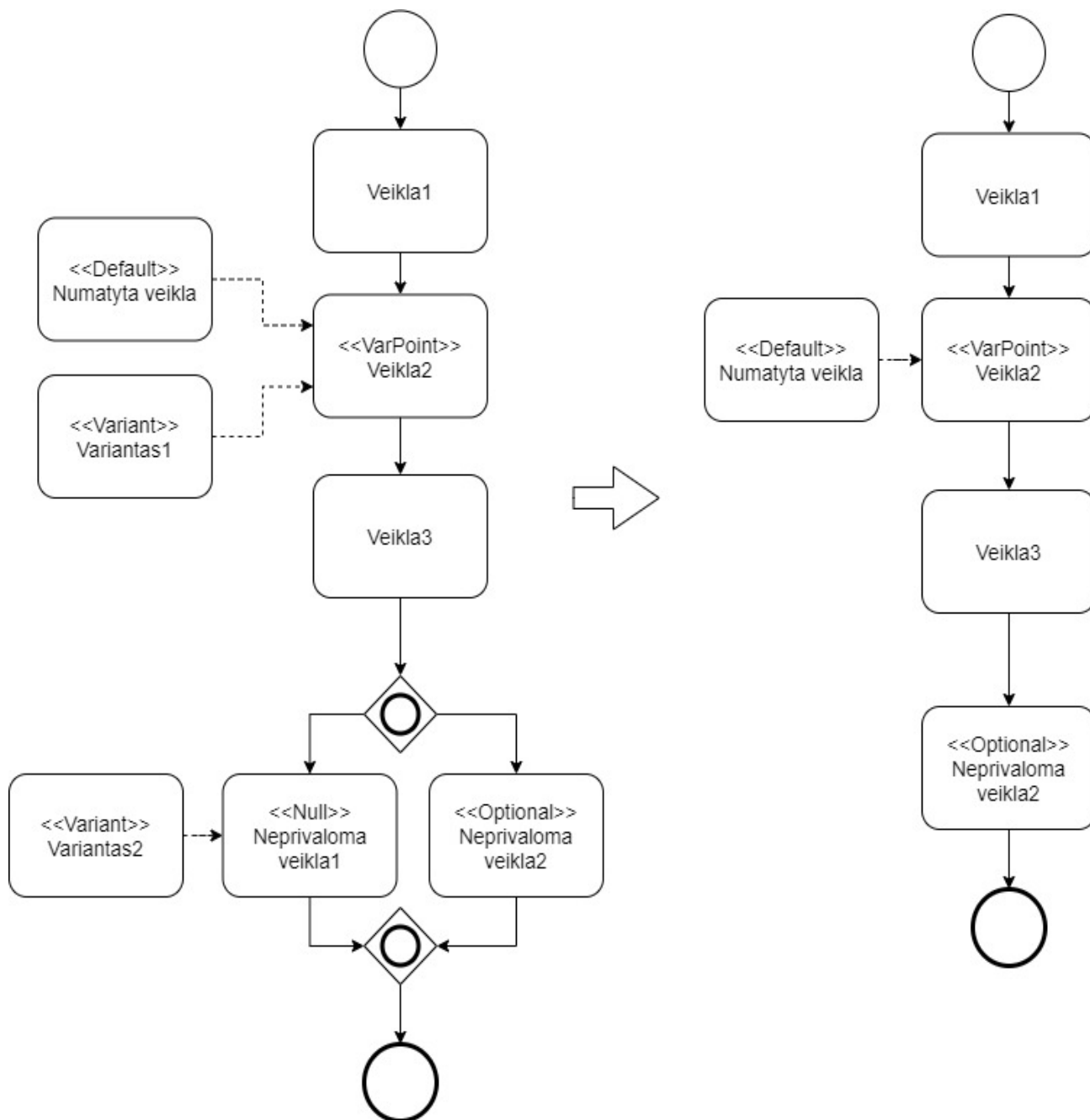
Adaptavimas pasiekiamas pasirenkant vieną ar daugiau variantų variaciniam taškui, o neprivalomos veiklos yra išjungiamos. Adaptavimas daromas tiek ant proceso modelio tiek ant dalykinės srities modelio. Maršruto pasirinkimas iš daugiau nei vieno varianto variaciniam taškui ir transformavimas, išvalant išjungtas neprivalomas veiklas, priklauso nuo konkrečios situacijos.



8 pav. Konfigūruojamų procesų modeliavimas. C – verslo charakteristika. V1, V2, V3 – charakteristikos reikšmės, NOT – loginis neigimas, | - loginė disjunkcija. Šis metodas yra eEPC metodikos plėtinys. Visi naudojami objektai yra tokie patys, kaip ir 7 pav.

### 2.5.6 PESOA metodas

Pagal PESOA (*Process Family Engineering in Service-Oriented Applications*), adaptuojamas proceso modelis yra koncepcinis proceso modelis, kur veiklos žymimos su stereotipais, nusakančiais kintamumą. Nors stereotipai yra UML notacijos išplėtimas, šiame metode jie yra pritaikomi tiek UML veiklos diagramose, tiek BPMN modeliuose.



9 pav. PESOA metodo pavyzdinė diagrama. Apskritimas žymi proceso pradžią, paryškintas apskritimas – proceso pabaigą, stačiakampis su apvaliais kampais – veiklą, rombai su apskritimu viduje – sietuvai.

Kintama proceso modelio veikla žymima kaip variacijos taškas su stereotipu <<*VarPoint*>>. Variacinis taškas pavaizduoja abstrakčią veiklą, pavyzdžiui, „Paruošti laikmeną redagavimui“, kuris turi būti specializuojamas konkrečiu variantu su stereotipu <<*Variant*>> iš pateiktų galimų aibės. Pavyzdžiui, „Veikla1“ gali būti specializuota į „Numatyta veikla“ ar „Variantas1“. Galima taip pat pažymėti numatytą variantą variaciniam taškui, naudojant stereotipą <<*Default*>>. Pavyzdžiui, „Numatyta veikla“ gali būti numatytasis „Veikla1“ variantas. Jeigu variantai yra uždari arba nesuderinami, kitaip tariant, jeigu tik vienas variantas yra priskirtas variaciniam taškui, tada naudojamas stereotipas <<*Abstract*>>, o ne <<*VarPoint*>>. Pažymėtas stereotipu <<*Null*>> variacijos taškas nusako neprivalomą veiklą, kuriai priskiriamas tik vienas variantas, o jo realizacija yra neprivaloma. Tokiu atveju veikla yra išimama iš modelio. Alternatyva <<*Null*>> stereotipo variacijos taškui ir jo galimų variantų yra jo atvaizdavimas tiesiai variacijos taške, naudojant stereotipą <<*Optional*>>.

Stereotipai priskiriami veikloms ir veiklų atributams. Sietuvai, įvykiai ir resursai šiame modelyje nėra adaptuojami. Adaptavimo metu kiekvienas variacijos taškas yra specializuojamas į vieną ar daugiau variantus. Nereikalingi variantai yra šalinami iš modelio. Išplėtimo mechanizmai nėra pritaikomi. PESOA metodika nepateikia transformacijos algoritmo išvesti adaptuotus modelius. Variacijos taškų, tokių kaip <<*Null*>> arba <<*Optional*>>, šalinimas ir adaptavimas, specializuojant į daugiau negu vieną variantą, gali sukelti modelio korektiškumo klaidas, taisomas rankiniu būdu. Visi metodo aspektai yra parodyti pavyzdinėje diagramoje (9 pav.).

### 2.5.7 Fragmentų adaptavimas

Šios grupės metodai adaptuoja *keitimo operacijas* (angl. *change operations*) ribojant arba išplečiant proceso modelį. Proceso modelis adaptuojamas naudojant dvi atominės keitimo operacijas. Viena yra *trinti* (angl. *delete*) fragmentą iš modelio, kitas yra *įdėti* (angl. *insert*) fragmentą į modelį. Sudėtingesnės operacijos kaip *perkelti* (angl. *move*) arba *pakeisti* (angl. *replace*) realizuojamos naudojant *trinti* ir *įdėti* operacijų kombinacijas. Trinamas arba įdedamas fragmentas turi turėti vieną įėjimo ir išėjimo tašką. Atitinkamai, sudėtingesnė operacija reikalauja dviejų proceso modelio srautų, kad būtų atskirtos trinamo arba įdedamo bazinio modelio dalys, kurios gali ir sutapti. Šie variaciniai, modelyje vadinami *reguliavimo* (angl. *adjustment*), taškai gali būti žymimi modelyje išreikštiniu būdu. Nepažymėjus, pagal nutylėjimą, kiekvienas srautas yra laikomas reguliavimo tašku. Po keitimo operacijų vykdoma atitinkama modelio transformacija. Trečioji keitimo operacija yra *modifikuoti* (angl. *modify*) keičia su veikla susijusius resursus arba objektus, pavyz-

džiui, pakeičia vieną resursą kitu arba priskiria veiklai naują resursą. Operacijos sudedamos į *operacijų sekas*, susiejamos su *dalykinės srities sąlygomis*, pagal kurias sprendžiama, ar esama seka turi būti adaptuota.

### 2.5.8 Provop metodas

*Procesų Variantai pagal Galimybes* (angl. *Process Variants by Options*, toliau *Provop*) metode adaptavimas vykdomas taikant keitimo operacijas baziniame modelyje su sužymėtais reguliavimo taškais (10 pav.). Bazinis modelis gali būti standartinis konkrečios dalykinės srities procesas, bendrinis (angl. *generic*) procesas (visų variantų viršaišbis) arba šių dviejų modelių sankirta.

Provop metode įvestos dvi atomines operacijos *trinti* ir *įdėti*, sudėtingesnė operacija *modifikuoti* (objektų ir resursų adaptavimui) ir papildoma ketvirta operacija *perkelti* (angl. *move*), skirta perkelti atskirtą dviem reguliavimo taškais bazinio modelio fragmentą į kitą modelio vietą, atskirtą kitais dviem reguliavimo taškais. Reguliavimo taškai apibrėžiami tik valdymo srautams, todėl resursams ir objektams kintamumas negali būti pavaizduotas. Provop metode, operacijų seka yra vadinama *variantu* (angl. *option*). Kelių variantų atveju vengiant adaptavimo klaidų, kaip pavyzdžiui abipusis paneigimas, yra naudojami *variantų apribojimai* (angl. *option constraints*).

Proceso modelio adaptavimas grindžiamas penkių žingsnių metodu, kuriame remiamasi dalykinės srities ypatybėmis:

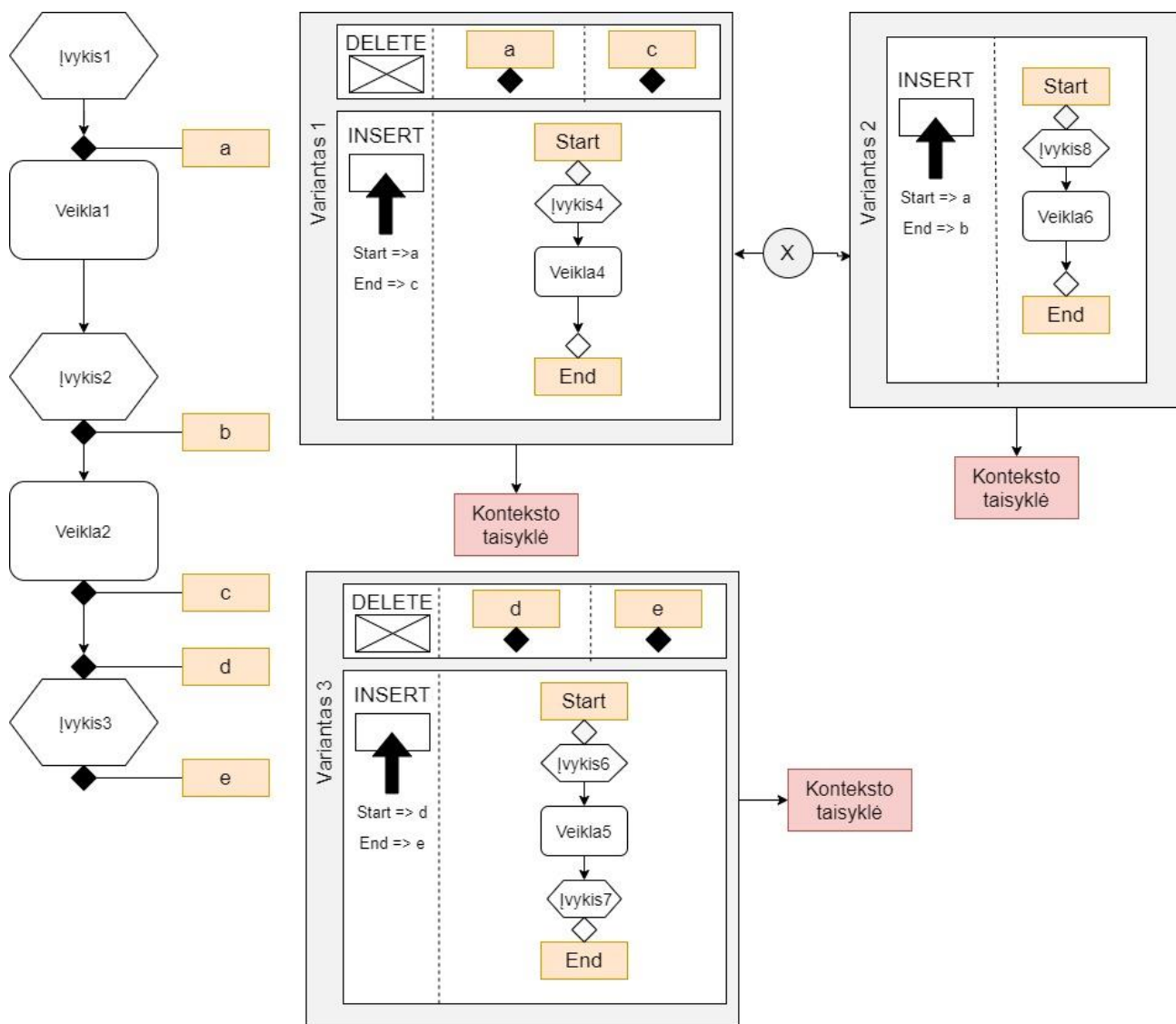
1. Naudotojas nustato visus įmanomus dalykinės srities kontekstus (angl. *contexts*). Kontekstas, Provop metode, yra dalykinės srities ypatybė, kuri yra pristatoma kaip kintamasis, pavydžiui, biudžetas, su visomis galimomis reikšmėmis (didelis, vidutiniškas, mažas). Tuo pačiu metu apibrėžiami dalykinės srities apribojimai loginių išraiškų forma, apribojantys sąveiką tarp kontekstų, pavyzdžiui,

$$\text{biudžetas} = \text{mažas} \Rightarrow \text{užbaigti} = \text{juosta}.$$

Kiekvienam variantui priskiriama dalykinės srities sąlyga (angl. *domain condition*) loginės išraiškos forma, apribojanti jos pritaikymą specifiniam verslo scenarijui.

2. Kiekvienam kontekstui yra automatiškai nustatoma aibė jį atitinkančių variantų.
3. Atrinktų variantų nuoseklumas yra tikrinamas pagal variantų apribojimus. Aptikus nuoseklumą naudotojui išvedamas pranešimas, kad, pavyzdžiui, varianto apribojimas prieštarauja konteksto apribojimui.
4. Galiojantys variantai pritaikomi baziniam modeliui atsižvelgiant į kiekvieną kontekstą.
5. Tikrinamas gauto rezultato korektiškumas.

Nekorektiški modeliai yra atmetami. Bazinio modelio adaptavimo kontekstai ir konteksto taisyklės yra pateiktos abstrakčiai, modelis nepateikia rekomendacijų, kaip rinktis variantus.



10 pav. ProVop metodo adaptuojamo proceso pavyzdys. Šešiakampis atvaizduoja įvykį, stačiakampis su apvaliais kampais – veiklą, gelsva etiketė su juodu rombu – reguliavimo taškas, gelsva etiketė su baltu rombu apačioje – fragmento pradžia, gelsva etiketė su baltu rombu viršuje – fragmento pabaiga, apskritimas su X viduje – variantų apribojimai

## 2.6. Adaptuojamų užduočių valdymas

Verslo procesų valdymas (angl. *Business Process Management*, toliau BPM) yra metodika pateikianti gerai patikrintus ir išgalėjusius tinkamai struktūrizuotų ir rutininių procesų analizės, optimizacijos, automatizacijos ir valdymo būdus. Adaptuojamų užduočių valdymas (angl. *Adaptive Case Management*, toliau ACM) kaip metodika siekia praplėsti BPM galimybes į silpnai struktūrizuotus žinių reikalaujančius verslo procesus [KSF+15]. Atvirkščiai negu klasikinis darbų sekų planavimas

(angl. *classical workflow management*, toliau WFM) arba įmonės resursų planavimo (angl. *Enterprise resource planning*, toliau ERP) sistemos, skirtos verslo procesų automatizacijai, ACM skirtas specifikuoti procesų kintamumą, adaptuojamumą vykdymo metu. ACM nėra laikomas BPM įpėdiniu. ACM, specifikuojantis silpnai struktūrizuotus žinių reikalaujančius verslo procesus, papildo klasikinį BPM padengdamas naują verslo procesų klasę.

### 2.6.1 CMMN notacija

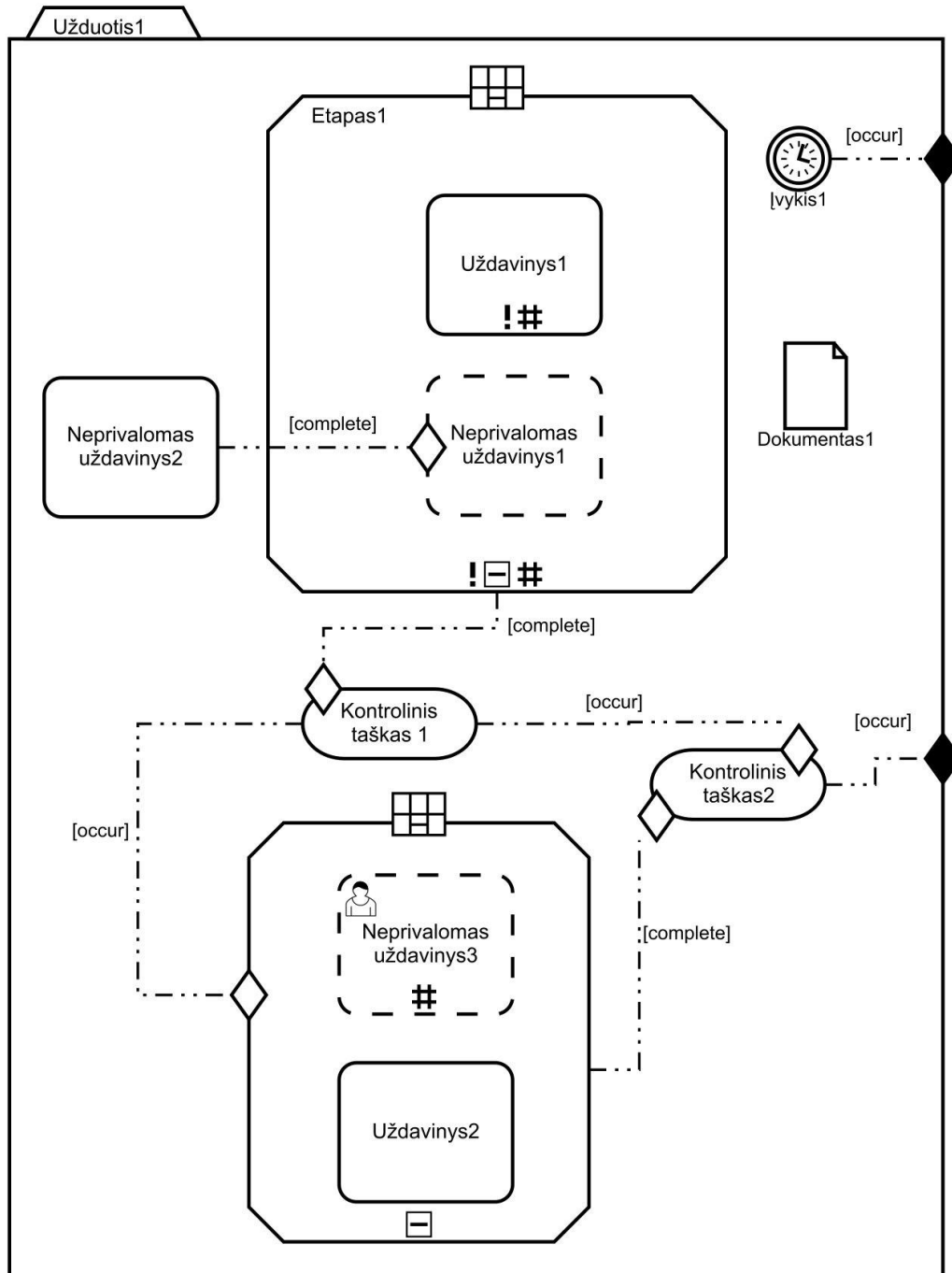
Object Management Group (toliau OMG) išleido užduočių (angl. *case*) valdymo standartą Atvejų valdymo modelį ir notaciją (angl. *Case Management Model and Notation*, toliau CMMN) [OMG20, OMG14]. CMMN yra ACM paradigmos vizuali realizacija, kurioje siekiama standartizuoti užduočių valdymą panašiai, kaip BPMN standartizuoja verslo procesų modeliavimą [HW11]. CMMN pateikia metamodelį, notaciją ir metamodelio semantiką. Pagrindinė šio metodo esybė yra užduotis, kuri yra visų CMMN modelio elementų konteineris. Proceso kūrimas pagal CMMN apima du etapus. Projektavimo etape yra modeliuojama užduotis, vykdymo metu užduoties vykdytojai siekia užduoties išskeltų tikslų.

### 2.6.2 CMMN taikymo pavyzdys

Pateiktame CMMN modelio taikymo pavyzdyje pristatomi pagrindiniai elementai su kintamumo ir adaptuojamumo mechanizmu [KSF+15] (11 pav.):

- Uždavins (angl. *task*) – veikla atliekama vykdymo metu (Uždavins1)
- Įvykis (angl. *event*) – gali inicijuotis patenkinus konkrečią sąlygą (Įvykis1)
- Etapas (angl. *stage*) – užduoties elementų konteineriai (Etapas1). Etapai turi beveik tokį patį gyvavimo ciklą ir vykdymo semantiką kaip uždaviniai
- Kontrolinis taškas (angl. *milestone*) – atitinką etapo tikslą, numato užduoties progreso tikrinimą (Kontrolinis taškas1)
- Sargai (angl. *sentry*) – numato loginę priklausomybę tarp etapų ir/arba uždavinių. Jei bet kuris iš minėtų elementų turi sargą pridėtą, jis gali būti pradėtas jei sargo sąlyga išpildyta (tušti rombai diagramoje).
- Dokumentai – etapų arba uždavinių išeiğa.
- Rolės – leidžia priskirti uždaviniui atliekančio aktorius klasę.
- Dekoratoriai
  - Šauktukas (!) nusako taisyklę, ar etapas/uždavins turi būti įvykdytas prieš įvykdant tėvinį etapą ar užduotį
  - Grotelės (#) nusako, kad etapas ar uždavins gali būti kartojamas.





11 pav. CMMN modelio pavyzdinė diagrama

Planiniai užduoties elementai yra automatiškai pridedami prie užduoties, kai užduotis yra iniciuojama. Kintamumas šiame metode įvedamas, kai prie užduoties papildomai pridėjami neprivalomi elementai, palikti atliekančio užduotį aktoriaus nuožiūrai (angl. *discretionary*). Tie elementai buvo iš anksto paruošti projektavimo metu, kai užduoties kūrėjai specifikuoja sąlygas, pagal kurias neprivalomi elementai, naudojant planavimo lenteles (angl. *planning tables*), yra pridedami prie užduoties vykdymo metu. Pavyzdžiui, planavimo lentelė specifikuoja, kad konkretus uždavinys bus pridėtas

prie užduoties, jei tam tikras įvykis bus inicijuotas vykdymo metu. Neprivalomi elementai yra vizualizuojami punktyru. Sargai yra pridedami kitos užduoties elemento krašte ir nusako priklausomybę (angl. *dependency*), jie reiškia užduoties pradžios kriterijų. Punktyrinė linija tarp sargo ir ankstesnio uždavinio nusako neprivalomą užuominą užduotį vykdančiam aktoriui. Sargas nusako, kad uždavinys yra atliekamas tik su išpildyta logine sąlyga (pradžios kriterijai). Tai yra modeliuojama kaip įvykis, kuris yra inicijuojamas esant patenkintai sąlygai. Sargai taip pat nusako užduoties baigimo kriterijus. Pavyzdyje yra kontrolinis taškas *Kontrolinis taškas 2*, kuris logiškai nusako užduoties pabaigą. Tai yra vizualizuojama juodu rombu. Taigi, pasiektas *Kontrolinis taškas 2* pavyzdyje yra interpretuojamas kaip užbaigta užduotis.

2 lentelė. Verslo procesų kintamumo modeliavimo metodų apibendrinimas

<b>Metodas</b>	<b>Modeliavimas</b>	<b>Kintamas elementas</b>	<b>Kintamumo specifikavimas/technika</b>
C-iEPCs	Įvykiais grindžiama proceso grandinė	Mazgas: veikla, sietuvas	Ribojimas: nereikalingos veiklos šalinamos, sietuvai ribojami
Konfigūruojamų procesų modeliavimas	Išplėstas įvykiais grindžiamo procesų grandinės notacijos (eEPC) modelis	Elementai: veikla, resursas, įvykis, objektas	Elementų anotacija: prie elemento pridedama dalykinės srities sąlyga. Jeigu sąlyga klaidinga, elementas šalinamas iš modelio
PESOA	UML veiklų diagrama, BPMN diagrama	Veikla	Stereotipai priskiriami veikloms ir veiklų atributams. Adaptavimo metu kiekvienas variacijos taškas specializuojamas į vieną ar daugiau variantų. Nereikalingi variantai šalinami iš modelio.
Provop	BPMN diagrama	Fragmentai: įvykis, veikla arba jų rinkiniai	Fragmentus, arba konkrečiai metode vadinamus reguliavimo taškus, galima trinti (apriboti), įdėti (išplėsti bazinį modelį), perkelti arba pakeisti (kas yra pirmų dviejų atominių operacijų junginys) ir modifikuoti (keisti su veikla susijusius resursus arba objektus)

CMMN	CMMN diagrama	Uždavinys, etapas	Minėti kintami elementai yra visada iš anksto paruošti. Naudojant logines sąlygas ir planavimo lenteles sprendžiama, kuris variantas pridodamas prie proceso vykdymo metu.
------	---------------	-------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.7. DiVA paslaugų kintamybių modeliavimas

DiVA metodas apibrėžia dinaminio kintamumo adaptuojamose paslaugomis grindžiamose sistemose valdymo bei kompleksiško kontrolės būdą [GCA+11]. DiVA metode adaptuojama sistema suprantama kaip dinaminė produktų linija, kurioje akcentuojamas sistemos kintamumas, o ne visos galimos konfigūracijos.

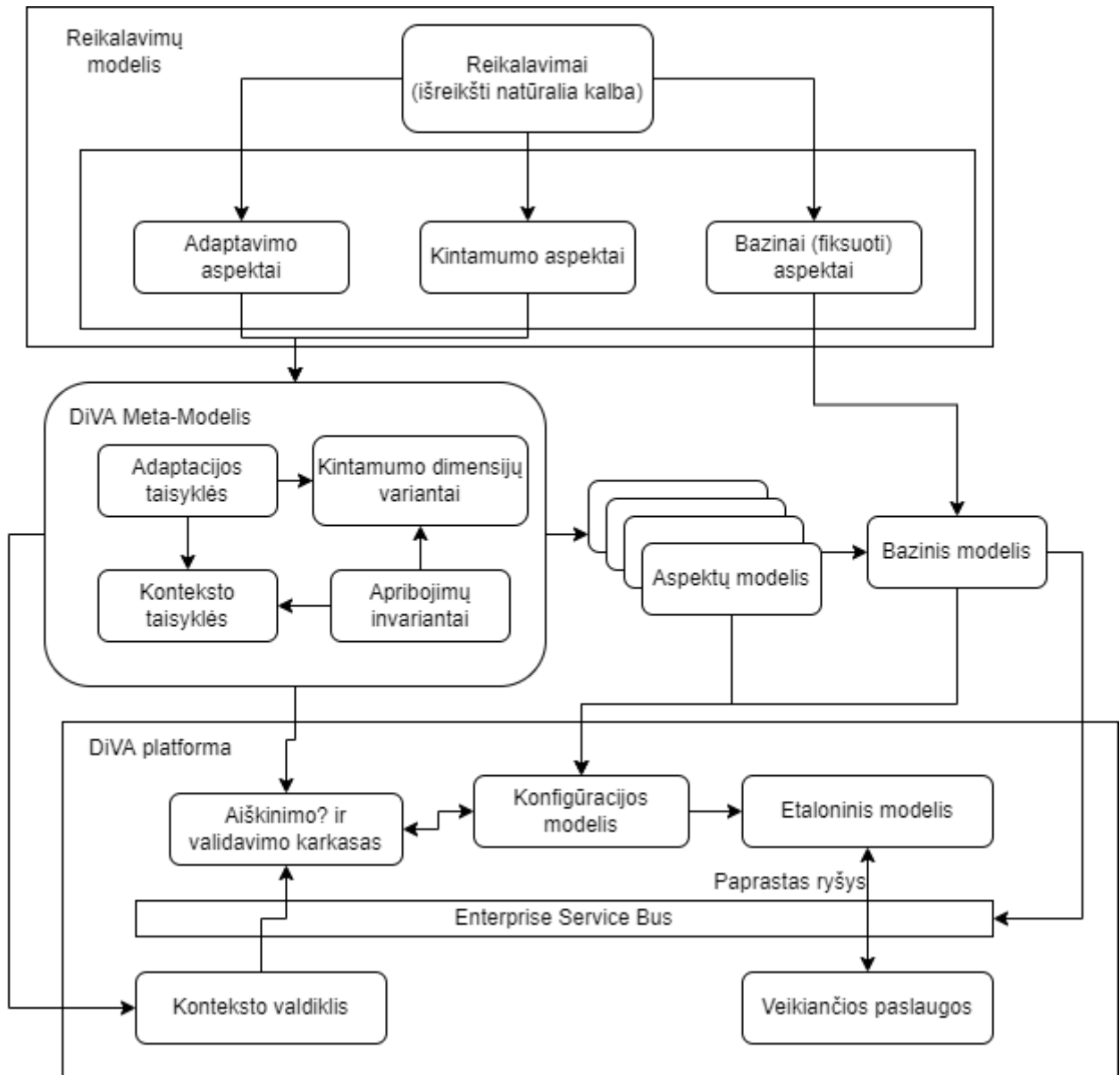
Pirmame DiVA proceso žingsnyje (12 pav.) iš natūralia kalba aprašytų pradinių reikalavimų išgaunami sistemos baziniai reikalavimai, kintamumo ir dinaminio adaptavimo aspektai. Šiame žingsnyje naudojami specifiniai įrankiai, kurie iš pradinių reikalavimų identifikuoja aukščiausio lygio sistemos funkcijas. Minėtos funkcijos yra potencialiai tiek pagrindinės paslaugos, tiek kintamumo taškai. Kartu identifikuojami susiję kontekstai ir apribojimai, užtikrinantys visų adaptacijų dinaminį pritaikymą ir eliminuojantys konfliktus vykdymo metu. Ši informacija yra išgaunama besinaudojant minėtais įrankiais pusiau automatizuotu būdu [Ra09, SRC+07].

Antrame žingsnyje, vadinamu DiVA meta-modeliu, yra konstruojamas operacinis, iš pirmo žingsnio gautų adaptuojamų reikalavimų, sistemos elgsenos modelis. Operacinis modelis pristato pagrindinį paslaugų sistemos veikimą, įtraukiant sistemos elgsenos aspektus. Jame modeliuojamos adaptacijos taisyklės, kintamumo dimensijų variantai, konteksto taisyklės ir apribojimų invarijantai. Papildomai, simuliacijos ir modelio tikrinimas yra naudojamas adaptacijos modeliui validuoti.

Trečias ir ketvirtas DiVA metodikos žingsniai apima sukurtų modelių validaciją ir adaptavimo modelių pavertimą vykdomaisiais modeliais. Šie modeliai yra validuojami, atsižvelgiant į skirtingus operacinius profilius, kurie atstovauja skirtingus programos ir platformos elgsenos elementus.

Aspektų modeliai yra architektūros elementai, iš kurių vykdymo metu sudaromas bazinis modelis. Komponuojant aspektus konfigūruojama prisitaikanti prie konteksto paslauga. DiVA platformos samprotavimo sistema (angl. *Reasoning framework*) konteksto duomenis surenka iš jutiklių. Samprotavimo sistema parenka geriausią konfigūraciją, atitinkančią adaptacijos taisyklės. Konfigūracijos modelis kuriamas vykdymo metu sudarant aspektų modelių kompoziciją. Pasikeitus konteksto duomenims, validavimo karkasas užtikrina saugų perėjimą iš vienos aspektų konfigūracijos prie kitos.

DiVA metodas turi keletą privalumų. Pirmas, nekontroliuojamas konfigūracijų skaičiaus augimas yra suvaldomas, modeliuojant kintamumo dimensijas ir variantus, o ne visą paslaugos konfigūraciją. Antras, adaptacijos logika yra aprašoma aukštame abstrakcijos lygyje, tokiu būdu atskiriama adaptacijos ir pačios sistemos logika. Trečias, aspektų modeliai yra validuojami ir verifikuojami anksčiau sistemos kūrimo gyvavimo ciklo etapuose, todėl galimos problemos atsiskleidžia projektavimo, o ne vykdymo metu, kai jų sprendimas taptų sudėtingas ir brangus.



12 pav. DiVA metodas (adaptuotas iš [GCA+11])

## 2.8. Literatūros analizės apibendrinimas

Apžvelgus literatūros šaltinius gauti atsakymai į literatūros analizei keliamus klausimus. Išvados suformuluotos atsakymų į pateiktus klausimus forma.

**Literatūros analizei keliami klausimai ir gauti atsakymai** (pateikti kursyvu):

1. Nustatyti paslaugų stiliaus sistemų reikalavimų inžinerijos savybes:

a. Kokios reikalavimų rūšys yra skiriamos?

*Reikalavimai kategorizuojami skirtingais būdais skirtingose dimensijose. Reikalavimai dalinami į ankstyvuosius ir vėlyvuosius, statinius ir dinامينius, adaptavimo ir stebėjimo reikalavimus. Ankstyvieji reikalavimai atspindi suinteresuotų asmenų intencijas. Kadangi jos kinta, tai jie turi būti nuolat atnaujinami. Vėlyvieji apibrėžia sistemos paslaugų veikimą, todėl kintamiems poreikiams turi būti adaptuojami. Kuriant sistemą turi būti nuspręsta, kurie sistemos aspektai statiniai (nekinantys), o kurie dinaminiai (adaptuojami). Stebėjimo reikalavimai yra išvedami iš adaptavimo reikalavimų ir turi būti nagrinėjama kaip kintamumą specifikuoti. Kokybės atributų svarba yra pripažįstama, bet jų specifikuojimas kintamumo kontekste nėra nagrinėjamas.*

b. Ką apima adaptuojamieji reikalavimai?

*Adaptuojamieji reikalavimai apima servisų adaptavimo aplinkybes, prioritetus ir adaptacijos veiklos tikslus.*

c. Kokius turinius turi apimti reikalavimų specifikacija?

*Specifikacija turėtų apimti tiek ankstyvuosius, tiek vėlyvuosius reikalavimų modelius. Specifikacija turi apimti visas galimas variacijas. Taip pat turi apimti tiek statinius, tiek dinامينius reikalavimus. Intencijomis grindžiamas paslaugos aprašas apima paslaugos interfeisą, elgseną, funkcinių aspektų kompoziciją ir paslaugos kokybę. Šiuo aprašu vėliau grindžiamas operacinis paslaugos aprašas, apimantis paslaugų sąveiką, koordinavimą ir funkcionalumą.*

2. Kokie kintami verslo proceso turiniai gali būti specifikuojami? Kaip?

*Iš verslo proceso perspektyvos, darbo eiga yra kintamas verslo proceso turinys.*

*Verslo procesų kintami darbo eigos dalys gali būti:*

- 1. **Mazgas.** C-iEPC metode – tai veikla ir sietuvas. Ribojimas: nereikalingos veiklos šalinamos, ribojami sietuvai.*
- 2. **Anotuojami elementai.** Pagal Konfigūruojamų procesų modeliavimą (išplėstas EPC metodas) tai yra: veikla, resursas, įvykis ir objektas. Anotuojami elementai pateikiami su dalykinės srities sąlygomis. Jeigu sąlygos rezultatas yra klaidingas, elementas iš modelio šalinamas.*

3. **Veikla.** PESOA metodikoje stereotipai priskiriami veikloms ir veiklų atributams. Adaptavimo metu nereikalingi variantai yra šalinami iš modelio.
4. **Fragmentas.** Provop metode įvykį, veiklą arba jų rinkinį galima trinti (apriboti bazinį modelį), įdėti (išplėsti bazinį modelį), perkelti arba pakeisti (kas yra pirmų dviejų atominių operacijų junginys) ir modifikuoti (keisti su veikla susijusius resursus arba objektus).
5. **Užduotys.** CMMN notacijoje visi kintami uždavinių, etapų variantai turi būti iš anksto paruošti. Remiantis loginėmis sąlygomis ir planavimo lentelėmis yra sprendžiama, kuris variantas yra vykdymo metu pridedamas prie proceso.

### 3. Kokie kintami paslaugų turiniai gali būti specifikuojami? Kaip?

*Iš paslaugos perspektyvos, kintami turiniai gali būti paslaugų kompozicija (kompozicijos kintamumas), tą pačią paslaugą atitinkantys skirtingi interfeisai (sąsajos kintamumas) ir skirtingos loginės paslaugos realizacijos (loginis kintamumas).*

*BP&SLA metodika nagrinėja paslaugų grindžiamų sistemų procesų specifیکavimą, tuo tarpu ir nefunkcinius reikalavimus, bet nekalbama apie kintamus turinius.*

*DiVA metodikoje valdomos funkcinių savybių kintamumas. Kintamų paslaugų konfigūracijų skaičiaus nekontroliuojamas augimas suvaldomas modeliuojant kintamumo dimensijas ir variantus, o ne visą paslaugos varianto konfigūraciją.*

### 4. Kaip specifikuojamos verslo procesų ir paslaugų nefunkcinės savybės?

*Šaltiniuose nagrinėti verslo procesų specifیکavimo metodai, pavyzdžiui, DiVA, nefunkcinių savybių specifیکavimą išreikštiniu būdu nenagrinėja.*

*Paslaugų nefunkcinės savybės yra fiksuojamos BP&SLA specifیکacijos SLA dalyje. Metodika apima keturis etapus: ankstyvųjų reikalavimų, verslo procesų hipergrafo ir hierarchijos išvedimą bei apribojimų SLA nustatymą.*

### 3. Kintamų reikalavimų specifikavimo metodas

Šiame skyriuje yra aprašomas dinamiškai kintančių verslo procesų specifikavimo ir modeliavimo metodas. Paprastai yra kuriamos ne atskiros paslaugos, o paslaugų šeimos, skirtos individualizuoti paslaugų teikimą, kai paslaugos adaptuojamos specifiniam naudojimui kontekstui. Naudojimo kontekstas (angl. *context of use*) apima naudotojas, veiklas, veikimo aplinką ir technologijas [ISO 9241-11:2018]. Verslo procesus įtakoja reikiamos suteikti paslaugos ir naudojimo konteksto kitimas, kas savo ruožtu sąlygoja funkcinių ir nefunkcinių reikalavimų kitimą.

Siūlomas metodas realizuojamas naudojantis modeliais grindžiama sistemų kūrimo paradigma (angl. *model driven architecture*) ir produktų šeimos kūrimo idėjomis. Produktų šeimos kintamumo specifikavimo ir modeliavimo metodika DiVA skirta modeliuoti, analizuoti, vertinti ir komponuoti kintančias prisitaikančių sistemų funkcinių savybių konfigūracijas paslaugos gyvavimo ciklo metu [GCA+11]. Ši metodika neapima paslaugas realizuojančio proceso automatizuoto specifikavimo, nefunkcinių savybių specifikavimo ir modeliavimo, todėl DiVA metodika yra modifikuojama ir išplečiama kintančių procesų nefunkcinių savybių kintamumui specifikuoti ir modeliuoti.

Pradžioje yra pateikiami paslaugos ir proceso metamodeliai, aprašomas siūlomas rezultatų gavimo procesas, toliau pateikiama jo konkretizacija ir pagrindžiantys argumentai. Pasiūlytas metodas iliustruojamas pavyzdžiu, specifikuojant konkrečios paslaugos - vartotojo web autentifikavimą – kintamą ir įgyvendinančią procesą.

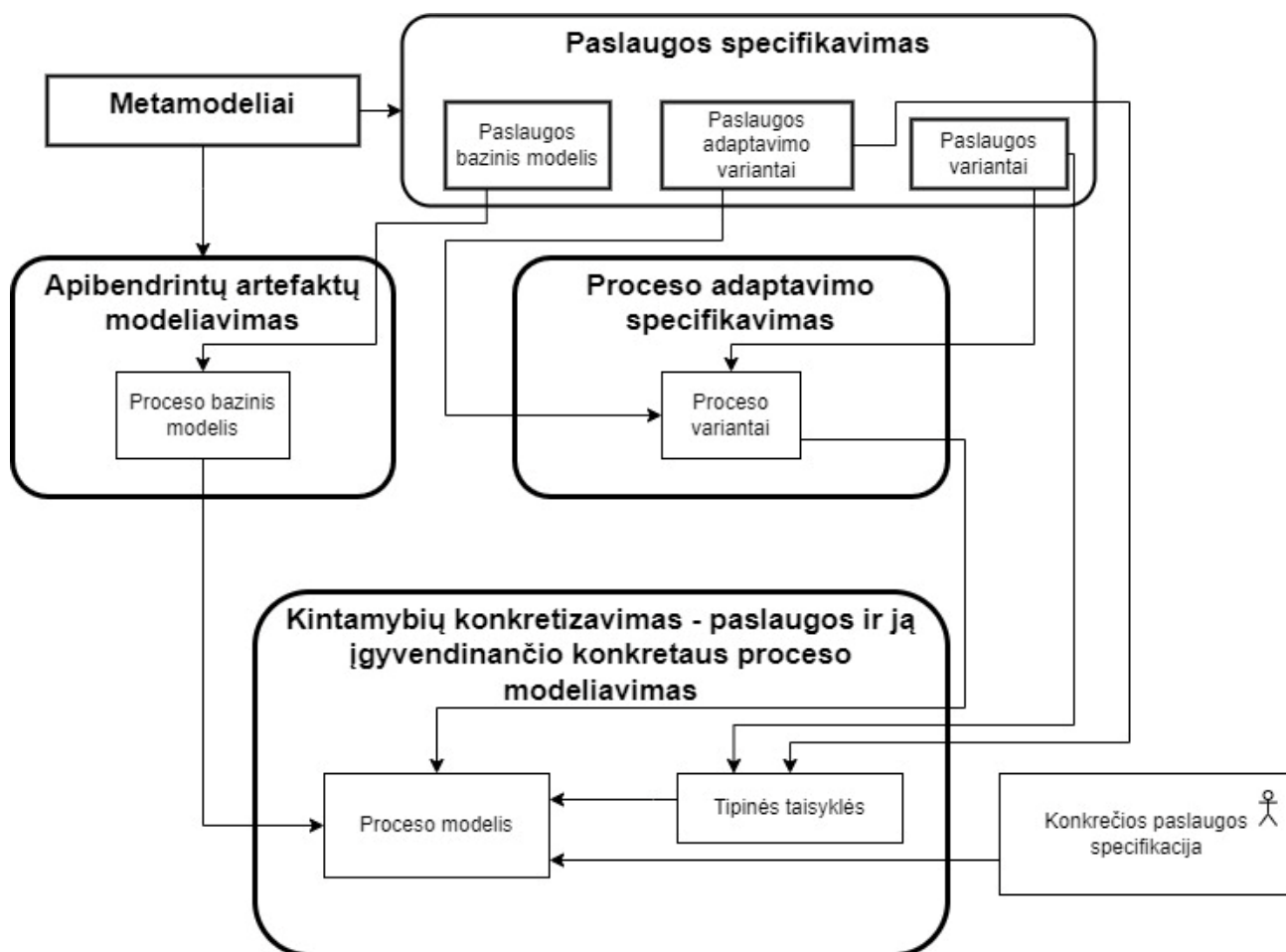
#### 3.1. Kintamųjų specifikavimo metodo bendra apžvalga

Paslaugas įgyvendinančiose sistemose skirtingi vartotojai gali gauti skirtingas to paties tipo paslaugas. Kad tai įgalinti, reikia identifikuoti kintamas paslaugų ir jas įgyvendinančių procesų dalis ir jas adaptuoti kiekvienam konkrečiam atvejui. Siūlomame metode paslaugos variantai yra išskiriami iš naudojimo scenarijų, kuriuose nagrinėjamos paslaugos nefunkcinės savybės. Metodas apima šiuos etapus: paslaugos specifikavimą, apibendrintų artefaktų modeliavimą, proceso adaptavimo specifikavimą ir kintamųjų konkretizavimą (13 pav.). Toliau šie etapai yra nagrinėjami detaliau.

Metamodeliais apibrėžiamos sąvokos naudojamos kintamųjų specifikavimo metode.

#### 3.2. Metamodeliai

Paslaugų ir procesų sampratų yra įvairių. Todėl šiame skyriuje apibrėžiama konceptų ir jų ryšių sistema, kuri turi būti vartojama modeliuojant paslaugas ir procesus, siekiant atsižvelgti į galimus jų variantus. Kitaip sakant, nustatoma, kaip bus konceptualizuojamos paslaugos ir procesai. Tik metamodeliu apibrėžti konceptai turi būti naudojami specifikuojant paslaugas ir procesus.

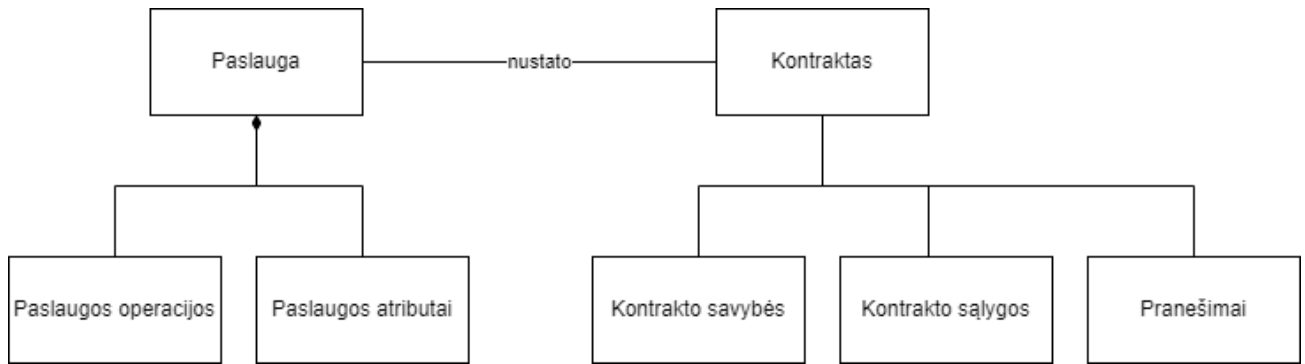


13 pav. Siūlomas kintamybių specifikavimo koncepcinis modelis. Stačiakampiai su apvalintais kampais žymi procesus. Paprasti stačiakampiai žymi artefaktus.

### 3.2.1 Paslaugos metamodelis

Paslauga suprantama kaip savybių ir funkcionalumo visuma (atributai ir operacijos); paslauga teikiama pagal kontrakte nustatytas sąlygas (14 pav.). Operacijos yra veiksmai susiję su teikiamais paslaugos variantais. Paslaugos atributai apibrėžia esybes ir jų savybes, kuriomis manipuluoja operacijos. Paslaugos kontrakte apibrėžiami paslaugos teikimo kokybės lygmenys ir teikiamų paslaugos variantų sąlygos.





14 pav. Paslaugos metamodelis.

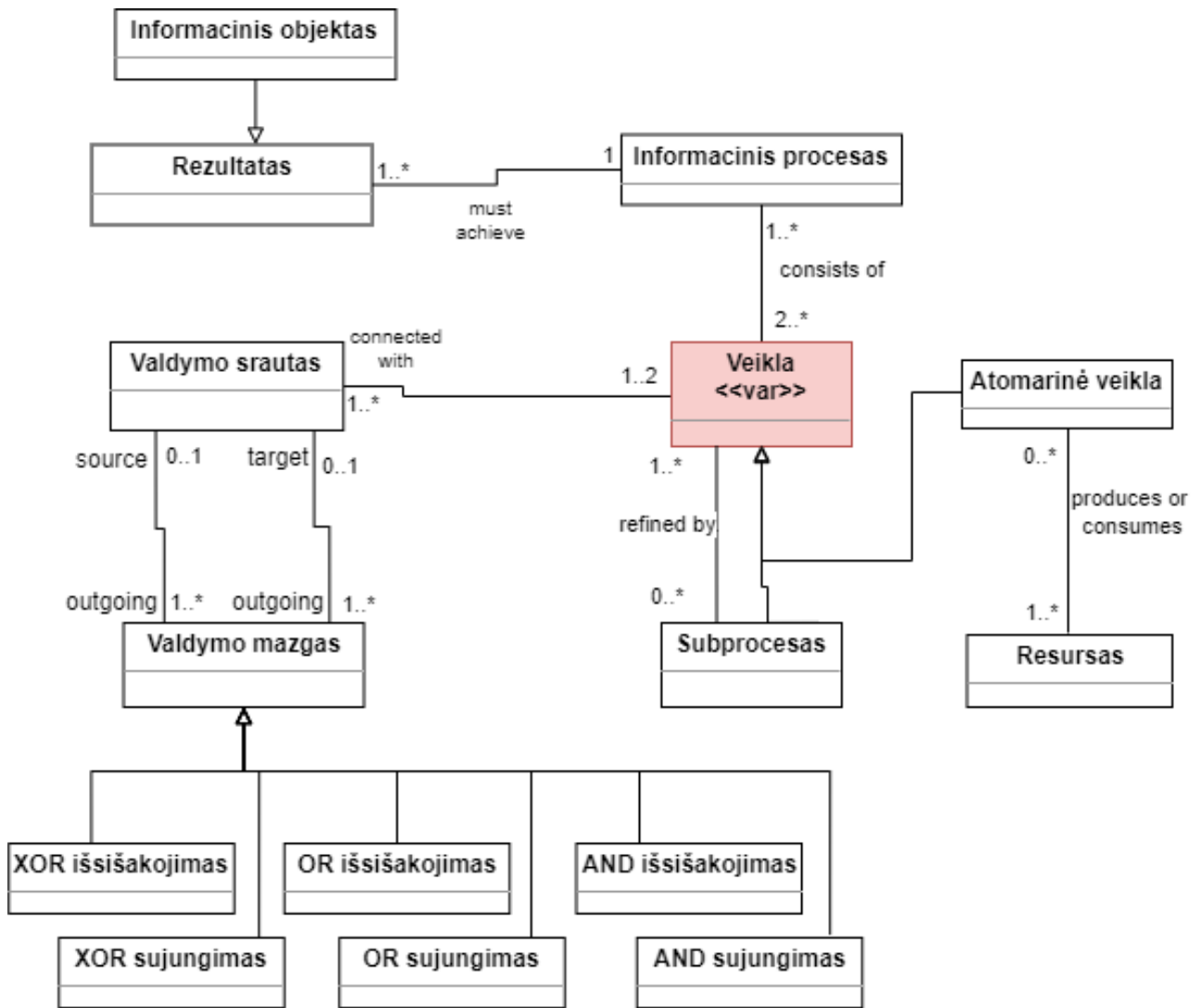
### 3.2.2 Proceso metamodelis

Projektuojant paslaugą su kintamybėmis, paslaugą realizuojantys verslo procesai yra sudaryti iš kintamų ir nekintamų veiklų (15 pav.). Siūlomą verslo proceso metamodelį (dalinai adaptuojamas iš [LK06]) 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 subprocesą. Atominė veikla yra nedaloma proceso dalis. Subprocesas, savo ruožtu, yra sudarytas iš kitų veiklų, dažnai atomarinių, rečiau dar iš kitų subprocesų. Iš paslaugomis grindžiamos sistemos perspektyvos, atominėje veikloje yra apdorjami informaciniai išteklių. Veiklų eiliškumą nusako 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 vykdymą ir visi srautai susilieja atgal į vieną srautą.
5. AND išsišakojimas – proceso šakos vykdomos lygiagrečiai.
6. AND sujungimas – proceso šakos baigia vykdymą ir visi srautai susilieja atgal į vieną srautą.

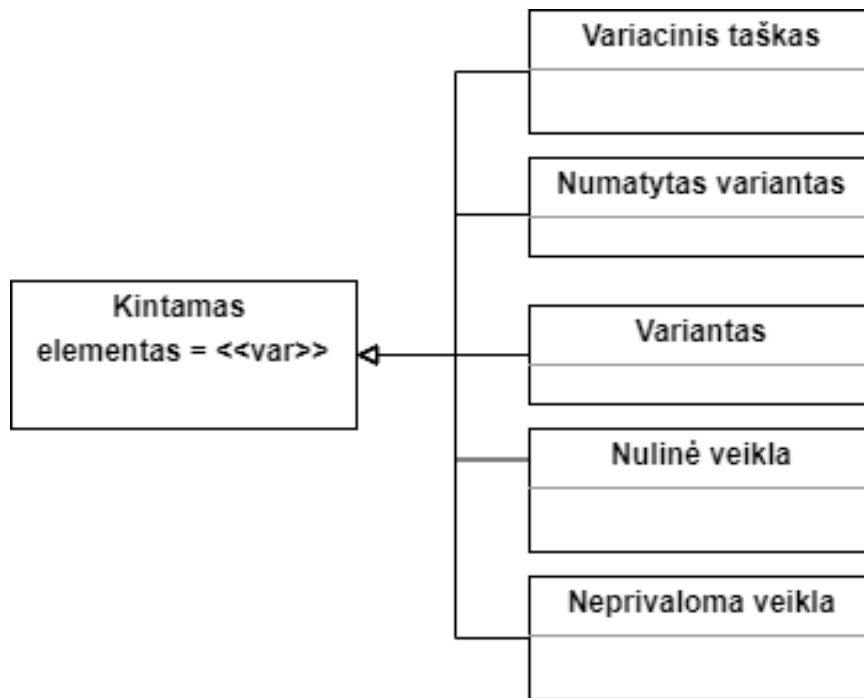


15 pav. Verslo proceso su kintamybėmis metamodelis (kintamas elementas pažymėtas raudonai)

### 3.2.3 Kintamumo mechanizmai

Atsižvelgiant į literatūros analizėje nagrinėtą medžiagą ir ją apibendrinant, nustatyti atliekamo tyrimo kontekste nagrinėtini kintamumo įgyvendinimo mechanizmai. Jie pateikti 4 pav. ir aprašyti UML veiklos diagrama, kurioje kintamumui žymėti įvedamos specialios anotacijos. Kintamumo mechanizmams įgyvendinti anotuotos veiklos gali būti:

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



16 pav. Kintamų elementų anotacijos metamodelis

**Variacinis taškas** (proceso bazinio modelio diagramoje (7 pav.) pažymėta <<VarPoint>>) nusakoma veiklos elementą, kuris specifikavimo 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** (proceso bazinio modelio diagramoje (7 pav.) pažymėta <<Default>>) yra veikla, kuri turėtų būti įkelti į variacinio taško vietą ir inicijuojama bendru atveju, jeigu iniciatorius nedaro išreikštinio sprendimo, kurį variantą naudoti.

**Variantas** (proceso bazinio modelio diagramoje (7 pav.) pažymėta <<Variant>>) 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 išreikštiniu būdu.

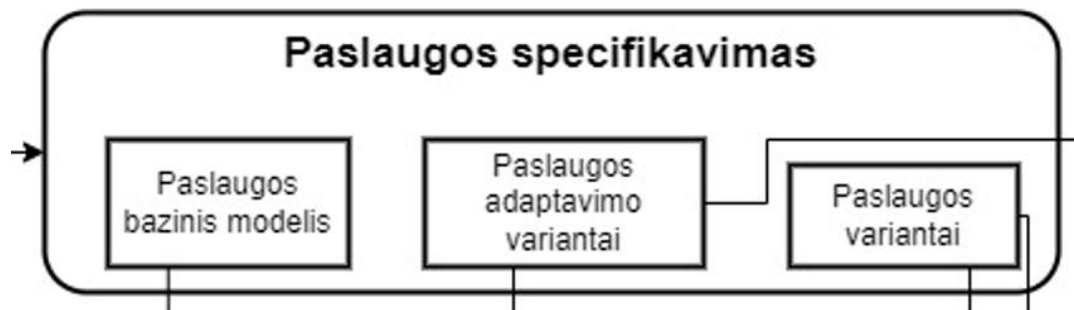
**Nulinė veikla** (proceso bazinio modelio diagramoje (7 pav.) pažymėta <<Null>>) yra panašus į variacinį tašką, tik gali būti praleistas, tai yra nepasirenkamas joks iš apibrėžtų variantų, jeigu konkrečioje situacijoje tam nėra poreikio.

**Neprivaloma veikla** (proceso bazinio modelio diagramoje (7 pav.) pažymėta <<Optional>>) 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.

### 3.3. Paslaugos specifikavimas

Šiame poskyryje yra nagrinėjamas siūlomo kintamybių specifikavimo modelio paslaugos specifikavimo dalis, kurią sudaro paslaugos bazinis modelis, adaptavimo variantai ir paslaugos variantai (17 pav.).



17 pav. Siūlomo kintamybių specifikavimo koncepcinio modelio (13 pav.) fragmentas

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

**Paslaugos adaptavimo variantai** yra susiję su paslaugos kintamumu ir dinaminiu 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 (angl. *feature*) 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 (3 lentelė), kurie yra apibrėžti paslaugos metamodelyje (14 pav.).

3 lentelė. Paslaugos specifikacija

<b>Paslaugos operacijos ir atributai</b>	<b>Apibrėžimas</b>
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 paslaugos bazinio lygmens veikimo matų
Pagrindiniai tikslų rodikliai	Bendra paslaugos nauda klientui

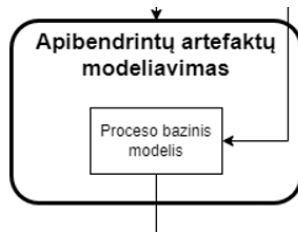
Paslauga apibrėžia kontraktą, kuris išvardina nefunkcines paslaugos savybes ir aprašo jos teikimo kokybės matus (4 lentelė). Vartotojas pasirenka atitinkantį jo poreikius paslaugos teikimo kokybės lygmenį. Paslaugų lygmenų kriterijai yra apibrėžiami, analizuojant naudojimo konteksto scenarijus.

4 lentelė. Paslaugos kontrakto aprašas

<b>Paslaugų adaptavimo variantai</b>	<b>Paslaugos lygmuo 1</b>	<b>Paslaugos lygmuo 2</b>	<b>...</b>	<b>Paslaugos lygmuo N</b>
Nefunkcinė charakteristika 1	Kriterijus <sub>11</sub>	Kriterijus <sub>21</sub>	...	Kriterijus <sub>N1</sub>
Nefunkcinė charakteristika 2	Kriterijus <sub>12</sub>	Kriterijus <sub>22</sub>	...	Kriterijus <sub>N2</sub>
...	...	...	...	...
Nefunkcinė charakteristika M	Kriterijus <sub>1M</sub>	Kriterijus <sub>2M</sub>		Kriterijus <sub>NM</sub>

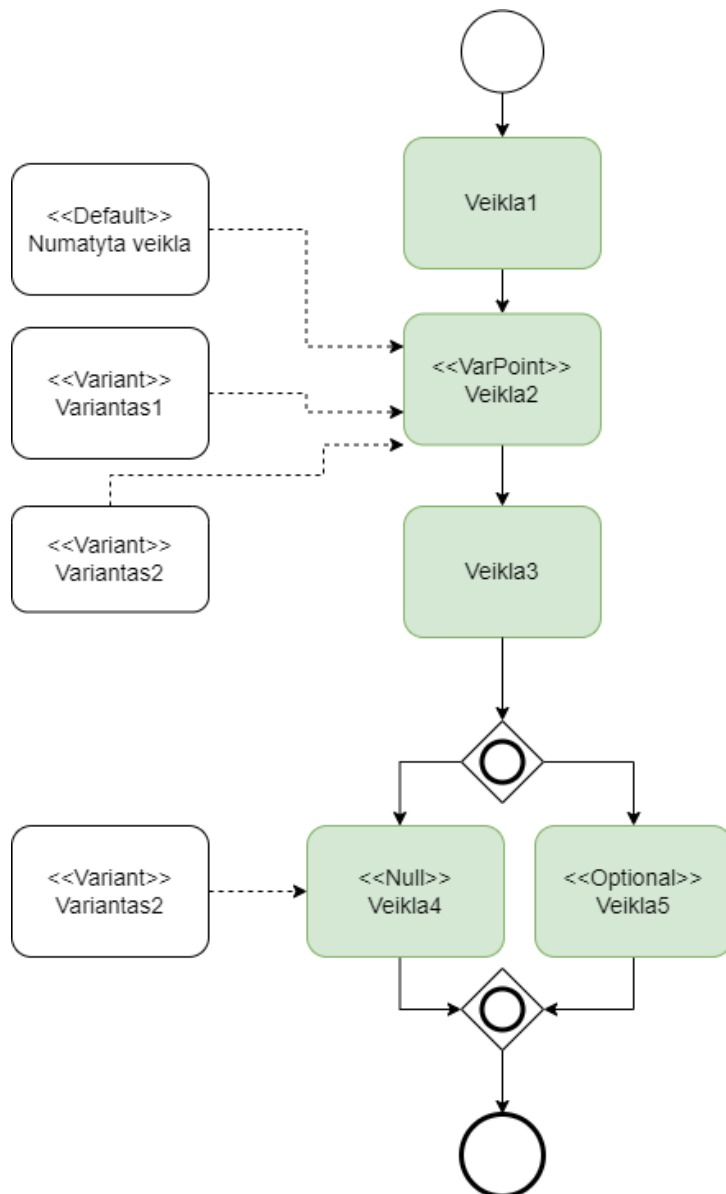
### 3.4. Apibendrintų artefaktų modeliavimas

Šiame etape, iš ankstesnės paslaugos specifikavimo dalies sudaryto paslaugos bazinio modelio, sudaromas proceso bazinis modelis (18 pav.). Proceso diagramos komponentai yra apibrėžti proceso metamodelyje (15 pav.). Proceso baziniame modelyje yra sukomponuojami paslaugos specifikacijoje apibrėžti paslaugos proceso bazinės veiklos ir jų eiliškumas.



18 pav. Siūlomo kintamybių specifikuojamo koncepcinio modelio (13 pav.) fragmentas

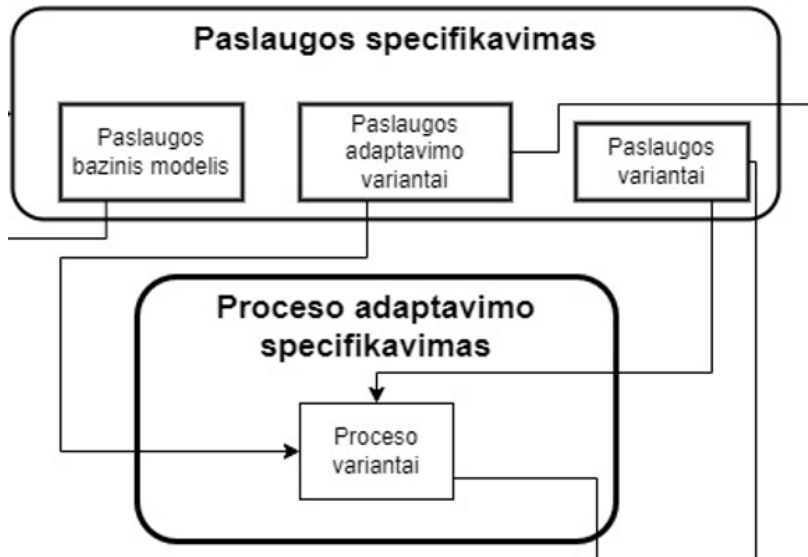
Proceso bazinio modelio pavyzdyje (19 pav.) parodytas procesas, kuriame apibrėžtos penkios veiklos. Veikla1 ir Veikla3 vykdomos vienodai, kitaip tariant, jos neturi kintamybių. Veikla2, Veikla4 ir Veikla5 turi kintamumo mechanizmus.



19 pav. Proceso bazinio modelio pavyzdys. Žalia spalva pažymėti proceso bazinio modelio elementai

### 3.5. Proceso adaptavimo specifikavimas

Šiame etape yra dekomponuojami paslaugos specifikacijoje išvardinti adaptavimo variantai, kurie susiejami su paslaugos variantais. Rezultate suformuojami proceso variantai (20 pav.).



20 pav. Siūlomo kintamybių specifikavimo koncepcinio modelio (13 pav.) fragmentas

Dekomponuojant adaptavimo variantus, yra analizuojamos nefunkcinės savybės (NF) ir identifikuojamos charakteristikos, įtakančios paslaugos varianto pasirinkimą (5 lentelė). Charakteristikoms yra priskiriamos vertės, gaunamos analizuojant paslaugos naudojimo kontekstą.

5 lentelė. Adaptavimo kriterijų (nefunkcinių savybių) dekompozicija

	NF <sub>1</sub>			NF <sub>2</sub>			NF <sub>M</sub>		
Paslaugos variantai	Ch <sub>11</sub>	...	Ch <sub>1N</sub>	Ch <sub>21</sub>	...	Ch <sub>2N</sub>	Ch <sub>M1</sub>	...	Ch <sub>MN</sub>
Variantas <sub>1</sub>	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas <sub>2</sub>	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas <sub>3</sub>	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
Variantas <sub>K</sub>	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1

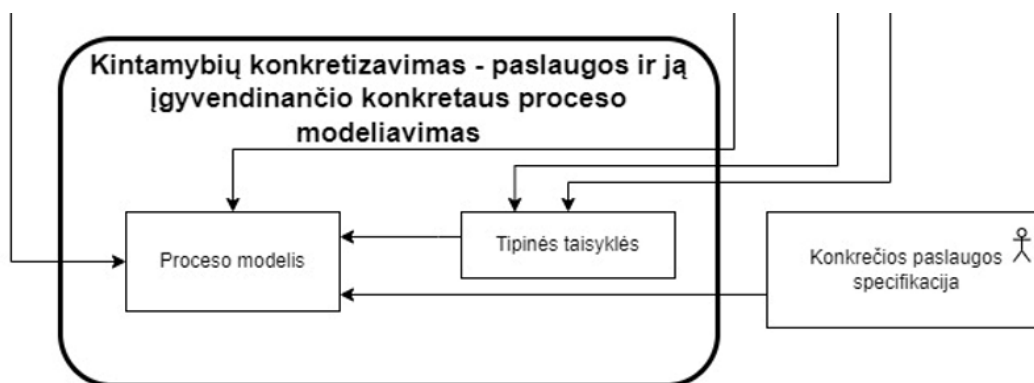
Mobiliajame įrenginyje veikiančio žaidimo adaptavimo pavyzdys, kuriame veikiantis žaidimo variantas priklauso nuo baterijos įkrovimo ir mobiliųjų duomenų išnaudojimo lygio (6 lentelė). Charakteristikos suteikiamos prioritetų vertės, skirtos konfigūruoti paslaugos teikimo kokybės lygį.

6 lentelė. Adaptavimo kriterijų dekompozicijos pavyzdys: mobiliajame įrenginyje veikinčio žaidimo varianto konfigūracija

	Našumas		Suderinamumas
Paslaugų adaptavimo variantai	Žemas baterijos įkrovimo lygis	Aukštas mobilių duomenų išnaudojimas	Prieinamas WiFi tinklas
Aukštos kokybės sąveikos režimas	0,2	0,2	1
Resursus taupantis režimas	1	1	0,5

### 3.6. Kintamybių konkretizavimas

Konkretizuojant kintamybes modeliuojamas konkretaus paslaugos varianto pasirinkimo procesas (21 pav.). Pasirinkimas yra apibrėžiamas tipinėse taisyklėse, kuriose susiejami adaptavimo variantai su paslaugos variantais. Proceso modelis nusako konkretaus paslaugos varianto proceso realizaciją.



21 pav. Siūlomo kintamybių specifikuojamo koncepcinio modelio (13 pav.) fragmentas

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 gali būti specifikuojamos atsižvelgiant į visas nefunkcines savybes (7 lentelė) arba į konkrečią vieną nefunkcinę savybę (8 lentelė).

7 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 cha-rakteristikų) kriterijų reikšmių intervalus, atrenkami proceso veiklos va-riantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos va-riantais (<<Variant>>, vienas arba daugiau).

8 lentelė. Variacinio taško, atsižvelgiančio į vieną nefunkcinę charakteristiką, tipinė taisyklė

Pavadinimas	VariacinisTaškas2
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkre-tinta veikla ir reikia užtikrinti vieną paslaugos nefunkcinę savybę.
Sprendinys	Pagal nustatytus leidžiamus vieno adaptavimo varianto (nefunkcinę cha-rakteristiką) kriterijų reikšmių intervalus, atrenkami proceso veiklos va-riantai (vienas arba daugiau), atitinkantys reikalavimus.
Rezultatas	Paslaugos tiekimo procesas su reikalavimus tenkinančiais veiklos va-riantais (<<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.

### 3.7. Apibendrinimas

Apibendrinant pabrėšime, kad 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 aukštesniame 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. Metodikos taikymo pavyzdys: autentifikacijos paslauga**

Šiame darbe sukurtame metode specifikuojamos teikiamos paslaugų šeimos nefunkcinės savybės, įtakojančios naudotojo poreikius atitinkančio paslaugos varianto pasirinkimą. Metodui iliustruoti pasirinkta web autentifikavimo paslaugų šeima, nes ši paslauga turi nusistovėjusius teikimo variantus, kurie pristatomi ir teikiami kaip atskiros paslaugos. Kuriant programų sistemas yra suprojektuojamas pasirinktas web autentifikavimo variantas, neprisitaikantis nei prie paskyroje saugomų duomenų jautrumo nei prie naudotojo naudojamo įrenginio. Sujungus skirtingus variantus į šeimą ir apibrėžus paslaugos kintamybes, paslaugos teikimas taptų lankstesnis. Kitas pasirinkimo motyvas yra jau atliktas išsamus Bonneau, Herley ir kitų autorių autentifikavimo schemų tyrimas [BHO+12], kuriame ištirti web autentifikacijos naudojimo kontekstai, identifikuoti web autentifikacijos variantai ir jų pasirinkimą įtakojančios nefunkcinės savybės. Šis tyrimas sudaro pagrindą specifikuoti web autentifikacijos paslaugų šeimos kintamumus ir konfigūruoti konkretaus varianto pasirinkimą. Taikant sukurtą metodiką kitai paslaugai, jos variantų ir nefunkcinių savybių analizė turi būti atlikta atskirai.

Web autentifikavimo paslaugos variantų pasirinkimus įtakoja dvi nefunkcinės savybės: saugumas ir panaudojamumas. Sistemos saugumo reikšmė tolygiai didėja, kartu didėja ir jos užtikrinimo kaštai. Verslai turi užtikrinti jų vartotojų duomenų saugumą, todėl adaptuojama paslauga turėtų suteikti priimtina saugumo lygį už priimtina kainą. Didinant saugumo reikalavimus, naudojimosi procedūra darosi sudėtingesnė vartotojui, kitaip tariant, blogėja jos panaudojamumas. Todėl reikalingas šių aspektų balansavimas, atsižvelgiant į naudojimo konteksto ypatybes. Jautriems duomenims reikalinga stipresnis saugumas, kurią naudodamas vartotojas sutinka su sudėtingesne prisijungimo procedūra, t. y. žemesniu panaudojamumo lygiu. Daugelis paskyrų neturi itin jautrių duomenų ir vartotojas apie tai žino, todėl šiuo atveju yra svarbesnis panaudojamumas, t. y. patogesnė prisijungimo procedūra nei atitikimas griežtiems saugumo standartams.

Apibendrinant, kuo aukštesnis saugumo lygis, tuo sudėtingiau paslauga naudotis. Saugumo ir panaudojamumo balansavimas sudaro pagrindą paslaugų kintamybėms apibrėžti.

### **4.1. Paslaugos specifikacija**

Šiame skyriuje pateikiamas pavyzdinės paslaugos – vartotojų web autentifikacijos – nefunkcinių reikalavimų įtraukimo į paslaugos kintamybių apibrėžimą ir paslaugos variantų formavimą pavyzdys. Toliau jo pagrindu apibrėžiama proceso specifikacija.

Paslaugos specifikacijos pavyzdyje (9 lentelė) pateikta, kad vartotojų web autentifikaciją sudaro kelios autentifikavimo schemos: tekstinė, grafinė, grindžiama telefonu ir biometrinė. Kiekviena iš šių schemų pasižymi konkrečiais privalumais skirtingose naudojimo kontekstuose [BHO+12].

Šiuo metu vartotojai autentifikuojami iš anksto pasirinkta schema, kuri nebūtinai atitinka vartotojo poreikius konkrečiame kontekste. Šis pavyzdys yra grindžiamas prielaida, kad autentifikavimo schema gali būti pritaikoma įvairiems naudojimui kontekstams<sup>1</sup>.

9 lentelė. Web autentifikacijos paslaugos aprašas

<b>Paslaugos pavadinimas</b>	<b>Prisitaikanti Web autentifikacija</b>
Paslaugos aprašas	Paslauga siūlo vartotojo autentifikavimą atsižvelgiant į panaudojamumo ir saugumo poreikius. Pagal šios kriterijus vartotojui siūlomi paslaugos teikimo variantai. Jei konkrečiam kontekstui tinka keli variantai, galutinį sprendimą priima vartotojas.
Bazinis paslaugos variantas	Vartotojo vardo ir slaptažodžio pateikimas
Paslaugos variantai	Tekstinis, grafinis, biometrinis ir telefonu grindžiamas slaptažodžio pateikimas
Paslaugos prieinamumas	99%
Paslaugos savininkas	Vardenis Pavardenis, +3709999999
Paslaugos naudotojai	Kompiuterizuoti verslai
Paslaugos veikimo kokybės matai	Panaudojamumas, saugumas
Pagrindiniai tikslų rodikliai	Prisitaikantis prisijungimas leidžia pritraukti naudotojus su įvairiais saugumo poreikiais ir IT gebėjimų įgūdžių lygmenimis. Tokiu būdu paslauga taps lankstesnė ir labiau prieinama verslų klientams.

#### 4.2. Kontrakto specifikacija

Web autentifikacijos kontakte apibrėžiami paslaugos teikimo lygmenys: patogus, vidutiniškas ir saugus. Paslauga autorizuoja ir verifikuoja vartotoją pagal kintamus autorizacijos būdus. Šie būdai išvedami iš naudojimo konteksto scenarijaus.

<sup>1</sup> Naudojimo kontekstas (angl. *context of use*) apima vartotojų charakteristikas, kompiuterizuojamą veiklą, veiklos aplinką ir naudojamą technologijas [ISO9241-11:2018].

#### 4.2.1 Pirmasis naudojimo scenarijus – patogus

Savininkas kuria nedidelę elektroninę parduotuvę, kurioje prekės yra tik reklamuojamos, o pirkimas realizuojamas, naudojant išorinę elektroninės bankininkystės paslaugą. Tokia paslauga nereikalauja aukšto lygio saugumo. Siekiant pritraukti klientus, parduotuvė turėtų užtikrinti lengvesnę prieigą, keliant ir redaguojant prekes iš bet kokio įrenginio. Šiuo atveju panaudojamumo lygis turėtų būti aukštesnis, o saugumo lygis gali būti minimalus.

#### 4.2.2 Antras naudojimo scenarijus – vidutiniškas (patogiai saugus)

Asmeninėje informacijos saugykloje (nuotraukos, filmai) gali būti jautri asmeninė informacija. Užtikrintas turėtų būti vidutinis saugumo lygis. Kadangi dažniausiai vartotojas saugykloje laiko asmeninę, privataus gyvenimo informaciją, ne komercinę, tai jam turėtų būti užtikrintas pakankamai patogi prieiga.

#### 4.2.3 Trečias naudojimo scenarijus – saugus

Vartotojas kuria vidutinę parduotuvę, kurioje bus atliekamos ir finansinės operacijos, tame tarpe vieno paspaudimo pirkimas. Tokios parduotuvės naudotojų paskyrose saugomi naudotojo mokėjimo kortelės duomenys. Elektroninės prekybos paslaugomis naudojama iš įvairaus dydžio įrenginių ir įvairių naršyklių. Tokiai paskyrai reikalingas aukštesnio lygio saugumas. Nors šiuo atveju naudotojui yra svarbus saugumas ir žemesnis panaudojamumo lygis būtų priimtinas, tačiau elektroninių parduotuvių rinkoje konkurencija yra labai aukšta, todėl panaudojamumo lygis turėtų būti bent vidutinis.

#### 4.2.4 Kontrakto specifikacijos apibendrinimas

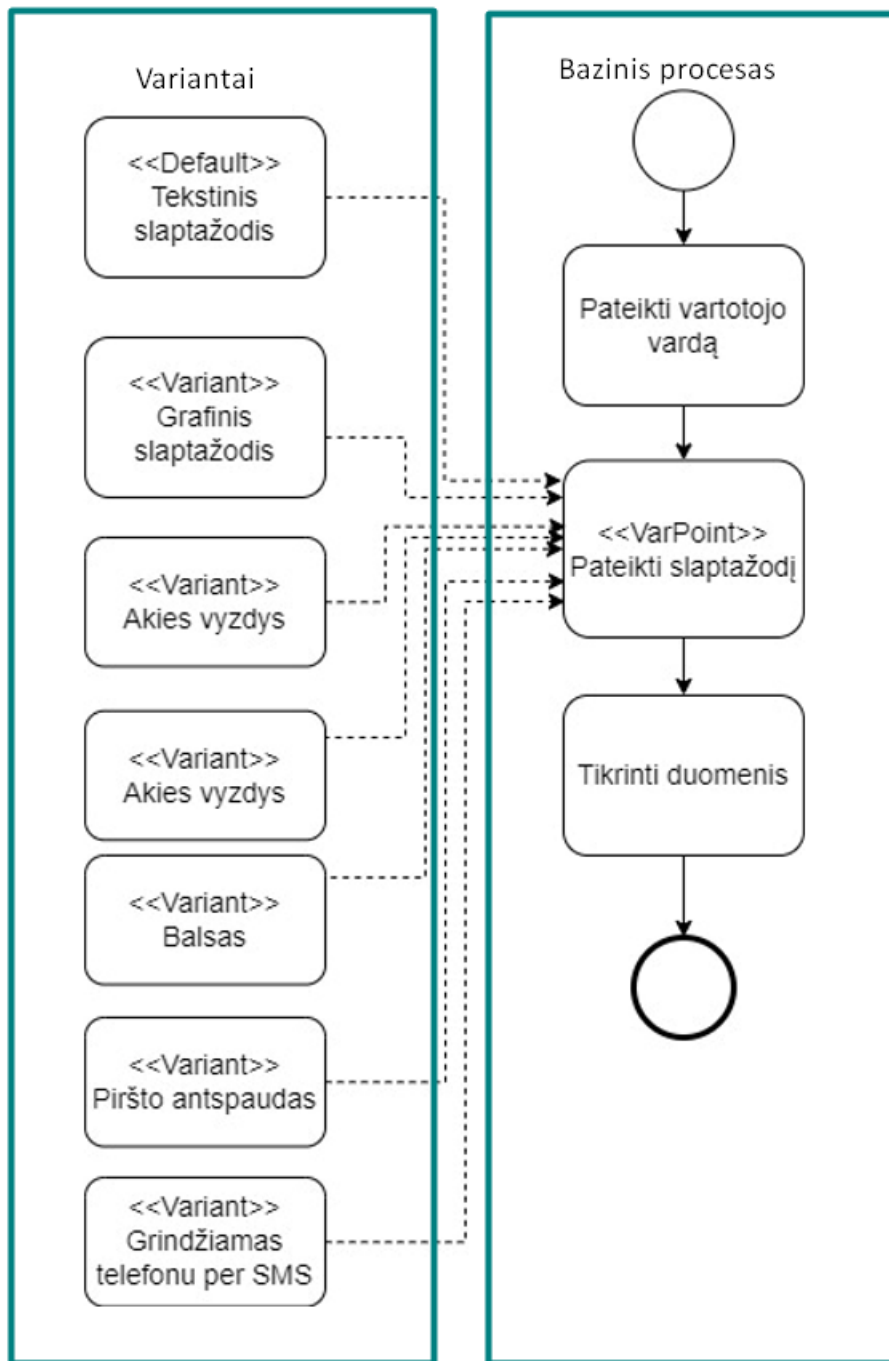
Adaptavimo variantai yra vertinami trijų lygių skalėje: žemas (iki 30%), vidutinis (nuo 30% iki 70%), aukštas (nuo 70% iki 100%).

10 lentelė. Web autentifikacijos kontrakto aprašas

<b>Paslaugų adaptavimo variantai</b>	<b>Patogus</b>	<b>Vidutiniškas</b>	<b>Saugus</b>
Panaudojamumas	Aukštas	Vidutinis	Vidutinis
Saugumas	Žemas	Vidutinis	Aukštas

### 4.3. Proceso bazinis modelis ir variantai

Šiame skyriuje formuojamas bazinis paslaugos teikimo proceso modelis. Web autentifikacijos pavyzdyje jį sudaro trys veiklos: dvi pastovios, viena su kintamybėmis (22 pav.). Aiškumo dėlei, su kintamybės tašku susieti paslaugos variantai. Kitame skyriuje yra detaliau nagrinėjamos nefunkcinės, kurių pagrindu bus toliau detalizuojami proceso variantai.



22 pav. Proceso bazinis modelis ir galimi variantai

#### 4.4. Nefunkcinių savybių analizė

Web autentifikacijos nefunkcines savybes sudaro panaudojamumas (angl. *usability*) ir saugumas (angl. *security*) [BHO+12]. Šios savybės yra pasirinktos todėl, kad jos savo prasme dažnai yra tarpusavy prieštaringos. Jei autentifikavimas yra patogus vartotojui, kitaip tariant panaudojamas (angl.

usable), pvz. vartotojas visose paskyrose naudoja vieną paprastą slaptažodį (pvz. *qwerty*), tuomet jis yra lengvai atspėjamas, todėl yra nesaugus. Atvirkščiai, jei slaptažodis yra saugus, jį sudaro didžiosios, mažosios raidės, skaitmenys ir simboliai, tuomet jis yra sunkiai atsimenamas. Paslaugos variacijos specifikavimo pavyzdyje šios savybės parinktos, kadangi jų derinys turi būti optimizuojamas, atsižvelgiant į paslaugos naudojimo kontekstą. Toliau apibrėžiami nefunkciniai panaudojamumo ir saugumo kriterijai, kurių pagrindu yra identifikuojamos paslaugos kintamybės. Toliau adaptavimo variantai yra dekomponuojami į matuojamas savybes, kurių pagrindų bus skaičiuojami paslaugos tiekimo variantai.

#### 4.4.1 Panaudojamumo matai

Panaudojamumo savybės nusako paslaugos patogumą vartotojui [BHO+12]:

- P1. *Atsimenamumo pastangos* nustato, kiek pastangų reikalauja autentifikacijos duomenų prisiminimas. Idealiu atveju nereikia nieko prisiminti (įvertis 1). Jei kiekvienai paskyrai naudotojas turi prisiminti skirtingus duomenis, tuomet aspektas netenkinamas (įvertis 0).
- P2. *Tinka dideliame paskyrų skaičiui* aspektas matuoja, ar stipriai didelis paskyrų skaičius apkrauna naudotoją (nagrinėjama tik naudotojo perspektyva). Geriausiu atveju didelis paskyrų kiekis neturi neigiamos įtakos naudotojo kognityvinėms pastangoms (įvertis 1). Blogiausias – prisiminimo reikalauja kiekviena paskyra (įvertis 0).
- P3. *Nereikalauja papildomų prietaisų* aspekte vertinama, ar naudotojui reikia turėti su savimi papildomą fizinį objektą (elektroninį prietaisą, mechaninį raktą, popieriaus lapą. Geriausiu atveju, naudotojui nieko nereikia turėti (įvertis 1). Tarpinis variantas būtų, jei objektas yra nešiojamas kartu bet kuriuo atveju, pvz. mobilusis telefonas, bet ne kompiuteris arba planšetė (įvertis 0,5). Blogiausias, prietaisas reikalingas (įvertis 0).
- P4. *Lengvai išmokstamas* matuoja, kiek schema reikalauja pastangų ją išmokti, pirmą kartą ją naudojant, o vėliau – ar lengva prisiminti, kaip ja naudotis. Geriausiu atveju, schema yra lengvai ir greitai išmokstama (įvertis 1). Blogiausias, mokymasis ir prisiminimas reikalauja didelių pastangų (įvertis 0).
- P5. *Naudojimo efektyvumas (prisijungimo trukmė)* matuoja, ar laikas, skiriamas kiekvienai autentifikacijai, yra priimtinaai trumpas. Laikas reikalingas kuriant naują paskyrą, galėtų būti ilgesnis, bet taip pat pakankamai trumpas. Geriausiu atveju, naudotojas prisijungdamas sugaišta mažai laiko (įvertis 1). Blogiausias, prisijungimas užtrunka daug laiko (įvertis 0).
- P6. *Lengvas atstatymas pametus* matuoja, ar lengva atstatyti savo autentifikacijos prieigą, jeigu autentifikacijos žetonas (angl. *token*) yra pamestas arba prisijungimo duomenys yra pamiršti.

Geriausiu atveju naudotojas gali greitai atstatyti prieigą (įvertis 1). Blogiausiu, atstatymas užtrunka ilgai ir reikalauja nemažai pastangų (įvertis 0).

P7. *Fizinių pastangų kiekis* matuoja fizinių veiksmų pastangas reikalingų pateikti autentifikacijos duomenis. Geriausiu atveju pastangos nėra didesnės, tarkime, vieno mygtuko paspaudimas (įvertis 1). Tarpiniu atveju papildomi veiksmai yra natūralūs ir nereikalauja nenatūralių judesių (įvertis 0,5). Kitais atvejais yra priskiriamas blogiausias įvertis (įvertis 0), pvz. kai reikia daug rašyti klaviatūra.

#### 4.4.2 Saugumo matai

Saugumo matai apima aspektus, kurie nusako [BHO+12]:

- S1. *Atsparus fiziniam stebėjimui* reiškia, jog kenkėjas negali apsimesti kitu naudotoju, stebėdamas jo autentifikavimąsi. Geriausiu atveju, net stebint daug kartų neįmanoma apsimesti kitu vartotoju (įvertis 1). Tarpiniu atveju, nulaužti schemą pakaktų daugiau nei 10–20 stebėjimų (įvertis 0,5). Blogiausiu atveju, po 1 ar 2 stebėjimų kenkėjas gali sėkmingai apsimesti tikru vartotoju (įvertis 0). Stebėjimai apima žiūrėjimą per petį, klaviatūros filmavimą, klavišų paspaudimo garsų įrašymą.
- S2. *Atsparus tikslingam apsimitinėjimui* kriterijuje nustatoma, ar naudotojo asmeninės informacijos žinojimas padeda kenkėjui juo apsimesti. Geriausiu atveju, žinomi asmeniniai duomenys nepadeda kenkėjui apsimesti kitu naudotoju (įvertis 1). Blogiausiu atveju, asmeninių duomenų žinojimas leidžia kenkėjui gali apsimesti kitu asmeniu (įvertis 0).
- S3. *Atsparus vidiniam stebėjimui* – pažymi, ar įmanoma kenkėjiškomis programomis nuskaityti vartotojo įvestį įrenginyje. Geriausiu atveju, tai padaryti neįmanoma (įvertis 1). Tarpiniu atveju kenkėjiška programa gali nustatyti autentifikacijos duomenis tik po 10–20 prisijungimų (įvertis 0,5). Blogiausiu atveju, pakanka ir vieno stebėjimo (įvertis 0).
- S4. *Atsparus sukčiavimui* (angl. *phishing*) – aprašo, ar kenkėjas imituodamas tikrą autentifikavimo paslaugos tiekėją gali surinkti prisijungimo duomenis, kurie vėliau būtų panaudoti autentifikacijai. Geriausiu atveju, surinkti įvesties duomenys nėra naudingi kitam prisijungimui apsimitant tikru naudotoju (įvertis 1). Blogiausiu atveju, pavogti duomenys gali būti panaudoti apsimitant tikru naudotoju (įvertis 0).
- S5. *Atsparus vagystei* kriterijus nustato, ar naudojamo autentifikavimui išorinio objekto vagystė gali padėti kenkėjui autentifikuotis kito naudotojo vardu. Geriausiu atveju, išorinis objektas nėra naudingas kenkėjui (įvertis 1). Blogiausiu atveju, minėtas objektas duoda prieigą prie vartotojo paskyros (įvertis 0).



## 4.5. Web autentifikavimo konfigūracijos procesų variantų analizė

Nefunkcinių web autentifikacijos savybių analizė yra grindžiama išsamiu Bonneau, Herley ir kitų autorių atliktų autentifikavimo schemų kategorijų tyrimu [BHO+12]. Šių autorių pateiktais įverčiais yra naudojamos apibrėžiant paslaugos adaptavimo būdus.

Įrenginių galimybių įvairovė leidžia rinktis šiuos autentifikacijos būdus:

1. Tekstinis prisijungimas, kuriuo metu naudotojas įveda simbolių eilutes, apibrėžiančias vartotojo vardą ir slaptažodį.
2. Grafinis prisijungimas, kuriuo metu vartotojas pasirenka slaptažodį paveikslą arba jo dalį, kuri ir yra autentifikacijos duomenys. Šis būdas sutrumpina vartotojo įvestį ir tinka liečiamiesiems ekranams.
3. Telefonu grindžiamo prisijungimo metu autentifikacijai naudojamas naudotojo telefonas.
4. Biometrinis prisijungimas išnaudoja žmogaus unikalius biometrinius duomenis.

Kiekvienas variantas nagrinėjamas panaudojamumo ir saugumo aspektais.

### 4.5.1 Tekstinis prisijungimas

Bazinė kategorija yra pasirenkama tekstinio slaptažodžio autentifikacijos būdas. Ši schema reikalauja atsiminimo pastangų (P1: 0). Esant dideliame paskyrų kiekiui, kiekvienos paskyros autentifikacijos duomenis kiekvieną kartą reikia prisiminti, kas stipriai apkrauna naudotojo mintines pastangas (P2: 0). Aišku, galima visur naudoti tuos pačius prisijungimo duomenis, bet tai yra saugumo spraga. Ši schema nereikalauja papildomų prietaisų (P3: 1). Žmonės ilgą laiką jau naudoja slaptažodžių autentifikacijos schemą, todėl vien iš patirties šis būdas yra lengvai išmokstamas (P4: 1). Slaptažodžiai yra efektyvūs naudojant, nes kelių simbolių įvedimas į tam skirtą laukelį neužtrunka ilgai (P5: 1). Paprastai slaptažodžiai yra lengvai atstatomi (P6: 1), tačiau tai reikalauja papildomų fizinių pastangų (P7:0).

Tekstiniai slaptažodžiai turi žemesnius saugumo įverčius, nes ši schema nėra atspari fiziniam stebėjimui (S1: 0). Net ir greitai įvedant slaptažodį, geros kokybės klaviatūros vaizdo įrašas jį parodys. Asmeninės informacijos žinojimas gali palengvinti atspėjimą (S1: 0). Slaptažodis paprastai nėra dažnai keičiamas, tai reiškia, jog jis nėra atsparus vidiniam stebėjimui (S3: 0). Slaptažodžiai nėra atsparūs sukčiavimui, nes dažnai pasitaiko, kad originalų web puslapį imituoja kenkėjiška programa ir slaptažodis yra lengvai perduodamas sukčiams (S4: 0). Visgi, kadangi slaptažodžiai nereikalauja išorinių objektų, jų neįmanoma pavogti (S5: 1).

#### 4.5.2 Grafiniai slaptažodžiai

Grafinės autentifikacijos schemas yra grindžiamos natūraliu žmogaus gebėjimu prisiminti vaizdus. Viena iš tokių schemas realizacijų yra PCCP (angl. *Persuasive Cued Click-Points*). Naudotojams yra pateikiami keli vaizdai vienas po kito, kuriuose jie pasirinka reikalingą tašką. Kai visur yra spaudžiamas teisingas taškas, autentifikacija pavyksta. Iš panaudojamumo perspektyvos, šis būdas yra nėra efektyvus nei atsimenamumo pastangų (P1: 0), nei tinkamumo dideliame paskyrų skaičiui (P2: 0) dėl didesnių atpažinimo pastangų. Šis schema nereikalauja išorinių prietaisų (P3: 1) ir yra lengvai išmokstama (P4: 1). Naudojimo efektyvumas yra blogesnis nei tekstinio slaptažodžio, nes autentifikacija užtrunka šiek tiek ilgiau (P5: 0,5). Reikalauja fizinių pastangų (P7:0).

Iš saugumo perspektyvos, ši schema nėra atspari stebėjimui, nes galima iš šono įrašyti naudotojo pasirinkimus (S1: 0). Tačiau ji atspari tikslingam apsimetinėjimui, nes asmeninės naudotojo informacijos žinojimas nepadedą atspėti slaptažodį (S2: 1). Kadangi statiniai slaptažodžių tipai yra pakartojami, todėl šis būdas nėra atsparus vidiniam stebėjimui (S3: 0). Šis būdas yra atsparus sukčiavimui, kadangi netikras puslapis turėtų gauti prieigą prie tikro įmonės serverio, kad galima būtų gauti naudotojo autentifikavimui reikalingus vaizdus (S4: 1). Schema atspari vagystei, nes joje išoriniai prietaisai nenaudojami (S5: 1).

#### 4.5.3 Telefonu grindžiamas prisijungimas

Autentifikuojant į naudotojo telefoną siunčiama žinutė su slaptažodžiu, kurį naudotojas įveda prisijungdamas. Ši schema yra pranašesnė už tekstinę autentifikaciją, nes nereikalauja atsiminti slaptažodžio (P1: 1), taip pat ir dideliame paskyrų skaičiui (P2: 1). Papildomas prietaisas – telefonas – naudojamas, tačiau tai nesukelia nepatogumo, kadangi įprasta jį su savimi turėti (P3: 0,5). Schema lengvai išmokstama (P4: 1). Naudojimo efektyvumas ženkliai sumažėja (P5: 0), nes reikalauja daugiau veiksmų ir turi būti perrašytas iš mobiliojo telefono, todėl reikalauja fizinių pastangų (P7:0).

Ši schema yra pakankamai saugi. Kadangi žinutėje siunčiamas vienkartinis slaptažodis, schema atspari fiziniam stebėjimui (S1: 1), tikslingam apsimetinėjimui (S2: 1) ir sukčiavimui (S4: 1). Schema nėra visiškai atspari vidiniam stebėjimui, nes SMS žinutė arba telefono skambutis gali būti perimtas kenkėjų, todėl yra priskiriamas dalinis atsparumas (S3: 0,5). Tačiau telefono vagystė leistų apsimesti kitu asmeniu (S5: 0).

#### 4.5.4 Biometrinė autentifikacija

Biometrinės prisijungimo schemas sudaro akies vyzdžio, piršto antspaudo arba balso validacija. Šios schemas nereikalauja atsiminimo (P1: 1), tinka dideliame paskyrų skaičiui (P2: 1) ir lengvai iš-

mokstamos (P4: 1). Pačiam slaptažodžiui nereikalingas išorinis įrenginys (P3: 1). Tačiau ne visi įrenginiai turi skenuojančią įrangą, todėl naudojimas yra ribotas. Naudojimo efektyvumą įtakoja nemaža klaidų tikimybė, kai nepavyksta nuskaityti duomenis (P5: 0,5). Fizinės pastangos apsiriboja žmogui natūraliais judesiais (P7: 0,5)

Saugumo požiūriu, biometrinė autentifikacija yra atspari fiziniam stebėjimui (S1: 1). Tačiau neatspari tikslingam apsimetinėjimui (S2: 0), vidiniam stebėjimui (S3: 0), sukčiavimui (S4: 0) ir vagystei (S5: 0), kadangi balsas gali būti įrašytas, piršto atspaudas nuimtas nuo prisiliesto paviršiaus specialiomis priemonėmis, veidas su akių rainelėmis nufotografuotas.

#### 4.5.5 Paslaugos variantų pasirinkimas

11 lentelėje apibendrinti autentifikacijos būdai pasižymi skirtingomis nefunkcinėmis savybėmis. Kuriant adaptuojamą autentifikavimo paslaugą, konkrečiam naudojimo kontekstui, tai yra konkrečioms naudotojų poreikiams, veikloms, veikimo aplinkai ir naudojamoms technologijoms gali būti siūlomas atitinkamas paslaugos variantas.

11 lentelė. Prisijungimo schemų nefunkcinių savybių vertinimas

Kategorija	Schema	Panaudojamumas							Saugumas				
		P1	P2	P3	P4	P5	P6	P7	S1	S2	S3	S4	S5
Bazinė	Tekstinė			1	1	1	1			0,5			1
Grafinis	PCCP			1	1	0,5	1			1		1	1
Grindžiamas telefonu	OTP per SMS	1	1	0,5	1		0,5		1	1	0,5	1	0,5
Biometrinis	Piršto antspaudas	1	1	1	1	0,5		0,5	1				
	Akies vyzdys	1	1	1	1	0,5		0,5	1				
	Balsas	1	1	1	1	0,5		0,5	1				

#### 4.6. Kintamybių konkretizavimas

Paslaugos kontrakte yra apibrėžti trys paslaugos teikimo lygmenys, todėl yra sudaromos trys taisyklės (12 lentelė, 13 lentelė, 14 lentelė).

12 lentelė. Pirmoji paslaugos lygmens „Patogus“ taisyklė

Šablono pavadinimas	Variacinis Taškas1
Taikymo sąlyga	Jeigu yra variacinio taško tipo nekonkretizuota „Pateikti slaptažodį“ veikla ir atsižvelgiama į panaudojamumo ir saugumo nefunkcines savybes.
Sprendinys	Proceso variantai yra atrenkami atsižvelgiant į paslaugos lygmenį: aukštą panaudojamumą ir žemą saugumą.
Rezultatas	Paslaugos tiekimo procesas su galimais veiklos biometriniais variantais: akies vyzdys, piršto antspaudas, balsas.

Pirmoje taisyklėje (12 lentelė) rezultatas skaičiuojamas šiuo būdu: biometrinių variantų panaudojamumo charakteristikų įverčių (11 lentelė) suma yra 5 (iš 7) arba 71%, todėl šie variantai priskiriami aukštam panaudojamumo lygiui. Biometrinių variantų saugumo charakteristikų įverčių suma yra 1 (iš 5) arba 20%, kas priskiriama žemam saugumo lygiui.

13 lentelė. Antra paslaugos lygmens „Vidutiniškas“ taisyklė

Šablono pavadinimas	Variacinis Taškas1
Taikymo sąlyga	Jeigu yra variacinio taško tipo nekonkretizuota „Pateikti slaptažodį“ veikla ir atsižvelgiama į panaudojamumo ir saugumo nefunkcines savybes.
Sprendinys	Proceso variantai yra atrenkami atsižvelgiant į paslaugos lygmenį: vidutinį panaudojamumą ir vidutinį saugumą.
Rezultatas	Paslaugos tiekimo procesas su galimais veiklos variantais: tekstinis, PCCP

Antros taisyklės (13 lentelė) rezultatas yra gautas sumuojant tekstinio varianto panaudojamumo įverčius. Jų suma sudaro 4 (iš 7) arba 57%, kas atitinka vidutinį panaudojamumo lygį. Tekstinio varianto saugumo įverčių suma yra 1,5 (iš 5) arba 30%, kas atitinka vidutinį saugumo lygį. Grafinio varianto panaudojamumo charakteristikų įverčių suma yra 3,5 (iš 7) arba 50%, kas atitinka vidutinį panaudojamumo lygį. Sumuojant grafinio varianto saugumo charakteristikų įverčius, gaunamas 3 iš 5 suminis įvertis, tai yra 60% ir patenka į vidutinį saugumo lygį.

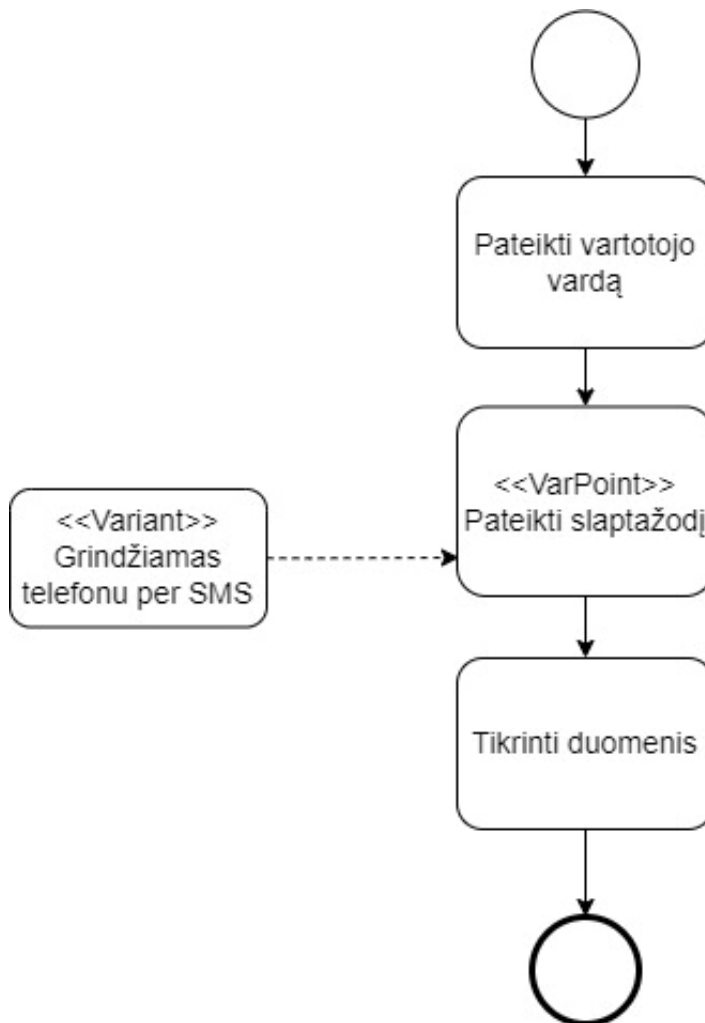
14 lentelė. Trečioji paslaugos lygmens „Saugus“ taisyklė

Šablono pavadinimas	Variacinis Taškas1
Taikymo sąlyga	Yra variacinio taško tipo nekonkretizuota „Pateikti slaptažodį“ veikla ir yra atsižvelgiama į panaudojamumo ir saugumo nefunkcines savybes.
Sprendinys	Proceso variantai yra atrenkami atsižvelgiant į paslaugos lygmenį: vidutinį panaudojamumą ir aukštą saugumą.

Rezultatas	Paslaugos tiekimo procesas su galimus veiklos variantu: grindžiamas telefonu variantas: OTP per SMS
------------	-----------------------------------------------------------------------------------------------------

Trečios taisyklės (14 lentelė) rezultatas gautas analogiškai. Grindžiamo telefonu varianto panaudojamumo įverčių suma yra 4 (iš 7) arba 57% ir atitinka vidutinį panaudojamumo lygį. Šio varianto saugumo įverčių suma sudaro 4 (iš 5) arba 80% ir atitinka aukštą saugumo lygį.

Vadovaujantis proceso bazinių modelių, proceso variantais (dekompozicija) ir tipinėmis taisyklėmis formuojamas konkretus paslaugos teikimo procesas. Jeigu jame yra alternatyvų, tada galutinės konkrečios paslaugos specifikaciją konkretizuoja vartotojas (23 pav.).



23 pav. Galutinis paslaugos teikimo procesas

## Rezultatai ir išvados

Šiame darbe sukurtas paslaugomis grindžiamų sistemų kintamumo specifikuojimo metodas, kuriame paslaugos variantai yra nustatomi analizuojant paslaugos nefunkcines savybes ir sudaromas individualizuotas paslaugos teikimo procesas.

Darbo rezultatai:

1. Apžvelgtos paslaugomis grindžiamų sistemų kintamumo modeliavimo ir kokybės atributų kintamumo specifikuojimo metodikos.
2. Paslaugomis grindžiamų sistemų kintamumo specifikuojimo metodas, kuriame paslaugos variantų specifikuojimas ir modeliavimas atsižvelgiant į jos nefunkcines charakteristikas.
3. Metodo taikymą iliustruojantis pavyzdys.
4. Paruoštas straipsnis, kuris buvo pristatytas „Lietuvos magistrantų informatikos ir IT tyrimai“ konferencijoje 2022 m. gegužės 16 dieną (1 priedas).

Darbe yra daromos šios išvados:

1. Atlikus literatūros analizę nustatyta, kad pagrindinis dėmesys atliktuose turimuose skiriamas funkcionalumo kintamumui (pvz., DiVA metodikoje modeliuojami tik funkcinių savybių kintami turiniai). BP&SLA metodikoje specifikuojami paslaugomis grindžiamų sistemų procesai, apimant ir nefunkcinius reikalavimus, bet nekalbama apie kintamus turinius.
2. Sukūrus paslaugomis grindžiamų sistemų paslaugų variantų kintamumo specifikuojimo metodą, nustatyta, kad proceso variantus galima konfigūruoti nefunkcines savybes nusakančių charakteristikų pagrindu ir naudojant tipines taisykles.
3. Taikant produktų šeimos idėjas ir tipines taisykles, konkrečiai paslaugai adaptuotas turinys, t. y., konkretaus proceso modelis, gali būti generuojamas automatizuotai.
4. Nustatyta, kad kreipiant pagrindinį dėmesį į kintamybių modeliavimą, nuo skirtingų kontekstų ir nuo skirtingų paslaugos savybių priklausantis variantų skaičiaus padidėjimas gali būti valdomas kontroliuojamu būdu.
5. Taikant metodą nustatyta, kad paslaugą realizuojančio proceso varianto konkretizavimas negali būti visiškai automatinis ir galutiniam sprendimui priimti gali būti reikalingas naudotojo įsikišimas.

## Šaltiniai

- [AG04] Aiello M., Giorgini P. *Applying the Tropos methodology for analysing web services requirements and reasoning about qualities of services*. Technical report DIT-04-034, University of Trento, 2004. 17 p.
- [Ars04] Arsanjani A. *Service-oriented modelling and architecture*, SOA and Web services Center of Excellence, IBM, 2004.
- [AFF+16] Alebrahim A., S. Faßbender, M. Filipczyk, M. Goedicke, M. Heisel, and U. Zdun. *Variability for Qualities in Software Architecture*. SIGSOFT Softw. Eng. Notes, 41, 1, 2016, pp. 32–35. DOI: <https://doi.org/10.1145/2853073.2853095>
- [BBC+10] Benbernou S., Brandic I., Cappiello C., Carro M., Comuzzi M., Kertesz A., Kritikos K., Parkin M., Pernic B., Plebani P. I. Modeling and negotiating service quality. In: Papazoglou, M., Pohl, K., Parkin, M., Metzger, A. (eds.) *Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems*, LNCS 6500, Berlin: Springer-Verlag, 2010, pp. 157–209.
- [BCD+09] Bucchiarone A., Cappiello C., Di Nitto E., Kazhamiakin R., Mazza V., Pistore M. Design for adaptation of service-based applications: main issues and requirements. In: *Proceedings of 5th International Workshop on Engineering Service-Oriented Applications: Supporting Software Service Development Lifecycles [WESOA]* Berlin: Springer-Verlag, 2009, pp. 467–476.
- [BHO+12] Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012, May). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. 2012 IEEE Symposium on Security and Privacy, pp. 553–567
- [BR10] Brocke J. von, Rosemann M. *Handbook on Business Process Management*, Springer, Heidelberg, Vol. 1, 2010.
- [BYY+18] Botangen K.A., J. Yu, S. Yongchareon, L.H. Yang, and Q. Bai. Specifying and reasoning about contextual preferences in the goal-oriented requirements modelling. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW '18)*. ACM, New York, USA, Art. 47, 2018, pp. 1–10. <https://doi.org/10.1145/3167918.3167945>
- [CK07] Chang S.H., S.D. Kim. A Variability Modeling Method for Adaptable Services in Service-Oriented Computing. In *Proceedings of the 11th Internationale Product Line Conference*, IEEE Computer Society, 2007, p. 261–266.

- [Erl05] Erl T., *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall, Upper Saddle River, NJ, 2005.
- [FMS07] Frankova G., F. Massacci & M. Seguran. *From Early Requirements Analysis towards Secure Workflows*. Technical report, University of Trento, 2007. [http://eprints.biblio.unitn.it/1220/1/SecureBPEL\\_FrankovaMassacciSeguran.pdf](http://eprints.biblio.unitn.it/1220/1/SecureBPEL_FrankovaMassacciSeguran.pdf)
- [FSG+11] Frankova, G., Séguran, M., Gilcher, F., Trabelsi, S., Dörflinger, J., & Aiello, M. Deriving business processes with service level agreements from early requirements. *Journal of Systems and Software*, 84(8), 2011, pp. 1351–1363.
- [FYS07] Frankova, G., Yautsiukhin, A., & Seguran, M. *From early requirements to business processes with service level agreements*. University of Trento, 2007.
- [GA13] Galster M., P. Avgeriou. Variability in Web Services. In: *Systems and Software Variability Management*, R. Capilla, J. Bosch, and K. Kang, eds., Springer Verlag, 2013, pp. 269–278,
- [GCA+11] Greenwood, P., Chitchyan, R., Ayed, D., Girard-Reydet, V., Fleurey, F., Dehlen, V., & Solberg, A. (2011). Modelling service requirements variability: The DiVA way. In *Service Engineering*, Springer, Vienna, pp. 55–84.
- [GL09] Gu Q., P. Lago, Exploring service-oriented system engineering challenges: a systematic literature review. *Service Oriented Computing Applications*, vol. 3, no.3, 2009, 171–188.
- [GMS05] Giorgini P., Mylopoulos J., Sebastiani R. Goal-oriented requirements analysis and reasoning in the Tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2), 2005, pp. 159–171.
- [GWT+14] M. Galster, D. Weyns, D. Tofan, B. Michalik, and P. Avgeriou. Variability in Software Systems – a Systematic Literature Review, *IEEE Transactions On Software Engineering*, vol. 40, no. 3, March 2014, pp. 282–306.
- [GZW+17] M. Galster, U. Zdun, D. Weyns, R. Rabiser, B. Zhang, M. Goedicke, and G. Perrouin. Variability and Complexity in Software Design: Towards a Research Agenda. *SIGSOFT Software Engineering Notes* 41, 6, 2017, pp. 27–30.  
DOI: <https://doi.org/10.1145/3011286.3011291>
- [HP04] Halmans G, Pohl K. Communicating the variability of a software product family to customers. *Inform – Forsch und Entwicklung* 18, 2004, pp. 113–131.
- [HW11] Harmon P., Wolf C. Business process modeling survey: December 2011. [http://www.bptrends.com/members\\_surveys/deliver.cfm?report\\_id=1005&target=Process\\_Modeling\\_Survey-Dec\\_11\\_FINAL.pdf](http://www.bptrends.com/members_surveys/deliver.cfm?report_id=1005&target=Process_Modeling_Survey-Dec_11_FINAL.pdf)



- [JRS05] J. Jiang, A. Ruokonen, T. Systa: Pattern-based variability management in web service development. In: Third European Conference on Web Services, Sweden, IEEE Computer Society, 2005, p. 83–94.
- [KBB+10] Kazhamiakin R., Benbernou S., Baresi L., Plebani P., Uhlig M., Barais O. Adaptation of service-based systems. In: Papazoglou, M., Pohl, K., Parkin, M., Metzger, A. (eds.) Service Research Challenges and Solutions for the Future Internet: S-Cube – Towards Engineering, Managing and Adapting Service-Based Systems, LNCS 6500, Berlin: Springer-Verlag, 2010, pp. 89–128.
- [KSF+15] Kurz Matthias, Schmidt Werner, Fleischmann Albert, Lederer Matthias. Leveraging CMMN for ACM: examining the applicability of a new OMG standard for adaptive case management. S-BPM ONE '15, 2015.
- [KSS09] M. Koning, C. Sun, M. Sinnema, P. Avgeriou. VxBPEL: supporting variability for Web services in BPEL. *Information and Software Technology*, 51, 2009, pp. 258–269.
- [LK06] List, B., & Korherr, B. An evaluation of conceptual business process modelling languages. *Proc. of the 2006 ACM Symposium on Applied Computing*, 2006, pp. 1532–1539.
- [LMC06] Lupeikiene A., Miliauskaite J., Caplinskas A. A view-based approach to quality of service modelling in service-oriented enterprise systems. In: Kirikova M., Grabis J. (eds.) Proceedings of the 2nd International Business and Systems Conference [BSC 2013], 5 November, Riga, Latvia. Riga Technical University, 2013, pp. 7–19.
- [LMC13] A. Lupeikienė, J. Miliauskaitė, A. Čaplinskas, A Model of View-Based Enterprise Business Service Quality Evaluation Framework. *INFORMATICA*, 2013, pp. 543–560.
- [LS11] Lane S., Richardson I. Process models for service-based applications: a systematic literature review. *Journal Information & Software Technology*, 53(5), 2011, pp. 424–439.
- [Lup07] Lupeikienė Audronė. *Teoriniai ir technologiniai informacinių sistemų aspektai: vadovėlis informatikos krypties doktorantams ir magistrams*, Vilnius: MII, 2007. 200 p.
- [MAL09] F.M. Medeiros, E.S. de Almeida, S.R. de Lemos Meira: Towards an approach for service-oriented product line architectures. In: Workshop on Service-oriented Architectures and Software Product Lines, SEI, San Francisco, CA, 2009, pp. 1–7.
- [MGA13] S. Mahdavi-Hezavehi, M. Galster, and P. Avgeriou: Variability in Quality Attributes of Service-Based Software Systems: A Systematic Literature Review. *Information and Software Technology*, vol. 55, no. 2, 2013, pp. 320–343.

- [Mil15] F. Milani, On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes and Variation, University of Tartu, 2015.
- [NP10] N. C. Narendra, K. Ponnalagu. Towards a Variability Model for SOA-based Solutions. IEEE International Conference on Services Computing, SCC 2010, Miami, Florida, USA, July 5-10, 2010. IEEE Computer Society, 2010, pp. 562–569.
- [OMG14] Object Management Group. Case management model and notation (CMMN). formal/2014-05-05. <http://www.omg.org/spec/CMMN/1.0>, 2014
- [OMG20] Object Management Group. Case management model and notation (CMMN). <http://www.omg.org/spec/CMMN/1.1>, 2020.
- [Per10] Pernici, B. Methodologies for design of service-based systems. In *Intentional perspectives on information systems engineering*, Springer, Berlin, Heidelberg, 2010, pp. 307–318.
- [PKB14] Pistore M., Kazhamiakin R., Bucchiarone A. (eds.) Integration framework baseline. S-Cube technical report CD-IA-3.1.1, 2009. 43p. [http://www.s-cube-network.eu/results/deliverables/wp-ia-3.1/CD-IA-3.1.1\\_Integration%20Framework%20Baseline.pdf](http://www.s-cube-network.eu/results/deliverables/wp-ia-3.1/CD-IA-3.1.1_Integration%20Framework%20Baseline.pdf)
- [PPS+06] Penserini L., Perini A., Sus A., Mylopoulos J. From stakeholder needs to service requirements. In: Proceedings of the Service-Oriented Computing: Consequences for Engineering Requirements [SOCCER'06], 12 September, Minneapolis, Minnesota, USA. IEEE Computer Society, 2006, pp. 1–10.
- [Ra09] Rayson, P. *Wmatrix: a web-based corpus processing environment*. Citeseer, 2009.
- [RA07] Rosemann M, Aalst W van der. A configurable reference modelling language. *Information Systems*, 2007, vol. 32, pp. 1–23
- [RAD+17] Rosa, M. L., Aalst, W. M. V. D., Dumas, M., & Milani, F. P. Business process variability modeling: A survey. *ACM Computing Surveys (CSUR)*, 50(1), 2017, pp. 1–45.
- [RKS10] Rolland C., Kirsch-Pinheiro M., Souveyet C. An intentional approach to service engineering. *IEEE Transactions on Services Computing Journal*, 3(4), 2010, pp. 292–305.
- [RMS06] Reference Model for Service Oriented Architecture 1.0 OASIS Standard, 12 October 2006
- [Sol82] Sol, H.G. (1982). Simulation in Information Systems Development. Doctoral Dissertation. University of Groningen, The Netherlands.
- [SRC+07] Sampaio, A., Rashid, A., Chitchyan, R., & Rayson, P. (2007). EA-Miner: towards automation in aspect-oriented requirements engineering. In *Transactions on aspect-oriented software development III*, Springer, Berlin, Heidelberg, pp. 4–39.

- [SRS+10] Sun C., R. Rossingb, M. Sinnemab, P.Bulanovb, M. Aiello, Modeling and Managing the Variability of Web-Service-Based Systems, *Journal of Systems and Software*, vol. 83, no. 3, Mar. 2010, pp. 502–516.
- [Tou03] Toulmin S.E. (2003). The uses of argument. Cambridge University Press, 247 p.
- [WL99] Weiss, David M. and Lai, Chi Tau Robert. *Software product-line engineering: a family-based software development process*. Addison-Wesley, 1999.

# Priedai

## 1 priedas. Darbo pagrindu sukurtas straipsnis

Lapin V. (2022) "Nefunkcinių reikalavimų kintamumas paslaugų architektūros stiliaus sistemoje Vieslav Lapin", *Vilnius University Open Series*, pp. 54-66. doi: [10.15388/LMITT.2022.6](https://doi.org/10.15388/LMITT.2022.6).

## Nefunkcinių reikalavimų kintamumas paslaugų architektūros stiliaus sistemoje

### Vieslav Lapin

Vilniaus universitetas, Matematikos ir informatikos fakultetas,  
Didlaukio g. 47, LT-08303 Vilnius  
[vieslav.lapin@mif.stud.vu.lt](mailto:vieslav.lapin@mif.stud.vu.lt)

---

**Santrauka.** Šiame straipsnyje aptariamas kintamų nefunkcinių reikalavimų specifikavimas ir modeliavimas kuriant paslaugų architektūros stiliaus programų sistemas. Paslaugų sistemų kūrime šiai problemai spręsti dar nėra skiriama pakankamai dėmesio. Pasiūlytas kintamų savybių specifikavimo 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 specifikavimas.

---

### 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 kintamomis 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 modeliui ir variacijų taškams aprašyti [5]. Paslaugų kokybės atributų kintamumo modeliavimas tik pradedamas 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 kūrimo 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 kompleksškumo kontrolės būdą [16]. Todėl nekontroliuojamas konfigūracijų skaičiaus augimas yra suvaldomas, modeliuojant 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 funkcinių savybių konfigūracijas paslaugos gyvavimo cikle [16]. Ši metodika neapima paslaugas realizuojančio proceso automatizuoto specifikavimo, nefunkcinių savybių specifikavimo ir modeliavimo.

### **3 Paslaugų ir procesų kintamų nefunkcinių savybių specifikavimas**

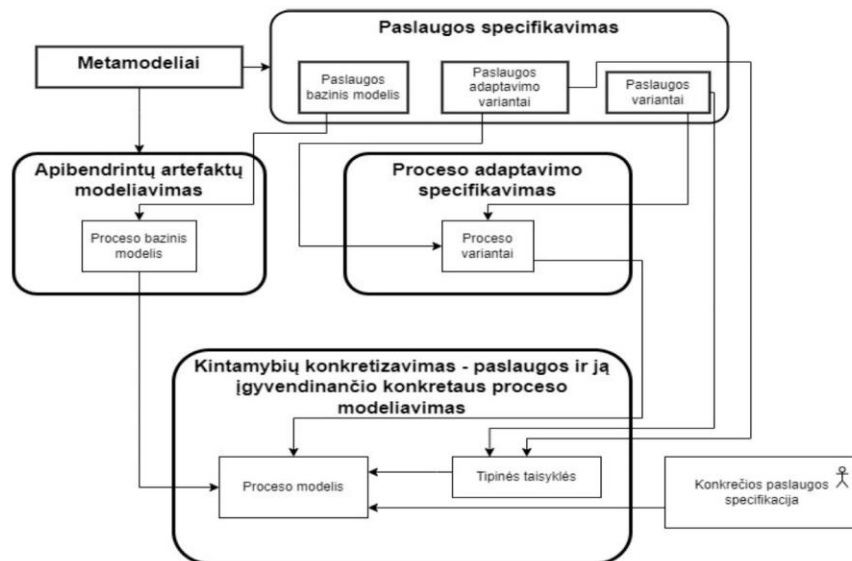
Paprastai yra kuriamos ne atskiros paslaugos, o paslaugų šeimos, skirtos individualizuoti paslaugų teikimą, kai paslaugos adaptuojamos specifiniam naudojimo kontekstui. Naudojimo kontekstas apima naudotojus, jų tikslus ir veiklas, veikimo aplinką ir technologijas [17]. Verslo procesus įtakoja suteikiamos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

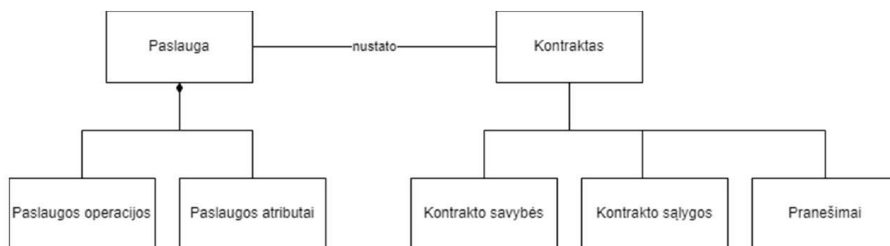
Paslauga suprantama kaip savybių ir funkcionalumo visuma (atributai ir operacijos); paslauga teikiama pagal kontrakte nustatytas sąlygas (2 pav.). Operacijos yra veiksmai susiję su teikiamais paslaugos variantais. Paslaugos atributai apibrėžia esybes ir jų savybes, kuriomis manipuliuoja operacijos. Paslaugos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

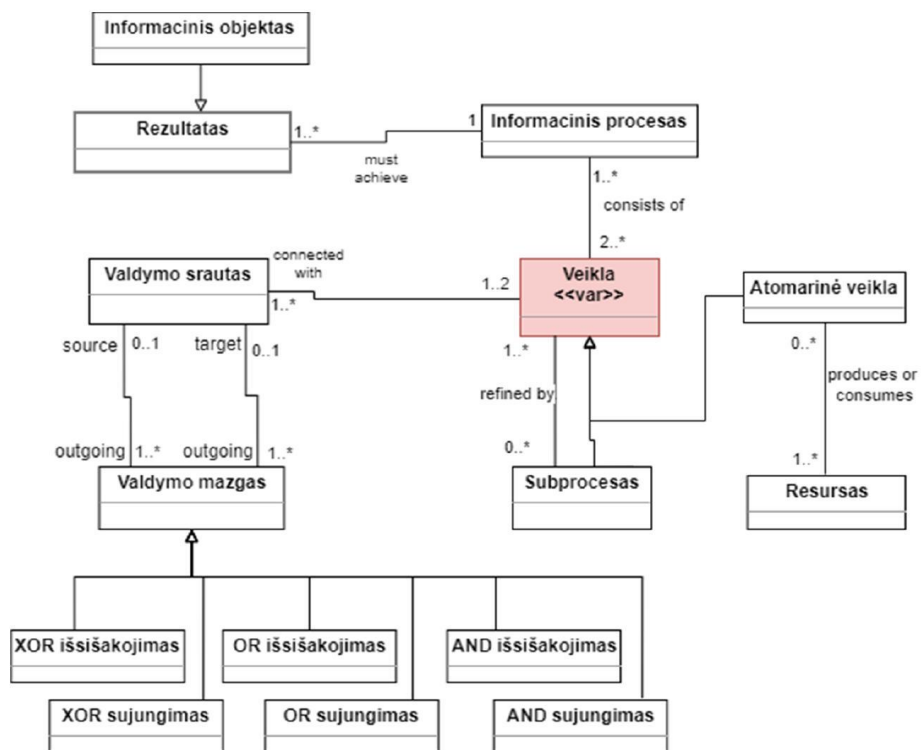
- 1) informacinis procesas,
- 2) veiklos (kintamos ir nekintamos),
- 3) atominė veikla,
- 4) subprocesas.

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 atominė, rečiau dar iš kitų subprocesų. Iš paslaugomis grindžiamos sistemos perspektyvos, atominėje veikloje yra apdorojami informaciniai ištekliai. Veiklų eiliškumą nusako 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 vykdymą ir visi srautai susilieja atgal į vieną srautą.
5. AND išsišakojimas – proceso šakos vykdomos lygiagrečiai.
6. AND sujungimas – proceso šakos baigia vykdymą 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 specifikuojama 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 išreikštiniu 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 ypatybės 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 matavimus. Vartotojas pasirenka atitinkantį jo poreikius paslaugos teikimo kokybės lygmenį. Paslaugų lygmenų kriterijai yra apibrėžiami, analizuojant naudojimo konteksto scenarijus (2 lentelė).

**1 lentelė.** Paslaugos specifikacija

<b>Paslaugos operacijos ir atributai</b>	<b>Apibrėžimas</b>
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

<b>Paslaugų adaptavimo variantai</b>	<b>Paslaugos lygmuo 1</b>	<b>Paslaugos lygmuo 2</b>	<b>...</b>	<b>Paslaugos lygmuo N</b>
Nefunkcinė charakteristika 1	Kriterijus <sub>11</sub>	Kriterijus <sub>21</sub>	...	Kriterijus <sub>N1</sub>
Nefunkcinis charakteristika 2	Kriterijus <sub>12</sub>	Kriterijus <sub>22</sub>	...	Kriterijus <sub>N2</sub>

### 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, įtakojančios paslaugos varianto pasirinkimą. Charakteristikoms yra priskiriamos vertės, gaunamos analizuojant paslaugos naudojimo kontekstą (3 lentelė).

**3 lentelė.** Adaptavimo kriterijų (nefunkcinių savybių) dekompozicija

	NF <sub>1</sub>			NF <sub>2</sub>			NF <sub>M</sub>		
Paslaugos variantai	Ch <sub>11</sub>	...	Ch <sub>1N</sub>	Ch <sub>21</sub>	...	Ch <sub>2N</sub>	Ch <sub>M1</sub>	...	Ch <sub>MN</sub>
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	Variacinis Taškas2
Taikymo sąlyga	Jeigu proceso variaciniame taške (<<VarPoint>>) turi būti sukonkrečinta 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 funkcinų savybių kintami turiniai ir specifikuojami 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*.