

**KAUNO TECHNOLOGIJOS
UNIVERSITETAS**

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Andrius Klimavičius

Įmonių sąveikumo procesų analizė ir sprendimas

Magistro darbas

Darbo vadovė: lekt. dr. Audronė Janavičiūtė

Kaunas, 2009

**KAUNO TECHNOLOGIJOS
UNIVERSITETAS**

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

TVIRTINU
Katedros vedėjas:

prof. dr. Egidijus Kazanavičius
2009 01

Įmonių sąveikumo procesų analizė ir sprendimas

Informacinių sistemų inžinerijos magistro baigiamasis darbas

Recenzentas

prof. dr. Lina Nemuraitė
2009 01

Vadovė

lekt. dr. Audronė Janavičiūtė
2009 01

Atliko

IFM-3/4 gr. stud. Andrius Klimavičius
2009 01

Kaunas, 2009

Turinys

1.IVADAS	6
1.1.Tyrimo sritis, objektas ir problema.....	6
1.2.Tyrimo tikslas ir uždaviniai.....	7
1.3.Analizės metodai.....	7
2.ĮMONIŲ SĄVEIKUMO PROCESŲ ANALIZĖ	8
2.1.Tyrimo objekto analizė.....	8
2.2.Problemos analizė.....	9
2.2.1Stagių pasikeitimų/inovacijų valdymas.....	10
2.2.2Prisitaikymas prie globalizacijos.....	10
2.2.3Dideli integracijos/sąveikavimo kaštai.....	11
2.2.4Sunkumai sprendimų priėmimo.....	11
2.2.5Argumentų trūkumas įmonių sąveikumui versle.....	12
2.2.6Atvirioji inovacija.....	12
2.3.Esamų įmonių sąveikumo sprendimų analizė.....	13
2.3.1Paslaugoms orientuota architektūra.....	13
2.3.2XML grindžiami sprendimai sąveikumui.....	15
2.3.3 SOAP.....	15
2.3.4BizTalk.....	16
2.3.5Portalai.....	17
2.4.El. verslas ir vertės grandinė.....	18
2.5.Dokumentų standartizavimas.....	19
2.5.1Elektroninių dokumentų mainai.....	20
2.5.2XML.....	21
2.5.3Elektroninio verslo XML.....	21
2.5.4Universali verslo kalba.....	22
2.5.5Bendroji verslo XML biblioteka.....	23
2.5.6Komercijos XML.....	23
2.6.Užsakymo proceso analizė.....	24
2.6.1Žodiniai užsakymai.....	25
2.6.2Komerciniai užsakymai.....	25
2.6.3Užsakymo operacijos.....	25
2.6.4Užsakymo dokumentų tipai.....	26
2.7.Užsakymas įmonės viduje.....	27
2.8.Analizės išvados.....	29
3.UŽSAKYMŲ APTARNAVIMO KOMPONENTO REIKALAVIMŲ ANALIZĖ IR SPECIFIKACIJA	30
3.1.Techninis pasiūlymas.....	30
3.1.1Sistemos apibrėžimas.....	30
3.1.2Bendras veiklos tikslas.....	30
3.1.3Sistemos pagrįstumas.....	30
3.1.4Prototipai ir pagalbinių informacija.....	31
3.1.5Siūlomas sprendimas.....	31
3.2.Galimybių analizė.....	33

3.2.1	Sistemos aktualumas.....	33
3.2.2	Komercine nauda.....	33
3.2.3	Techninės galimybės.....	34
3.2.4	Vartotojų pasiruošimo analizė.....	35
3.3	Duomenų analizė.....	35
3.3.1	UBL dokumentų duomenų tipai.....	35
3.3.2	UBL-Oder dokumento analizė.....	35
3.3.3	Dalykinės srities modelis.....	39
3.4	Reikalavimų specifikacija.....	39
3.4.1	Reikalavimai įmonės duomenų bazei.....	39
3.4.2	Sistemos funkcijos.....	40
3.4.3	Kiti funkciniai reikalavimai.....	44
3.4.4	Apribojimai ir nefunkciniai reikalavimai.....	45
3.4.5	Duomenų ir rezultatų specifikacija.....	46
3.4.6	Vartotojui (pirkėjui) keliami reikalavimai.....	46
3.5	Techninė specifikacija.....	47
4.	UŽSAKYMO APTARNAVIMO KOMPONENTO PROJEKTAS.....	48
4.1	Duomenų bazės modelis.....	48
4.2	XSL transformacijos.....	50
4.2.1	Vardų erdvės.....	50
4.2.2	Datų formatai.....	51
4.2.3	Transformuotų UBL dokumentų struktūra.....	51
4.2.4	XSL transformacijų šablonai įmonės duomenų bazei.....	55
4.3	Komponento sąsajos aprašymas.....	58
4.4	Sistemos architektūra.....	60
4.5	Klasių diagramos.....	60
4.6	Dinaminė sistemos elgsena.....	64
4.7	Realizacijos modelis.....	70
4.8	Komponento nustatymų aprašymas.....	71
4.9	Diegimo instrukcijos.....	72
5.	UŽSAKYMO APTARNAVIMO KOMPONENTO TYRIMAS.....	73
5.1	Testavimo modelis.....	73
5.2	Pradiniai duomenys.....	75
5.3	Rezultatai ir išvados.....	76
6.	IŠVADOS.....	77
7.	LITERATŪRA.....	78
8.	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	81
9.	PRIEDAI.....	83
9.1	IT pritaikymas įmonės vertės grandinėje.....	83
9.2	UBL dokumentų duomenų tipai aprašai.....	85
9.3	Užsakymų aptarnavimo komponento sąsajos WSDL.....	89
9.4	UBL dokumentai naudojami testavimui.....	93
9.5	Sugeneruotos SQL užklauskos.....	99

1. Įvadas

Įmonės, norėdamos įgyvendinti savo tikslus, turi bendradarbiauti tarpusavyje. Kai kurioms įmonėms veikla globalioje rinkoje tapo gyvybiškai svarbiu faktoriumi jų išlikime, o kai kurios iš jų atranda naujų galimybių sukongcentruodamos savo veiklą vietinėje rinkoje. Dažnai įmonėms, tiek didelėms, tiek ir mažoms, reikia užmegzti bendradarbiavimo ryšius. Mažo ir vidutinio dydžio įmonės, kurios specializuojasi pelningoje veiklos srityje, tam, kad padidintų jų pačių pridėtinę vertę, turi apjungti savo jėgas ir konkuruoti rinkoje drauge. Šiandien įmonių konkurencingumas yra didelio masto ir apima galimybę nenutrūkstamai sąveikauti su kitomis įmonėmis.

1.1. *Tyrimo sritis, objektas ir problema*

Tyrimo sritis apima:

- verslo procesus,
- žiniatinklio technologijas,
- paslaugoms orientuotą architektūrą,
- įmonių sąveikumą,
- duomenų integravimą.

Tyrimo objektas yra įmonių sąveikumo procesai.

Per pastarąjį dešimtmetį įmonių sąveikumas žymiai patobulėjo, ypač informacinių ir telekomunikacijų technologijų dėka. Pavyzdžiui, internetas palengvino informacijos paiešką ir duomenų mainus tarp įmonių. Taip pat keletas programinės įrangos gamintojų (IBM, Microsoft, Oracle, SAP) pradėjo dominuoti įmonių programinės įrangos rinkoje, teikdami įmonių funkcijų integracijos ir efektyvumo didinimo paslaugas. Tačiau kyla nemažai klausimų, susijusių su šių, gamintojų pagrįstais sprendimais [1]. Išlieka atskirtumas tarp įmonių sąveikumo, nes integravimo procesai vis dar yra sudėtingi ir brangūs. Dabartinės žiniatinklio technologijos nėra nei pakankamai išstobulintos tam, kad palaikytų įmonių sąveikumą, nei yra plačiai naudojamos vidiniuose įmonės procesuose.

Esamų technologijų tyrimas turėtų padėti sukurti sprendimą įmonei, norinčiai įgalinti tam tikro verslo proceso sąveikumą.

1.2. Tyrimo tikslas ir uždaviniai

Ištirti įmonės sąveikumo procesus ir pasiūlyti, bei pritaikyti įmonių sąveikumo sprendimą.

Uždaviniai:

1. Išsiaiškinti įmonių sąveikumą ir jo problemas;
2. Išanalizuoti esamus įmonių sąveikumo sprendimus;
3. Pasiūlyti sąveikumo sprendimą, pasirinktam verslo procesui;
4. Pritaikyti konkrečiai įmonei sukurtą sprendimą.

1.3. Analizės metodai

Analizei atlikti bus naudojamas mokslinės literatūros analizės ir apibendrinimo metodas. Šis metodas pasirinktas, todėl, kad problema yra labai aktuali, o ją sprendžiant yra rašoma daug straipsnių, rengiamos konferencijos. Europos Komisija parengė gaires [1], kuriose aprašomi sąveikumo problemos galimi sprendimo būdai ir kryptys. Viso to rezultatas yra didelė sutelkta žinių bazė. Mokslinių straipsnių įvairovė leidžia gerai įsigilinti į problemą. Juose naudojami įvairūs standartiniai žymėjimai (UML, BPMN), kurie yra suprantami ne tik šios srities ekspertams, bet ir mažiau kompetencijos turintiems žmonėms.

2. Įmonių sąveikumo procesų analizė

2.1. Tyrimo objekto analizė

Verslo procesas – tai rinkinys tarpusavyje susijusių užduočių, skirtų tam tikrai problemai spręsti [2]. Egzistuoja keturi verslo procesų tipai [3]:

- Produkto kūrimo procesai – produktų ir paslaugų kūrimo ir testavimo procesai;
- Pardavimų ir marketingo procesai – susiję su organizacijos veikla rinkoje, pardavimai, marketingas, ryšiai su klientais.
- Valdymo procesai – kurie daro įtaką sistemos veiklai. Tipiniai tokių procesų pavyzdžiai yra „Korporacinis valdymo būdas“ ir „Strateginis valdymas“.
- Tiekimo grandinės procesai – tai procesai susiję su žaliavų apsirūpinimu, produktų gamyba ir išplatinimu.
- Įgalintieji procesai – tokie procesai palaiko pagrindinę įmonės veiklą. Pavyzdžiui, apskaita, darbuotojų paieška, informacinių technologijų palaikymas.

Verslo procesų sąveikumas (VPS) įgalina bendradarbiaujančias įmonės dalintis procesais, tam, kad pasiektų bendrą tikslą ar sukurtų kitas paslaugas [4]. Kaip įprasta, VPS yra sutinkamas tada, kai procesas atitinka standartus, kurie jį įgalina pasiekti savo tikslą, nepaisant to, kam jis priklauso, jo vietos, versijos ar konstrukcijos naudojamoje kompiuterinėje sistemoje. Tai labai svarbus reikalavimas sąveikumui.

Norint patikrint, ar procesas tinkamas sąveikumui, įmonės turi nustatyti, ar įgyvendinant proceso tikslą galima atsisakyti žmogaus darbo jėgos. Tas konkretus tikslas turi būti aiškiai apibrėžtas nuo pradžios iki galo. Tačiau pradžia ir pabaiga yra labai subjektyvios. Vienoje organizacijoje procesas gali prasidėti, kai klientas užsako produktą, ir baigtis, kai produktas pristatomas klientui, o kitoje organizacijoje tas pats procesas gali prasidėti nuo produkto gamybos ir tęstis tol, kol baigiasi jo garantija.

Apibūdinant verslo procesų sąveikumą, kiekvienas procesas turi būti apsvarstomas nuo pradžių iki galo, įskaitant ir darbą, kuris atsiranda dėl programinės įrangos nesuderinamumo, tokių kaip duomenų surinkimas iš vienos sistemos ir pervedimas į kitą, ir ataskaitų paruošimas, į kurias įeina duomenys iš nesuderinamų sistemų. Procesas turi vykti nenutrūkstamai, nepaisant naudojamų jį palaikančių

kompiuterizuotų sistemų. Jei egzistuoja ne būtinas žmogaus darbas bet kuriame taške, verslo procesas nėra sąveikus.

Šiais laikais yra įprasta tarp įmonių užmegzti ryšius ir bendradarbiauti vienai su kita. Viena iš šių laikų tendencijų yra vis didėjantis bendradarbiavimas per visą produkto gyvavimo ciklą. Tai susiję su verslo varomuoju varikliu, panašiai kaip kaštų mažinimas, lankstumas, produkto naujoviškumas ir pan. Viso to rezultatas gali būti kas nors nuo gana stabilaus partnerių aljanso resursų tiekimo grandyje iki labiau laikino bendradarbiavimo virtualioje įmonėje. Susidaro verslo ekosistemos, kuriose organizacijos bendradarbiauja arba konkuruoja viską apimančiame karkase, kuris evoliucionuoja ir išlieka.

2.2. Problemos analizė

Pastarajame dešimtmetyje įvyko ženklių pasikeitimų įmonių sąveikavimo srityje. Pavyzdžiui, Internetas (ir naršyklės technologijos) palengvino informacijos paiešką, pasikeitimą informacija tarp organizacijų. Tai prisidėjo prie naujo verslo modelio sukūrimo įmonių bendradarbiavimui. Be to, keletas programinės įrangos kūrėjų (pvz., Microsoft, IBM, Oracle ir SAP) užėmė pirmaujančias pozicijas tarp įmonėms skirtos programinės įrangos (PI) kūrėjų. Šie kūrėjai prisidėjo prie organizacijų funkcijų integracijos ir efektyvumo augimo. Tačiau lieka atviri klausimai kokią įtaką įmonių sąveikumui šių gamintojų sukurta PI. Praėjus daugiau nei dešimčiai metų, po to kai įvairiose bendruomenėse buvo pradėta kalbėti apie įmonių sąveikumą, sąveikavimo problema organizacijose išlieka. Egzistuoja sąveikumo „salos“. Integracijos procesai yra sudėtingi ir brangūs. Dažnai ši problema nėra aiškiai matoma patiems sąveikumo naudotojams, ypač vidutinio ir mažo dydžio įmonėms.

Siekiant charakterizuoti įmonių sąveikumo probleminę sritį, galima išskirti šiuos svarbius aspektus [1]:

- Staigių pasikeitimų/inovacijų valdymas.
- Prisitaikymas prie globalizacijos.
- Dideli integracijos/sąveikavimo kaštai.
- Sunkumai sprendimų priėmime (pvz., kada sąveikauti su kitomis organizacijomis).
- Argumentų trūkumas įmonių sąveikumui versle.
- Pokyčiai bendradarbiavimo modelyje einantys link atviros inovacijos.

2.2.1 Staigių pasikeitimų/inovacijų valdymas

Turbūt svarbiausia problema, su kuria organizacijos susiduria šiandien yra pokyčių ir inovacijų pagreitinėjus tempas. Esant tokios sąlygoms, grėsmių ir galimybių atpažinimas egzistuojančiam verslo modeliui, didelis įmonių lankstumas ir didesnis noras keisti įmonės strategija, tapo kritiniu. Vis dėlto, įrodyta, kad toks pasikeitimų procesas yra sunkiai suprantamas [5]. Didelės organizacijos turi sunkumų atpažįstant pasikeitimus rinkoje ir suvaldant šį procesą. Kita vertus mažo ir vidutinio verslo įmonės yra šiek tiek lankstesnės, tačiau dažnai turi nepakankamai resursų stebėti rinkos pasikeitimus ir investuoti verslo adaptavimui. Be to, įmonių sąveikumo kontekste pasikeitimo proceso valdymas nėra paremtas mokslinėmis teorijomis ir principais ir yra pasiekiamas kaskart specifiniu būdu. Atsižvelgiant į naujas verslo galimybes, naujų verslo ryšių sukūrimas yra brangus ir sunkus. Ypač tai taikytina mažoms ir vidutinėms įmonėms. Joms reikalingi prisitaikantys sąveikavimo sprendimai tam, kad geriau susitvarkytų su kintančia verslo aplinka.

Šis pasikeitimų ir inovacijų valdymo procesas yra būtinas sėkmingam verslui sudėtingose rinkose. Tik tos įmonės, kurios sugeba sėkmingai atsinaujinti, galės išlikti ilgam. Ekonomikos gebėjimo atsinaujinti rodiklis yra pajamos, gautos iš naujų produktų ir paslaugų. Tai ekonomikos dinamiškumo ir funkcionavimo efektyvumo matas tokiose sudėtingose rinkose. Šis matas yra labiau informatyvus nei, pavyzdžiui, tyrimams ir diegimui skirtų lėšų santykis su pardavimais. Be to, pajamos iš naujų produktų ir paslaugų rodo gamybos apimtį ir todėl yra labiau orientuotos į rezultatą.

2.2.2 Prisitaikymas prie globalizacijos

Antras aspektas yra globalizacija. Ekonomikų integracija prisidėjo prie didėjančio pasaulinio konkurencingumo kai kuriose pramonės ir paslaugų sferose, ir tai lėmė augantį kainų spaudimą bei reikalavimus našumui daugelyje pramonės šakų. Informacinių technologijų dėka mažo ir vidutinio dydžio įmonės turi galimybę atlikti tarptautines operacijas. Taip pat dėmesys skiriamas ir pavojams, kuriuos kelia veikla perkelta į mažesnio darbo užmokesčio šalis. Mažai dėmesio skiriama regioniniam atskirtumui. Darbo užmokesčio skirtumai tarp regionų ir šalių rodo, kad yra didėjančios galimybės įmonėms iš pažangių šalių geriau panaudoti jų išteklius ir gauti naudą iš regioninių sąnaudų skirtumo. Deja, šis globalizacijos procesas dažnai reikalauja kurti ir naudoti

sudėtingus ir brangius specifinius projekto sprendimus, kurie sunkiai prieinami daugeliui mažų ir vidutinių įmonių. Taip nėra gerai žinoma, kokia metodika ir priemonės yra reikalingos siekiant įmonių sąveikumo globalizacijos kontekste.

Taigi, įmonių sunkumai valdant pasikeitimus ir inovacijas, bei padidėjusi konkurencingumą, atsiradusį dėl globalizacijos, yra linkę augti ateityje. Tai ypač taikytina mažo ir vidutinio verslo įmonės dėl to, kad jos negali skirti daug pinigų tyrimams, todėl jų galimybės sąveikauti su kitomis organizacijomis yra ribotos. Jeigu būtų geresni, pigesni įmonių sąveikumo sprendimai, tai mažesnės organizacijos galėtų transformuoti naujo produkto idėjas į naujus produktus ir paslaugas, nešančias pelną.

2.2.3 Dideli integracijos/sąveikavimo kaštai

Trečia problema yra susijusi su didelėmis išlaidomis, susijusiomis su informacinių sistemų integravimu ir sąveikumu su kitomis įmonėmis. Tokie projektai, kurie reikalauja aukšto lygio sistemų integracijos, dažnai yra sudėtingiausio ir rizikingiausio tipo, apie tai liudija ir dažnas numatytų išlaidų viršijimas. Taigi, siekdamas sumažinti riziką, įmonės buvo linkusios priimti pirmaujančių programinės įrangos teikėjų siūlomus sprendimus, taip vengdamos sudėtingų ir rizikingų sistemų integracijų.

Apibendrinant, integravimo išlaidos, vėlavimas ir rizika yra didelė kliūtis siekiant įgyvendinti įmonių sąveikumo sprendimus. Vis daugiau įmonių procesų migruos į interneto technologijomis pagrįstus sprendimus, kurie turės standartizuotas sąsajas ir lankstumą, bei leis mažinti integracijos išlaidas. Tačiau nepaisant to, kad yra daugybė patentuotų sprendimų, integravimo išlieka didelės išlaidos, o verslo sąveikumui nėra pakankamų techninių sprendimų.

2.2.4 Sunkumai sprendimų priėmime

Ketvirtasis aspektas susijęs su sprendimų priėmimo procesu. Vadybininkai taip pat susiduria su sunkumais priimant sprendimus, susijusius su įmonių sąveikumu. Sandorių nustatymas tarp įmonių paprastai yra atliekamas tik po derybų akis į akį tarp organizacijų atstovų. Dauguma sprendimų, susijusių su transakcijomis tarp organizacijų, reikalauja vadybininkų intervencijos. Pastarieji, atsižvelgdami į jų patirtį, žinias ir intuiciją, gali geriau suformuoti kompetentingus sprendimus naujų verslo sąveikų

kontekste ir gali geriau atpažinti sandorio su kita įmone naudą. Pavyzdžiui, dauguma sandorių tarp įmonių paremti asmeninėmis žiniomis ir pasitikėjimu tarp vadovų.

Tokias operacijas yra daug sunkiau klasifikuoti informacinių sistemų kontekste. Ypač kai vadybininko įsikišimas yra būtinas sandoriams, kuriems reikia dar neaprašytų žinių, ir verslo problemoms, kurios yra naujos ar skiriasi nuo įprastų. Laiko aspektas taip pat svarbus, nes įmonės turi kintančius poreikius - užtikrinti sąveikumą laikui bėgant. Iki šiol daugelyje sričių (su pastebima išimtimi finansuose) sunkumai, su kuriais susiduriama gerinant sprendimų priėmimo procesą naudojant informacines technologijas, skirtas įmonių sąveikumui, buvo neįveikiami, teigiama Europos Komisijos parengtose gairėse [1]. Tačiau geresnės priemonės, metodika ir gairės gali būti plėtojamos siekiant padėti vadybininkams įmonių sąveikumo sprendimo priėmime.

2.2.5 Argumentų trūkumas įmonių sąveikumui versle

Yra gerai žinomas faktas, kad verslo sėkmė dažnai priklauso nuo prieigos prie vertingos ribotos informacijos šaltinio ar žinių. Hipotezės, kad verslo sėkmė dažnai priklauso nuo sprendimus priimančių vadybininkų, turinčių prieigą prie reikiamos informacijos ir žinių realiu laiku, kad informacija yra prieinama bet kokiam verslui, turi akivaizdžių pasekmių įmonių sąveikumui. Dabartinės sistemos nėra pasiruošusios susidoroti su tokiais informacijos kiekiais.

Egzistuoja įsitikinimas, kad *įmonių sąveikumas* yra neaiški sąvoka, neturinti aiškaus panaudojimo versle. Organizacijoms trūksta pavyzdžių, praktikos ir gairių, apie tai, kaip sukuriama didžiausia vertė per įmonių sąveikumą. Sąveikumo operacijų sudėtingumas teisiniu ir logistiniu požiūriu atrodo neįveikiamas, ypač tarptautiniame kontekste. O sprendimus priimančiams asmenims, turint ribotus išteklius ir ribotą laiką, priimti įmonių sąveikumo sprendimą nėra realu.

2.2.6 Atvirioji inovacija

Termino *atvira inovacija* autorius yra Henry Chesbrough, kuris savo knygoje [6] tikina, kad kompanijos nebėra apribuotos savo pačių idėjų rinkinio ir gali pasirinkti geriausias idėjas iš įvairesnių šaltinių. Tai gali būti tiekėjai ar konsultantai, kitos firmos produktai, jungtiniai tyrimų projektai su kitomis firmomis. Pagrindinė idėja yra

atvirumas, įgalinantis turtingesnę konkuruojančių idėjų rinkinį, kuriuo grindžiami naujoviški projektai, taip pagerinant jų vertę ir tuo pat metu dalinantis rizika ir išlaidomis.

Apibendrinant, egzistuoja daugybė sunkumų pritaikant įmonių sąveikumo sprendimus. Trūksta funkcinių sprendimų ir įrodymų, koks svarbus yra įmonių sąveikumas.

2.3. *Esamų įmonių sąveikumo sprendimų analizė*

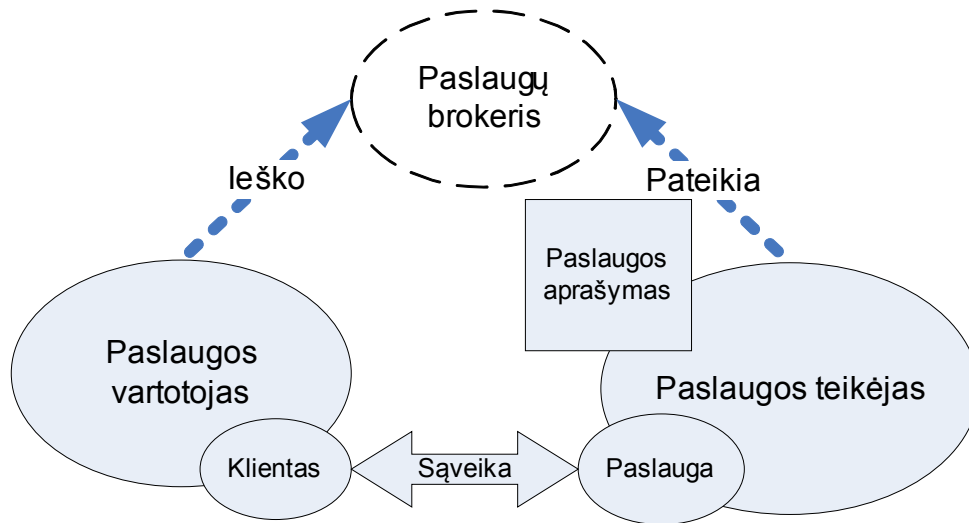
2.3.1 Paslaugoms orientuota architektūra

Neatsiejama įmonių sąveikumo dalis yra paslaugoms orientuota architektūra (SOA). Norint užtikrinti, kad verslo procesas vykdytų tikslą automatizuodamas žmogaus darbą, įmonių sąveikumo procesai pasitelkia paslaugoms orientuotą architektūrą, kuri susitelkia ties procesais, o ne ties technologijomis, reikalingomis automatizuoti juos [7].

Paslaugoms orientuota architektūra nusako rinkinį komponentų, kurių sąsajos gali būti apibūdinamos, publikuojamos, pasiekiamos ir iškviestos per tinklą. SOA siekia paremti programinės įrangos kūrimą tokiu būdu, kuris paveikia dinaminių sistemų kūrimą. Jos lengvai gali būti pritaikomos nepastovioms aplinkoms ir būti palaikomoms. Sistemos perskirstymas į sudedamąsias dalis įgalina sistemos komponentų perkonfigūravimą atsižvelgiant į galinio vartotojo poreikius ir sistemos aplinką. Be to, panaudojimas plačiai priimtų standartų ir protokolų, kurie yra paremti XML ir veikiančiais su interneto standartais (HTTP, SMTP ir t.t.), pagerina sąveikavimą [8].

SOA susideda iš paslaugos vartotojo ir paslaugos teikėjo (2.3.1.1 pav.), taip pat dar galimas ir paslaugų brokeris, kuris teikia paslaugų paiešką vartotojui.

Pagrindinis paslaugoms orientuotos architektūros tikslas yra palengvinti suvaldyti stambių įmonių sistemų augimą, palengvinti interneto paslaugų aprūpinimą ir naudojimą bei sumažinti organizacijų tarpusavio bendradarbiavimo kaštus. Su SOA galima pasiekti valdomą, struktūrizuotą aplikacijos-su-aplikacija bendravimą be pakeitimų pačiose programose .



2.3.1.1 pav. Paslaugoms orientuota architektūra.

Galima išskirti tokius SOA principus [9]:

- Pakartotinis panaudojimas, sąveikumas, portabilumas, susiskaidymas, komponentizavimas;
- Standartų atitikimas (tiek bendrų, tiek ir pramonės šakų);
- Paslaugų identifikavimas, kategorizavimas, paslaugų stebėjimas ir sekimas.

SOA svarba yra ta, kad, ji suteikia paprastą kintamą paradigmą organizacijų sistemų dideliems tinklams, kuriems reikia sąveikavimo. Iš tikrųjų SAO yra kintama, nes ji sukelia mažiausiai galimų prielaidų apie tinklą ir minimizuoja visas patikimas prielaidas, kurios yra dažnai netiesiogiai gaunamos mažesnėse sistemose. Architektas naudojantis SOA principus yra geriau pasiruošęs sistemų, kurios yra kintamos, evoliucionuojančios ir valdomos, kūrimui. Tūrėtų būti lengviau apsispręsti, kaip integruoti funkcionalumą peržengiant nuosavybės ribas. Pavyzdžiui, didesnė kompanija, kuri įsigijo mažesnę kompaniją, privalo nustatyti, kaip integruoti įsigytą jos IT infrastruktūra yra bendrą įmonės IT paketą. Per šią neatskiriamą savybę kisti ir evoliucionuoti, SOA įgalina IT paketą, kuris taip pat yra lengvai pritaikomas įvairiems specifinės probleminės srities poreikiams arba procesų architektūrai. Taigi SOA gali būti stiprus pagrindas verslo judrumui ir pritaikomumui.

2.3.2 XML grindžiami sprendimai sąveikumui

XML (angl. *Extensible Markup Language* – išplečiama žymių kalba) skirta transportuoti ir kaupti duomenis, tobulinti žiniatinklio funkcionalumą, teikiant informaciją būdais, kurie yra tikslesni, lankstesni ir lengviau pritaikomi. Skirtingai nei HTML (angl. *Hyper-text Markup Language* – hiperteksto žymų kalba), ji neturi fiksuoto formato. XML yra išplečiama ir naudoja "metakalbą". Metakalba - tai kalba, kuri aprašo kitas kalbas. Įvertinant šią savybę, XML leidžia vartotojams kurti savo žymėjimo kalbas įvairių rūšių dokumentams [10].

Tradicinis taikomųjų uždavinių sprendimas dažnai reiškia, jog reikia kurti programinę įrangą skirtingiems prietaisams. Skirtingi prietaisai – skirtingos jų formos, ir tai labai įtakoja PĮ kūrimą. Jei turime 800x600 ekrano dydį ir žiniatinklio sąsają, ji niekada neveiks ant 4 eilučių po 20 simbolių ekraną turinčio įrenginio. Ne XML grindžiamoje PĮ tenka kurti atskirus klientus bendraujančius su serveriu [11].

Taikant XML grindžiamas technologijas, egzistuoja viena aplikacija, kuri bendrauja su duomenų baze ir pateikia XML srautą. Tam tikras transliatorius priima šį srautą ir, pritaikydamas XSL transformaciją, atvaizduoja informaciją. Ši transformacija vykdoma serveryje, todėl ypatingi reikalavimai kliento pusei nėra keliami.

Ši „centro ir atšakų“ architektūra suteikia didžiulį našumą. Jei rinkoje atsirado naujas prietaisas, kuriam reikia pritaikyti programą, mums nereikia keisti PĮ kodo, mums tereikia pranešti transliatoriui, kokią transformaciją naudoti (transformacija turi būti sukurta).

Taigi naudojant tokią architektūrą užtikrinamas rezultatas, kad elektroninio verslo aplikacijos sugebės veikti plačiame prietaisų rate. Teisingai suprojektuota sistema, turi būti lanksti. Suprojektuotoje sistemoje pritaikius XML technologijas – pagrindines sąveikavimo technologijas [12] - galima visa tai įgyvendinti sutaupant lėšų. Plačiau apie XML sprendimus skyrelyje *Dokumentų standartizavimas*.

2.3.3 SOAP

SOAP (angl. Simple Object Access Protocol) yra paprastas objektų prieigos protokolas. Jis skirtas struktūrizuotos informacijos apsikeitimui decentralizuotoje, paskirstytoje aplinkoje [13].

XML grindžiami SOAP pranešimai pakeičia būdą, kuriuo mes kuriame paskirstytas aplikacijas, ir tai, kaip mes vykdomė duomenų apsikeitimą. SOAP taip pat yra vienas pamatų paslaugoms orientuotai architektūrai. SOAP yra dar vienas mechanizmas leidžiantis nuotolinį procedūrų iškvietimą (panašus funkcionalumas kaip DCOM arba IIOP). Pagal protokolą SOAP nepalaiko šių funkcijų:

- Paskirstytų sistemų šiukšlių surinkimas;
- Abipusio HTTP komunikavimo;
- Pranešimų konvejerio (keleto metodų iškvietimo viename pranešime);
- Komponentų kūrimas ir sujungimo su komponentais sukūrimo.

SOAP nepriklauso nuo platformos, programavimo kalbos ir jokio objektinio modelio. Taigi, aplikacijos, kuriose įdiegtas SOAP, gali aprėpti keletą operacinių sistemų, susidedančių iš objektų iš skirtingų gamintojų, parašytų skirtingomis kalbomis ir paremtų skirtingais objektiniais modeliais. Tai tikriausiai ilgai laukta universali priemonė tikram komponentų pakartotiniam panaudojimui. Senesnės protokolo versijos būdavo pririštos prie HTTP protokolo, tačiau dabar situacija kiek kitokia – SOAP nebėra pririštas prie tam tikro transportavimo protokolo. Galima siųsti SOAP pranešimas per SMTP ar FTP. Nors ir naudojami skirtingi transportavimo protokolai, tačiau SOAP pranešimams yra ir bus daugiausiai naudojamas HTTP.

Taigi, SOAP yra specifikacija, kuri aprašo duomenų pasikeitimo aplinką tarp sistemų, ir yra vienas iš sąveikumo standartų. Taip pat SOAP įgalintos aplikacijos yra lengviau diegiamos nei analogiškos sukurtos su DCOM ir pan.

2.3.4 BizTalk

BizTalk yra Microsoft verslo procesų integracijai skirtas sprendimas, grindžiamas XML, kur pagrindinis dėmesys skirtas elektroniniam verslui [14]. BizTalk susideda iš šių dalių:

- Schemų saugyklos (BizTalk.org);
- Karkaso;
- Įgalinančių įrankių (BizTalk serveris, schemų atvaizduotojo).

BizTalk.org schemų saugykla yra numatyta paskatinti schemų atradimą, pakartotiną panaudojamumą ir standartizavimą. BizTalk.org yra nepriklausomas organas,

palaikomas daugybės korporacijų įskaitant Microsoft, SAP, CommerceOne, Boeing, BP/Amoco ir kt. BizTalk karkasas yra specifikacija, kuri aprašo kaip XML pranešimo „apvalkalas“ turi atrodyti. Ji pateikia standartinę išsidėstymą informacijai, tokiai kaip siuntėjas/gavėjas, adresai ir priedai, kuri gali būti nuskaityta ir veikti BizTalk palaikančiuose įrankiuose.

BizTalk teikia šias pagrindines funkcijas:

- verslo procesų automatizavimas;
- verslo procesų modeliavimas;
- verslas-verslui komunikavimas;
- įmonės taikomojo uždavinio integravimas (angl. *Enterprise Application Integration*)
- pranešimų brokeris.

BizTalk serveris yra našus, patikimas ir galintis kisti. Jis sukurtas taip, kad sugeba apdoroti šimtus verslo dokumentų per sekundę [15]. Šiandien daugiau nei 7000 organizacijų yra pasirinkę BizTalk, tam, kad suintegruotų savo sistemas ir automatizuotų procesus per globalią tiekimo grandinę.

2.3.5 Portalai

Įmonių portalai taip pat vienas iš galimų sąveikavimo sprendimų. Portalas tai karkasas informacijai, žmonėms ir procesams integruoti. Visa reikalinga integruota informacija pateikiama viename ekrane [16]. Jis pagrįstas vieno prisijungimo principu, kai reikalingas tik pradinis vartotojo identifikavimas, įgalinantis priėjimą prie skirtingų sistemų, dažniausiai tinklapio sąsaja paremta forma. Informacijai agreguoti naudojami portletai. Taigi, tam, kad įgalintų sąveikumą, reikalingi standartinės portletų sąsajos bei komunikavimo mechanizmai. Egzistuoja du pagrindiniai standartai: *JSR168* [17] ir *OASIS WSRP* [18].

JSR168 yra Java portletų specifikacija, kuri aprašo Java API sąsajų portalams rinkinį bei turinio agregaciją, suasmeninimą, prezentaciją ir saugumą. Ši specifikacija išleista 2003 metų vasarą. Tuo tarpu kai JSR128 aprašo portletų programavimą, OASIS WSRP (angl. *Web Services for Remote Portlets*) aprašo protokolo tarpusavio komunikavimą.

OASIS WSRP aprašo XML ir žiniatinklio paslaugų standartą, kuris įgalina žiniatinklio paslaugų su portalais ar kitomis tarpinėmis žiniatinklio aplikacijomis „įjunk-ir-naudokis“ principą.

Žiniatinklio paslaugų teikėjas naudoja WSRP SOAP tam, kad bendrautų su pačiu portalu. Teikėjas SOAP užklausas paverčia į JSR168 Java metodų iškvietimą. Tam tikras portletas yra įvykdomas ir grąžinamas turinys transliatoriui kaip WSRP SOAP pranešimas.

Taigi šie du standartai *JSR168* ir *OASIS WSRP* dirba kartu tam, kad užtikrintų portletų palaikymą ir portalų sąveikavimą. Taip pat reiktų pažymėti, kad šie du standartai nereikalauja vienas kito. Portalas gali palaikyti JSR168 ir nepalaikyti WSRP. Portalas, kuriame įgalintas WSRP gali palaikyti tiek Java tiek ir .NET portletus.

Įmonių portalai bus neatskiriama verslo dalis. Įsigyjant portalą teks pasirašyti ilgalaikę bendradarbiavimo sutartį, todėl reikia labai atidžiai pasirinkti portalo gamintoją. Šiuo metu rinkoje yra šie portalų gamintojai: Apache, BEA, BroadVision, eXo, IBM, Liferay, Microsoft, Oracle, Plone, Red Hat (JBoss), SAP, Sun, uPortal, Vignette. Dauguma jų yra rizikingi sprendimai organizacijai [19].

2.4. El. verslas ir vertės grandinė

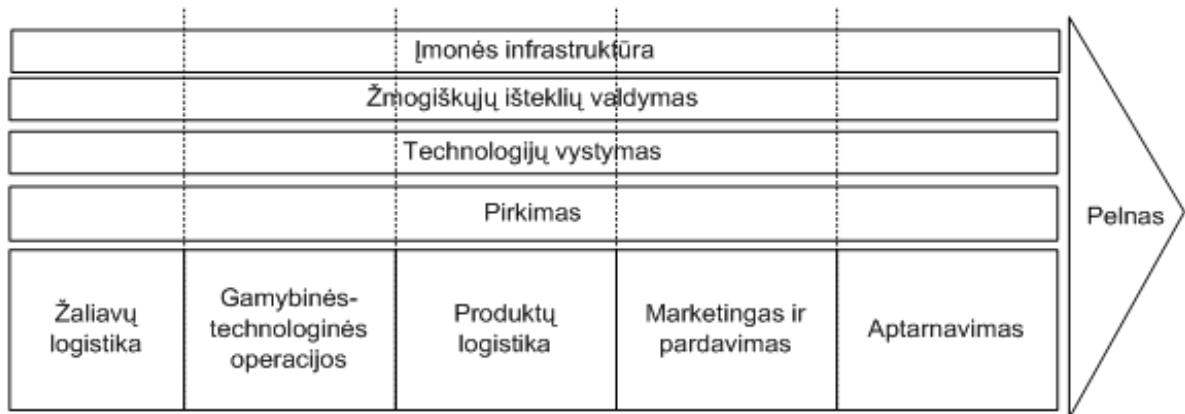
Ateityje versle informacinėmis technologijomis pagrįstas elektroninis verslas bus naudojamas ne tik įvairių veiksmų ir dalyvių vertės sistemoje ryšiams palaikyti, bet ir jos darbui optimizuoti realiaame laike (angl. *real time*). Remiantis informacija iš apjungtų veiksmų ir bendrų organizacijų, bus priimami sprendimai. Gamybos užduotys bus padalijamos pagal įmonių pajėgumus ir resursus, kuriuos galima gauti iš tiekėjų. Kadangi pirmieji šios stadijos diegimo etapai apims santykinai paprastą aprūpinimo ištekliais, gamybos, logistikos ir aptarnavimo operacijų optimizavimą, pilnesnis optimizavimas apims ir prekių kūrimą [20].

Verslo sąveikumo procesai garantuos įmonei sėkmingą konkurenciją. Verslo efektyvumo gerinimas prasideda nuo analizės, kurią galima atlikti M.Porter pasiūlytu „Vertės grandinės“ analizės metodu. Konkurenciniai pranašumai, diegiant elektroninį verslą atskiruose pridėtinės vertės grandinės etapuose, nelemia tvirto konkurencinio pranašumo sukūrimo, o skatina sekti pramonės šakos lyderiu ir siekti konkurencinės lygybės. Todėl, formuojant tvariusius konkurencinius pranašumus, elektroninės

prekybos diegimas neturi apsiriboti vien atskirų pridėtinės grandinės vertės etapų efektyvumo didinimu, o integruotai didinti visos grandinės efektyvumą. Be abejo, toks teiginys nereiškia, kad įmonės privalo diegti elektroninę prekybą kiekviename etape, tačiau įmonė turi žiūrėti į tvirtų konkurencinių pranašumų formavimą plačiau negu į atskirų operacijų efektyvumo didinimą.

M. Porter išskirtas pridėtinės vertės grandinės veiklų grupės, pritaikytos sugrupuojant konkurencinius pranašumus, kurie įgyjami panaudojant elektroninį verslą:

- Konkurenciniai pranašumai suformuojami pagrindinėje veikloje;
- Konkurenciniai pranašumai suformuojami pagalbinėje veikloje.



2.4.1 pav. M.Porter vertės grandinė.

Kiekvienoje įmonėje vertės grandinė yra sava, 9.1 priede pateikiame pavyzdžius, kurie iliustruoja IT pritaikymą, didinant efektyvumą operacijų bei formuojant konkurencinius pranašumus.

Informacinių technologijų poveikis gamybos procesams gali būti traktuojamas kaip konkurencinio pranašumo formavimo šaltinis. Bendrai laikomasi nuomonės, kad informacinės technologijos iš prigimties yra lanksti technologija, įgalinanti lanksčiau pritaikyti prekes vartotojų poreikiams ir leidžianti pasiekti masto ekonomiją, esant mažesnėms gamybos apimtims. Dėl to ji gali pakeisti gamybos ekonomiką ir palengvinti prekių diferenciaciją pagal unikalias savybes [20].

2.5. Dokumentų standartizavimas

Dokumentų standartizavimas tikriausiai yra labiausiai išspręsta įmonių sąveikumo problema. Tačiau kartu galima teigti, kad kuo daugiau standartų toje pačioje srityje, tuo

daugiau ir sumaištis. Egzistuoja daug standartų, ir beveik visi jie remiasi XML, skirtu verslo dokumentų mainams.

Istoriškai pirmieji bandymai apsiekti verslo dokumentais elektroninių būdų buvo elektroniniu paštu ir faksu, nors duomenys struktūrizuoti, tačiau nebuvo vieningos formos – standarto. Atsidaro pirmasis standartas – *elektorinių dokumentų mainai – EDI* (angl. *Electronic Data Interchange*), kiek vėliau XML ir juo grįstos verslo dokumentų kalbos, kurios ir nagrinėjamos šiame skyrelyje.

2.5.1 Elektroninių dokumentų mainai

Naujų technologijų naudojimo versle keliu 1980 metais Amerikoje didmenininkės pradėjo eiti pasirinkusios *EDIFACT* (EDI pirmtakė) technologiją, kuri leido perduoti užsakymo dokumentus naudojant telefono liniją. Ši technologija vėliau tobulėjo ir taip atsirado nauja - EDI (elektroninių dokumentų mainų) technologija paremta kitais standartais ir kitomis perdavimo priemonėmis. Todėl galima sakyti, jog *EIDIFACT* buvo perversmo elektroninių dokumentų pasaulyje pradininkė.

Formaliai, EDI yra pasikeitimas informacija standartiniu formatu tarp kompiuterių be žmogaus intervencijos [21]. EDI yra laikomas kaip techninė *verslo dialogo* reprezentacija tarp dviejų tiek vidinių, tiek išorinių organizacijos esybių.

EDI ganėtinai senas (devintame dešimtmetyje sukurtas standartas), tai rodo, kad jis yra labai brandus standartas, ir turi keturis pagrindinius rinkinius:

- Jungtinių Tautų rekomenduojamas *UN/EDIFACT* standartas (pirmasis EDI).
- JAV standartas *ANSI ASC X12*;
- *TRADACOMS* standartas taikomas Jungtinėje karalystėje;
- *ODETTE* naudojamas Europos automobilių pramonėje.

Reiktų paminėti, kad EDI nėra XML grindžiamas ir yra daug sunkiau žmogui suprantamas formatas, kita vertus, tiesioginis operatorius duomenimis yra kompiuteris, o ne žmogus. EDI pavyzdys:

NM1*WT*1* Klimavicius*Andrius*J.~N3*J. Borutos 4- 28*~N4*Kaunas*LT*46115

Šiame pavyzdyje aprašomas asmuo: pavardė, vardas, adresas, šalis, pašto kodas.

EDI naudojami dokumentai yra užsakymai, sąskaitos-faktūros, pristatymo pranešimai, banko pavedimų ataskaitos ir kiti.

Apibendrintai EDI galima apibūdinti taip:

- Pasenęs, bet vis dar naudojamas standartas;
- Brangiai pritaikomas sprendimas organizacijose;
- Duomenys perduodami sunkiai suvokiamu formatu;
- Daug mažesni duomenų kiekiai, lyginant su XML;
- Naudojamas senų kompanijų ir vyriausybių (pvz., JAV sveikatos apsaugos pramonė).

2.5.2 XML

Apie XML grindžiamus sprendimus įmonių sąveikumui rašyta 2.3.2 skyrelyje. XML pagrindu sukurta daug verslo dokumentų standartų. Žinomiausi jų būtų ebXML, UBL, xCBL, cXML.

2.5.3 Elektroninio verslo XML

Elektroninio verslo XML - ebXML (angl. *Electronic Business using eXtensible Markup Language*) yra modulinis rinkinys specifikacijų, kurios bet kokio dydžio, bet kokios geografinės padėties įmonės įgalina tvarkyti verslą per Internetą. Naudojant ebXML, kompanijos turi standartizuotą metodą biznio pranešimų pasikeitimui, prekybos ryšių užmezgimui, aprašymui ir verslo procesų registravimui [22].

Viena iš galingiausių ebXML savybių yra galimybė pasiekti *ad hoc* verslo sąveikavimus. Iš pirmo žvilgsnio, terminas *ad hoc* gali sukelti neigiamas emocijas, bet tai yra tiksliai ta ebXML savybė, kuri paverčia jį ypač galingu elektroninio verslo tvarkymui [23].

ebXML pradžia yra 1999 metai OASIS organizacijos iniciatyva. Pradinis projektas pristatė penkias esmines duomenų sluoksnių specifikacijas, įskaitant ir XML standartus, skirtus:

- verslo procesams,
- pagrindiniams duomenų komponentams,
- bendradarbiavimo protokolų susitarimams,
- bendravimui pranešimais,
- registrams ir saugykloms.

Sukurti iš viso penki ISO standartai, kuriuos prižiūri dvi organizacijos – tai OASIS ir UN/CEFACT [24].

Taip pat reiktų paminėti ir RossetaNet – tai standartas teikiantis pranešimo turinio, „choreografijos“ ir transportavimo specifikacijas modernių technologijų kūrėjams. Nors tai kiek labiau subrendęs standartas nei ebXML, jis yra „vertikalus“ – nukreiptas į tam tikrą pramonės šaką. ebXML yra „horizontalus“ [25].

2.5.4 Universali verslo kalba

UBL (angl. *Universal Business Language*) – universali verslo kalba yra standartizuotų elektroninių dokumentų, tokių kaip *užsakymas* ar *sąskaita-faktūra*, rinkinys. UBL skirta teikti universalią ir suprantamą komercinę sintaksę teisiškai įpareigojantiems verslo dokumentams veikti standartinio verslo rėmuose, teikiant visapusišką, standartais paremtą infrastruktūrą, kuri praplėstų esamų visų dydžių sistemų naudą [26]

UBL sukūrė ir plėtoja OASIS Techninis Komitetas, kartu dalyvaujant daugybės pramonės šakų duomenų standartų organizacijoms. UBL leidžiama naudotis be jokių mokesčių ir teisinių įsipareigojimų autoriams. UBL sukurtas pasinaudojant kitais dokumentų standartais (ebXML, xCBL).

Naujausia UBL versija yra *UBL 2.0 Errata 01* išleista 2008 gegužę, joje apibrėžiamas 31 dokumentas, kuriuos galima sugrupuoti taip:

- Prieš-pardavimo (prekių katalogų užklauso, katalogas, kainos)
- Užsakymo (užsakymas, užsakymo atsakymas, užsakymo pakeitimas, užsakymo atšaukimas).
- Pristatymo (važtaraštis, kilmės sertifikatas, siuntimo instrukcijos, transportavimo būseną, pažeistų ir trūkstančių prekių sąrašai)
- Sąskaitų (priminimas, sąskaitą-faktūra, primokėjimo, ar trūkstamo mokėjimo pranešimai, apmokėjimo būseną)
- Apmokėjimo (pranešimas apie apmokėjimą)
- Kiti (sisteminiai pranešimai, prisegti dokumentai)

UBL specifikacijoje UML diagramomis aprašyti ne tik dokumentų struktūros, bet ir pagrindiniai elektroninių dokumentų mainų apibendrinti procesai:

- Prekių katalogų pateikimas;
- Užsakymo procesas;
- Užsakymo įgyvendinimo procesas;
- Sąskaitų išrašymas;
- Atsiskaitymo procesas

Šie procesai skaidomi į smulkesnius, pavyzdžiui, prekių katalogų pateikimas į užsakovo inicijuotą ir pardavėjo katalogo pateikimą.

Išskiriamos ir UBL vartotojų rolės: *pirkėjas, pardavėjas, pristatytojas, sąskaitos gavėjas, sutarties dalyvis, tiekėjas, prekių gavėjas ir siuntėjas, dokumentų siuntėjas ir gavėjas, krovinio persiuntėjas, transportuotojas, žirantas, importuotojas, eksportuotojas*. Reiktų paminėti, kad UBL dokumentų rinkinys buvo sukurtas kaip *ebXML Pagrindinių Komponentų Techninės Specifikacijos 2.01* įgyvendinimas. Šie pagrindiniai komponentai sudaro specialius dokumentus, tokius kaip *užsakymas* ar *sąskaita-faktūra* [27]. UBL schemas (dokumentus aprašantys taisyklių rinkiniai) yra tinkamos daugkartiniam panaudojimui, praplėtimui ir agregavimui.

2.5.5 Bendroji verslo XML biblioteka

xCBL (angl. *XML Common Business Library*) - Bendroji verslo XML biblioteka – yra rinkinys XML specifikacijų, skirtų naudoti el. versle [28]. Naujausioje 4.0 versijoje, tiesa, gan senoje, išleistoje 2003 metais, yra 44 dokumentai. xCBL taip, kaip ir UBL, šaknys yra EDI standarte.

xCBL palaiko visus pagrindinius dokumentus ir sandorių tipus pasaulinėje el. komercijoje, įskaitant kelių įmonių tiekimo grandinės automatizavimą, tiesioginius ir netiesioginius viešuosius pirkimus, aukcionus bei sąskaitų pateikimą ir apmokėjimą tarptautinėje aplinkoje.

Nors ir standartas atrodo daug apimantis ir galintis daug, visgi reiktų pastebėti, kad naujesnių standartų buvimas išstūmė xCBL.

2.5.6 Komercijos XML

cXML (angl. *Commerce XML*) – komercijos XML standartas – labai skiriasi nuo kitų elektroninių verslo protokolų. cXML yra supaprastintas protokolas skirtas, nuolatiniai verslo dokumentų kaitai tarp el. komercijos mazgų ir tiekėjų. Protokolas neapima visų

sąveikų, kurias norėtų matyti kai kurios šalys, dalyvaujančios el. prekyboje. Vis dėlto per išorės elementų naudojimą cXML yra praplečiama. Šie išplėtimai yra būtini komunikavimui taškas-su-tašku [29].

cXML pranašumai:

- Lengva įgyvendinti automatizuoto užsakymo gavimą, užsakymo įgyvendinimo atnaujinimą ir katalogo siuntimą.
- Dauguma pardavimo sprendimų jau yra aprašyti cXML
- cXML yra plačiai priimtas B2B protokolas
- Išplečiamumas: jei papildomi duomenys yra reikalingi, jie gali būti perduodami taip pat.
- cXML yra vienintelis sąveikumo standartas, kuriame išlikusi didelė dalis EDI sintaksės.

cXML palaiko katalogus, tiekėjų aprašymus, sutartis, užsakymus, užsakymų pavirtinimus, pristatymo žinutes, sąskaitas ir kitas įvairias sinchronizacijos ir integracijos transakcijas.

cXML yra išleistas remiantis informacija iš daug įmonių. Šis protokolas yra nemokamo naudojimo. Jis atviras be jokių apribojimų.

2.6. Užsakymo proceso analizė

Panagrinėsime tipinį verslo procesą – užsakymą. Terminas užsakymas (verslo kontekste) žodynuose traktuojamas įvairiai, tačiau prasmė išlieka ta pati. Apibendrintai galime apibrėžti taip – *užsakymas, tai raštu arba žodžiu užfiksuotas įsipareigojimas tam tikrų paslaugų ar produktų komercinis sandoris*. Žiūrint iš pirkėjo taško jis išreiškia ketinimą pirkti ir vadinamas *pirkimo užsakymu* (angl. *purchase order*). Iš pardavėjo taško jis išreiškia ketinimą parduoti ir asocijuojamas su *pardavimo užsakymu* (angl. *sales order*). Tuomet, kai pirkėjo pirkimo užsakymas ir pardavėjo pardavimo užsakymas sutampa, užsakymas virsta kontraktu tarp pirkėjo ir pardavėjo.

Organizacijos ribose, terminas užsakymas gali būti siejamas su darbo užsakymu gamybai, profilaktiniam patikrinimui ar remonto paslaugų užsakymu organizacijai.

Dauguma įmonių užsakymai yra naudojami surinkti ir pateikti išlaidų ir pajamų ataskaitas, dėl aiškiai apibrėžtų tikslų. Tuomet yra įmanoma parodyti, kokiais tikslais buvo patirtos išlaidos.

2.6.1 Žodiniai užsakymai

Tokios įmonės kaip parduotuvės, degalinės ir restoranai vykdo veiklą su savo klientais priimdamos žodinius užsakymus, arba suprantamus iš vartotojo veiksmų. Prekių krepšelio pristatymas į kasą yra suprantamas kaip vartotojo ketinimas įsigyti prekes. Išsinešimo arba valgymo vietoje užsakymo sudarymas restorane yra žodinis pirkimo užsakymas. Taip pat ir degalų prisipylimas benzino kolonėlėje yra suprantamas kaip užsakymas. Pardavėjas dažniausiai tikisi apmokėjimo iškart grynais, čekiu ar kreditine kortele už pirkinius, ir pardavėjas pateikia apmokėjimo čekį.

2.6.2 Komerciniai užsakymai

Komercinėje veikloje, įvairūs verslo dokumentai yra naudojami užrašyti derybų sutartį pirkti ir parduoti, užrašyti pačią sutartį ir užrašyti kontrakto nutraukimo ir laikymosi sąlygas. Susitarimas pirkti ir parduoti yra kontrakto forma.

Yra penki būtini reikalavimai tam, kad egzistuotų kontraktas tarp dviejų šalių: susitarimas, prievartos nebuvimas, abipusis patenkinimas, galimumas ir legalumas. Taip pat dar gali būti ir šeštasis reikalavimas, kai viskas turi būti įtvirtinta *raštu*.

Pagrindinis reikalavimas komerciniams užsakymams, kad jie būtų kontraktais, yra tai, kad turi būti susitarimas (pasiūlymas ir jo priėmimas). Prieš sudarydamos kontraktus, įmonės dažnai užrašo derybų detales naudodamos prašymus kainoms arba kainų siūlymams. Kai tik susitarimas arba kontraktas yra parengtas, įmonės turi įrašyti juos kaip patvirtintus pirkimų ar pardavimų užsakymus.

2.6.3 Užsakymo operacijos

Norėdami išsiaiškinti, kaip vyksta užsakymo procesas, remsimės *universalialia verslo kalba* (UBL), kuri struktūrizuotai aprašo užsakymo procesą. Galime išskirti tokias standartines užsakymo proceso operacijas:

- pateikimas,
- gavimas,
- priėmimas,

- atmetimas,
- atšaukimas,
- pakeitimas,
- detalių pridėjimas,
- atsakymo priėmimas.

Paprasčiausia užsakymo operacijų seka yra tokia – pirkėjas pateikia užsakymą pardavėjui, pastarasis jį priima ir apdoroja, ir, jei užsakymas įmanomas, jį priima. Beje, tokia paprasčiausia seka būna tada, kai pirkėjas nelaukia atsakymo į užsakymą, jei jis sėkmingas. Tačiau dažnai atsitinka taip, kad reikia užsakymą koreguoti dėl vienos ar kitos priežasties, pavyzdžiui, nepakankamas prekių kiekis sandėlyje.

Šie procesai plačiau paaiškinti kitame skyrelyje.

2.6.4 Užsakymo dokumentų tipai

Be operacijų taip pat reikia apibrėžti ir dokumentų, dalyvaujančių tose operacijose tarp abiejų šalių, tipus:

- užsakymas
- užsakymo patvirtinimo atsakymas,
- užsakymo atsakymas,
- užsakymo nutraukimas,
- užsakymo pakeitimas.

Užsakymas – pradinis turinys, kurį pirkėjas pateikia pardavėjui. *Užsakymo patvirtinimo atsakymas* – tai pranešimas, kad užsakymas, kurį padarė pirkėjas, patvirtintas be pakeitimų, priešingu atveju pastarasis yra atmetamas.

Pardavėjo siūlomi pakeitimai užsakyme yra siūlomi naudojant *užsakymo atsakymą*. Jis yra siūlomas kaip pakaitalas pirkėjo pasiūlytam. Jis atspindi kitą užsakymo transakcijos būseną. Jis taip pat reiškia, kokiais ištekliais pardavėjas sutinka arba pateikia papildomas su užsakymu susijusias detales pirkėjui, kurios prekės nebuvo galimos užsakymo metu. Į tai įeina:

- pristatymo data (pasiūlyta pardavėjo, jei pirkėjas nereikalavo),
- kainos,
- nuolaidos,

- mokesčiai,
- prekių klasifikavimo kodai.

Pardavėjas taip pat gali informuoti, kokios prekės bus pakeičiamos į kitas, ar būtinus pakeitimus, naudojant užsakymo atsakymą.

Pirkėjas gali pakeisti savo užsakymą dviem būdais: siųsdamas *užsakymo pakeitimą*, arba nutraukdamas užsakymą ir sudarydamas naują užsakymą, pakeisiantį senąjį.

Užsakymo pakeitimas atspindi esamą užsakymo transakcijos būseną. Pirkėjas gali padaryti užsakymo pakeitimą dėl įvairių priežasčių, tokių kaip, keitimas užsakytų prekių, kiekio, pristatymo datos, pristatymo adreso ir pan. Tiekėjai gali priimti arba atmesti *užsakymo pakeitimą*, naudodami *užsakymo atsakymą* arba *užsakymo patvirtinimo atsakymą*.

Bet kuriuo proceso metu pirkėjas gali atšaukti sudarytą užsakymą naudodamas *užsakymo atšaukimo* dokumentą. Teisėti kontraktai, pardavimo partnerių susitarimai ir verslo taisyklės apriboja, ties kuriuo proceso tašku *užsakymo atšaukimas* bus ignoruojamas (pvz., gamybos arba pristatymo metu). Su duotu susitarimu ir taisyklėmis, užsakymo atšaukimas gali būti automatizuota arba neautomatizuota verslo transakcija. Kontrakto informacijos sąlygos verslo įsipareigojimui diktuos kurie, jei išvis tokių bus, iš šių apribojimų ar nuostatų bus taikomos.

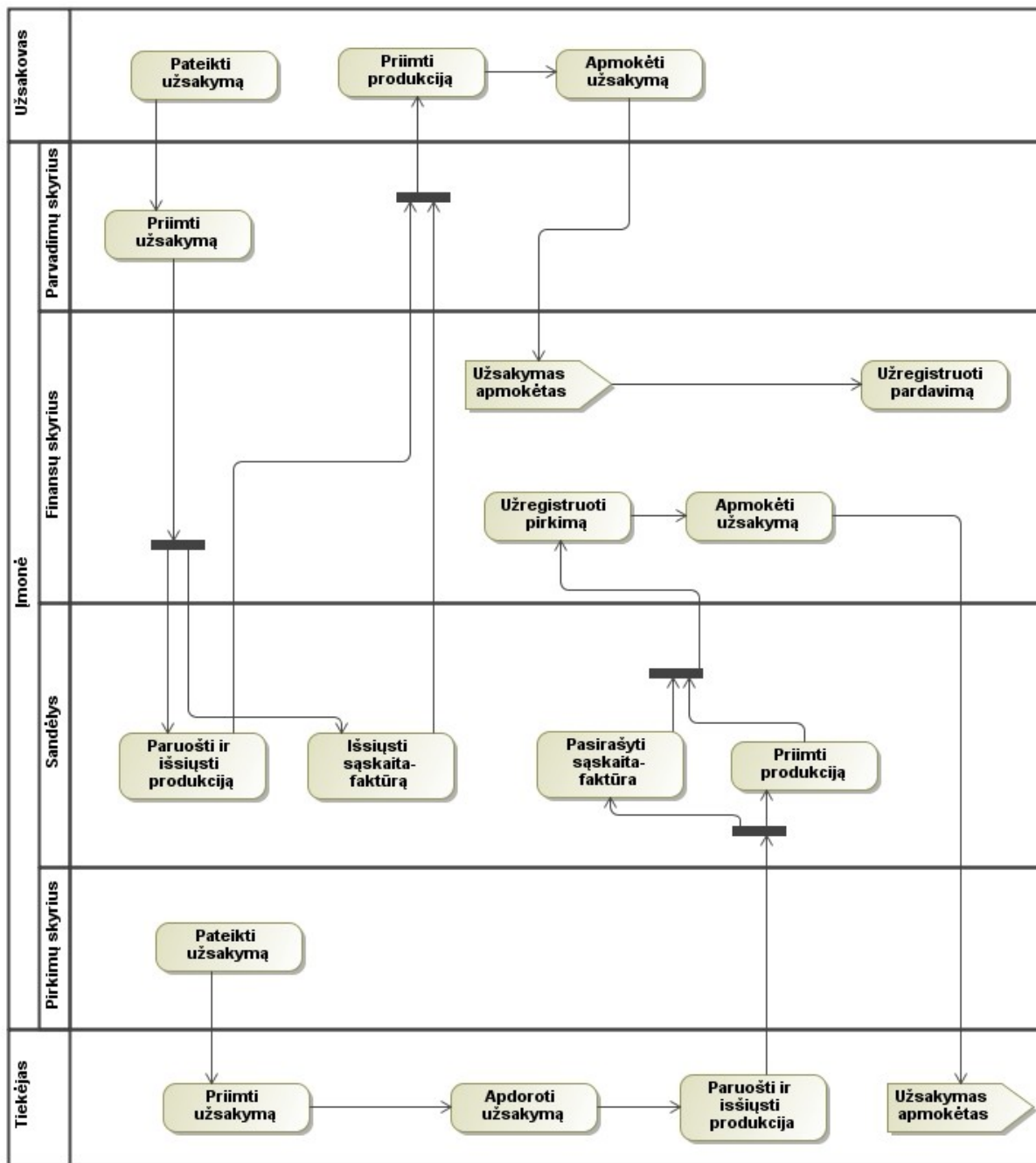
2.7. Užsakymas įmonės viduje

Norint suprasti, kaip vykdomas užsakymas, reikia išnagrinėti, kas vyksta įmonėje: kokie procesai ir kas už juos atsakingas. Sudarėme supaprastinta užsakymo vykdymo veiklos diagramą. Joje įmonė vaizduojama dviem būdais: įmonė-užsakovas ir įmonė-tiekėjas (2.7.1 pav.).

Išskyrėme keturis organizacinius vienetus, dalyvaujančius pirkimuose ir pardavimuose:

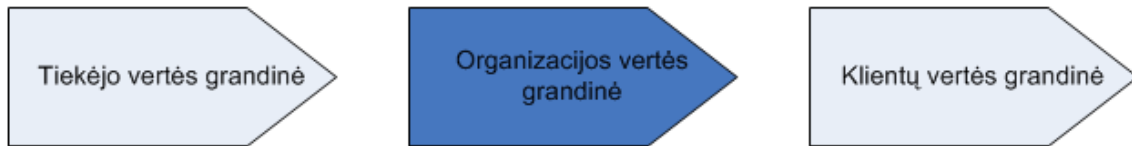
1. Pardavimų skyrius – atsakingas už užsakymo priėmimą.
2. Finansų skyrius – atsakingas už apskaitą, užsakymų apmokėjimą ir užsakymo apmokėjimo priėmimą.
3. Sandėlys – atsakingas už prekių išrašymą ir priėmimą, įskaitant ir sąskaitų-faktūrų išrašymą ir pasirašymą.

4. Pirkimų skyrius – atsakingas už užsakymo pateikimą tiekėjui.



2.7.1 pav. Užsakymo vykdymo veiklos diagrama.

Žiūrint iš vertės grandinių perspektyvos (2.7.2 pav.), įmonė gali būti tiek klientas, tiek tiekėjas. Ir iš tikrųjų, norint įgyvendinti elektroninius užsakymus užtenka realizuoti užsakymų sistemą kiekvienoje organizacijoje ir toji grandinė IT pagalba egzistuos.



2.7.2 pav. Verčių grandinė.

Gali atsitikti ir taip, kad reikia vykdyti automatinius užsakymus, pavyzdžiui, sumažėjus atsargoms sandėlyje automatiškai užsakyti žaliavų. Tokios paprastos užsakymo sistemos čia nepadės, nes bus labai sunku integruoti tiekėjo informacinę sistemą su užsakovo. Taip pat, jei įmonės-užsakovai naudoja įmonės informacinius portalus, joms svarbu turėti vieningą prieigą ir prie partnerių-tiekėjų užsakymų sistemų. Todėl, yra poreikis integruoti užsakymų sistemas. Šioms ir panašioms problemoms spręsti naudojami servais, kurių pagalbą įmonės teikia įvairias paslaugas – tarp jų ir užsakymo.

2.8. Analizės išvados

Atlikus analizę galima daryti išvadą, įmonių sąveikumas suteikia konkurencinį pranašumą, padeda sukurti naujus produktus ir paslaugas. Jis daro verslą lankstesnį ir leidžia lengviau prisitaikyti prie pasikeitimų rinkoje.

Įmonių sąveikumas yra aktuali problema. Ji labiau susijusi su technologinio sprendimo priėmimu, nei technologijų nebuvimu. Taip yra, todėl, kad sąveikumo taikymas yra sudėtingas procesas, reikalaujantis didelių išlaidų. Sunkumai kyla sprendimų priėmime, pavyzdžiui užsakymo procese standartiškai būtinos vadybininkų akis-į-akį derybos, todėl tokias operacijas yra sunku klasifikuoti informacinių sistemų kontekste. Trūksta įrodymų apie sąveikumo naudą.

Paanalizavus verslo dokumentų standartus (EDI, ebXML, UBL, cXML, xCBL) galima iškirti UBL, kadangi jis yra nuolat tobulinimas, sukurtas kitų standartų pagrindu, turi praplėtimo galimybes. Jo kūrime dalyvauja įvairių pramonės šakų atstovai.

Užsakymo proceso analizė parodė, galima sukurti automatinę užsakymo aptarnavimo sistemą. Taip bus įgalintas įmonių sąveikumas ir pagal M.Porterio vertės grandinės skaidymą, informacinių technologijų taikymas suteiks konkurencinį pranašumą.

3. Užsakymų aptarnavimo komponento reikalavimų analizė ir specifikacija

3.1. Techninis pasiūlymas

3.1.1 Sistemos apibrėžimas

Užsakymų aptarnavimo komponentas – tai sistema, įgalinanti įmonės klientams automatizuotai užsisakyti prekes. Ji teiks užsakymo aptarnavimo paslaugas pasitelkdama verslo dokumentų standartą UBL.

3.1.2 Bendras veiklos tikslas

Bendras veiklos tikslas yra sukurti sistemą, kuri įgalina įmonių sąveikumą ir kurią būtų galima pritaikyti organizacijoje, turinčioje savo informacinę sistemą ir naudojančią populiarią reliacinę duomenų bazę užsakymų informacijai saugoti. Sistema turi būti sukurta komponento principu, t.y., kad minimaliomis pastangomis ją būtų galima įdiegti įvairiose įmonėse.

3.1.3 Sistemos pagrindumas

Remiantis įmonių sąveikumo analizės išvadomis galima teigti, kad esama situacija šioje srityje vis dar problemiška. Išanalizavus kaip versle vyksta užsakymo procesas remiantis UBL, įmonei galima pritaikyti sąveikumo sprendimą, grindžiamą esamomis technologijomis ir standartais, kuris leistų palengvinti žmogaus darbą jį automatizuojant.

Paprasčiausias pavyzdys, kuomet mažmeninėje įmonėje dirbantis asmuo yra atsakingas už tai, kad sandėliuose netruktų prekių. Darbuotojas nuolat turi sekti informaciją apie sandėlio atsargas ir joms besibaigiant padaryti užsakymą iš tiekėjo. Dažnai prekių užsakymai vykdomi per įvairias didmeninę prekybą besiverčiančių įmonių informacines sistemas, t.y., darbuotojas turi jungtis prie jų sistemos ir ten įvykdyti užsakymą. Norint palengvinti žmogaus darbą galima jį automatizuoti. Tam, kad automatizavimas būtų įmanomas, pirmiausiai reikia sukurti paslaugą, kuri leistų vykdyti tokio tipo verslo sandorius.

Šios paslaugos teikimas suteikia versle pranašumą prieš kitas konkuruojančias organizacijas. Kaip parodė analizė, įmonių sąveikumas labai aktualus mažoms ir

vidutinėms organizacijoms. Todėl, remiantis anksčiau pateiktu pavyzdžiu, galima teigti, jog tokią paslaugą teikiančią sistema kurti verta.

Sistemos kūrimą galima pagrįsti ir numatoma sąlyginai didele pritaikymo sritimi. Užsakymo proceso analizė (žr. 2.7 skyrius) parodė, kad tokį komponentą galima taikyti bet kurioje grandies dalyje. Didžioji dalis firmų patenka į tokią vertės grandinę. Taip pat labai dažnai įvairios prekės ir paslaugos į mūsų namus patenka ne iš didmenine prekyba besiverčiančių įmonių, bet iš tarpininkų, kurie yra potencialūs tiesioginiai vartotojai.

3.1.4 Prototipai ir pagalbiniė informacija

Programinė įranga bus kuriama nuo pradžių, t.y., nesinaudojant paruoštais ar pradėtais prototipais, dėl tokių, projekto tikslo įgyvendinimui tinkančių prototipų nebuvimo.

Projektas bus išskaidytas į logiškai nepriklausomus modulius pagal jų funkcionalumą ir nefunkcinius reikalavimus. Moduliai bus įgyvendinami po vieną ir apjungiami galutinėje sistemos kūrimo fazėje.

Kuriamam komponentui bus panaudota nauja ir pažangi pagalbiniė programinė įranga, skirta tiek ją kurti, tiek ir diegti bei palaikyti jau sukurtą sistemą. Tai leis ne tik pagreitinti projekto įgyvendinimą, bet ir leis pagerinti jo kokybę.

Taip pat projekto kūrime bus naudojami kelių tipų programų generatoriai:

- Teksto nagrinėjimo programų ir leksinių analizatorių generatoriai kalbos apdorojimui.
- CASE priemonių kodo generatoriai.

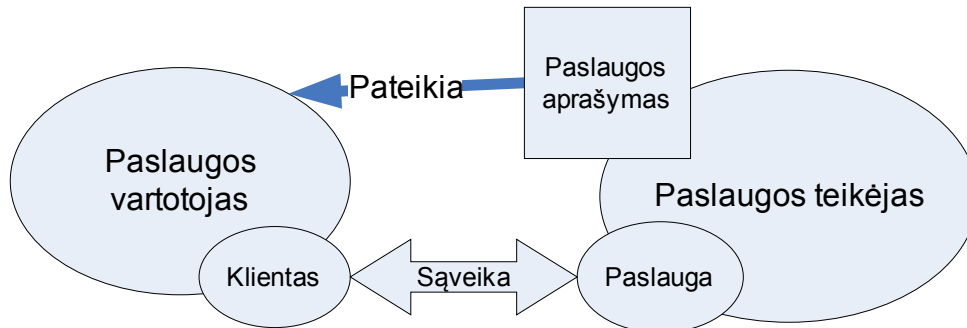
Jie leis sumažinti kaštus ir ženkliai palengvinti programuotojų darbą ir sumažinti klaidų skaičių.

3.1.5 Siūlomas sprendimas

Įgyvendinimui siūloma naudoti paslaugoms orientuota architektūra, nes kuriamas komponentas, kuris yra dokumentams orientuotas, tinkamas pakartotiniam panaudojimui, o tai yra vieni iš SOA architektūros principų.

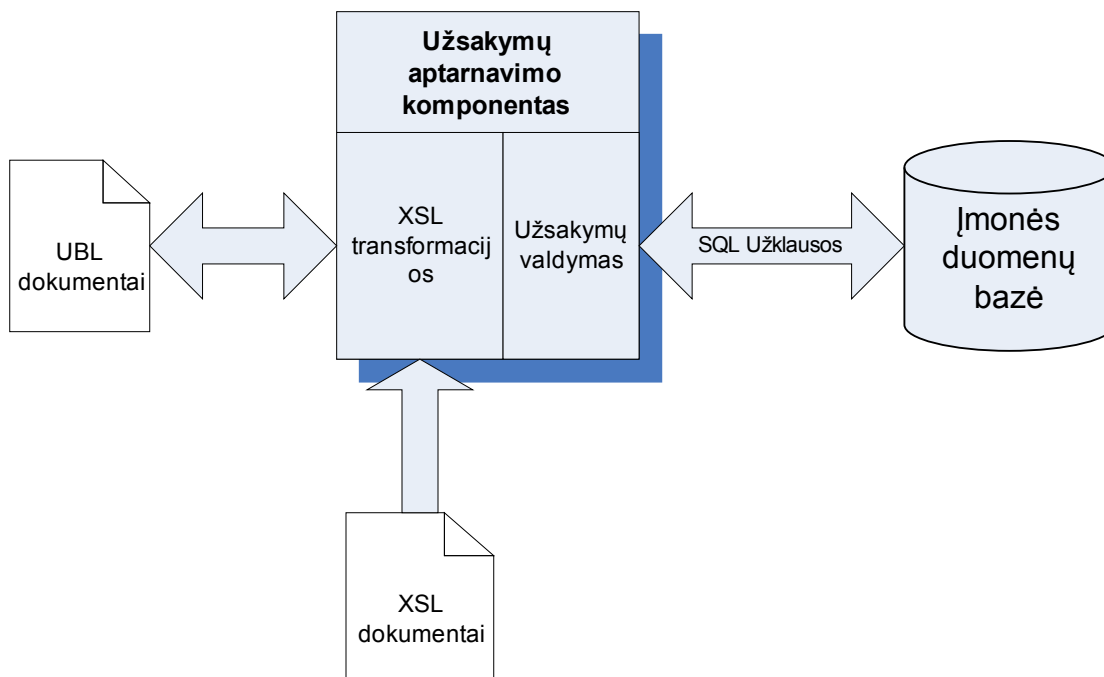
Architektūros modelis (3.1.5.1 pav.) aprašo sąveiką tarp dviejų šalių: paslaugos teikėjo, kuris publikuoja paslaugos aprašą ir tiekia paslaugos realizaciją ir paslaugos vartotojo, kuris naudodamas tiesioginę nuorodą į paslaugos aprašymą, susisieja (angl.

bind) ir iškviečia (angl. *invoke*) paslaugą. Kitaip nei 2.3.1 skyriuje aprašytoje architektūroje, čia nėra paslaugų brokerio, kuris yra visai nebūtinai planuojamai sistemai, nes sistema labiau orientuota į glaudžiai susijusių organizacijų ratą.



3.1.5.1 pav. Paslaugoms orientuotos architektūros modelis.

Duomenų mainams siūloma naudoti UBL dokumentus (žr. 2.5.2.2 ir 3.3 skyriai). Taigi panaudojant paslaugoms orientuotą architektūrą siūlomas sprendimo koncepcinis modelis (3.1.5.2 pav.) turės du pagrindinius modulius – tai XSL transformacijos ir užsakymų valdymas. XSL transformacijų modulis naudos XSL dokumentus, kuriuose bus saugoma informacija apie tai, kaip nuo UBL pereiti prie įmonės vidinės duomenų bazės struktūros. Užsakymų valdymo modulis naudos transformacijų modulio gautus rezultatus ir pavers juos SQL užklausomis.



3.1.5.2 pav. Užsakymų patarnavimo komponento koncepcinė schema.

Kaip matome komponento kontekstą sudaro UBL dokumentai, XSL dokumentai ir įmonės duomenų bazė, kurioje saugoma užsakymų informacija. UBL dokumentai dalyvaus .komunikavime su įmonės išore (mažmenine prekyba besiverčiančiomis įmonėmis). XSL dokumentus reikės rašyti individualiai kiekvienai organizacijai, remiantis įmonės duomenų bazės struktūra.

3.2. Galimybių analizė

3.2.1 Sistemos aktualumas

Sistema bus pakartotinio panaudojimo, t.y., skirta ne vienintelei organizacijai. Programavimo inžinerija anksčiau buvo labiau orientuota į originalų programinės įrangos kūrimą, bet dabar pripažinta, kad norint geriau, pigiau ir greičiau sukurti programinę įrangą, reikia pritaikyti projektavimo procesą, pagrįstą sistemingu pakartotiniu panaudojimu.

Pakartotinis panaudojimas suteikia šiuos privalumus:

- Padidintas patikimumas;
- Atitikimas standartams;
- Mažesni kūrimo kaštai.

Kaip jau minėta skyriuje *Sistemos pagrįstumas*, kuriamas komponentas įmonei suteiks konkurencinį pranašumą ir veiks nepriklausomai nuo organizacijos, kurioje ji įdiegta. Taigi, tiek įmonėje, kurioje bus diegiama ši sistema, tiek ir įmonėms, kurios naudosis jo teikiamomis paslaugomis, turėtų būti aktualus tokios sistemos egzistavimas.

3.2.2 Komercinė nauda

Projektuojama sistema nėra skirta komerciniai veiklai. Ji bus kuriama atviro kodo principu, t.y., bus galima naudotis be jokių teisinių ar finansinių įsipareigojimų.

Sistema komercinę naudą turėtų teikti ją įsidiegusioms įmonėms. Ji įgalins įmonių sąveikumą. Ypač tai naudinga mažojo ir vidutinio verslo įmonėms, kurios, norėdamos išlikti šiuolaikinėje rinkoje, turi apjungti paslaugas, pritaikyti sąveikumo sprendimus savo versle.

3.2.3 Techninės galimybės

Pagal pasirinktą architektūrą galima įvertinti techninio įgyvendinimo galimybes. Egzistuoja daug paslaugoms orientuotų sprendimų įgyvendinimų. Vieni tokių būtų:

- *Apache Axis 2* – brandus ir labai paplitęs žiniatinklio paslaugų teikimo variklis.
- *RESTful* paslaugos – pats paprasčiausias sprendimas, kuris gali būti įvairiomis žiniatinklio technologijomis sukurtas puslapis.
- *GlasFish, Metro, XFire* ir kiti paslaugų teikimo varikliai.

Pagal architektūrą paslaugos teikėjas turi pateikti paslaugos aprašymą. Tam tikslui yra naudojamas *WSDL* formatas. Ko gero visos paslaugomis orientuotos sistemos pateikia savo aprašą šiuo formatu. Taigi ir vartotojui, kuriančiam šios sistemos klientą, tai neturėtų būti naujovė.

Kuriamos sistemos pagrindas yra spragos užpildymas tarp UBL dokumentų ir įmonės duomenų bazės. Dokumentai į saugyklą pateks ir iš jos bus sugeneruojami ir atiduodami per projektuojamą paslaugos komponentą pasitelkiant XSL transformacijas (angl. *Extensible Stylesheet Language*).

XSL – tai XML grindžiama kalba, skirta transformuoti XML dokumentus į kitus XML dokumentus arba žmonėms suprantamus formatus. XSL dažniausiai naudojamas konvertuoti duomenis tarp skirtingų XML schemų (pvz., iš UBL į įmonės duomenų bazės schemą) ar konvertuoti duomenis į HTML dokumentus žiniatinkliui. XSL transformacijos procese dalyvauja XML dokumentai, XSL transformacijų šablonai ir XSL transformacijos variklis. Egzistuoja komerciniai ir ne komerciniai transformacijų varikliai, pvz.: *Saxon, JAXP*.

Duomenų manai vyks XML formatu, skirtingai nei duomenų bazėse saugomi duomenys. Vadinasi, XML duomenis reikės atvaizduoti į bazės lenteles. Šiam tikslui būna speciali pagalbinė programinė įrangė, pavyzdžiui, *Oracle XSU utility*.

Planuojama, kad komponentas veiks su bet kuria populiaria releacine duomenų bazių valdymo sistema (*Oracle, MySQL, MS SQL, PostgreSQL* ir t.t.). Visos jos palaiko vieningą užklausų kalbą SQL, todėl sunkumu suderinti duomenų baze neturėtų kilti.

Sistemos vartotojas, turėdamas paslaugos aprašą, gali pasinaudoti paslauga nepriklausomai nuo to, kokia įmonėje informacinė sistema veikia. WSDL formato apdorojimas įgyvendintas daugelyje programinės įrangos platformų.

Taigi, kaip matome, technologiniu požiūriu ši sistema yra įgyvendinama.

3.2.4 Vartotojų pasiruošimo analizė

Sukurta paslauga bus diegiama įmonėje. Joje dirbantys IT specialistai ir turintys žiniatinklio paslaugų diegimo patirties su sunkumais susidurti neturėtų. Taip pat bus reikalingos XSL transformacijų žinios bei UBL dokumentų struktūros suvokimas (žr. 3.3 skyrių).

Paslaugos vartotojai bus taip pat įmonės. Jų IT darbuotojai turės mokėti pasinaudoti žiniatinklio paslaugomis, aprašytais WSDL formatu. Visos kitos žinios reikalingos automatizuoti jų sistemas yra ne kuriamo komponento kontekste.

3.3. Duomenų analizė

3.3.1 UBL dokumentų duomenų tipai

UBL dokumentai sudaryti iš primityvių (klasifikuotų ir neklasifikuotų) ir sudėtingų duomenų tipų. Primityvūs duomenų tipai yra sudaryti praplečiant numatytuosius XML schemų duomenų tipus. Iš viso jų yra 20. Jų aprašymai yra 9.2 priede.

Klasifikuoti duomenų tipai sudaryti praplečiant *CodeType* tipą. Pastarieji dar labiau detalizuoja informaciją saugomą dokumentuose. Sudėtingi duomenų tipai susideda iš primityvių duomenų tipų. Jie gali būti kitų sudėtinių tipų dalimi, taip sudarydami UBL dokumentus.

Svarbu paminėti ir kelis sudėtinius duomenų tipus: *LineItem* – linijos prekė, ir *Party* – šalies, dalyvaujančios sandoryje, aprašymai pateikti atitinkamai 9.2.2 ir 9.2.3 lentelėse. Šie tipai yra esminės sudedamosios užsakymo dalys.

Taigi galima sakyti, kad UBL dokumentai susideda iš daugybės hierarchiškai susijusių primityvių duomenų tipų, kuriuos galima atvaizduoti į kitos kalbos duomenų tipus.

3.3.2 UBL-Order dokumento analizė

Iš viso užsakymo procese dalyvauja 5 tipų UBL dokumentai:

- *UBL-Order* – užsakymas;
- *UBL-OrderResponseSimple* – užsakymo patvirtinimas arba paneigimas;

- *UBL-OrderResponse* – nuodugnus užsakymo patvirtinimas arba paneigimas, su pardavėjo prekių pakeitimais arba siūlomomis alternatyvomis;
- *UBL-OrderChange* – sutikimas apie užsakymo pakeitimą;
- *UBL-OrderCancellation* – užsakymo atšaukimas.

UBL-Order yra pagrindinis dokumentas, nuo kurio nedaug skiriasi *UBL-OrderResponse*. Kiti yra daug paprastesni, jie savyje saugo informaciją apie sutikimą arba nesutikimą su užsakymo sąlygomis, todėl jų nenagrinėsime.

UBL naudojamo užsakymo dokumento struktūra pateikiama 3.2.1 lentelėje.

3.2.1 lentelė. UBL-Order dokumento struktūra.

UBL elemento pavadinimas	UBL tipas	UBL klasė (sudėtinis tipas)	Kardin alumas	Aprašymas
UBLVersionID	Identifier		0..1	UBL 2 schemas, kuri aprašo visus šiame UBL-Order dokumente esančius elementus, versijos numeris.
CustomizationID	Identifier		0..1	Identifikuoja vartotojo nustatytus UBL pakeitimus reikalingus specifiniam panaudojimui.
ProfileID	Identifier		0..1	Identifikuoja vartotojo nustatytą specifinių pakeitimų profilį.
ID	Identifier		1	Pirkėjo priskirtas užsakymo identifikatorius.
SalesOrderID	Identifier		0..1	Pardavėjo priskirtas užsakymo identifikatorius.
CopyIndicator	Indicator		0..1	Identifikuoja ar šis užsakymas yra kopija ar ne.
UUID	Identifier		0..1	Universalus unikalus šio dokumento identifikatorius.
IssueDate	Date		1	Pirkėjo priskiriama šio užsakymo paskelbimo data.
IssueTime	Time		0..1	Pirkėjo priskiriama šio užsakymo paskelbimo laikas.
Note	Text		0..n	Laisvos formos tekstas, taikomas užsakymui. Šis elementas gali būti skirtas užrašams ar bet kuriai kitai panašiai informacijai, kuri nėra jokiam kitame elemente.

3.2.1 lentelės tęsinys.

UBL elemento pavadinimas	UBL tipas	UBL klasė (sudėtinis tipas)	Kardin alumas	Aprašymas
RequestedInvoice CurrencyCode	Currency Code		0..1	Sąskaitų-faktūrų susijusių su šiuo užsakymu valiuta.
DocumentCurren cy Code			0..1	Šio užsakymo valiuta.
PricingCurrency Code			0..1	Valiuta naudojama visoms kainoms užsakyme.
TaxCurrencyCod e			0..1	Su šiuo užsakymu susijos sąskaitos-faktūros reikalaujamų mokesčių valiuta.
CustomerReferen ce	Text		0..1	Papildoma nuoroda užsakymui.
AccountingCostC ode	Code		0..1	Pirkėjo sąskaitos kodas taikomas užsakymui.
AccountingCost	Text		0..1	Pirkėjo sąskaitos kodas taikomas užsakymui. Išreikštas tekstu
LineCountNumeri c	Numeric		0..1	Eilučių dokumente kiekis.
ValidityPeriod		Period	0..n	Užsakymo galiojimo laiko periodas.
Quotation Document Reference		Document Reference	0..1	Nuoroda į prekių katalogą.
Order Document Reference			0..n	Nuoroda į kitą Užsakymą
Originator Document Reference			0..1	Nuoroda į kūrėjo dokumentą.
Additional Document Reference			0..n	Nuoroda į papildomą dokumentą.
Contract		Contract	0..n	Sutartis
Signature		Signature	0..n	Parašas
BuyerCustomer Party		Customer Party	1	Pirkėjo šalies aprašymas
SellerSupplierPart y		Supplier Party	1	Pardavėjo šalies aprašymas
Originator CustomerParty		Customer Party	0..1	Užsakymo šalies pradininko aprašymas
Freight ForwarderParty		Party	0..1	Krovinio šalies gabentojo aprašymas

3.2.1 lentelės tęsinys.

UBL elemento pavadinimas	UBL tipas	UBL klasė (sudėtinis tipas)	Kardin alumas	Aprašymas
Accounting CustomerParty		Customer Party	0..1	Šalis kuriai turi būti nusiųsta sąskaita-faktūra.
Delivery		Delivery	0..n	Pristatymo aprašymas
DeliveryTerms		Delivery Terms	0..1	Pristatymo terminai.
PaymentMeans		Payment Means	0..1	Apmokėjimo būdai.
Transaction Conditions		Transaction Conditions	0..1	Visam užsakymui taikomos pirkimo ar pardavimo sąlygos.
AllowanceCharge		Allowance Charge	0..n	Visam užsakymui taikoma nuolaida.
DestinationCountry		Country	0..1	Paskirties šalis (muito mokesčio tikslais).
TaxTotal		Tax Total	0..n	Visa užsakymo suma (paskaičiuota pirkėjo).
Anticipated MonetaryTotal		Monetary Total	0..1	Pirkėjo mokama suma anksčiau laiko.
OrderLine		Order Line	1..n	Užsakymo eilute.

Iš 3.2.1 lentelėje aprašytos struktūros galima išskirti penkis pagrindinius būtinus užsakymo elementus:

- Užsakymo identifikatorius (*ID*);
- Užsakymo paskelbimo data (*IssueDate*);
- Pirkėjo šalies aprašas (*BuyerCustomerParty*);
- Pardavėjo šalies aprašas (*SellerSupplierParty*);
- Užsakymo eilutė (*OrderLine*).

Šių elementų buvimas UBL-Order dokumente leidžia atlikti užsakymą. Detalesni Pirkėjo ir Pardavėjo šalių, bei Užsakymo eilutės (sudėtingiausia duomenų struktūra) aprašymai pateikiami 9.2 priede.

Išanalizavus UBL dokumentų struktūras galima iškelti reikalavimus įmonės duomenų bazei, kad būtų galima įgyvendinti planuojamą užsakymo aptarnavimo

komponentą. Šios sudėtingos struktūros bus atvaizduotos į paprastesnius XML dokumentus ir tarnaus kaip tarpininkai tarp UBL ir įmonės duomenų bazės.

3.3.3 *Dalykinės srities modelis*

Pagal duomenų analizę galima išskirti tris sistemos esybes:

- Klientas – skirta informacijai apie klientą saugoti;
- Užsakymas- skirta užsakymo informacijai saugoti;
- Prekė – skirta prekių sąrašui saugoti.

Pastarųjų tarpusavio ryšiai ir rolės pavaizduojamos esybių ryšių diagrama, pateikiama žemiau (3.3.3.1 pav.):



3.3.3.1 pav. Sistemos esybių-ryšių modelis.

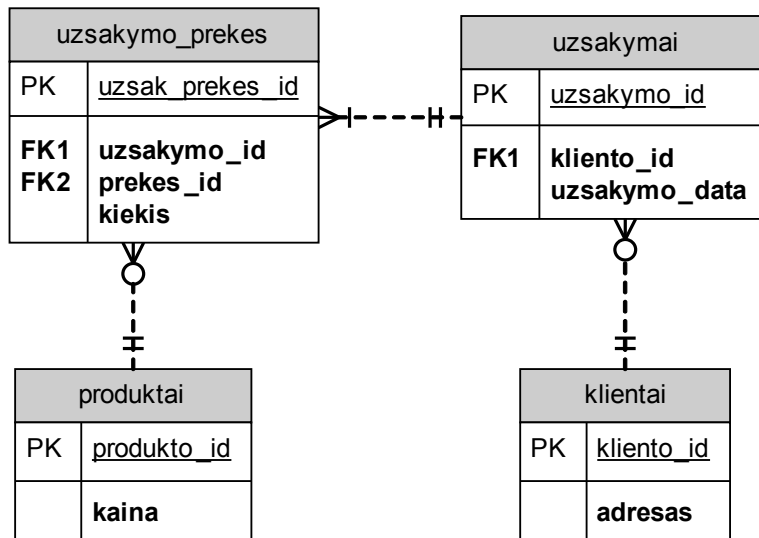
3.4. *Reikalavimų specifikacija*

3.4.1 Reikalavimai įmonės duomenų bazei

Remiantis *UBL-Order* dokumento elementų analize ir dalykinės srities modeliu galime suformuluoti tokius neformalius reikalavimus įmonės duomenų bazei (DB):

- Užsakymas turi turėti unikalų identifikatorių;
- Užsakymas saugoti informaciją apie tai, kada jis sukurtas;
- Turi būti galimybė į užsakymą įtraukti daug skirtingų prekių ir kiekių;
- Prekė turi būti unikaliai identifikuojama ir turėti kainą;
- Klientas (pirkėjas) turi turėti unikalų identifikatorių bei adresą;

Pagal šiuos minimalius reikalavimus galima išskirti saugomus atributus: *kliento_id*, *adresas*, *užsakymo_id*, *produkto_id*, *kaina*, *kiekis* ir suformuoti tokią normalizuotą duomenų bazės schemą (3.4.1.1 pav.):



3.4.1.1 pav. Primityvi užsakymų sistemos duomenų bazė schema.

Reikalavimai yra minimalūs, todėl veikiančių užsakymų sistemų duomenų bazių schemas turėtų juos tenkinti.

3.4.2 Sistemos funkcijos

Funkcinius reikalavimus pateiksime panaudos atvejų diagrama (3.4.2.1 pav.).

Sistemoje išskiriami du aktoriai:

- Pirkėjo IS – tai tiesioginis sistemos vartotojas, kuris pateikia užsakymus.
- Įmonės IS – tai įmonės informacinė sistema, kurioje bus diegiamas komponentas.



3.4.2.1 pav. Panaudos atvejų diagrama.

3.4.2.1 lentelė. Panaudojimo atvejo „Užsisakyti“ specifikacija.

1. PANAUDOJIMO ATVEJIS: Užsisakyti.	
Aktorius	Pirkėjo informacinė sistema (IS).
Aprašas	Apima procesą, kurio metu pateikiamas užsakymas sistemai, jis gali būti atšauktas arba gautas jo pakeitimas.
Prieš sąlyga	Pirkėjas žino savo identifikatorių sistemoje ir prekių katalogą.
Sužadinimo sąlyga	Pirkėjas nori pateikti užsakymą.
Po sąlyga	Sistemoje užregistruojamas naujas užsakymas ir žymima jo būsena.

3.4.2.2 lentelė. Panaudojimo atvejo „Pateikti užsakymą“ specifikacija.

2. PANAUDOJIMO ATVEJIS: Pateikti užsakymą.	
Aktorius	Pirkėjo informacinė sistema.
Aprašas	Apima užsakymo pateikimo procesą.
Prieš sąlyga	Pirkėjas žino savo identifikatorių sistemoje ir prekių katalogą.
Sužadinimo sąlyga	Įmonė pateikia užsakymą.
Po sąlyga	Sistemoje užregistruojamas naujas užsakymas. ir pažymimas kaip naujai registruotas.

3.4.2.3 lentelė. Panaudojimo atvejo „Atšaukti užsakymą“ specifikacija.

3. PANAUDOJIMO ATVEJIS: Atšaukti užsakymą.	
Aktorius	Pirkėjo informacinė sistema.
Aprašas	Apima užsakymo atšaukimo procesą.
Prieš sąlyga	Pirkėjas prieš tai buvo pateikęs užsakymą.
Sužadinimo sąlyga	Įmonė pateikia užsakymo atšaukimą.
Po sąlyga	Sistemoje užsakymas pažymimas kaip atšauktas.

3.4.2.4 lentelė. Panaudojimo atvejo „Pakeisti užsakymą“ specifikacija.

4. PANAUDOJIMO ATVEJIS: Pakeisti užsakymą.	
Aktorius	Pirkėjo informacinė sistema.
Aprašas	Apima užsakymo pakeitimo procesą.
Prieš sąlyga	Pirkėjas prieš tai buvo pateikęs užsakymą ir nori padaryti pakeitimų. Užsakymas negali būti priimtas.
Sužadinimo sąlyga	Įmonė pateikia užsakymo pakeitimą.
Po sąlyga	Sistemoje užsakymas pakeičiamas ir pažymimas kaip pakeistas.

3.4.2.5 lentelė. Panaudojimo atvejo „Aptarnauti užsakymą“ specifikacija.

5. PANAUDOJIMO ATVEJIS: Aptarnauti užsakymą.	
Aktorius	Įmonės informacinė sistema (IS).
Aprašas	Procesas, kuriuo metu vyksta pakeitimai įmonės duomenų bazėje, ir registruojami pakeitimai susiję su užsakymų aptarnavimu.
Prieš sąlyga	Nėra
Sužadinimo sąlyga	Pirkėjo IS pateikia atlieka užsakymą.
Po sąlyga	Įmonės IS įvyksta pakeitimai susiję su užsakymų aptarnavimu.

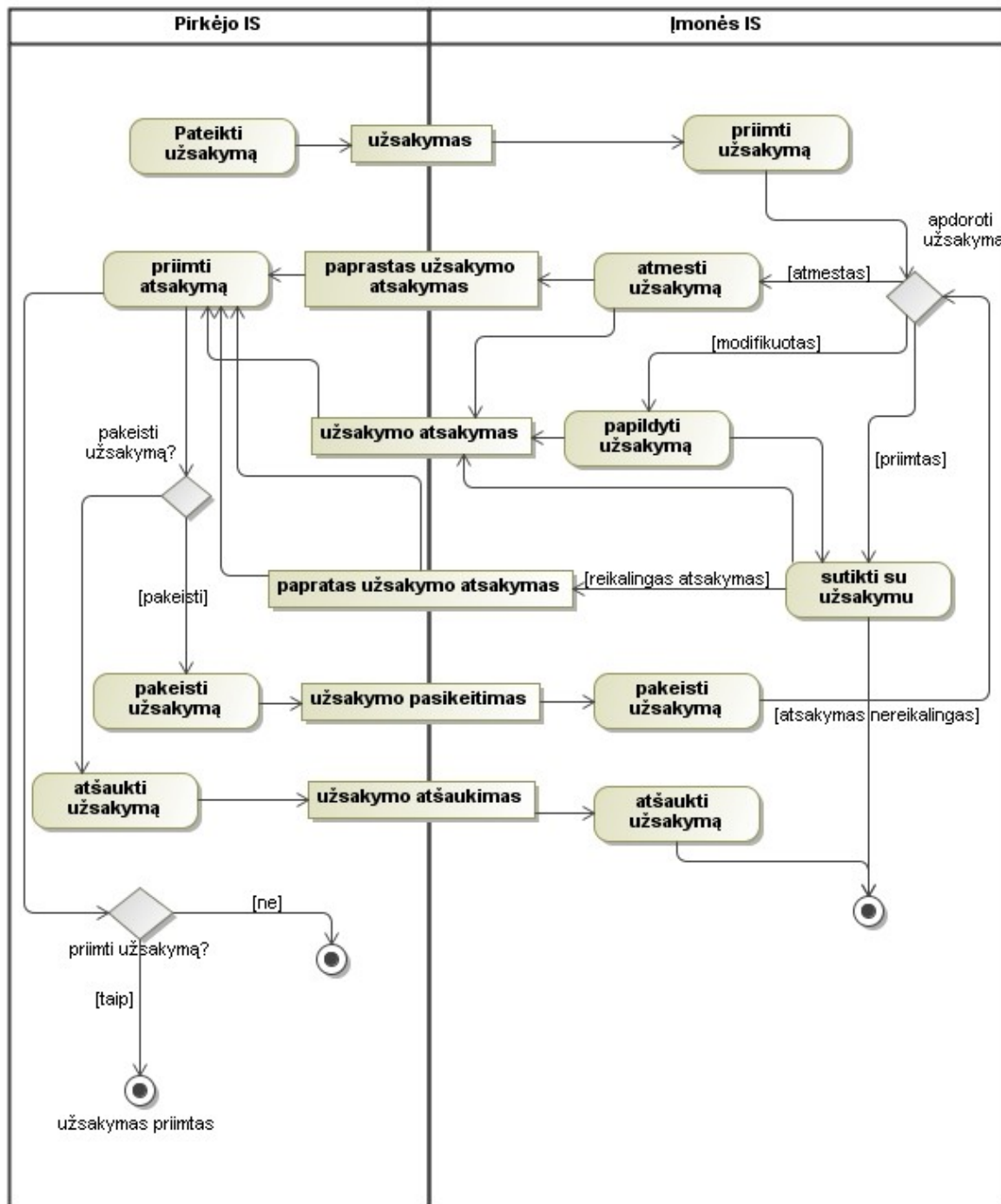
3.4.2.6 lentelė. Panaudojimo atvejo „Priimti užsakymą“ specifikacija.

6. PANAUDOJIMO ATVEJIS: Priimti užsakymą.	
Aktorius	Įmonės informacinė sistema (IS).
Aprašas	Procesas, kuriuo metu vyksta Įmonės IS registruojamas naujas užsakymas.
Prieš sąlyga	Nėra.
Sužadinimo sąlyga	Pirkėjo IS pateikia užsakymą.
Po sąlyga	Įmonės IS užregistruojamas naujas užsakymas

3.4.2.7 lentelė. Panaudojimo atvejo „Pakeisti užsakymo būseną“ specifikacija.

7. PANAUDOJIMO ATVEJIS: Pakeisti užsakymo būseną.	
Aktorius	Įmonės informacinė sistema (IS).
Aprašas	Procesas, kuriuo metu pakeičiama užsakymo būseną Įmonės IS.
Prieš sąlyga	Nėra.
Sužadinimo sąlyga	Pirkėjo IS pateikia užsakymą, jo atšaukimą, arba pakeitimą.
Po sąlyga	Įmonės IS užregistruojamas užsakymo būsenos pakeitimas.

Užsakymo priėmimą, atšaukimą ir pakeitimą apimanti veiklos diagrama (3.4.2.2 pav.):



3.4.2.2 pav. Užsakymo priėmimą, atšaukimą ir pakeitimą apimanti veiklos diagrama.

Procesai „atmesti užsakymą“, „apdoroti užsakymą“, „sutikti su užsakymu“ yra ne kuriamo komponento kontekste. El. sistema negali spręsti ar priimti užsakymą ar jį atmesti. Šie sprendimai paliekami įmonės darbuotojams, ar veikiančioms sistemoms.

Taigi iš panaudos atvejų (3.4.2.1. Pav.) ir veiklos (3.4.2.2 pav.) diagramų ir galima išskirti pagrindines užsakymo aptarnavimo komponento funkcijas:

- priimti užsakymą;
- pakeisti užsakymą;
- atšaukti užsakymą.

3.4.3 Kiti funkciniai reikalavimai

1. Turi būti galimybė nustatyti užsakymų būseną.
2. Įmonės duomenų bazėje turi būti sukurta papildoma lentelė užsakymų būsenai saugoti, kurios struktūra pateikiama žemiau:

3.4.3.1 lentelė. Papildomos lentelės struktūra.

Atributas	Būtinas	Tipas	Aprašymas
<u>uuid</u>	taip	string	Raktinis atributas, kurio reikšmė yra <i>UBL Order</i> dokumento <i>UUID</i> reikšmė, unikalčiai identifikuojanti UBL dokumentą.
užsakymo id	taip	string	Atributas saugoti užsakymo įmonės IS raktinę reikšmę.
UBL id	ne	string	Atributui iš UBL dokumentų <i>SalesOrderID</i> saugoti.
data	taip	date	Atributas būsenos registravimo, pakeitimo datai saugoti.
komentaras	ne	string	Atributas komentarui apie būsenos pakeitimus saugoti.
būsena	taip	integer	Atributas skirtas būsenai saugoti

3. Papildomos lentelės *būsenos* atributo galimos reikšmės ir prasmės:

- 1 – naujas užsakymas;
- 2 – užsakymas priimtas;
- 3 – užsakymas atmestas;
- 4 – užsakymas ištrintas;
- 5 – užsakymas pakeistas.

4. Turi būti įmanoma priderinti įvairių datos formatą naudojama UBL dokumentuose prie įmonės duomenų bazėje naudojamo datos formato.

3.4.4 Apribojimai ir nefunkciniai *reikalavimai*

Sistema turi tenkinti šiuos nefunkcinius reikalavimus ir apribojimus:

- Komponentas turi būti realizuotas Java aplinkoje.
- Duomenų mainams naudoti UBL standartą.
- Komponentą sukurti taip, kad jo diegimas įmonėje nereikalautų esamos duomenų struktūros keitimo.
- Komponentas turi būti realizuotas paslaugoms orientuota architektūra, t.y., suteikti galimybę užsakovams užsisakyti prekes tiesiogiai nedalyvaujant tiekėjui.
- Sukurtas komponentas turi veikti su SOAP 1.1 ir SOAP 1.2 protokolais.
- Duomenų transformacijoms naudoti XSL transformacijas.
- XSL transformacijos turi būti saugomos komponento išorėje.
- Komponento diegimas, atmetus XSL transformacijų rašymą, turi būti lengvas, t.y., trukti ne ilgiau nei darbo dieną, įmonės informacinę sistemą prižiūrinčiam darbuotojui.
- Visi komponento nustatymai turi būti saugomi vienoje byloje.
- Vienu metu vartotojas gali pateikti iki 5 užsakymų.
- Sistemos reakcijos laiką į užklausą negali viršyti 5 sekundžių.
- Papildomos lentelės ir atributų pavadinimai (žr. 3.4.3 skyrelį) gali būti bet kokie, atitinkantys SQL vardų reikalavimus.

3.4.5 Duomenų ir rezultatų specifikacija

Kadangi yra nefunkcinis reikalavimas „naudoti UBL standartą“, tai duomenų specifikaciją yra UBL dokumentų rinkinys, kuri sudaro toki dokumentai:

3.4.5.1 lentelė. Duomenų specifikacija

Duomenų srauto tipas	Apibūdinimas	Susiję procesai	Gavėjas	Siuntėjas
Užsakymas	Dokumentas turintis informaciją tiesiogiai susijusią su produktų užsakymu.	Užsakymo pateikimas	Sistema	Pirkėjo IS
Užsakymo pasikeitimas	Dokumentas turintis informaciją tiesiogiai susijusią su produktų užsakymo pakeitimu.	Pakeisti užsakymą	Sistema	Pirkėjo IS
Užsakymo atšaukimas	Dokumentas patvirtinantis užsakymo atšaukimą	Užsakymo atšaukimas	Sistema	Pirkėjo IS

Rezultatų specifikacija pateikiama 3.4.5.2 lenteleje:

3.4.5.2 lentelė. Rezultatų specifikacija

Duomenų srauto tipas	Apibūdinimas	Susiję procesai	Gavėjas	Siuntėjas
Atsakymas	Atsakymas į pirkėjo užklausa, TAIP arba NE (<i>boolean</i>)	Užsakymo pateikimas, pakeitimas, atšaukimas.	Pirkėjo IS	Sistema
Astakymo klaida	Pranešimas apie klaidą, kodėl buvo atmesta pirkėjo užklausa.	Užsakymo pateikimas, pakeitimas, atšaukimas.	Pirkėjo IS	Sistema

3.4.6 Vartotojui (pirkėjui) keliami reikalavimai

1. Visi *UBL Order* dokumentai turi turėti UUID atributą, kurio struktūra tokia:

XXXX-YYYY-ZZZZZZZZ, čia:

XXXX - įmonės IS saugomas pirkėjo ID (ilgis neturi viršyti 10 simbolių)

YYYY - pirkėjo unikalus užsakymo identifikatorius (ilgis neturi viršyti 20 simbolių.)

ZZZZZZZZ - užsakymo data formatu yyyyMMDD.

2. Pirkėjas privalo užpildyti *IssueDate* UBL elementą.

3. Pirkėjas privalo pateikti prekių pristatymo adresą arba *DeliveryAddress* arba *BuyerCustomerParty* elementuose.
4. Pirkėjas privalo pateikti identifikatorių įmonės IS *SupplierAssignedAccountID* elementą dokumente.
5. Pirkėjas turi žinoti įmonės prekių katalogą su jas identifikuojančiomis raktinėmis reikšmėmis ir jas užrašyti UBL dokumentų *SellersItemIdentification* elemente.
6. Pirkėjas turi būti registruotas įmonės informacinėje sistemoje.
7. Pirkėjas kartu su UBL dokumentu visada turi siusti tokias dvi SOAP antraštes:
 - 7.1. FROM – nurodo savo įmonės pavadinimą
 - 7.2. TO – nurodo įmonės kuriai siunčia pavadinimą.

3.5. Techninė specifikacija

Dėl nefunkcinio reikalavimo „Komponentas turi būti realizuotas Java aplinkoje“, tarnybinės stoties taikomųjų programų serveriui minimalūs reikalavimai turi būti tokie:

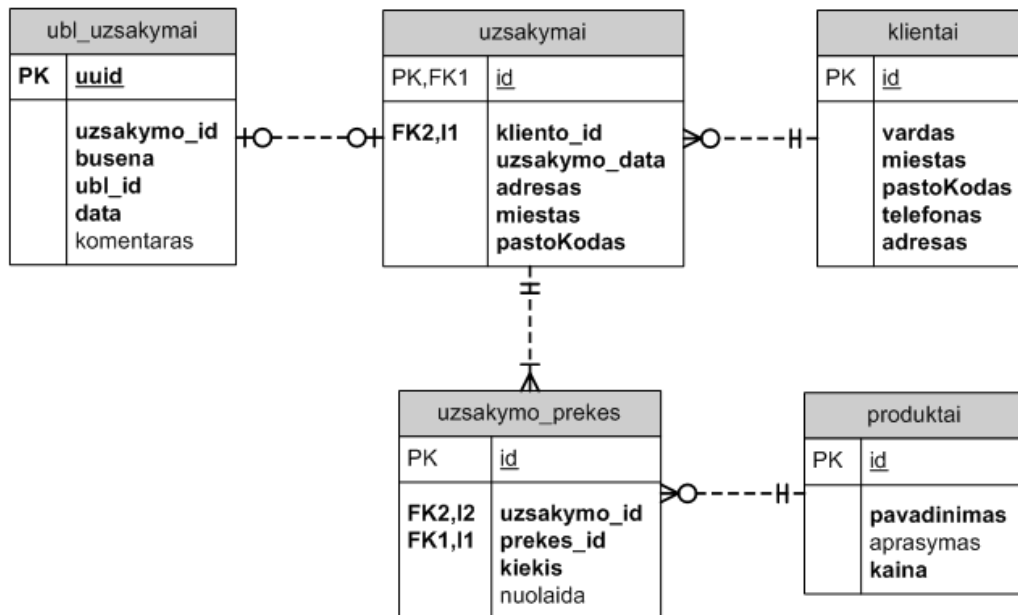
- Įdiegta Java Runtime Environment 5.0,
- Nustatytas JAVA_HOME aplinkos kintamasis, rodantis į Java instaliacijos vietą
- 1 GHz ir greitesnis procesorius,
- 512 MB atminties,
- Laisvos vietos kietajame diske: 400 MB.

Sistemos naudotojams keliami reikalavimai nėra apibrėžiami, jie priklauso nuo vartotojo sistemos sudėtingumo ir pasirinktų technologijų pasinaudojant projektuojamu užsakymo apdorojimo komponentu. Pagrindinis reikalavimas tai interneto ryšys, kurio sparta būtų pakankama tam, kad būtų galima įgyvendinti elektroninius užsakymus.

4. Užsakymo aptarnavimo komponento projektas

4.1. Duomenų bazės modelis

Pagal 3.4.1 skyriuje sudarytą minimalų duomenų bazės modelį, realizacijai buvo naudojamas modifikuotas DB modelis (4.1.1 pav.). Jis papildytas naujais atributais, tam, kad atitiktų realios sistemos fragmentą, susijusį su užsakymų aptarnavimu. Modelis papildytas lentele atsižvelgiant 3.4.3 skyriuje aprašytus reikalavimus.



4.1.1 pav. Įmonės duomenų bazės modelis.

Duomenų bazės modelio aprašymas pateiktas žemiau esančiose lentelėse

4.1.1 lentelė. Duomenų bazės lentelės *ubl_uzsakymai* aprašymas.

Atributas	Būtinis	Tipas	Aprašymas
uuid	taip	varchar	Unikalus UBL dokumento identifikatorius. <i>UUID</i> reikšmė.
uzsakymo_id	taip	varchar	Šąsajos su užsakymu atributas.
busena	taip	int	Užsakymo būsenos saugojimas .
ubl id	ne	varchar	UBL užsakymo <i>ID</i> elemento reikšmė.
data	taip	datetime	Užsakymo būsenos keitimo data ir laikas.
komentaras	ne	varchar	užsakymo pašalinimo arba pakeitimo komentaras.

4.1.2 lentelė. Duomenų bazės lentelės *uzsakymai* aprašymas.

Atributas	Būtinasis	Tipas	Aprašymas
id	taip	int	Unikalus užsakymo identifikatorius. Generuojamas automatiškai
kliento_id	taip	int	Kliento identifikatorius.
uzsakymo_data	taip	date	Užsakymo pateikimo data.
adresas	taip	varchar	Užsakymo prekių pristatymo adresas.
miestas	taip	varchar	Užsakymo prekių pristatymo miestas
pastoKodas	taip	varchar	Atributas skirtas saugoti pašto kodui.

4.1.3 lentelė. Duomenų bazės lentelės *klientai* aprašymas.

Atributas	Būtinasis	Tipas	Aprašymas
id	taip	int	Unikalus kliento identifikatorius. Generuojamas automatiškai.
vardas	taip	varchar	Kliento pavadinimas (įmonės pavadinimas).
miestas	taip	varchar	Miestas kuriame įsikūrusi įmonės būstinė
pastoKodas	taip	varchar	Įmonės būstinės pašto kodas
telefonas	taip	varchar	Kontaktinio asmens telefonas
adresas	taip	varchar	Įmonės būstinės adresas

4.1.4 lentelė. Duomenų bazės lentelės *produktai* aprašymas.

Atributas	Būtinasis	Tipas	Aprašymas
id	taip	int	Unikalus prekės identifikatorius.
pavadinimas	taip	varchar	Prekės pavadinimas.
aprasymas	ne	varchar	Prekės aprašymas.
kaina	taip	real	Prekės kaina.

4.1.5 lentelė. Duomenų bazės lentelės *uzsakymo_prekes* aprašymas.

Atributas	Būtinasis	Tipas	Aprašymas
id	taip	int	Unikalus užsakymo eilutės identifikatorius. Generuojamas automatiškai.
uzsakymo_id	taip	int	Susijusio užsakymo identifikatorius.
prekes_id	taip	int	Prekės identifikatorius.
kiekis	taip	real	Prekių kiekis.
nuolaida	ne	real	Taikoma nuolaida prekėms.

4.2. XSL transformacijos

Pirmiausiai reiktų pažymėti, kad XSL transformacijoms buvo naudojama 1.0 XSL versija. Naujausia yra 2.0, tačiau jai nėra stabilus nemokamo XSL variklio Java platformoje. Transformacijos, tai paprasčiausios XML bylos, kuriose aprašytos taisyklės, kaip vieną dokumentą atvaizduoti į kitą. XSL transformacijų byla žymima šakniniu elementu:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Toliau jame aprašomos taisyklės. Taisyklės pavyzdys:

```
<kliento_id>
<xsl:value-of select="//cbc:SupplierAssignedAccountID"/>
</kliento_id>
```

Ši taisyklė reiškia, kad sukuriamas elementas, o jo turinys paimamas iš UBL dokumento `SupplierAssignedAccountID` elemento. Elementai aprašomi XPath kalba. Taip pat, prieš elementą dar yra sutrumpinimas `cbc` reiškiantis vardų erdvę kuriai priklauso elementas (žr. 4.2.1 skyrių). Plačiau apie XSL transformacijas galima skaityti <http://www.w3.org/Style/XSL/>.

4.2.1 Vardų erdvės

Bendrai paėmus, XML bylose elementai su tais pačiais vardais gali būti skirtingai interpretuojami. Tam, kad elementas vienareikšmiai būtų suprastas, naudojamos vardų erdvės (angl. *namespace*). Užsakymo aptarnavimo komponento naudojamos vardų erdvės yra:

- `xmlns="http://service.ubl.ak/rowset"`
- `xmlns:col="http://service.ubl.ak/collection"`

UBL dokumentai aprašyti naudojant šias vardų erdves, kurias reikia įtraukti į kiekvieną transformaciją:

- `xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDatatypes-2"`
- `xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentParameters-2"`
- `xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"`

- xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
- xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaModule:2"
- xmlns:ord="urn:oasis:names:specification:ubl:schema:xsd:Order-2"

Be jau aprašytų reikalingos ir dar kelios papildomos vardų erdvės pačiai transformacijai atlikti:

- xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
- xmlns:xs="http://www.w3.org/2001/XMLSchema"
- xmlns:fn="http://www.w3.org/2005/xpath-functions"

Taigi, apjungus erdves gauname tokią XSL transformacijos antraštę:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns="http://service.ubl.ak/rowset" xmlns:col="http://service.ubl.ak/
collection"
xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDataTypes-2"
xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentParameters-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaModule:2"
xmlns:ord="urn:oasis:names:specification:ubl:schema:xsd:Order-2">
```

4.2.2 Datų formatai

UBL schemoje, datų formatai gali būti įvairūs pvz.: 2008-10-12, arba 121008, todėl, kad juos suprastų įmonės informacinė sistema, reikia datas transformuoti į vieną bendrą formatą, kuris yra užrašytas nustatymų byloje, pvz.: *yyyyMMDD* – reikštų 20081012 datą.

4.2.3 Transformuotų UBL dokumentų struktūra

UBL-Order transformuojamas į tokią XML struktūrą:

```

<ROWSET xmlns="http://service.ubl.ak/rowset"
xmlns:col="http://service.ubl.ak/collection"
table="užsakymo lentelės pavadinimas"
uuid="uuid reikšmė"
buyer_id="pardavejo užsakymo reikšmė">
  <ROW num="1">
    <užsakymo_atributas>reikšmė</užsakymo_atributas>
    <užsakymo_atributas>reikšmė</užsakymo_atributas>
    ...
    <col:COLLECTION table="uzsakymo_prekiu_lent_pavadinimas"
relation="rysio_su_uzsakymo_lentele_atributas">
      <col:ITEM>
        <atributo_vardas>reikšmė</atributo_vardas>
        <atributo_vardas>reikšmė</atributo_vardas>
      </col:ITEM>
      <col:ITEM>
        <atributo_vardas>reikšmė</atributo_vardas>
        <atributo_vardas>reikšmė</atributo_vardas>
      </col:ITEM>
      ...
    </col:COLLECTION>
  </ROW>
</ROWSET>

```

Šakninis elementas yra *ROWSET*, vienam užsakymui jis turi viena *ROW* elementą. *ROWSET* atributas *table* yra lentelės kurioje saugomi užsakymai pavadinimas, *uuid* – saugomas UBL dokumento *UUID* elementas, o *buyer_id* – *SalesOrderID*. *ROW* elemente surašomi užsakymo lentelės atributai ir jų reikšmės. Surašyti turi būti visi atributai, kurie yra privalomi lentelėje. Jeigu raktinis atributas generuojamas automatiškai jo rašyti nereikia.

ROW elemente kolekcijos (užsakymo prekės) surašomos į *COLLECTION* elementą. Pastaba: vardų sritis yra "http://service.ubl.ak/collection". Užsakymo prekės aprašomos *ITEM* elementais, prekės atributai surašomi analogiškai kaip *ROW* elemente. *COLLECTION* atributas *table* nurodo, kurioje lentelėje saugomos užsakymo prekių įrašai, atributas *relation* nurodo atributo vardą, kuriuo susietos užsakymo prekės ir užsakymas. *ITEM* elemente turi būti išvardinami visi privalomi lentelės atributai.

Visos elementų reikšmės imamos iš atitinkamų UBL-Order dokumento elementų. Žemiau pateikiamas struktūros pavyzdys, pritaikytas duomenų bazės schemai aprašyti 4.1 skyriuje:

```

<ROWSET xmlns="http://service.ubl.ak/rowset"
xmlns:col="http://service.ubl.ak/collection" table="uzsakymai"
uuid="191-U195-20081122"
buyer_id="U195">

```

```

    <ROW num="1">
      <kliento_id>1</kliento_id>
      <uzsakymoData>20081120</uzsakymoData>
      <pristatymoData>20081122</pristatymoData>
      <adresas>PIEVU G. 12</adresas>
      <miestas>Jonava</miestas>
      <pastoKodas>78941</pastoKodas>
      <col:COLLECTION table="uzsakymo_prekes"
        relation="uzsakymo_id">
        <col:ITEM>
          <kiekis>100</kiekis>
          <prekes_id>1</prekes_id>
        </col:ITEM>
        <col:ITEM>
          <kiekis>10</kiekis>
          <prekes_id>2</prekes_id>
        </col:ITEM>
      </col:COLLECTION>
    </ROW>
  </ROWSET>

```

UBL-OrderChange dokumentas transformuojamas taip pat kaip ir UBL-Order, tačiau ROWSET elementas papildomai turi *key_column* atributą, kuriame saugomas užsakymo lentelės rako pavadinimas, ir *update_columns*, kuriame saugomi kableliu atskirti užsakymo lentelės atributai. ROWSET taip pat turi papildomą NOTE elementą, kuriame paprastu tekstu saugomas pakeitimo aprašymas. Pastaba *uuid* reikšmė yra nuoroda į užsakymą kuris yra keičiamas. Pasinaudojus papildoma lentele, sistema atskiria kurį užsakymą reikia keisti.

Transformuoto UBL-OrderChange pavyzdys:

```
<ROWSET xmlns="http://service.ubl.ak/rowset"
xmlns:col="http://service.ubl.ak/collection table="uzsakymai"
uuid="191-U195-20081122"
buyer_id="U195"
key_column="id"
update_columns="kliento_id,uzsakymo_data,adresas,miestas,
pastoKodas">
  <NOTE>
    Pakeičiami prekių kiekiai.
  </NOTE>
  <ROW num="1">
    <kliento_id>1</kliento_id>
    <uzsakymoData>20081120</uzsakymoData>
    <pristatymoData>20081122</pristatymoData>
    <adresas>PIEVU G. 12</adresas>
    <miestas>Jonava</miestas>
    <pastoKodas>78941</pastoKodas>
    <col:COLLECTION table="uzsakymo_prekes"
      relation="uzsakymo_id">
      <col:ITEM>
        <kiekis>50</kiekis>
        <prekes_id>1</prekes_id>
      </col:ITEM>
      <col:ITEM>
        <kiekis>5</kiekis>
        <prekes_id>2</prekes_id>
      </col:ITEM>
    </col:COLLECTION>
  </ROW>
</ROWSET>
```

UBL-OrderCancellation dokumentas formuojamas panašiai, jo šablonas pateikiamas žemiau:

```
<CANCELLATION order_uuid="uuid nuoroda į užsakymą">
  <NOTE>
    Atšaukimo priežastis
  </NOTE>
</CANCELLATION>
```

Čia šakninis elementas yra *CANCELLATION*, jis turi vienintelį atributą *order_uuid*, kurio reikšmė turi būti norimo atšaukti užsakymo *uuid*. *NOTE* elemente aprašoma, atšaukimo priežastis.

4.2.4 XSL transformacijų šablonai įmonės duomenų bazei.

Iš viso reikalingos keturios XSL transformacijos, trys jų aprašytos aukščiau ir viena tarpinė transformacija, kurios modifikuoti adaptuojant komponentą kitai įmonei nereikia nereikia:

1. Užsakymo (UBL-Order):

```
<xsl:stylesheet version="1.0">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <ROWSET table="uzsakymai">
      <xsl:attribute name="uuid"><xsl:value-of
select="//cbc:UUID"/></xsl:attribute>
      <xsl:attribute name="buyer_id"><xsl:value-of
select="//cbc:ID"/></xsl:attribute>
      <xsl:call-template name="root"/>
    </ROWSET>
  </xsl:template>
  <xsl:template name="root" match="/ord:Order">
    <ROW num="1">
      <kliento_id>
        <xsl:value-of
select="//cbc:SupplierAssignedAccountID"/>
      </kliento_id>
      <uzsakymo_data>
        <xsl:value-of select="//cbc:IssueDate"/>
      </uzsakymo_data>
      <adresas>
        <xsl:value-of
select="//cac:DeliveryAddress/cac:AddressLine/cbc:Line"/>
        <xsl:text>, </xsl:text>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:StreetName"/>
        <xsl:text> </xsl:text>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:BuildingNumber"/>
      </adresas>
      <miestas>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:CityName"/>
      </miestas>
      <pastoKodas>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:PostalZone"/>
      </pastoKodas>
      <col:COLLECTION table="uzsakymo_prekes"
relation="uzsakymo_id">
        <xsl:for-each select="//cac:OrderLine/cac:LineItem">
          <col:ITEM>
            <kiekis>
              <xsl:value-of
select="./cbc:Quantity"/>
            </kiekis>
            <prekes_id>
              <xsl:value-of
select="./cac:Item/cac: SellersItemIdentification/cbc:ID"/>
            </prekes_id>
          </col:ITEM>
        </xsl:for-each>
      </col:COLLECTION>
    </ROW>
  </xsl:template>

```

```

        </col:COLLECTION>
    </ROW>
</xsl:template>
</xsl:stylesheet>

```

2. Užsakymo atšaukimo (UBL-OrderCancellation):

```

<xsl:stylesheet version="1.0">
  <xsl:output method="xml" version="1.0" encoding="UTF-8"
indent="yes"/>
  <xsl:template match="/">
    <CANCELLATION>
      <xsl:attribute name="order_uuid"><xsl:value-of
select="//cac:OrderReference/cbc:UUID"/></xsl:attribute>
      <NOTE>
        <xsl:value-of select="//cbc:CancellationNote"/>
      </NOTE>
    </CANCELLATION>
  </xsl:template>
</xsl:stylesheet>

```

3. Užsakymo pakeitimo (UBL-OrderChange):

```

<xsl:stylesheet version="1.0">
  <xsl:output method="xml" version="1.0" encoding="UTF-8"
indent="yes"/>
  <xsl:template match="/">
    <ROWSET table="uzsakymai" key_column="id"
update_columns="kliento_id,uzsakymo_data,adresas,miestas,pastoKodas">
      <xsl:attribute name="uuid"><xsl:value-of
select="//cac:OrderReference/cbc:UUID"/></xsl:attribute>
      <NOTE>
        <xsl:value-of
select="//ord:OrderChange/cbc:Note"/>
      </NOTE>
      <xsl:call-template name="root"/>
    </ROWSET>
  </xsl:template>
  <xsl:template name="root" match="//ord:OrderChange">
    <ROW num="1">
      <id/>
      <kliento_id>
        <xsl:value-of
select="//cbc:SupplierAssignedAccountID"/>
      </kliento_id>
      <uzsakymo_data>
        <xsl:value-of select="//cbc:IssueDate"/>
      </uzsakymo_data>
      <adresas>
        <xsl:value-of
select="//cac:DeliveryAddress/cac:AddressLine/cbc:Line"/>
        <xsl:text>, </xsl:text>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:StreetName"/><xsl:text> </xsl:text>
        <xsl:value-of
select="//cac:DeliveryAddress/cbc:BuildingNumber"/></adresas>

```



```

        <miestas>
            <xsl:value-of
select="//cac:DeliveryAddress/cbc:CityName"/></miestas>
            <pastoKodas>
                <xsl:value-of
select="//cac:DeliveryAddress/cbc:PostalZone"/></pastoKodas>
                <col:COLLECTION table="uzsakymo_prekes"
relation="uzsakymo_id">
                    <xsl:for-each
select="//cac:OrderLine/cac:LineItem">
                        <col:ITEM>
                            <kiekis>
                                <xsl:value-of select="./cbc:Quantity"/>
                            </kiekis>
                            <prekes_id>
                                <xsl:value-of
select="./cac:Item/cac:SellersItemIdentification/cbc:ID"/>
                                </prekes_id>
                            </col:ITEM>
                        </xsl:for-each>
                    </col:COLLECTION>
                </ROW>
            </xsl:template>
</xsl:stylesheet>

```

4. Užsakymo skaidymo (vidinė transformacija, naudojama atskirti užsakymą ir užsakymo eilutės prekes):

```

<xsl:stylesheet version="1.0"
  <xsl:output method="xml" version="1.0" encoding="UTF-8"
indent="yes"/>
  <xsl:template match="/">
    <ROWSET>
      <xsl:attribute name="table"><xsl:value-of
select="//col:COLLECTION/@table"/></xsl:attribute>
      <xsl:attribute name="relation"><xsl:value-of
select="//col:COLLECTION/@relation"/></xsl:attribute>
      <xsl:apply-templates select="//col:COLLECTION"/>
    </ROWSET>
  </xsl:template>
  <xsl:template match="//col:COLLECTION/col:ITEM">
    <ROW>
      <xsl:copy-of select="*" />
    </ROW>
  </xsl:template>
</xsl:stylesheet>

```

Pastaba. Visų transformacijų šakniniai elementai *xsl:stylesheet* yra sutrumpinti. Jis visoms transformacijoms vienodas ir aprašytas 4.2.1 skyriuje.

4.3. Komponento sąsajos aprašymas

Pagal reikalavimų analizėje iškirtas pagrindines sistemos funkcijas buvo realizuotas komponentas. Šio komponento sąsaja aprašyta WSDL – žiniatinklio paslaugų aprašymo kalba.

Tam, kad lengviau būtų suprantama (WSDL dokumentas yra XML kalba) sąsaja, jos aprašymą galima pavaizduoti diagrama (4.3.1 pav.). Joje vaizduojami esminiai paslaugos elementai.

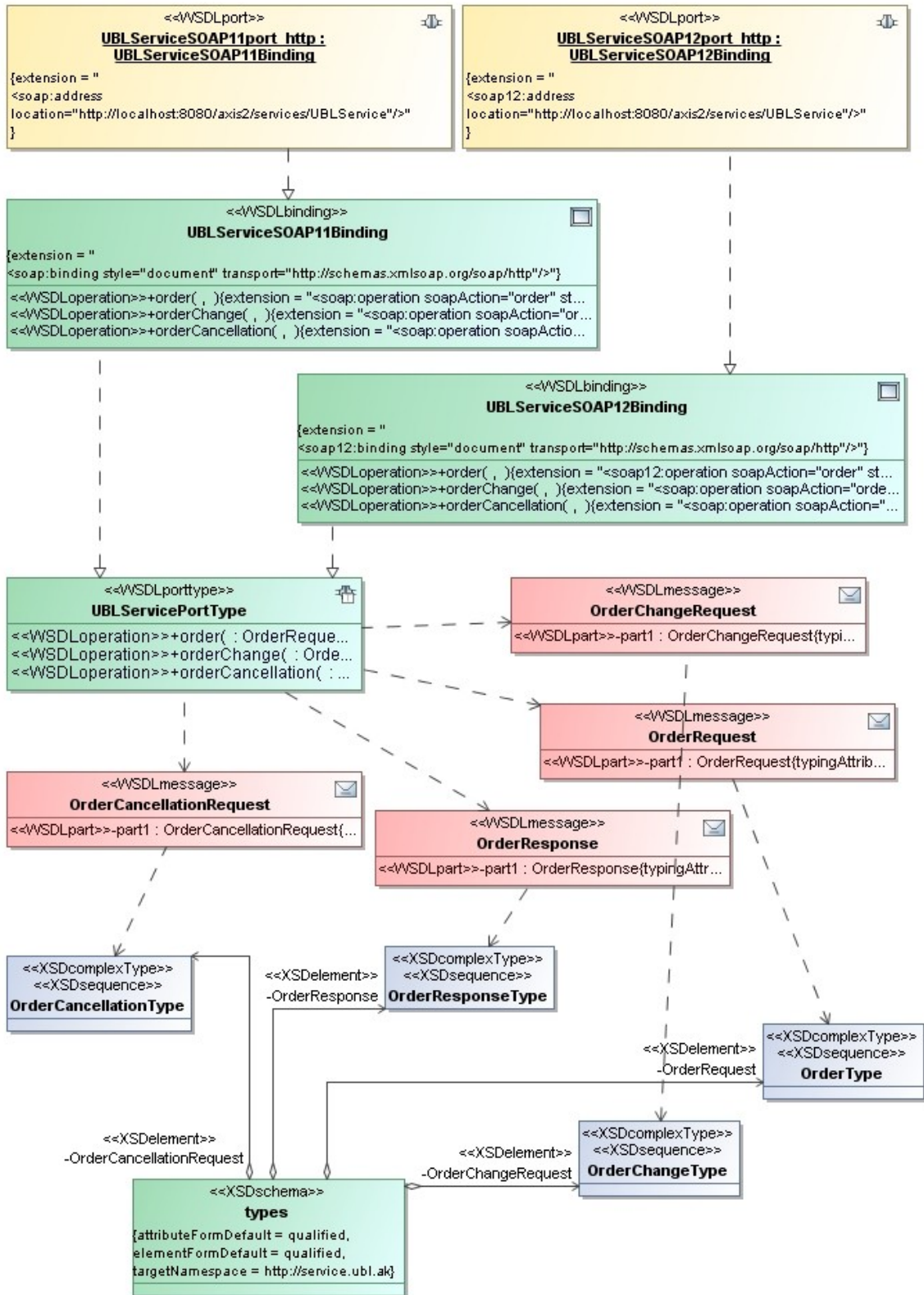
Taigi pradėsime nuo žemiausio lygmens – duomenų tipų. Duomenų tipai aprašomi XSD schema (*XSDschema* stereotipo elementas diagramoje). Yra keturi duomenų tipai: *OrderType*, *OrderCancellationType*, *OrderChangeType* ir *OrderResponseType*. Primieji trys yra skirti duomenims priimti (UBL dokumentai), o paskutinis – išsiųsti (patvirtinimas/paneigimas apie sėkmingą užklausą). Jie vaizduojami *XSDcomplextyp*e stereotipo elementais.

Šie duomenų tipai naudojami pranešimuose, diagramoje pavaizduoti elementai su stereotipu *WSDLmessage*. Kiekvienam pranešimui po vieną duomenų tipą.

Sąsajos operacijos aprašomos prievado tipu (angl. *port type*), yra trys operacijos: *order*, *orderChange* ir *orderCancellation*, šių operacijų parametrai yra atitinkami pranešimai: *OrderRequest*, *OrderChangeRequest* ir *OrderCancellationRequest*. Visų operacijų grąžinama reikšmė yra *OrderResponse*. Šis elementas pavaizduotas *WSDLporttype* stereotipo elementu.

Kylant aukšty, prievadai realizuoti SOAP 1.1 ir SOAP 1.2 protokolų saistais (*WSDLbinding* stereotipo elementai). O pastarieji turi savo prievadus, kurie pasiekiami adresu <http://localhost:8080/axis2/services/UBLService>.

Reiktų paminėti, kad pagal *duomenų ir reikalavimų specifikaciją* aprašyti duomenų tipai, UBL dokumentai perduodami MTOM būdu, tai optimizuotas duomenų perdavimo būdas naudojantis SOAP protokolą.

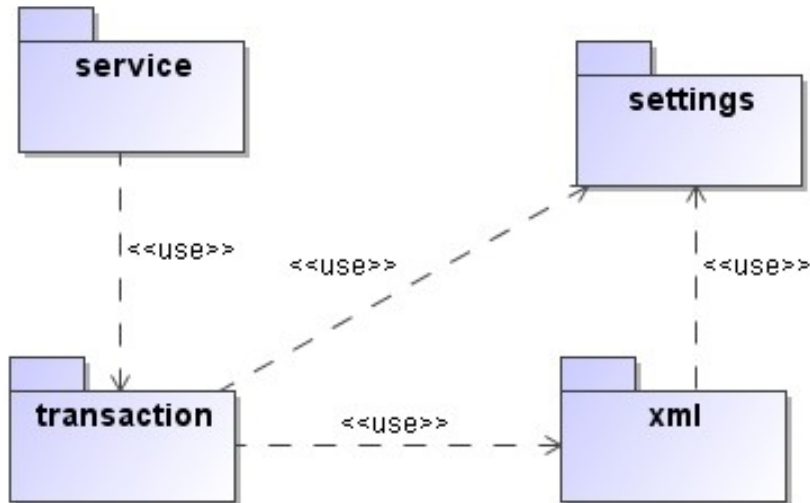


4.3.1 pav. Užsakymo aptarnavimo komponento WSDL diagrama.

4.4. Sistemos architektūra

Užsakymo aptarnavimo komponento architektūra pavaizduota paketų diagrama

4.4.1 pav.:



4.4.1 pav. Paketų diagrama.

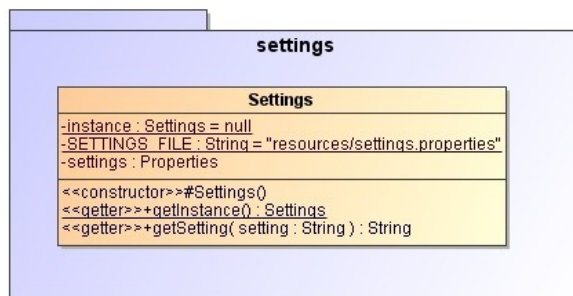
Sistemoje skiriami 4 paketai:

1. *service* – skirtas paslaugos sąsajai įgyvendinti.
2. *transaction* – skirtas valdyti užsakymo transakcijoms, talpinti XML dokumentus SQL bazėse. Jį naudoja *service* paketas.
3. *xml* – skirtas apdoroti UBL dokumentus ir paversti juos XML užklausomis, jį naudoja *transaction* paketas.
4. *settings* – komponento nustatymams tvarkyti, skirtas paketas, jį naudoja *transaction* ir *xml* paketai.

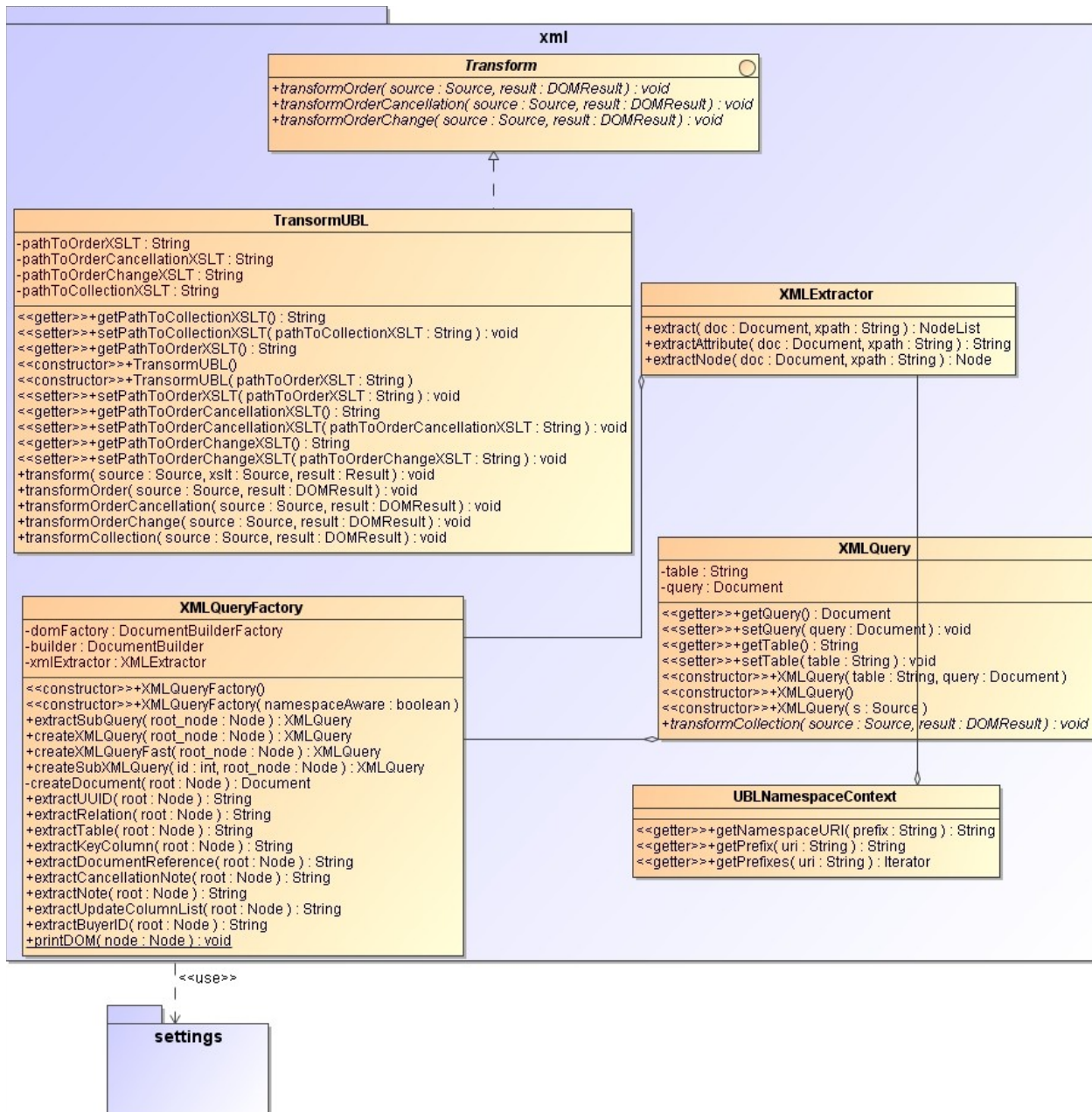
4.5. Klasių diagramos

4.5.1, 4.5.2, 4.5.3 ir 4.5.4 pav. pateikiamos paketų klasių diagramos, kuriuose pavaizduojami ir ryšiai su kitais paketais:

Paketas *settings* turi vieną klasę *Settings*, kuri veikia *Singleton* principu, t.y., jos sukurtas objektas yra vienintelis sistemos gyvavimo laikotarpiu. Naudojantis metodu *getSetting* gaunami komponento nustatymai.

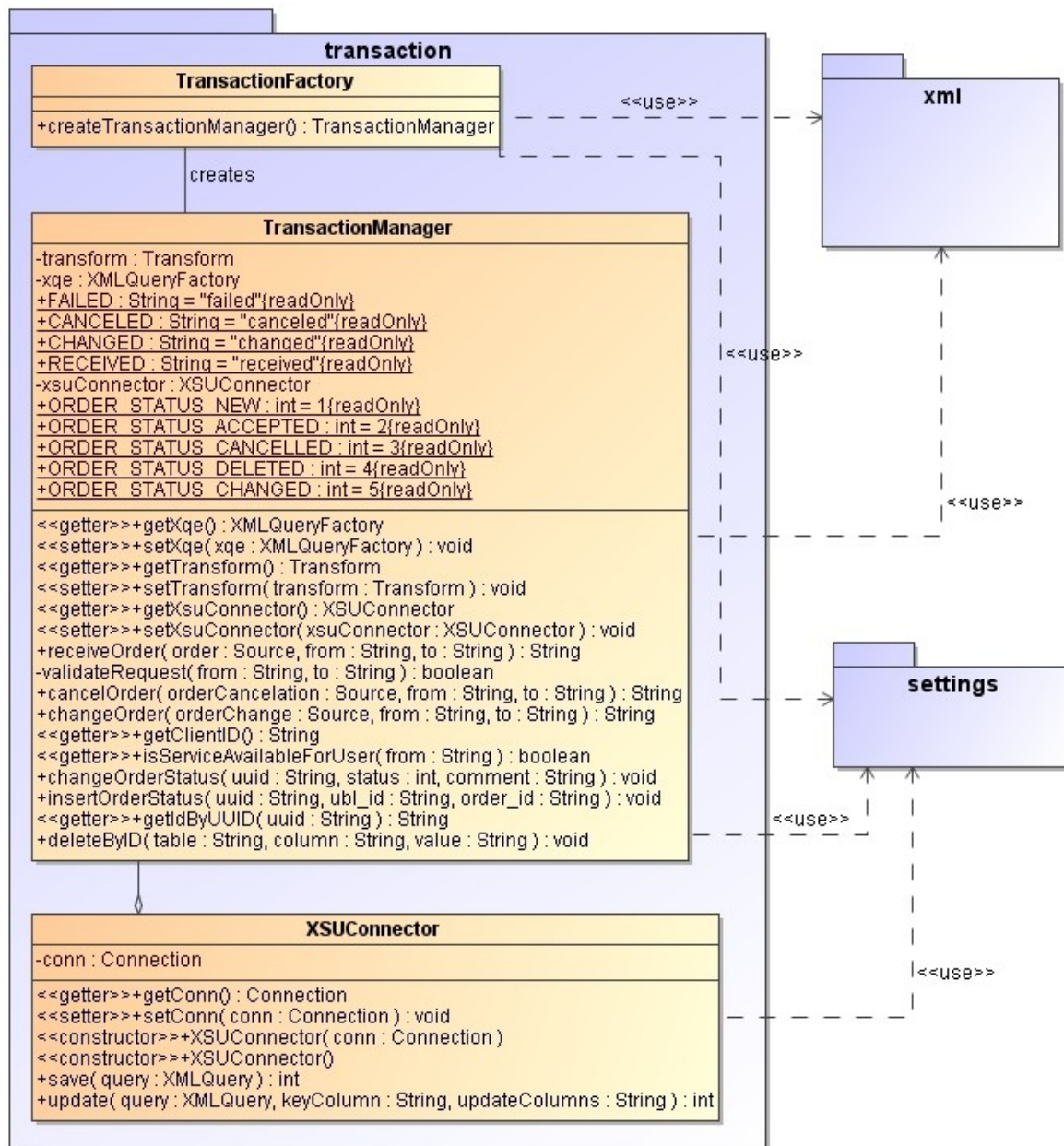


4.5.1 pav. Paketo settings klasių diagrama.



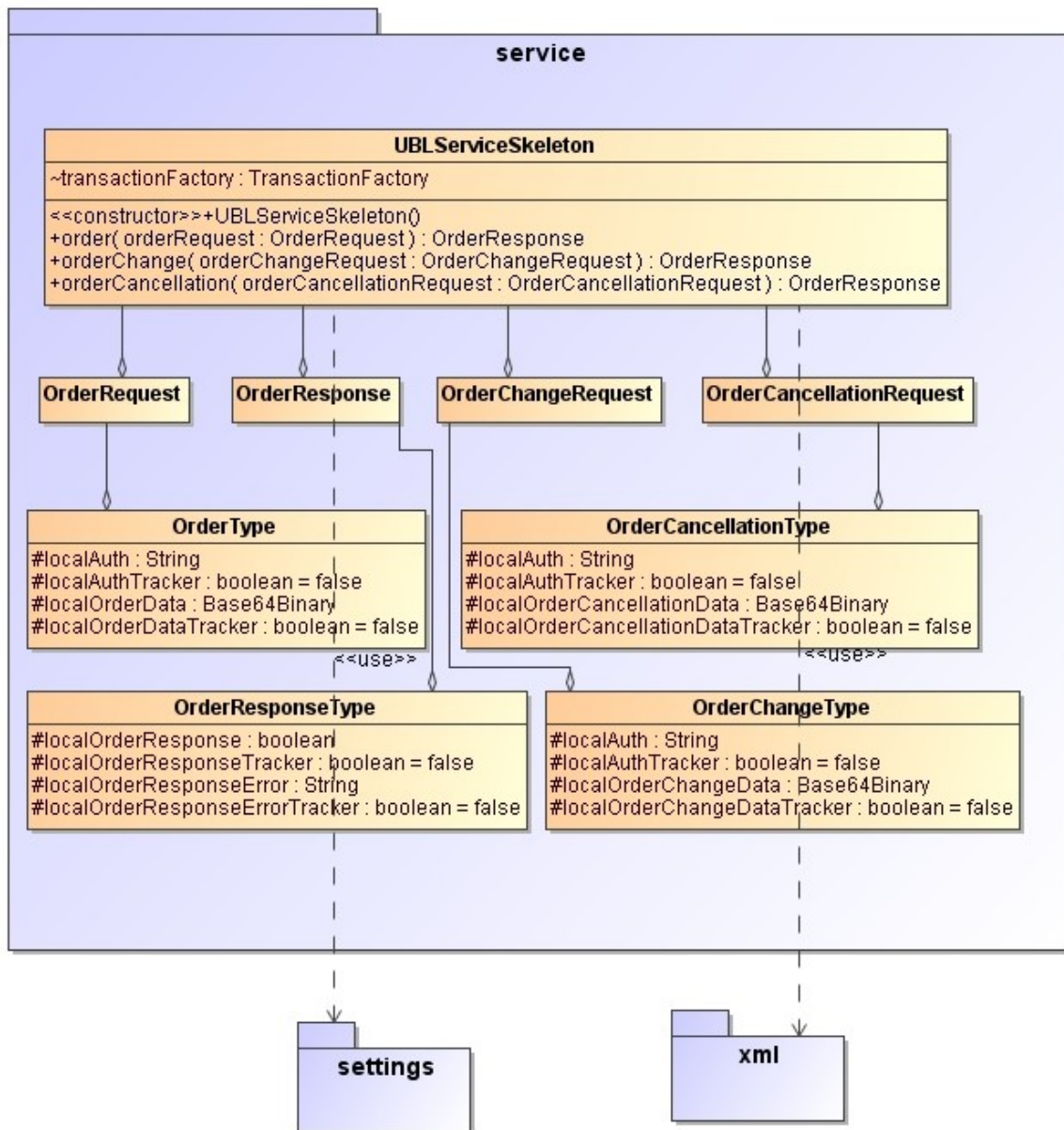
4.5.2 pav. Paketo xml klasių diagrama.

Šiame pakete esančios klasės tarnauja UBL dokumentų transformavimui, informacijos gavimui iš XML bylų, bei XML užklausų kūrinimui. Sąsaja *Transform* realizuoja *TransformUBL* klasė (transformacijų operacijų klasė). *XMLQueryFactory*, kuria *XMLQuery* objektus ir tarnauja kaip supaprastina sąsaja gauti tam tikrą informaciją iš XML dokumentų. *XMLExtractor*, naudojamas informacijos ir XML dokumentų gavimui pasitelkiant XPath. Ši klasė naudoja *UBLNamespaceContext*, tai klasė skirta atpažinti komponento naudojamas vardų erdves XML dokumentuose. *XMLQueryFactory* naudoja *Settings* klasę iš *settings* paketo.



4.5.3 pav. Paketo transaction klasių diagrama.

Šio paketo klasės tarnauja UBL užklausų atvaizdavimui i duomenų bazę. *TransactionFactory* klasė sukuria *TransactionManager* objektą. Pastarasis naudoja turi *XSUConnector* objektą tam, kad galėtų atlikti įterpimo ir modifikavimo veiksmus su duomenų baze. Visos paketo klasės naudoja nustatymų paketą. Veiksams su XML dokumentas *TransactionFactory* ir *TransactionManager* naudoja *xml* paketą.



4.5.4 pav. Paketo service klasių diagrama.

Paketas naudodamas pagalbinius *settings* ir *xml* paketus, veikia kaip užsakymo aptaravimo komponentas. *UBLServiceSkeleton* klasės metodai tiesiogiai susiję su

paslaugos iškvietimu. Šios klasės metodai atitinka komponento sąsajoje aprašytas operacijas. Operacijų parametrai yra *OrderRequest*, *OrderChangeRequest* ir *OrderCancellationRequest* objektai. Operacijų grąžinamos reikšmės yra *OrderResponse* objektai.

4.6. Dinaminė sistemos elgsena

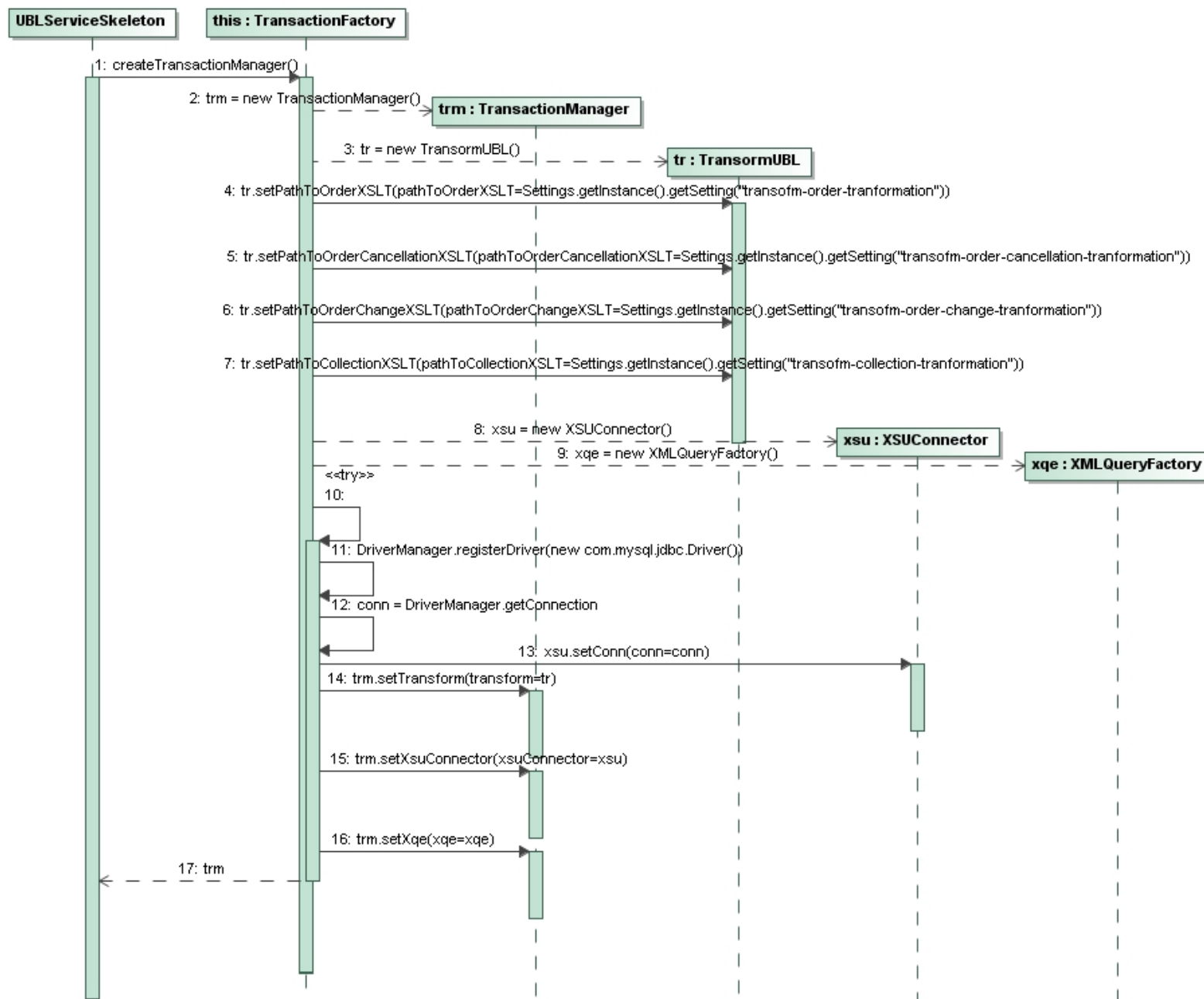
Užsakymo komponento dinamika pavaizduota sekų diagramomis, jos vaizduoja kaip sistemos objektai sąveikauja tarpusavyje.

Pirmiausiai pateikiama klasės *UBLServiceSkeleton* metodo *order()* sekos diagrama (4.6.1 pav.). Čia metodo iškvietimo iniciatorius yra *Paslaugos vartotojas*, kuris pateikia savo UBL-Order dokumentą aptarnavimo komponentui (1 veiksmas). *Order()* metode yra paaimamos SOAP *from* ir *get* antraštės (2-4 žingsniai), kuriuos naudojamos patikrinti ar teikiama paslauga tai siuntėjo įmonei, bei ar teisingas adresatas. 5 žingsnyje sukuriama transakcijų valdymo objektas. Jo sukūrimas pavaizduotas 4.6.2 pav. 7 žingsnyje yra iškviečiama užsakymo patalpinimo į duomenų bazę funkcija (4.6.3 pav.), kurios vykdymo metu įvykusios klaidos fiksuojamos 9 ir 10 žingsnyje. Toliau 11-16 vykdomas atsakymo į operaciją sukūrimas ir užpildymas rezultatais. Galiausiai, išsiunčiamas atsakas vartotojui (17 žingsnis). Tiek *orderCancellation()*, tiek ir *orderChange()* yra analogiški metodai, tik skiriasi 7 žingsnis, kuriame iškviečiama kita *TransactionManager* funkcija atitinkamai *cancelOrder()* ir *changeOrder()*.

TransactionManager – transakcijų iš UBL į SQL valdymo klasė sukuriama *TransactionFactory* pagalba (4.6.2 pav.), ši klasė turi vienintelį metodą *createTransactionManager()*. Jį iškviečia (1 žingsnis) *UBLServiceSkeleton* klasės objektas, vykdydamas bet kokia transakcija. Transakcijų valdymui reikalingos transformacijos, jas atlieka *TransformUBL* klasė, kuriai inicijuoti reikia nustatymų apie XSLT bylas (3-7 žingsniai). Taip pat reikalinga *XSUConnector* klasė, kuri atsakinga už XML dokumentų talpinimą į duomenų bazę. Užklausoms gaminti naudojama *XMLQueryFactory* klasė, kurios sukurti *XMLQuery* objektai naudojant *XSUConnector* patenka į duomenų bazę. 11-12 žingsniuose sukuriama susijungimas su duomenų baze. Nustatomi reikalingi objektai naujai sukurtam *TranformManager* egzemplioriui (14-16) ir jis yra grąžinamas (17).



4.6.1 pav. Klasės `UBLServiceSkeleton` metodo `order()` sekos diagrama.



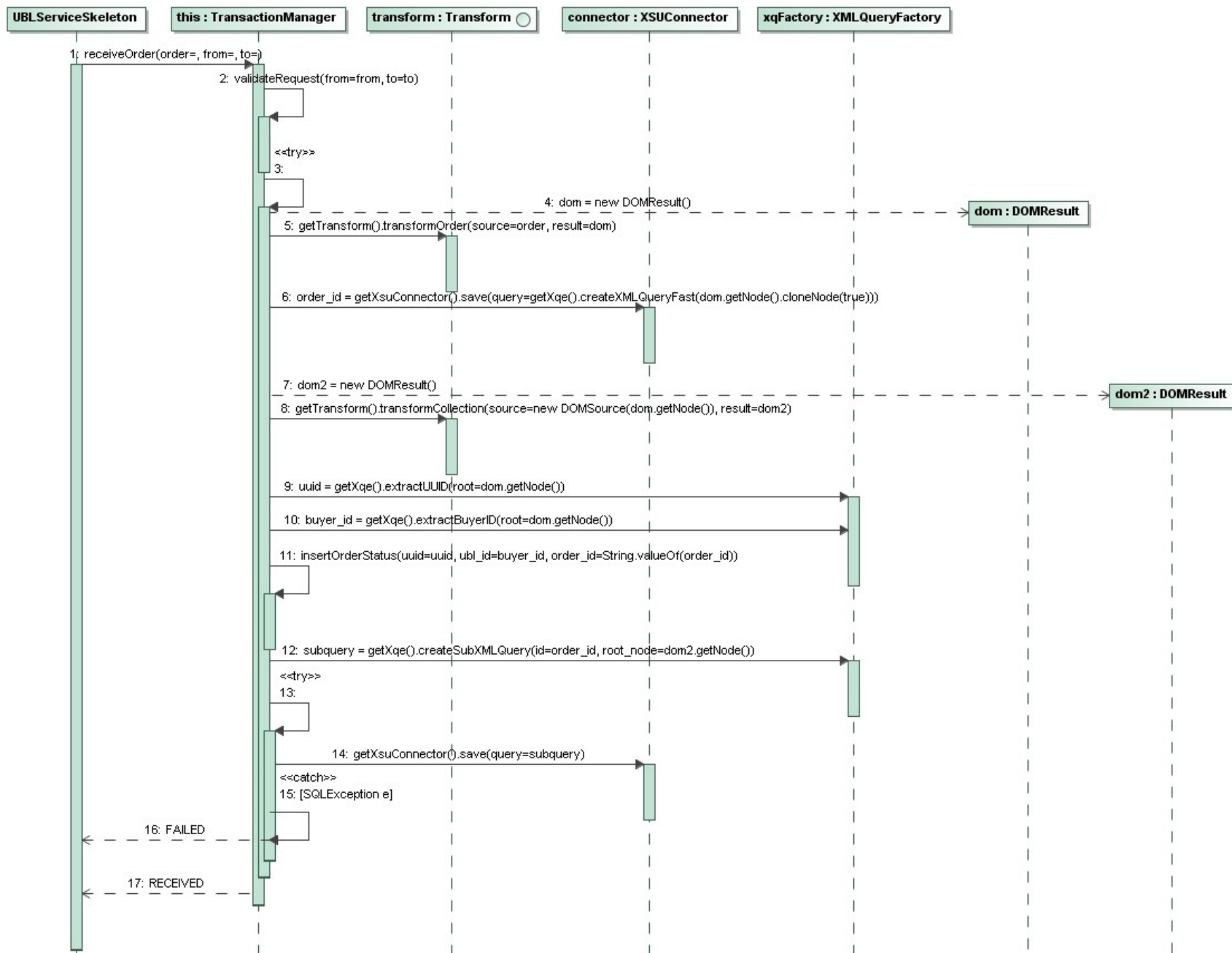
4.6.2 pav. Klasės `TransactionFactory` metodo `createTransactionManager()` sekos diagrama.

4.6.3 pav. pavaizduota kaip vyksta *TransactionManager receiveOrder()* metodas – esminė užsakymo priėmimo operacija. Ją inicijuoja *UBLServiceSkeleton* egzempliorius (1 žingsnis). Antrame žingsnyje (*validateRequest()*) yra patikrinama ar paslauga yra teikiama užsakovui, ir ar tinkamas UBL dokumento adresatas. Pirmiausiai transformuojamas UBL dokumentas į užsakymoXML (4-5 žingsniai). Tada iš jo yra sukuriama užklausa (metodas *createXMLQueryFast()*). Ji perduodama *XSUConnector* egzemplioriui iškviečiant *save()* metodą (6 žingsnis). Ši operacija neišsaugo užsakymo eilučių duomenų bazėje, ji išsaugo tik patį užsakymą. Toliau, su jau transformuotu UBL dokumentu, vykdoma dar viena transformacija (7-8 žingsniai), kuriuos metu ištraukiamas užsakymo prekių (eilučių) sąrašas XML pavidale. 9-10 žingsniuose išgaunama informacija apie UBL dokumento identifikatorius (*uuid* ir *buyer_id*). Pasinaudojus gauta informacija, metodu *insertOrderStatus()* duomenų bazėje sukurtoje papildomoje lentelėje išsaugoma būsenos informacija apie UBL užsakymą. 12-14 žingsniuose sukuriama prekių sąrašo gauta 8 žingsnyje užklausa ir išsaugoma duomenų bazėje. Jei įvyko klaidų metodo pabaigoje grąžinamas pranešimas FAILED, jei jų nebuvo – RECEIVED.

Užsakymo pakeitimą iliustruoja 4.6.4 pav. pateiktas *orderChange()* metodas. Čia pirmi 5 žingsniai analogiški *receiveOrder()*. 6-7 žingsnyje transformuojamas jau turimas XML, taip išgaunant prekių eilučių sąrašą, kuris gali būti pakeistas. 8-14 žingsniuose išgauna informacija sąsaja su originaliu dokumentu (vidinis patalpinto užsakymo identifikatorius duomenų bazėje), bei užsakymo pakeitimo komentaras (kodėl jis keičiamas). Tada užsakymas pakeičiamas. Pašalinamos eilutės prekės (16 žingsnis) ir sukuriama nauja užklausa skirta įvesti naujas užsakymo eilutes į duomenų bazę. Po pakeitimo 19 žingsnis pakeičia užsakymo statusą. Apie ratas klaidas grąžinamas pranešimas (21 žingsnis), jei klaidų nebuvo grąžinamas sėkmingai įvykdyto užsakymo pranešimas (22 žingsnis).

Užsakymo atšaukimas vykdomas *cancelOrder()* pagalba. Jis, kaip ir prie tai dvi operacijos, patikrina ar galima paslauga. Tada pakeičia užsakymo statusą, papildomoje lentelėje iškviesdamas *changeOrderStatus()* metodą.

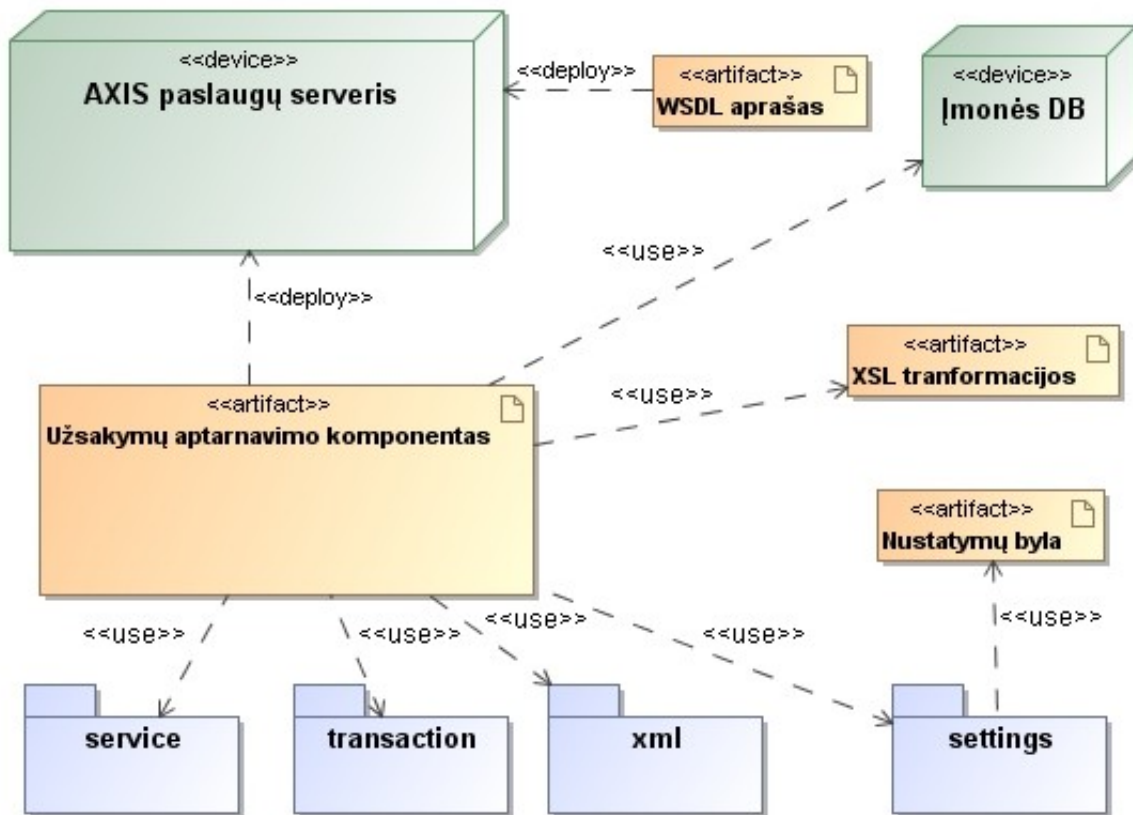
Taigi, aprašytos sudėtingesnės užsakymo aptarnavimo komponento objektų sąveikos. Kiti objektai yra paprastesni, todėl jų neaprašinėsimė.



4.6.3 pav. Klasės `TransactionManager` metodo `receiveOrder()` sekos diagrama.

4.7. Realizacijos modelis

Užsakymų aptarnavimo komponento realizacijos modelis pateikiamas žemiau (4.7.1 pav.):



4.7.1 pav. Realizacijos modelis.

Sukurtas *Užsakymo aptarnavimo komponentas* yra diegiamas *AXIS paslaugų serveryje*, kuris naudoja *WSDL paslaugos aprašą*, tam, kad susietų komponento operacijas su ateinančiomis į serverį užklausomis. Komponentas naudodamasis keturiais paketais realizuoja funkcionalumą. Paketas *settings* naudoja *nustatymų bylą*, tam, kad perduotų nustatymus sistemai. Komponentas tiesiogiai bendrauja su *Įmonės DB*, pateikdamas SQL užklausas. Pakartotiniam panaudojimui, reikia keisti XSL transformacijas ir nustatymų bylą, jei naudojama kita įmonės duomenų bazė.

4.8. Komponento nustatymų aprašymas

Nustatymų byla yra *settings.properties*, kuri yra *resources* kataloge. Žemiau pateikiamas kiekvieno nustatymo paaiškinimas. Šios bylos pavyzdys yra pateiktas 5.2 skyriuje.

- `service-provider` – Paslaugos teikėjo pavadinimas, jis naudojamas SOAP pranešimo TO antraštės tikrinime.
- `service-users` – Paslaugos naudotojai, įmonių pavadinimai ar kiti identifikatoriai išvardinti, atskiriant kableliu. Naudotojas turi sutapti su SOAP antraštės FROM turiniu.
- `date-format` – Datos formatas naudojamas transformuotose UBL dokumentuose.
- `transofm-order-tranformation` – Kelias iki UBL-Order XSL transformacijos.
- `transofm-order-cancellation-tranformation` – Kelias iki UBL-OrderCancellation XSL transformacijos.
- `transofm-order-change-tranformation` – Kelias iki UBL-OrderChange XSL transformacijos.
- `transofm-collection-tranformation` – Kelias tarpinės transformacijos naudojamos išgauti užsakymo prekių sarašą.
- `db-url` – URL nuoroda į įmonės duomenų bazę.
- `db-user` – Įmonės duomenų bazės vartotojo vardas.
- `db-pass` – Įmonės duomenų bazės vartotojo slaptažodis.
- `db-ubl-order-table-name` – Papildomos lentelės duomenų bazėje pavadinimas.
- `db-ubl-order-status-column` – Papildomos lentelės duomenų bazėje būsenos atributo vardas.
- `db-ubl-order-date-column` – Papildomos lentelės duomenų bazėje datos atributo vardas.
- `db-ubl-order-id-column` – Papildomos lentelės duomenų bazėje užsakymo identifikatoriaus atributo vardas. Šis atributas sieja užsakymo identifikatorių su papildomos lentelės įrašu.
- `db-ubl-order-ubl-id-column` – Papildomos lentelės duomenų bazėje UBL dokumento ID elementui saugoti skirtas atributo vardas.
- `db-ubl-order-uuid-column` – Papildomos lentelės duomenų bazėje UBL dokumento UUID elementui saugoti skirtas atributo vardas.
- `db-ubl-order-comment-column` – Papildomos lentelės duomenų bazėje komentarui (apie atšaukimą/pakeitimą) saugoti skirtas atributo vardas.

4.9. Diegimo instrukcijos

Jei sistema diegiama kitoje įmonėje (duomenų bazės struktūra skiriasi nuo aprašytos 4.1 skyriuje), prieš tai reikia pagal 4.2 skyriuje pateiktą aprašymą parengti transformacijas kitai įmonės duomenų bazei. Prieš diegimą ir paleidimą, reikia įsitikinti, kad paleistas duomenų bazių serveris.

Diegimo instrukcijos žingsnis po žingsnio:

1. Parsisiųsti ir instaliuoti naujausią Java virtualiąją mašiną (jei jos dar nėra).
2. Nurodyti aplinkos kintamąjį `JAVA_HOME` į vietą, kur instaliuota Java (jei tokio kintamojo nėra).
3. Įmonės duomenų bazėje sukurti papildomą lentelę pagal 3.4.3 skyriuje aprašytus reikalavimus.
4. Išpakuoti komponento archyvą.
5. Nurodyti aplinkos kintamąjį `AXIS_HOME` į vietą, kur išpakuotas komponentas.
6. Nustatymų byloje nurodyti prisijungimo prie duomenų bazės nuorodą, vartotojo vardą ir slaptažodį.
7. Nustatymų byloje nurodyti sukurtos lentelės ir jos atributų pavadinimus. Nustatymų bylą galima rasti *resources/settings.properties*.
8. Nustatymų byloje nurodyti, koks datos formatas naudojamas transformuotose dokumentuose.
9. Nustatymų byloje nurodyti, koks yra įmonės pavadinimas, kuris bus naudojamas atskirti ar ateinantys UBL dokumentai patenka tikrai į tą įmonę.
10. Nustatymų byloje nurodyti įmonių pavadinimus, kurios gali naudotis komponento teikiamomis paslaugomis. Įmones užrašyti atskiriant kableliu.
11. Sistema paleidžiama naudojant išpakuotame kataloge aktyvuojant *bin/axis2server.bat* arba *bin/axis2server.sh* paleidžiamųjų sekų bylas, atitinkamai *Windows* ir *Unix* aplinkose.

5. Užsakymo aptarnavimo komponento tyrimas

5.1. Testavimo modelis

Užsakymo aptarnavimo komponentui tirti, buvo sukurtas specialus klientas, imituojantis kitos įmonės sistemą. Jo pagalba, galima siųsti užklausas paslaugų serveriui. Šis klientas parašytas Java kalba ir paleidžiamas per komandinę eilutę. Paleidimo komanda yra:

```
>java UBLSeviceClient [užklausa] [kelias iki UBL dokumento]
[užsakovas] [paslaugos teikėjas]
```

Galimos užklausos reikšmės:

- uzsakyti – užsakymo pateikimas;
- atsaukti – užsakymo atšaukimas;
- pakeisti – užsakymo pakeitimas.

Kelias iki UBL dokumento gali būti tiek santykinis (kliento buvimo vietos failinėje sistemoje atžvilgiu), tiek ir absoliutus. Užsakovas – įmonės pavadinimas, kuri vykdo užsakymą (nurodoma tarp kabučių). Paslaugos teikėjas – įmonės teikiančios paslaugą pavadinimas (nurodoma tarp kabučių).

Naudojimo pavyzdys:

```
java UBLServiceClient uzsakyti UBL-Order-.xml „UAB Test
Klientas“ „UAB Test Tiekėjas“
```

Įvykdę šią užklausa klientas atvaizuos tokius rezultatus:

```
=== Pradedamas paslaugos iskvietimas: uzsakyti
=== Dokumentas: test/TEST/UBL-Order-2.0-Example.xml
=== Atliekamas uzsakymas....
=== Atlikta.
=== Uzsakymas priimtas
=== Baigta.
```

Testavimo metu, naudojantis aukščiau aprašyti klientu, siunčiamos užklausos užsakymo aptarnavimo komponentui. Jei gautas patvirtinimas apie sėkmingą operaciją, tada duomenų bazėje tikrinami užsakymo duomenys. Taip pat patikrinama ar papildomoje būsenų lentelėje, atsirado naujas įrašas arba pakeitimas, priklausomai nuo operacijos tipo. Paslaugos operacija laikoma pavykusia, kai visi būtini užsakymo lentelės atributai yra užpildyti ir padarytas atitinkamas įrašas apie operaciją būsenų lentelėje.

Užsakant, pakeičiant ar atšaukiant, gali įvykti klaida. Tuomet gaunamas klaidos pranešimas. Klaida gali reikšti, kad užsakymo komponentas yra blogai sukonfigūruotas, blogos jo transformacijos arba blogi duomenys esantys UBL dokumentuose. Klaidų pranešimai ir jų paaiškinimai pateikiami 5.1.1 lentelėje. Joje taip pat aprašyti ir galimi klaidų pašalinimo būdai.

5.1.1 lentelė. Klaidų pranešimai ir jų paaiškinimai.

Klaidos pranešimas	Priežastis	Pašalinimas
Nerastas <i>transformacijos</i> failas.	Blogai nurodyta transformacijos failo vieta, arba jis neegzistuoja.	Patikrinti ar transformacijų failai egzistuoja, ar teisingai nurodyti jų vardai, ir ar teisingai nurodytos vietos, kur jie yra.
Bloga transformacija.	Transformacijoje yra sintaksės klaidų.	Rekomenduojama naudoti XSL transformacijų redaktorių tam, kad patikrintumėte, kurioje vietoje yra klaida.
Nepavyko prisijungti prie DB.	Blogi prisijungimo duomenys arba nepaleista/neinstaliuota duomenų bazių valdymo sistema.	Komponentas diegiamas įmonėje, kurioje yra duomenų bazių valdymo sistema, todėl greičiausiai nurodytas neteisingas prisijungimo vardas ir/arba slaptažodis. Taip pat reiktų patikrinti ar duomenų bazių vartotojas turi teisę prisijungti ir vykdyti SQL užklausas.
Nėra tokios lentelės.	Transformacijos faile nurodytos lentelė duomenų bazėje neegzistuoja.	Patikrinti ar transformacijoje įrašytas lentelės pavadinimas sutampa su duomenų bazėje esančiu. Taip pat gali būti nesukurta papildoma lentelė užsakymo būsenai saugoti.
Raktas unikalumo pažeidimas.	UBL dokumentuose esanti informacija apie identifikatorius yra bloga, arba jos nėra.	Patikrinti ar UBL dokumentas atitinka 3.4.6 skyriuje nurodytus reikalavimus. Tikėtinas UBL UUID atributo unikalumo pažeidimas.
Trūksta duomenų.	UBL dokumente, neužpildyti būtini laukai.	Patikrinti ar tikrai UBL dokumentuose užpildyti būtini laukai (3.4.6 skyrius).
Nėra tokio atributo.	Klaida transformacijoje.	Patikrinti ar transformacijoje įvesti atributų vardai atitinka duomenų bazėje esančius atributus.

5.1.1 lentelės tęsinys.

Klaidos pranešimas	Priežastis	Pašalinimas
Trūksta atributo.	Klaida transformacijoje.	Patikrinti ar teisinga atitinkama transformacija. Galima klaida surašant atributų vardus.
Užsakymas neegzistuoja	Blogai nurodytas užsakymo UUID	Patikrinti ar nuoroda į užsakymą yra gera ir ar jis egzistuoja.
Neteisintas adresatas.	Siunčiama užklausa ne tam paslaugos teikėjui	Patikrinti ar nustatymų byloje teisingai įvestas paslaugos teikėjas, jei taip, tada patikrinti ar teisingai jis nurodomas siunčiant užklausa.
Blogas datos formatas.	Neteisingai užrašytas datos formatas nustatymų byloje.	Datos formatą reiktų aprašyti naudojant SimpleDateFormat [30] formatą.

5.2. Pradiniai duomenys

Komponentas testuotas naudojant tokius nustatymus:

```
#Paslaugos teikejo pavadinimas, kuris nurodomas, SOAP pranesimo TO
antrasteje.
service-provider=UAB Test Tiekėjas

#Paslaugos naudotojai, atskirti kableliu.
service-users=UAB Test Uzsakovas1, UAB Test Uzsakovas2

#Datos formatas ateinantis is UBL Order
date-format=yyyyMMDD

#Tranformacijos
transofm-order-tranformation=resources/xslt/Order.xslt
transofm-order-cancellation-
transformation=resources/xslt/OrderCancellation.xslt
transofm-order-change-tranformation=resources/xslt/OrderChange.xslt
#Vidine tranformacija
transofm-collection-tranformation=resources/xslt/COLLECTION.xslt

#Db nustatymai
db-url=jdbc:mysql://localhost/ubl
db-user=root
db-pass=labas123

#Db papildomos lenteles nustatymai
db-ubl-order-table-name=ubl_uzsakymai
db-ubl-order-status-column=busena
db-ubl-order-date-column=data
db-ubl-order-id-column=uzsakymo_id
db-ubl-order-ubl-id-column=ubl_id
db-ubl-order-uuid-column=uuid
db-ubl-order-comment-column=komentaras
```

Testuojant išbandytos visos trys operacijos. UBL dokumentai naudojami užklausoje pateikiami 9.4 priede. Transformacijos, kurios naudojamos testuojant pateiktos 4.2.4 skyriuje.

5.3. Rezultatai ir išvados

Naudojantis sukurtu klientu, buvo išsiųstos užklausoje, kurias priėmė užsakymo aptarnavimo komponentas ir jas patalpino įmonės duomenų bazėje. Iš UBL dokumentų sugeneruotos ir įvykdytos SQL užklausoje pateikiamos 9.5 priede.

Atlikus testavimą galima daryti tokias išvadas:

- Sukurtas užsakymų aptarnavimo komponento klientas, gali būti tiesiogiai integruojamas į įmonės sistemą.
- Sistemai reikia patobulinti klaidų apdorojimą, t.y., reikia padaryti sistemos konfigūravimo patikrinimą, dar prieš aptarnaujant užsakymus. Taip pat reikia atskirti sistemos konfigūravimo ir su vartotojo siunčiamų duomenų teisingumu susijusias klaidas. Klaidų pranešimus padaryti labiau orientuotus į vartotoją.
- Sukurtas komponentas vykdo savo tikslą, nes pagal parengtus XSL šablonus ir pateiktus UBL dokumentus išsaugo juos įmonės duomenų bazėje.
- Komponentas apima tik užsakymo priėmimą, atšaukimą ir pakeitimą įmonės duomenų bazėje, todėl galima ją patobulinti – įtraukti kitų su užsakymu susijusių procesų (pvz., užsakymo priėmimo patvirtinimas) palaikymą.

6. Išvados

Atlikus darbą galima daryti tokias išvadas:

1. Išanalizavus įmonių sąveikumo problemas galima teigti, kad pagrindinės problemos yra susijusios ne su technologijų trūkumu, o su sprendimų priėmimu, sąveikumo naudos įrodymo trūkumu, dideliais integracijos/sąveikumo kaštais.
2. Ištyrus esamus įmonių sąveikumo sprendimus, galima teigti, kad paslaugoms orientuota architektūra ir XML grindžiamos technologijos yra kartinės įmonių sąveikumui įgalinti. Šių technologijų taikymas versle didina konkurencinį pranašumą.
3. Pasiūlytas, suprojektuotas ir įgyvendintas užsakymų aptarnavimo komponentas, kuris įmonės duomenų bazėje automatiškai patalpina užsakymus, ateinančius UBL dokumentų pavidalu, galintis tris operacijas: priimti, pakeisti ir atšaukti užsakymą.
4. Atlikus sukurto užsakymo komponento tyrimą, buvo sukurtas klientas, kuris gali būti pritaikomas įmonėse, norinčiose pasinaudoti komponento teikiama užsakymo aptarnavimo paslauga.
5. Sukurtą sistemą galima modifikuoti į visą užsakymo procesą apimančią užsakymo aptarnavimo sistemą.

7. Literatūra

1. Information society technologies 2006 June 6. Working document Version 3.0 “Research Roadmap”, Enterprise Interoperability [žiūrēta 2008-12-06]. Prieiga per internetā:
<ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/ebusiness/20060606_roadmap_v30_en.pdf>.
2. Sparx Systems. The Business Process Model. 2004. [žiūrēta 2009-01-05]. Prieiga per internetā:
<http://www.sparxsystems.com.au/downloads/whitepapers/The_Business_Process_Model.pdf>
3. Business Process Types. Iš BPTrends [interaktyvus]. [žiūrēta 2008-12-06]. Prieiga per internetā: <<http://www.bptrends.com/>>
4. Department of Finance and Administration. The Australian Government Business Process Interoperability Framework. 2007. [žiūrēta 2009-01-06]. Prieiga per internetā:
<http://www.finance.gov.au/publications/agimo/docs/Business_Process_Interoperability_Framework.pdf>
5. Benoît Müller. eGovernment, Interoperability and Innovation. 2005. [žiūrēta 2008-12-06]. Prieiga per internetā: <<http://interop-esa05.unige.ch/INTEROP/Proceedings/eGovScientific/papers/6b1.pdf>>
6. Chesbrough W. Henry. Open Innovation The New Imperative for Creating and Profiting from Technology. Iš Harvard Business [interaktyvus]. 2003, kovas.
7. Upadrasta Venkatesh. An Approach to Enterprise Interoperability Using SOA. Iš BPTrends [interaktyvus]. 2006, liepa [žiūrēta 2008-12-06]. Prieiga per internetā:
<http://www.bptrends.com/deliver_file.cfm?fileType=publication&fileName=07-06-ART-EntInteropSOA-Venkatesh.pdf>
8. State-of-the art for Interoperability architecture approaches / Arne-Jørgen Berre, Axel Hahn, David Akehurst, Jean Bezivin, Aphrodite Tsalgatiidou, François Vermaut, Lea Kutvonen, Peter F. Linington. Information Society, 2004 November 19. 362 p. [žiūrēta 2008-12-06]. Prieiga per internetā: <.

vlab.eu/deliv/dap-domain-architecture-and-platforms/D91/attachment_download/file
>

9. Yvonne Balzer. Improve your SOA project plans. 2004, liepa. [žiūrēta 2008-12-06]. Prieiga per internetu: <<http://www-128.ibm.com/developerworks/webservices/library/ws-improvesoa/>>
10. Josienita Borlongan. About XML. 2008. [žiūrēta 2008-12-06]. Prieiga per internetu: <http://www.ehow.com/about_4565629_xml.html>
11. Loesgen Brian. XML Interoperability. Iš Wrox conferences. 2000. [žiūrēta 2008-12-06]. 10 p. Prieiga per internetu <http://www.topxml.com/conference/wrox/2000_vegas/text/brianl_xml.pdf>
12. Wrightson, Ann. Enterprise Interoperability Architecture using XML Industry Standards. Iš RenderX [interaktyvus]. [žiūrēta 2008-12-06]. Prieiga per internetu <http://www.idealliance.org/papers/dx_xmle04/papers/03-06-04/03-06-04.pdf>
13. SOAP Version 1.2 Part 1: Messaging Framework. 2007, balandis. [žiūrēta 2008-12-06]. Prieiga per internetu <<http://www.w3.org/TR/2007/REC-soap12-part1-20070427/#intro>>
14. Herring, C. Milosevic, Z. Implementing B2B Contracts using BizTalk. Iš 34th Annual Hawaii International Conference on System Sciences. 2001, 10 p.
15. Mehta, B. Levy, M. Meredith, G. Andrews, T. Beckman, B. Klein, J. Mital, A. BizTalk Server 2000 Business Process Orchestration. The 17th International Conference on Data Engineering April 2–6, 2001 Heidelberg, Germany. 35 – 39 p.
16. Vickers, Sue. Portal Standards The answer to portal interoperability? Iš JAVA.SYS – CON [interaktyvus]. 2005 [žiūrēta 2008-12-06]. Prieiga per internetu: <<http://java.sys-con.com/read/47686.htm>>
17. JSR-000168 Portlet Specification. 2003. [žiūrēta 2008-12-06]. Prieiga per internetu: <<http://jcp.org/aboutJava/communityprocess/review/jsr168/>>
18. OASIS Web Services for Remote Portlets (WSRP) TC. [žiūrēta 2008-12-06]. Prieiga per internetu: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
19. Boye, Janus. Enterprise Portal Market Overview 2008. Iš KMWorld [interaktyvus]. 2008, liepa. [žiūrēta 2008-12-06]. Prieiga per internetu: <<http://www.kmworld.com/Articles/Editorial/Feature/Enterprise-Portal-MARKET-Overview-2008--48954.aspx>>

20. Všį „, Ekonominiai ir statistiniai tyrimai“. Elektroninio verslo plętra Lietuvoje ir įtaka šalies įmonių konkurencingumui. 2005, Vilnius. 298 p.
21. EDI vs. XML. Iš EDIDevelopment [interaktyvus]. [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.edidev.com/XMLvsEDI.html>>
22. UN/CEFACT, OASIS. Iš ebXML [interaktyvus].]. [žiūrėta 2008-12-06]. Prieiga per internetą: < <http://www.ebxml.org/>>
23. Dournaee, Blake. Introduction to ebXML. Iš ORACLE technology network [interaktyvus]. 2004, birželis [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.oracle.com/technology/pub/articles/dev2arch/2004/12/ebXML.html>>
24. Mertz, David. Understanding ebXML. Is IBM [interaktyvus]. 2001, birželis [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.ibm.com/developerworks/xml/library/x-ebxml/>>
25. Masud, Suhayl. RosettaNet's place in the overall Web services picture. Iš SearchSOA [interaktyvus].2003, spalį [žiūrėta 2008-12-06]. Prieiga per internetą: <http://searchsoa.techtarget.com/expert/KnowledgebaseAnswer/0,289625,sid26_cid555414,00.html#>
26. Universal Business Language v2.0 Standard, 12 December 2006. [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html>>
27. On UBL. Iš Unimaze [interaktyvus]. [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.unimaze.com/on+ubl.aspx>>
28. About xCBL. Iš xCBL [interaktyvus]. [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.xcbl.org/about.shtml>>
29. cXML FAQ. Iš cXML [interaktyvus]. [žiūrėta 2008-12-06]. Prieiga per internetą: <<http://www.cxml.org/prnews/faq.cfm>>
30. SimpleDateFormat. Iš Sun [interaktyvus]. [žiūrėta 2009-01-06]. Prieiga per internetą: <<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html>>

8. Terminų ir santrumpų žodynas

- Ad Hoc – tai decentralizuotas tinklas.
- BPMN - Business Process Modeling Notation – verslo procesų modeliavimo notacija.
- CASE – Computer-Aided Software Engineering – kompiuterizuota IS inžinerija.
- cXML – Commerce XML – komercijos XML.
- DB – duomenų bazė.
- DCOM – Distributed Component Object Model – paskirstytų komponentų objektinis modelis.
- ebXML - Electronic Business using XML - elektroninio verslo XML.
- EDI – Electronic Data Interchange – elektroninių dokumentų mainai.
- EDIFACT - Electronic Data Interchange For Administration, Commerce, and Transport – elektroninai duomenų mainai valdymui, komercijai ir transportui.
- Egzempliorius – programinės įrangos kontekste, tai klasės objektas.
- FTP – File Transfer Protocol – failų persiuntimo protokolas.
- HTTP - Hypertext Transfer Protocol – hiperteksto persiuntimo protokolas.
- ID – identifikatorius.
- IIOP - Internet Inter-Orb Protocol – Protokolas leidžiantis kurti paskirstytas sistemas naudojant skirtingas programavimo kalbas.
- Įmonių sąveikumas – Enterprise Interoperability – veiklos sritis, kurios tikslas gerinti įmonių sąveikavimą su kitomis įmonėmis, organizacijomis ar kitais tos pačios organizacijos padaliniais, tam, kad būtų galima plėsti verslą.
- IS – informacinė sistema.
- IT – informacinės technologijos.
- JSR168 – portletų specifikacija.
- PĮ – programinė įranga.
- Verslo procesas – tai rinkinys tarpusavyje susijusių užduočių, skirtų tam tikrai problemai spręsti.

- RESTful – sistemos sukurtos REST principu, tai principas kuomet kuriamos paprasčiausiomis žiniatinklio technologijomis.
- SMTP – Simple Mail Transfer Protocol – paprastas el. pašto persiuntimo protokolas.
- SOA – Service Oriented Architecture – paslaugoms orientuota architektūra.
- SOAP – Simple Object Access Protocol – paprastas objektų prieigos protokolas.
- SQL – duomenų bazių užklausų kalba.
- UML – Unified Modeling Language – unifikuota modeliavimo kalba.
- UUID – universalus unikalus identifikatorius (UBL dokumentų kontekste).
- Užsakymas – tai raštu arba žodžiu užfiksuotas įsipareigojimas tam tikrų paslaugų ar produktų komercinis sandoris.
- Užsakymų aptarnavimo komponentas – tai sistema, įgalinanti įmonės klientams automatizuotai užsisakyti prekes.
- VPS – verslo procesų sąveikumas.
- WSDL – Web Services Definition Language – žiniatinklio paslaugų aprašymo kalba.
- WSRP – portletų specifikacija sukurta OASIS.
- xCBL – XML Common Business Library – bendroji verslo XML biblioteka.
- XML – Extensible Markup Language – išplečiamą žymių kalba.
- XSD – XML Schema Definition – kalba skirta aprašyti XML dokumentų turinį.
- XSL – kalba skirta XML transformacijoms aprašyti.

9. Priedai

9.1. IT pritaikymas įmonės vertės grandinėje.

9.1.1 Lentelė. IT pritaikymas įmonės vertės grandinėje.

VEIKLA	ELEKTRONINIO VERSLO PRITAIKYMAS
Žaliavų logistika	Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas ištekliams planuoti ir valdyti, apsikeisti operatyvia informacija tarp įmonės ir tiekėjų, atsargų ištekliams valdyti, išteklių judėjimui stebėti.
Gamybinės-technologinės operacijos	Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti diegiamas, siekiant efektyviai keisti informacija tarp atskirų padalinių, įvairių procesų automatizavimui bei kontrolės užtikrinimui.
Produktų logistika	Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas vartotojo inicijuotoms, pardavėjo ar kanalo partnerio užsakymų operacijoms, taip pat informacijai pateikti, komunikavimui ir t.t..
Marketingas ir pardavimas	<p>Informacinėmis technologijomis pagrįstas elektroninis verslas taikomas naujiems pardavimo kanalams sukurti, elektroninėms parduotuvėms steigti. Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas informacijai apie vartotojus bei konkurentus rinkti.</p> <p>Galimybė išnaudoti naują reklamą bei galimybės nukreipti pastangas į tikslinius rinkos segmentus.</p> <p>Didinti komunikavimo efektyvumą. Pasinaudojant informacinėmis technologijomis pagrįsto elektroninio verslo galimybėmis, galima geriau išanalizuoti ir patenkinti klientų poreikius. Klientui gali būti suteikta didesnė pagalba prekes išsirenkant, perkant ir naudojant prekę, pateikta daugiau informacijos, užtikrintas greitesnis atsakymas į užklausimus, ir geresnis, išsamesnis prekės panaudojimo aprašymo pateikimas. Labai svarbu yra tai, kad klientas informaciją gali gauti kada tik panorėjęs ir neišėidamas iš savo darbo vietos.</p> <p>Informacinėmis technologijomis pagrįstas elektroninis verslas leidžia įvertinti kiekvieno vartotojo reikmes – o tai leidžia jam pateikti tikslia, išsamią informaciją apie tik jį dominančias prekes ir paslaugas. Informacinėmis technologijomis pagrįstas elektroninis verslas sąlygoja tobulą kainų diskriminaciją, pateikiant kiekvienam vartotojui skirtingą kainą ir tokiu būdu maksimizuoti pelną.</p>

Aptarnavimas	<p>Informacinėmis technologijomis pagrįstas elektroninis verslas panaudojamas informacijai klientams pateikti, juos konsultuoti, teikti papildomas paslaugas. Pirmasis eksperimentas šioje srityje buvo vartotojams suteikta galimybė sekti savo užsakymų būklę. Paprastas pavyzdys: transporto kompanijos <i>FedEx</i> klientai pradėjo naudotis internetu, kad galėtų stebėti savo pervežamo siuntinio vietą. <i>FedEx</i> nustatė, kad sumažėjo jų išlaidos, nes iš 800 prieš tai dirbusių atsakinėjančių į klientų skambučius operatorių buvo reikalingi tik keliasdešimt. Tačiau tikrasis ir ilgalaikis <i>FedEx</i> privalumas kompanijai buvo ne informacinio centro personalo kaštų mažinimas ar klientų aptarnavimo gerinimas (klientams nebereikėjo laukti eilėse, kol atsilieps operatorė), o kompanijos augimo galimybės ir lankstumas, kurias suteikė supaprastintas procesas.</p>
Įmonės infrastruktūra	<p>Informacinėmis technologijomis pagrįsto elektroninio verslo pagrindu grindžiamos įmonės materialinių ir finansinių išteklių valdymo sistemos, ryšių su išoriniais partneriais palaikymo sistemos.</p>
Žmogiškųjų išteklių valdymas	<p>Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas administravimui, apmokymams, informacijos apie įmonę skleidimui, ataskaitų pateikimui, duomenų bazės apie potencialius darbuotojus sudarymui. Informacinėmis technologijomis pagrįstas elektroninis verslas padeda įmonei iš įvairių geografinių vietų rinkti informaciją apie norinčius įsidarbinti, ir taip įgyti galimybę tiesiogiai kontaktuoti su ieškančiais darbo asmenimis. Be to, tuo pačiu įmonė gali disponuoti informacija ne apie vieną asmenį, o apie visus pareiškusius norą įsidarbinti.</p>
Technologijų vystymas	<p>Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas naujoms prekėms kurti, bendradarbiaujant su įmonės filialais ir pridėtinės vertės grandinės dalyviais. Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas informacijai apie naujausius pasiekimus rinkti ir paskleisti įmonėje, taip pat ir operatyviai informacijai rinkti.</p>
Pirkimas	<p>Informacinėmis technologijomis pagrįstas elektroninis verslas gali būti panaudojamas susiejant žaliavų įsigijimo, atsargų ir prognozavimo sistemas su tiekėjų sistemomis, tiesiogiai ir netiesiogiai aprūpinant per prekyvietes, biržas, aukcionus.</p> <p><i>Apsirūpinimas materialiniais ištekliais.</i> Pasinaudodama elektroninio verslo sprendimais, įmonė gali surinkti kur kas daugiau informacijos apie galimus tiekėjus nei paprastu būdu. Vienas iš privalumų – galimybė tiesiogiai susisiekti su tiekėju prašant papildomos informacijos apie siūlomas prekes.</p> <p><i>Apsirūpinimas technologiniais ištekliais.</i> Labai sparčiai tobulėjant technologijoms, svarbu yra stebėti jų kaitą, rinkti informaciją apie naujausias iš jų. Kadangi technologijų plėtra yra globalus reiškinys, todėl, siekiant laiku gauti naujausią informaciją, būtina naudotis pasaulinės informacijos kanalais, iš kurių operatyviausi yra pagrįsti informacinėmis technologijomis</p>

9.2. UBL dokumentų duomenų tipai aprašai

9.2.1 Lentelė. UBL dokumentų primityvūs neklasifikuotų duomenų tipai.

Primityvus tipas	XML praplėstas tipas	Aprašymas
AmountType	decimal	Piniginių vienetų kiekis aprašomas su kartu su valiuta.
BinaryObjectType	base64Binary	Dvejetainio objekto tipas.
GraphicType	base64Binary	Diagrama, grafikas, matematinės kreivės ar panaši reprezentacija.
PictureType	base64Binary	Analogiškas <i>GraphicType</i> .
SoundType	base64Binary	Garso saugojimui skirtas tipas.
VideoType	base64Binary	Video saugojimui skirtas tipas.
CodeType	string	Ženklų sekai, kuri naudojama trumpinti tam tikras reikšmes – kodui saugoti.
DateTimeType	dateTime	Naudojamas datai ir/arba laikui saugoti.
DateType	date	Naudojamas datai saugoti.
TimeType	time	Naudojamas laikui saugoti.
IdentifierType	string	Identifikatoriui (ženklų seka, skirta identifikuoti, išskirti iš kitų) saugoti.
IndicatorType	boolean	Loginių reikšmių sąrašas susidedantis iš <i>taip</i> ir <i>ne</i> .
MeasureType	decimal	Skaitinė reikšmė naudojama išreikšti objekto matą, kartu su matavimo vienetu.
NumericType	decimal	Skaitiniai informacijai saugoti.
ValueType	decimal	Analogiškai <i>ValueType</i> .
PercentType	decimal	Procentais išreikštai informacijai saugoti.
RateType	decimal	Reitingu išreikštai informacijai saugoti.
QuantityType	decimal	Kiekui saugoti skirtas tipas, ne piniginei informacijai saugoti.
TextType	string	Tekstui saugoti.
NameType	string	Asmens, vietos, daikto ar sąvokos pavadinimui saugojimui skirtas tipas.

9.2.2 lentelė. Linijos prekės tipo aprašymas.

UBL elementas	UBL duomenų tipas	Aprašymas
ID	Identifier	Pirkėjo priskirtas linijos prekės identifikatorius.
SalesOrderID	Identifier	Pardavėjo suteiktas identifikatorius linijai.
UUID	Identifier	Unikalus universalus identifikatorius.
Note	Text	Laisvos formos tekstas, taikomas užsakymui. Šis elementas gali būti skirtas užrašams ar bet kuriai kitai

		panašiai informacijai, kuri nėra jokiam kitame elemente.
LineStatusCode	Code	Linijos būsenos kodas, atsižvelgiant į pradinę būseną.
Quantity	Quantity	Linijos prekių kiekis.
LineExtensionAmount	Amount	Bendra kaina įskaitant nuolaidas, bet atmetus mokesčius.
TotalTaxAmount	Amount	Bendra suma su mokesčiais.
MinimumQuantity	Quantity	Minimalus norimas prekės kiekis.
MaximumQuantity	Quantity	Maksimalus norimas prekės kiekis.
MinimumBackorderQuantity	Quantity	Minimalus grąžinimo dydis.
MaximumBackorderQuantity	Quantity	Maksimalus grąžinimo dydis.
InspectionMethodCode	Code	Patikrinimo metodo kodas.
PartialDeliveryIndicator	Indicator	Nurodo ar galimas dalinis pristatymas.
BackOrderAllowedIndicator	Indicator	Nurodo ar galimas grąžinimas.
AccountingCostCode	Code	Pirkėjo apskaitos kodas prekei.
AccountingCost	Text	Pirkėjo apskaitos kodas prekei, išreikštas tekstu.
Delivery	Delivery	Pristatymo informacija.
DeliveryTerms	DeliveryTerms	Pristatymo terminai.
OriginatorParty	Party	Gamintojo šalies aprašymas.
PricingReference	PricingReference	Kainos nuoroda.
AllowanceCharge	AllowanceCharge	Priemokos/nuolaidos informacija.
Price	Price	Kaina (kainų periodai, apmokėjimo būdai ir t.t.).
Item	Item	Prekė (pavadinimas, aprašymas, savybės, kilmė, klasifikacija ir t.t.).

9.2.3 lentelė. Šalies tipo aprašymas.

UBL elementas	UBL duomenų tipas	Aprašymas
WebsiteURI	Identifier	Šalies internetinė nuoroda.
LogoReferenceID	Identifier	Nuoroda į šalies logotipą.
EndpointID	Identifier	Nurodo nuorodą į galinį tašką.
PartyIdentification	PartyIdentification	Šalies identifikatorius.
PartyName	PartyName	Šalies pavadinimas.
Language	Language	Kalba.
PostalAddress	PostalAddress	Pašto adresas.
PhysicalLocation	PhysicalLocation	Fizinė buvimo vieta.
PartyTaxScheme	PartyTaxScheme	Šalies mokėjimo schema.
PartyLegalEntity	PartyLegalEntity	Informacija apie juridinį asmenį.
Contact	Contact	Kontaktinė informacija.
Person	Person	Konatinis asmuo.
AgentParty	Party	Nuoroda į kitą įgaliotinę šalį.

9.2.4 lentelė. Pirkėjo šalies aprašas.

UBL elementas	UBL duomenų tipas	Aprašymas
CustomerAssignedAccountID	Identifier	Pirkėjo priskirtas sąskaitos identifikatorius.
SupplierAssignedAccountID	Identifier	Pardavėjo priskirtas sąskaitos identifikatorius.
AdditionalAccountID	Identifier	Papildomos sąskaitos identifikatoriai
Party	Party	Informacija organizacijai aprašyti (pavadinimas, vietovė, kontaktai, atsakingi asmenys ir t.t.).
DeliveryContact	Contact	Kontaktinė informacija apie už pristatymą atsakingą asmenį arba įmonės skyrių.
BuyerContact	Contact	Kontaktinė informacija apie pirkėjo atsakingą asmenį arba įmonės skyrių.
AccountingContact	Contact	Kontaktinė informacija apie atsakingą asmenį arba skyrių už įmonės sąskaitos tvarkymą.

9.2.5 lentelė. Pardavėjo šalies aprašas.

UBL elementas	UBL duomenų tipas	Aprašymas
CustomerAssignedAccountID	Identifier	Pirkėjo priskirtas sąskaitos identifikatorius.
SupplierAssignedAccountID	Identifier	Pardavėjo priskirtas sąskaitos identifikatorius.
DataSendingCapability	Text	Informacija apie duomenų siuntimo galimybę.
Party	Party	Informacija organizacijai aprašyti (pavadinimas, vietovė, kontaktai, atsakingi asmenys ir t.t.).
DespatchContact	Contact	Kontaktinė informacija apie už išsiuntimą atsakingą asmenį arba įmonės skyrių.
SellerContact	Contact	Kontaktinė informacija apie pardavėjo atsakingą asmenį arba įmonės skyrių.
AccountingContact	Contact	Kontaktinė informacija apie atsakingą asmenį arba skyrių už įmonės sąskaitos tvarkymą.

9.2.6 lentelė. Užsakymo eilutės aprašymas.

UBL elementas	UBL duomenų tipas	Aprašymas
SubstitutionStatusCode	Code	Pakeitimų (ne)buvimo kodas.
LineItem	LineItem	Eilutės prekė.
SellerProposedSubstituteLineItem	LineItem	Pardavėjo siūlomas prekės pakeitimas.
SellerSubstitutedLineItem	LineItem	Pardavėjo padarytas prekės pakeitimas
CatalogueLineReference	Line Reference	Nuoroda į eilutę.
QuotationLineReference	Line Reference	Nuorodą į prekių katalogą, kuria naudojantis padarytas užsakymas
DocumentReference	Document Reference	Informacija apie dokumentą, nurodytą kito dokumento.

9.3. Užsakymų aptarnavimo komponento sąsajos WSDL

```
<wsdl:definitions xmlns:tns="http://service.ubl.ak" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://service.ubl.ak">
  <wsdl:types>
    <xsd:schema xmlns="http://schemas.xmlsoap.org/wsdl/" attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://service.ubl.ak">
      <xsd:import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmime.xsd"/>
      <xsd:complexType name="OrderType">
        <xsd:sequence>
          <xsd:element minOccurs="0" name="auth" type="xsd:string"/>
          <xsd:element minOccurs="0" name="orderData" type="xmime:base64Binary"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="OrderChangeType">
        <xsd:sequence>
          <xsd:element minOccurs="0" name="auth" type="xsd:string"/>
          <xsd:element minOccurs="0" name="orderChangeData" type="xmime:base64Binary"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="OrderCancellationType">
        <xsd:sequence>
          <xsd:element minOccurs="0" name="auth" type="xsd:string"/>
          <xsd:element minOccurs="0" name="orderCancellationData" type="xmime:base64Binary"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="OrderResponseType">
        <xsd:sequence>
          <xsd:element minOccurs="0" name="OrderResponse" type="xsd:boolean"/>
          <xsd:element minOccurs="0" name="OrderResponseError" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:element name="OrderRequest" type="tns:OrderType"/>
      <xsd:element name="OrderChangeRequest" type="tns:OrderChangeType"/>
    </xsd:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        <xsd:element name="OrderCancellationRequest"
type="tns:OrderCancellationType"/>

        <xsd:element name="OrderResponse"
type="tns:OrderResponseType"/>

        </xsd:schema>
    </wsdl:types>
    <wsdl:message name="OrderRequest">
        <wsdl:part name="part1" element="tns:OrderRequest"/>
    </wsdl:message>
    <wsdl:message name="OrderChangeRequest">
        <wsdl:part name="part1" element="tns:OrderChangeRequest"/>
    </wsdl:message>
    <wsdl:message name="OrderCancellationRequest">
        <wsdl:part name="part1"
element="tns:OrderCancellationRequest"/>
    </wsdl:message>
    <wsdl:message name="OrderResponse">
        <wsdl:part name="part1" element="tns:OrderResponse"/>
    </wsdl:message>

    <wsdl:portType name="UBLServicePortType">
        <wsdl:operation name="order">
            <wsdl:input message="tns:OrderRequest"
wsaw:Action="order"/>
            <wsdl:output message="tns:OrderResponse"
wsaw:Action="orderRespose"/>
        </wsdl:operation>
        <wsdl:operation name="orderChange">
            <wsdl:input message="tns:OrderChangeRequest"
wsaw:Action="orderChange"/>
            <wsdl:output message="tns:OrderResponse"
wsaw:Action="orderRespose"/>
        </wsdl:operation>
        <wsdl:operation name="orderCancellation">
            <wsdl:input message="tns:OrderCancellationRequest"
wsaw:Action="orderCancellation"/>
            <wsdl:output message="tns:OrderResponse"
wsaw:Action="orderRespose"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="UBLServiceSOAP11Binding"
type="tns:UBLServicePortType">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="order">
            <soap:operation soapAction="order" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

```

```

        <wsdl:operation name="orderChange">
            <soap:operation soapAction="orderChange"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="orderCancellation">
            <soap:operation soapAction="orderCancellation"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="UBLServiceSOAP12Binding"
type="tns:UBLServicePortType">
        <soap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="order">
            <soap12:operation soapAction="order"
style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="orderChange">
            <soap:operation soapAction="orderChange"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="orderCancellation">
            <soap:operation soapAction="orderCancellation"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

```

```
    <wsdl:service name="UBLService">
      <wsdl:port name="UBLServiceSOAP11port_http"
binding="tns:UBLServiceSOAP11Binding">
        <soap:address
location="http://localhost:8080/axis2/services/UBLService"/>
        </wsdl:port>
      <wsdl:port name="UBLServiceSOAP12port_http"
binding="tns:UBLServiceSOAP12Binding">
        <soap12:address
location="http://localhost:8080/axis2/services/UBLService"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

9.4. UBL dokumentai naudojami testavimui

UBL-Order.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<Order
xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDataty
pes-2"
xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentP
arameters-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComp
onents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregate
Components-2"
xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaM
odule:2" xmlns="urn:oasis:names:specification:ubl:schema:xsd:Order-2">
  <cbc:UBLVersionID>2.0</cbc:UBLVersionID>
  <cbc:ID>AEG012345</cbc:ID>
  <cbc:SalesOrderID>CON0095678</cbc:SalesOrderID>
  <cbc:CopyIndicator>>false</cbc:CopyIndicator>
  <cbc:UUID>191-AEG012345-20081229</cbc:UUID>
  <cbc:IssueDate>20081229</cbc:IssueDate>
  <cbc:Note>testuojamas uzsakymas</cbc:Note>
  <cac:BuyerCustomerParty>

    <cbc:SupplierAssignedAccountID>1</cbc:SupplierAssignedAccountID>
    <cac:Party>
      <cac:PartyName>
        <cbc:Name>UAB Test Pirkejas</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:StreetName>Borutos g.</cbc:StreetName>
        <cbc:BuildingNumber>56A</cbc:BuildingNumber>
        <cbc:CityName>Kaunas</cbc:CityName>
        <cbc:PostalZone>46115</cbc:PostalZone>
        <cac:AddressLine>
          <cbc:Line>24 kab.</cbc:Line>
        </cac:AddressLine>
      </cac:PostalAddress>
      <cac:Contact>
        <cbc:Name>Andrius Klimavicius</cbc:Name>
        <cbc:Telephone>8 37 55 15 16</cbc:Telephone>
      </cac:Contact>
    </cac:Party>
  </cac:BuyerCustomerParty>
  <cac:SellerSupplierParty>

    <cbc:CustomerAssignedAccountID>CO001</cbc:CustomerAssignedAccountID>
    <cac:Party>
      <cac:PartyName>
        <cbc:Name>Test Tiekejas</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:StreetName>Vytatuo g.</cbc:StreetName>
```

```

        <cbc:BuildingNumber>23</cbc:BuildingNumber>
        <cbc:CityName>Vilnius</cbc:CityName>
        <cbc:PostalZone>21456</cbc:PostalZone>
    </cac:PostalAddress>
    <cac:Contact>
        <cbc:Name>Saulius</cbc:Name>
        <cbc:Telephone>8 37 578 432</cbc:Telephone>
        <cbc:Telefax>8 37 578 433</cbc:Telefax>

    <cbc:ElectronicMail>uzsakymai@tiekejas.lt</cbc:ElectronicMail>
    </cac:Contact>
</cac:Party>
</cac:SellerSupplierParty>
<cac:Delivery>
    <cac:DeliveryAddress>
        <cbc:StreetName>Lazdynu g.</cbc:StreetName>
        <cbc:BuildingNumber>124A</cbc:BuildingNumber>
        <cbc:CityName>Kaunas</cbc:CityName>
        <cbc:PostalZone>45678</cbc:PostalZone>
        <cac:AddressLine>
            <cbc:Line>3a., 512 kabinetas</cbc:Line>
        </cac:AddressLine>
    </cac:DeliveryAddress>
</cac:Delivery>
<cac:OrderLine>
    <cbc:Note>Uzsakymo eilutes pvz</cbc:Note>
    <cac:LineItem>
        <cbc:ID>1</cbc:ID>
        <cbc:Quantity>100</cbc:Quantity>
        <cac:Item>
            <cbc:Description>Mobilusis telefonas Nokia
3310</cbc:Description>
            <cbc:Name>Nokia 3310</cbc:Name>
            <cac:SellersItemIdentification>
                <cbc:ID>2</cbc:ID>
            </cac:SellersItemIdentification>
        </cac:Item>
    </cac:LineItem>
    <cac:LineItem>
        <cbc:ID>2</cbc:ID>
        <cbc:Quantity>200</cbc:Quantity>
        <cac:Item>
            <cbc:Description>Mobilusis telefonas Motorola
L3</cbc:Description>
            <cbc:Name>Motorola L3</cbc:Name>
            <cac:SellersItemIdentification>
                <cbc:ID>1</cbc:ID>
            </cac:SellersItemIdentification>
        </cac:Item>
    </cac:LineItem>
</cac:OrderLine>
</Order>

```

UBL-OrderChange.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderChange
xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDatatypes-2"
xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentParameters-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaModule:2"
xmlns="urn:oasis:names:specification:ubl:schema:xsd:OrderChange-2">
  <cbc:UBLVersionID>2.0</cbc:UBLVersionID>

  <cbc:CustomizationID>urn:oasis:names:specification:ubl:xpath:Order-2.0:sbs-1.0-draft</cbc:CustomizationID>
  <cbc:ProfileID>bpid:urn:oasis:names:draft:bpss:ubl-2-sbs-order-with-simple-response-draft</cbc:ProfileID>
  <cbc:ID>AEG012346</cbc:ID>
  <cbc:SalesOrderID>CON0095680</cbc:SalesOrderID>
  <cbc:CopyIndicator>>false</cbc:CopyIndicator>
  <cbc:UUID>191-AEG012346-20081229</cbc:UUID>
  <cbc:IssueDate>20090103</cbc:IssueDate>
  <cbc:Note>Uzsakymo pakeitimas</cbc:Note>
  <cac:OrderReference>
    <cbc:UUID>191-AEG012345-20081229</cbc:UUID>
  </cac:OrderReference>
  <cac:BuyerCustomerParty>

    <cbc:SupplierAssignedAccountID>1</cbc:SupplierAssignedAccountID>
    <cac:Party>
      <cac:PartyName>
        <cbc:Name>UAB Test Pirkejas</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:StreetName>Borutos g.</cbc:StreetName>
        <cbc:BuildingNumber>56A</cbc:BuildingNumber>
        <cbc:CityName>Kaunas</cbc:CityName>
        <cbc:PostalZone>46115</cbc:PostalZone>
        <cac:AddressLine>
          <cbc:Line>24 kab.</cbc:Line>
        </cac:AddressLine>
      </cac:PostalAddress>
      <cac:Contact>
        <cbc:Name>Andrius Klimavicius</cbc:Name>
        <cbc:Telephone>8 37 55 15 16</cbc:Telephone>
      </cac:Contact>
    </cac:Party>
  </cac:BuyerCustomerParty>
  <cac:SellerSupplierParty>
```

```

<cbc:CustomerAssignedAccountID>CO001</cbc:CustomerAssignedAccountID>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>Test Tiekėjas</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Vytatuo g.</cbc:StreetName>
      <cbc:BuildingNumber>23</cbc:BuildingNumber>
      <cbc:CityName>Vilnius</cbc:CityName>
      <cbc:PostalZone>21456</cbc:PostalZone>
    </cac:PostalAddress>
    <cac:Contact>
      <cbc:Name>Saulius</cbc:Name>
      <cbc:Telephone>8 37 578 432</cbc:Telephone>
      <cbc:Telefax>8 37 578 433</cbc:Telefax>

    <cbc:ElectronicMail>uzsakymai@tiekejas.lt</cbc:ElectronicMail>
  </cac:Contact>
</cac:Party>
</cac:SellerSupplierParty>
<cac:Delivery>
  <cac:DeliveryAddress>
    <cbc:StreetName>Kauno g.</cbc:StreetName>
    <cbc:BuildingNumber>15</cbc:BuildingNumber>
    <cbc:CityName>Vilnius</cbc:CityName>
    <cbc:PostalZone>21233</cbc:PostalZone>
    <cac:AddressLine>
      <cbc:Line>212a kabinetas</cbc:Line>
    </cac:AddressLine>
  </cac:DeliveryAddress>
</cac:Delivery>
<cac:OrderLine>
  <cbc:Note>Užsakymo eilutes pvz</cbc:Note>
  <cac:LineItem>
    <cbc:ID>1</cbc:ID>
    <cbc:Quantity>200</cbc:Quantity>
    <cac:Item>
      <cbc:Description>Mobilusis telefonas Nokia
3310</cbc:Description>
      <cbc:Name>Nokia 3310</cbc:Name>
      <cac: SellersItemIdentification>
        <cbc:ID>2</cbc:ID>
      </cac: SellersItemIdentification>
    </cac:Item>
  </cac:LineItem>
  <cac:LineItem>
    <cbc:ID>2</cbc:ID>
    <cbc:Quantity>200</cbc:Quantity>
    <cac:Item>
      <cbc:Description>Mobilusis telefonas Motorola
L3</cbc:Description>
      <cbc:Name>Motorola L3</cbc:Name>
      <cac: SellersItemIdentification>
        <cbc:ID>1</cbc:ID>

```



```
        </cac:SellersItemIdentification>
    </cac:Item>
</cac:LineItem></cac:OrderLine></OrderChange>
```

UBL-OrderCancellation.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderCancellation
  xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDatatypes-2"
  xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentParameters-2"
  xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaModule:2" xmlns="urn:oasis:names:specification:ubl:schema:xsd:Order-2">
  <cbc:UBLVersionID>2.0</cbc:UBLVersionID>

  <cbc:CustomizationID>urn:oasis:names:specification:ubl:xpath:Order-2.0:sbs-1.0-draft</cbc:CustomizationID>
    <cbc:ProfileID>bpid:urn:oasis:names:draft:bpss:ubl-2-sbs-order-with-simple-response-draft</cbc:ProfileID>
      <cbc:ID>AEG012347</cbc:ID>
      <cbc:SalesOrderID>CON0095681</cbc:SalesOrderID>
      <cbc:CopyIndicator>>false</cbc:CopyIndicator>
      <cbc:UUID>191-AEG012347-20081229</cbc:UUID>
      <cbc:IssueDate>20090103</cbc:IssueDate>
      <cbc:CancellationNote>Nebereikalingos
prekes</cbc:CancellationNote>
      <cac:OrderReference>
        <cbc:UUID>191-AEG012345-20081229</cbc:UUID>
      </cac:OrderReference>
      <cac:BuyerCustomerParty>

        <cbc:SupplierAssignedAccountID>1</cbc:SupplierAssignedAccountID>
        <cac:Party>
          <cac:PartyName>
            <cbc:Name>UAB Test Pirkejas</cbc:Name>
          </cac:PartyName>
          <cac:PostalAddress>
            <cbc:StreetName>Borutos g.</cbc:StreetName>
            <cbc:BuildingNumber>56A</cbc:BuildingNumber>
            <cbc:CityName>Kaunas</cbc:CityName>
            <cbc:PostalZone>46115</cbc:PostalZone>
            <cac:AddressLine>
              <cbc:Line>24 kab.</cbc:Line>
            </cac:AddressLine>
          </cac:PostalAddress>
          <cac:Contact>
            <cbc:Name>Andrius Klimavicius</cbc:Name>
            <cbc:Telephone>8 37 55 15 16</cbc:Telephone>
          </cac:Contact>
        </cac:Party>
      </cac:BuyerCustomerParty>
```

```
<cac:SellerSupplierParty>
  <cbc:CustomerAssignedAccountID>C0001</cbc:CustomerAssignedAccountID>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>Test Tiekėjas</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Vytatuo g.</cbc:StreetName>
      <cbc:BuildingNumber>23</cbc:BuildingNumber>
      <cbc:CityName>Vilnius</cbc:CityName>
      <cbc:PostalZone>21456</cbc:PostalZone>
    </cac:PostalAddress>
    <cac:Contact>
      <cbc:Name>Saulius</cbc:Name>
      <cbc:Telephone>8 37 578 432</cbc:Telephone>
      <cbc:Telefax>8 37 578 433</cbc:Telefax>
      <cbc:ElectronicMail>uzsakymai@tiekejas.lt</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:SellerSupplierParty>
</OrderCancellation>
```

9.5. Sugeneruotos SQL užklauso

Užsakymas.

Įvykdyta užklausa:

```
>java UBLSeviceClient uzsakyti test/TEST/UBL-Order-2.0-Example.xml
„UAB Test Klientas“ „UAB Test Tiekėjas“
```

Sugeneruotos SQL užklauso:

```
INSERT INTO `uzsakymai` (`id`, `kliento_id`,
`uzsakymo_data`, `adresas`, `miestas`, `pastoKodas`)
VALUES
(228, 1, '2008-01-29 00:00:00', '3a., 512 kabinetas,
Lazdynu g. 124A', 'Kaunas', '45678');

INSERT INTO `uzsakymo_prekes` (`id`, `uzsakymo_id`,
`prekes_id`, `kiekis`, `nuolaida`) VALUES
(97, 228, 2, 100, NULL),
(98, 228, 1, 200, NULL);

INSERT INTO `ubl_uzsakymai` (`uzsakymo_id`, `busena`,
`ubl_id`, `data`, `komentaras`, `uuid`) VALUES
(228, 1, 'AEG012345', '2008-12-29 23:12:45', NULL, '191-
AEG012345-20081229');
```

Užsakymo pakeitimas.

Įvykdyta užklausa:

```
>java UBLSeviceClient pakeisti test/TEST/UBL-OrderChange-Example.xml
„UAB Test Klientas“ „UAB Test Tiekėjas“
```

Sugeneruota SQL užklausa:

```
UPDATE uzsakymai SET uzsakymo_data = '2008-01-29
00:00:00', adresas = '212a kabinetas, Kauno g. 15',
miestas = 'Vilnius', pastoKodas= 21233
WHERE id = 228;

DELETE FROM uzsakymo_prekes WHERE uuid = 228;

INSERT INTO `uzsakymo_prekes` (`id`, `uzsakymo_id`,
`prekes_id`, `kiekis`, `nuolaida`) VALUES
(99, 228, 2, 200, NULL),
(100, 228, 1, 200, NULL);
```

```
UPDATE ubl_uzsakymai SET busena = 5, data = NOW(),  
komentaras = 'Uzsakymo pakeitimas' WHERE uuid='191-  
AEG012345-20081229';
```

Uzsakymo atšaukimas.

Įvykdyta užklausa:

```
>java UBLSeviceClient atsaukti test/TEST/UBL-OrderCancellation-  
Example.xml „UAB Test Klientas“ „UAB Test Tiekėjas“
```

Sugeneruota SQL užklausa:

```
UPDATE ubl_uzsakymai SET busena = 3, data = NOW(),  
komentaras = 'Nebereikalingos  
prekes' WHERE uuid='191-AEG012345-20081229'
```