

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**KOMPIUTERIŲ KATEDRA**

Vytautas Macaitis

**Verslo taisyklių suderinimas įmonių sąveikumo  
sprendimuose**

Magistro darbas

Darbo vadovas

doc. Nerijus Morkevičius

Kaunas, 2011

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**  
**KOMPIUTERIŲ KATEDRA**

Vytautas Macaitis

**Verslo taisyklių suderinimas įmonių sąveikumo  
sprendimuose**

Magistro darbas

Recenzentas  
prof. dr. Lina Nemuraitė

2011-05-

Darbo vadovas  
doc. Nerijus Morkevičius

2011-05-

Atliko  
IFM-9/4 gr. Stud.  
Vytautas Macaitis

2011-05-

Kaunas, 2011

## SUMMARY

### **Business rules reconciliation in enterprise interoperability solutions**

Written by Vytautas Macaitis

This paper investigates the interoperability solutions and specific methods of the reconciliation of business rules, small and medium-sized business (hereinafter referred to as SMB) management in information systems, as well as the problems and opportunities related to such reconciliation.

The main objective of the research is to review and analyse the problems of business rules reconciliation and their solutions. The paper will attempt to create a method based on results, which will allow users to reconcile business rules in enterprise interoperability solutions more effectively.

The goal of the paper is to investigate existing reconciliation methods of business rules, their limitations and possibilities, as well as offer a method which will be more effective in reconciling negotiation rules of two or more companies. Utilising the proposed method, the paper aims to find solutions to the problems related to automated negotiations where previously outlined negotiation rules between two or more parties are automatically reconciled and applied to the exchanged business documents with a minimal participation of people.

The need for such research arose from the increased competition between SMBs, the necessity to enhance companies' interoperability when fulfilling orders, as well as the pursuit to offer more effective, better quality services.

The paper consists of the introduction, six parts and the conclusion. The first four parts encapsulate the relevance of the problem, the definition of the paper's goals and tasks, acquaint with the structure of the research, offer comprehensive overview and analysis of existing solutions, define the concept of the intended solution and describe the requirements' specifications. The system plan corresponding to the established requirements is presented. The method of business rules reconciliation, the conditions, which the experiment requires, and the novelty of the research are presented in Part 5. The research on business rules reconciliation method is generalised in Part 6.

# Turinys

<b>1. ĮVADAS.....</b>	<b>5</b>
<b>2. VERSLO TAISYKLIŲ SUDERINIMO ĮMONIŲ SAŲVEIKUMO SPRENDIMUOSE ANALIZĖ....</b>	<b>7</b>
2.1. ANALIZĖS TIKSLAS .....	7
2.2. ANALIZĖS METODAI .....	7
2.3. TYRIMO OBJEKTO ANALIZĖ .....	8
2.4. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA.....	9
2.5. TYRIMO TIKSLAS, AKTUALUMAS IR UŽDAVINIAI .....	9
2.6. TYRIMO PLANAS.....	11
2.7. VARTOTOJŲ ANALIZĖ .....	11
2.7.1. Vartotojų aibė, tipai ir savybės.....	11
2.7.2. Vartotojų tikslai ir problemos.....	12
2.8. ESAMŲ SPRENDIMŲ ANALIZĖ.....	12
2.8.1. SaŲveikumo sprendimai skirti SVV.....	12
2.8.2. SaŲveikumo sprendimuose naudojami dokumentų standartai .....	19
2.8.3. Derybų taisyklių suderinimo sprendimai .....	24
2.9. SIEKIAMAS SPRENDIMAS .....	28
2.10. ANALIZĖS IŠVADOS .....	29
<b>3. SISTEMOS REIKALAVIMŲ SPECIFIKACIJA IR ANALIZĖ .....</b>	<b>30</b>
3.1. REIKALAVIMŲ SPECIFIKACIJA .....	30
3.1.1. Funkciniai reikalavimai.....	31
3.1.2. Nefunkciniai reikalavimai .....	37
3.2. REIKALAVIMŲ ANALIZĖS APIBENDRINIMAS.....	38
<b>4. SISTEMOS PROJEKTAS .....</b>	<b>39</b>
4.1. SISTEMOS ARCHITEKTŪROS PROJEKTAS .....	40
4.1.1. Sistemos loginė architektūra.....	40
4.1.2. Panaudojimo atvejų specifikacija .....	41
4.1.3. Veiklos paslaugu architektūra.....	44
4.2. DETALUS PROJEKTAS .....	45
4.2.1. Panaudojimo atvejų realizacijos – analizės klasių diagramos .....	45
4.3. SISTEMOS ELGSENOS MODELIS .....	47
4.3.1. Sistemos ir jos elementų būsenų modeliai.....	47
4.3.2. Veiklos proceso modeliai .....	47
4.3.3. Veiklos transakcijų modelis.....	49
4.3.4. Veiklos esybių būsenų diagramos .....	50
<b>5. REALIZACIJA.....</b>	<b>51</b>
5.1. VERSLO TAISYKLIŲ SUDERINIMO SPRENDIMO SAVYBIŲ ANALIZĖ BEI VEIKIMO APRAŠAS .....	51
5.1.1. Užsakymo dokumentai.....	51
5.1.2. Taisyklės (formalizavimas, sukūrimas) .....	55
5.1.3. Vartotojų susiduriamos problemos .....	59
5.1.4. Taisyklių pritaikymas .....	62
5.1.5. Taisyklių nesuderinamumas.....	65
5.1.6. Taisyklių suderinimas .....	67
5.2. EKSPERIMENTO REALIZACIJOS APRAŠYMAS .....	73
5.2.1. Sprendimo reikalavimų specifikacija .....	73
5.2.2. Eksperimento realizacijos koncepcija.....	74
5.3. TESTAVIMO MODELIS BEI DUOMENYS, KONTROLINIS PAVYZDYS.....	80
<b>6. EKSPERIMENTINIS SISTEMOS TYRIMAS.....</b>	<b>83</b>
6.1. EKSPERIMENTO PLANAS BEI GAUTI REZULTATAI.....	83
6.2. SUKURTO METODO IR REALIZACIJOS APIBENDRINIMAS .....	86
<b>7. IŠVADOS .....</b>	<b>87</b>
<b>8. LITERATŪRA.....</b>	<b>88</b>
<b>9. PRIEDAI .....</b>	<b>90</b>

## 1. Įvadas

Veiklos perkėlimas į internetą šiandien yra nebe naujovė, bet būtinybė. Internetas tapo vienu iš pagrindinių informacijos apsikeitimo būdų, prieinamų dideliame žmonių ir įmonių skaičiui.

Vienas pagrindinių verslo įmonių gyvavimo reikalavimų – nuolatinis, organizuotas tarpusavio bendradarbiavimas ir naujų partnerių ieškojimas. Vyksta savotiškas įmonių tarpusavio sąveikumas. T.y. atskirų ir iš esmės skirtingų organizacijų gebėjimas bendradarbiauti siekiant abipusės naudos ir užsibrėžtų bendrų tikslų. Tai visų suinteresuotųjų grupių puoselėjama vertybė ir nuolat vertinama vykdomos veiklos kokybinė charakteristika, kurią galima išreikšti per išmatuojamus kiekybinius rodiklius (arba metrikas).

Bendravimas (bendradarbiavimas) dažniausiai vyksta žodžiu (įvairūs dalykiniai verslo susitikimai bei telefoniniai pokalbiai) arba elektroniniu paštu. Užmezgamas savotiškas bendro tikslo siekimas. Šis bendravimas neatsiejamas nuo sąveikumo. T.y. bendradarbiaujančios įmonės neišvengiamai pradeda „operuoti“ įvairiais dokumentais, strategijomis, apribojimais ir panašiai. Vyksta derybų procesai, kurių metu įmonės nustato tam tikras taisykles viena kitos atžvilgiu (suteikia nuolaidas patikimiems klientams, taiko lanksčią kainų politiką ir pan.). Tačiau tai atima labai daug įmonių vadovų bei atsakingų asmenų laiko, be to tokios verslo taisyklės dažniausiai sąlygoja sudėtingą dokumentų pildymą bei, galimai didelį jų koregavimo skaičių. Tokias derybas tikslinga automatizuoti, kad sąveikumo sistema pati, turėdama iš anksto nustatytas įmonės verslo (derybų) taisykles, suderintų jas su kitų įmonių taisyklėmis taip sumažindama derybininkų apkrovą. Verslo taisyklė tai tokia taisyklė, kuri apima bei įgyvendina verslo politiką ir praktiką, priimdama sprendimus ar išvesdama naujus duomenis iš jau esamų duomenų. Dažniausiai pasitaiko dviejų tipų verslo taisyklės, t.y. „Jei - Tai“ taisyklės, bei „Veiksmų“ taisyklės. Pirmuoju atveju atsižvelgiama į tam tikrus, realiu laiku gaunamus pranešimus ar duomenis (pvz.: jei užsakymo suma viršija šimtą litų, tai taikoma dešimties litų nuolaida), antruoju - taisyklės vykdomos neatsižvelgiant į gaunamus duomenis.

Verslo įmonių sąveikumo sprendimai, jau ne pirmi metai bandomi vykdyti internetu ir yra gana didelė tokių sistemų įvairovė – nuo paprastų informacinių puslapių iki didelių, sudėtingų sistemų. Tačiau Lietuvoje šiuo metu tokios sistemos nėra pakankamai išvystytos efektyviam verslo plėtojimui, nors paklausa yra didelė. Dar daugiau, egzistuojančiuose sąveikumo sprendimuose dažnai visai nėra atsižvelgiama į automatinio derybų ir verslo taisyklių suderinimo poreikį, todėl vykdant verslo derybas įmonės dažniausiai pasikliauja tik įprastomis bendravimo priemonėmis tokiomis kaip telefonas, faksas, elektroninis paštas.

Pasaulinio lygio verslo derybų suderinimo sistemos taip pat dar nėra išvystytos. Jos kol kas tik mokslinių tyrimų ir prototipų lygyje. Žmonės, susiję su šiomis sistemomis, sprendžia įvairias problemas, ieško geriausių sprendimų. Jie nori pasiekti tokio rezultato, kuris tenkintų daugumą pasaulio įmonių. Tikisi, kad sistemos nebūtų pritaikytos tam tikram specifiniam įmonių ratui, tam tikrai kalbai, siekia, kad būtų efektyviai naudojamos nedidelėse, specifine veikla užsiimančiose įmonėse.

Sukurti internetinę sistemą, kurioje būtų galima reglamentuoti verslo derybų taisyklės nėra sunku, tačiau to nepakanka. Surašytas ir sureglamentuotas derybų taisyklės reikia efektyviu, greitu ir patikimu būdu suderinti tarpusavyje, kas nėra labai paprasta. Tą gali padaryti tik tinkama, aukštos kokybės programinė įranga.

Šiame darbe siekiame maksimaliai automatizuoti verslo derybų taisyklių suderinimą, iš anksto nustatant tik tam tikras taisyklės (reikalavimus), taip nustatant kokioms dalyvių grupėms reikia siųsti tam tikrų tipų pranešimus.

Kitas šio darbo tikslas suvienodinti verslo taisyklių reikalavimus, standartizuoti juos, pritaikyti lietuviškai rinkai. Dabar skirtingoms įmonėms reikalingos skirtingos taisyklės, kurios tarpusavyje sunkiai derinamos. Kitaip tariant, kiek įmonių, tiek skirtingų poreikių ir reikalavimų. Turint galimybes nustatyti atitinkamus taisyklių parametrus, galima juos įvairiais būdais suderinti. Reikalui esant, šias taisyklės galima pakoreguoti atsižvelgiant į rinkos pokyčius, pavyzdžiui, papildyti įvedamų taisyklių reikalavimus konkretesniais punktais. Pagal nustatytas, standartizuotas derybų taisyklės sistema kontaktuoja su kitomis įmonėmis ieškodama reikalavimus atitinkančių rezultatų. Taip sistema pati save „valdo“. Automatizuotai atliekamas derybų taisyklių derinimas ir rezultatų pateikimas derybininkams palieka tik galutinio įvertinimo ir pasirinkimo galimybę.

## **2. Verslo taisyklių suderinimo įmonių sąveikumo sprendimuose analizė**

### **2.1. Analizės tikslas**

Analizės tikslas – išsiaiškinti ir išanalizuoti teorinius bei esamus smulkaus ir vidutinio verslo įmonių sąveikumo sprendimus, jų apribojimus, galimybes. Išanalizuoti smulkaus ir vidutinio verslo derybų specifiką naudojant tam tikras taisykles. Nustatyti reikalavimus tokių taisyklių suderinimui. Apžvelgti sąveikumo sprendimų bei jų reikalavimų privalumus ir trūkumus, rasti egzistuojančius būdus šiems ir kitiems trūkumams pašalinti.

### **2.2. Analizės metodai**

Prieš pradėdant tyrimą, pirmiausiai pasirenkame metodus bei priemones jam atlikti. Tokių analizės metodų yra keletas, todėl pasirinktas tas, kuris tiksliausiai ir išsamiausiai atskleidžia esamą tiriamos srities situaciją.

Tyrimo objekto bei esamų sprendimų analizei atlikti gali būti taikomi tokie metodai:

- mokslinės literatūros analizė arba teorinės analizės ir apibendrinimo metodas,
- apklausa (anketinė, interviu, ekspertų apklausa),
- stebėjimas,
- dokumentų turinio analizė.
- eksperimentas,
- matematinė statistika.

Tačiau norint tiksliausiai atskleisti pirminės analizės rezultatus, metodas pasirenkamas toks, kuris apžvelgia didžiausią, eksperimentais bei stebėjimais pagrystą, informacijos kiekį. Taip pat jis turi apimti įvairių apklausų surinktus duomenis, dokumentų apžvalgas bei testavimus.

Taigi siekiant atlikti kokybišką bei išsamią tiriamos srities analizę, pasirinktas jos atlikimo metodas turi būti plataus pobūdžio, apimantis keletą aspektų. Todėl buvo pasirinktas “mokslinės, teorinės, apibendrinančiosios literatūros” analizės metodas.

### 2.3. Tyrimo objekto analizė

Tyrimų pagrindą sudaro įmonių verslo taisyklių suderinimo problemų ir sprendimų apžvalga ir analizė.

Įmonės, plėtodamos savo verslą, turi nuolat ieškoti partnerių, tiekėjų, vykdytojų ir panašiai. Nuolat priimti tam tikrus, efektyviam bendradarbiavimui skirtus sprendimus.

„Sąveikumas (angl. *Interoperability*) – atskirų ir skirtingų organizacijų gebėjimas bendrai veikti siekiant abiem šalims naudingų ir sutartų tikslų, įskaitant informacijos ir žinių mainus tarp organizacijų, atliekant veiklos procesus, kuriuos jos gali vykdyti, duomenų mainams naudojant tų organizacijų IRT sistemas“ [1] (IRT – Informacinės ir ryšių technologijos).

Tam, kad būtų užtikrinamas sąveikumas, tarp smulkaus ir vidutinio verslo organizacijų, turi būti parengti sąveikumo pagrindai, kurie apibrėžia galimus arba iš anksto susitartus įmonių/institucijų tarpusavio bendradarbiavimo metodus ir atitinkamų standartų parinkimo bei naudojimo tvarką. Šie pagrindai gali apibrėžti kriterijus, specifikacijas, standartus, protokolus, procedūras, apribojimus, lengvatas ir kita, kurie naudojami tarp-institucinių sprendimų priėmimui. Pavyzdžiui verslo procesai gali būti aprašyti tokiais standartais kaip BPE, BPEL, o verslo dokumentų apsikeitimui – EDI, UBL, ebXML ir panašūs standartai.

Kaip papildoma priemonė, šiuose sprendimuose galima sukurti verslo taisykles, kriterijus, pagal kuriuos būtų galima realizuoti automatines partnerių derybas panaudojus automatinį šių derybų suderinimą. Verslo taisyklė (angl. *Business Rule*) – ribojimas. Tai – sutarta taisyklė, kurios laikomasi verslo metu. Tokios taisyklės gali nustatyti kainų ar nuolaidų strategiją, vartotojų galimybes įvairiose situacijose, darbuotojų teises bei pareigas ir pan. Verslo taisyklių aiškus formulavimas naudingas visada, tačiau jis tampa absoliučiai būtinas jei imama programuoti jų automatišką laikymąsi kompiuterio pagalba. Kiekviena įmonė reglamentuotų savas taisykles. Atliekant paiešką pagal pateiktus nurodymus būtų atrinkti tik tokie rezultatai, kurie juos tenkintų.

Toks verslo taisyklių derinimas įmonių sąveikumo sprendimuose būtų labai efektyvus, tačiau tai nėra paprasta įgyvendinti. Problema yra šių taisyklių suderinime, kadangi kiekvienos įmonės taisyklės gali būti labai skirtingos ir tarpusavyje sunkiai susiejamos. Jas reikėtų perversti į vienodą visom taisyklėm standartą bei rasti būdą, kaip jas identifikuoti ir palyginti. Išsprendus šį uždavinį būtų galima išgauti tinkamą atsakymą.



## 2.4. Tyrimo sritis, objektas ir problema

### *Tyrimo sritis*

Smulkaus ir vidutinio verslo įmonių sąveikumo sprendimai, verslo taisyklių suderinimo problema įmonių sąveikume.

### *Tyrimo objektas*

Tyrimų pagrindą sudaro įmonių verslo taisyklių suderinimo problemų ir sprendimų apžvalga ir analizė. Remiantis analizės rezultatais darbe bus siekiama sukurti metodiką, leisiančią efektyviau suderinti verslo taisykles įmonių sąveikumo sprendimuose.

### *Tyrimo problema ir numatomas sprendimas*

Praktiškai įgyvendintuose ir įdiegtuose verslo įmonių sąveikumo sprendimuose dažnai iškyla derybų problema. Kiekviena įmonė dažnai taiko įvairias nuolaidas ir apribojimus priklausomai nuo to, su koku partneriu ji bendradarbiauja. Atliekant verslo transakcijas per sąveikumo sprendimus patogiau kai įmonės gali nustatyti ir naudoti formaliai apibrėžiamas derybų taisykles. Šiame darbe bus sprendžiama automatinių derybų problema, kai dvejų ar daugiau partnerių iš anksto suformuluotos derybų taisyklės automatiškai suderinamos ir pritaikomos perduodamiems verslo dokumentams, nedalyvaujant žmonėms.

## 2.5. Tyrimo tikslas, aktualumas ir uždaviniai

**Tikslas:** Ištirti egzistuojančių verslo taisyklių suderinimo metodus, jų apribojimus ir galimybes bei pasiūlyti metodiką, gebančią automatiškai suderinti dvejų ar daugiau įmonių derybų taisykles sąveikumo sistemose.

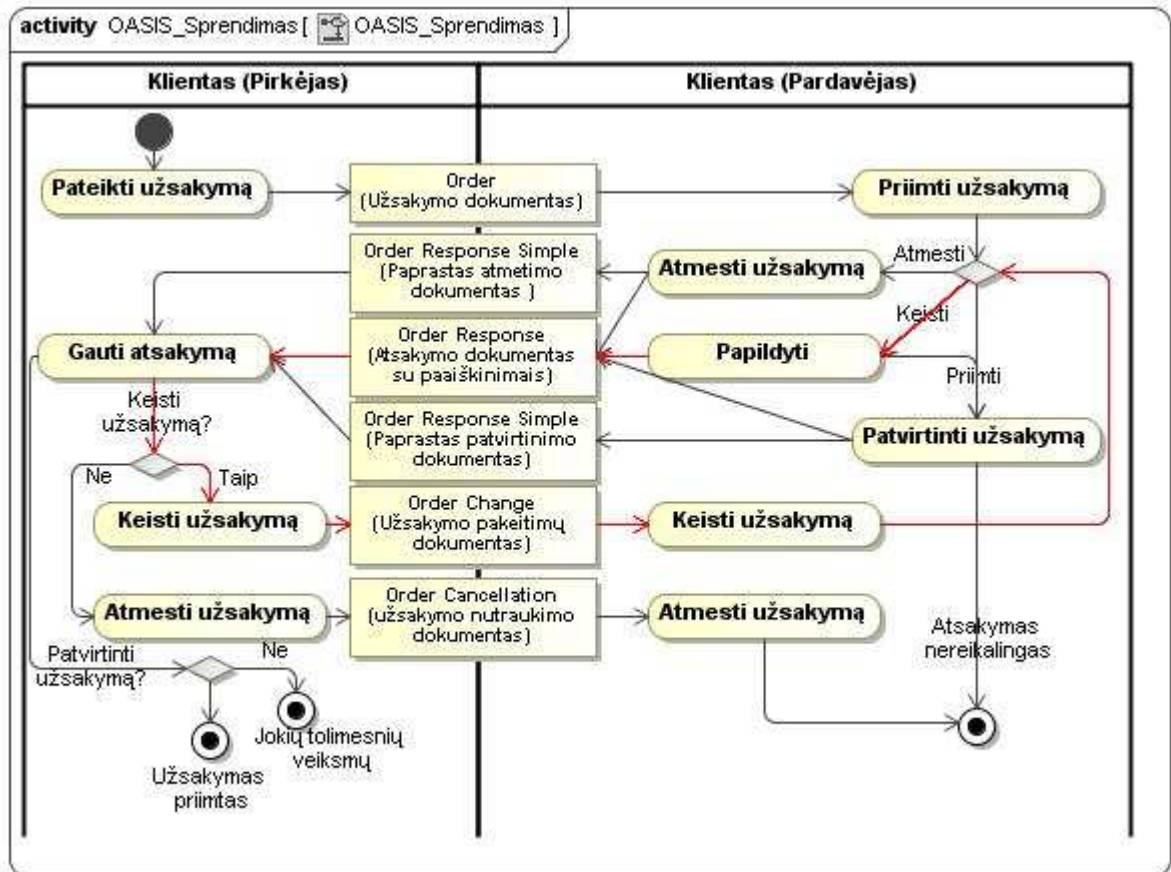
**Aktualumas:** Šiuo metu reta didesnė įmonė savo reikmėms nenaudoja integruotų kompiuterinių ERP sistemų, kurios skirtos vykdyti centralizuotą įmonės veiklos apskaitą: buhalterinę, logistikos, gamybos, paslaugų valdymo, žmogiškųjų resursu valdymo ir pan. Šias sistemas galima suskirstyti į dvi pagrindines grupes: sukurtas pagal individualų užsakymą įmonių konkretiems poreikiams tenkinti (1), arba sukurtas tenkinti standartizuotus verslo poreikius (2). Antrosios grupės sistemos perduodamos kaip verslo valdymo sistemos, pritaikytos apibendrintiems verslo procesams. Vėliau tokia sistema jau yra pritaikoma konkrečiam atvejui (parametrizuojama, keičiama verslo logika). Šiame darbe nagrinėjamos būtent antrosios grupės ERP sistemos.

Pastaruoju metu verslas sparčiai auga, dinamiškai keičiasi, o kartu su juo turi keistis ir verslą remianti verslo valdymo sistema – tam, kad galėtų operatyviai realizuoti augančius ir besikeičiančius verslo poreikius.

Bene svarbiausi įmonės verslo procesai vienokiu ar kitokiu būdu susiję su užsakymais. Priklausomai nuo situacijos, jie gali pareikalauti labai daug laiko, kas sumažina

maksimalų derybų efektyvumą. Laiko praradimai atsiranda dėl nesugebėjimo iš pirmo karto paruošti dokumentus, kurie tenkintų abi užsakymu suinteresuotas „šalis“. Dėl šios priežasties procesas kartojamas keletą kartų.

Tą pačią problemą išvelgia ir pasaulinis „OASIS“ (*Organization for the Advancement of Structured Information Standards*) konsorciumas. Išnagrinėjęs realiai šiuolaikiniame pasaulyje vykstančius derybų procesus, jis pateikia schemą (žr. 2.1 pav.), kurioje galima matyti, koks yra užsakymo gyvavimo ciklas.



2.1 pav. OASIS konsorciumo pateiktas užsakymo procesas [2]

Galima pastebėti, jog egzistuoja net keletas, specialiai deryboms pritaikytų dokumentų, kurie yra neišvengiami, norint tinkamai paruošti užsakymo dokumentą. Problemiškiausi ir daugiausia laiko užimantys iš jų yra „Order Response“ bei „Order Change“ dokumentai. Priklausomai nuo situacijos sąveikaujant dviem ar daugiau įmonių, jie gali pasikartoti keletą ir daugiau kartų, ko pasėkoje užsakymas vis tiek gali būti nesudarytas, o laikas išekvotas negrįžtamai. Be to pastebime, jog gali susidaryti uždaras ciklas, kurio metu vis koreguojamas užsakymas ir tikslinami dokumentai. Ciklo egzistavimas procese taip pat gali sudaryti didžiulius laiko nuostolius. Dėl šios priežasties būtina spręsti automatinių derybų klausimus, kas iš esmės sumažintų užsakymo sudarymui reikalingo laiko nuostolius.

### ***Uždaviniai:***

- 1) Išanalizuoti derybų taisyklių suderinimo problemas verslo įmonių sąveikumo sprendimuose.
- 2) Ištirti egzistuojančius verslo taisyklių suderinimo sprendimus.
- 3) Nustatyti reikalavimus verslo įmonių derybų taisyklių suderinimui.
- 4) Sukurti verslo įmonių derybų taisyklių suderinimo metodiką.
- 5) Sukurti verslo įmonių derybų taisyklių suderinimo programinės įrangos prototipą.
- 6) Ištirti pasiūlytos metodikos veiksmingumą atliekant eksperimentus su realiais duomenimis.

## **2.6. Tyrimo planas**

- Susijusių mokslinių darbų analizė
- Verslo ir derybų taisyklių analizė
- Egzistuojančių verslo taisyklių suderinimo metodų analizė.
- Reikalavimų verslo įmonių derybų taisyklių suderinimui nustatymas
- Verslo įmonių derybų taisyklių suderinimo metodikos kūrimas
- Derybų taisyklių suderinimo prototipo kūrimas
- Sukurtos metodikos bei prototipo testavimas

## **2.7. Vartotojų analizė**

### **2.7.1. Vartotojų aibė, tipai ir savybės**

Kadangi šiame darbe bus sprendžiama automatinių derybų problema, minimaliai dalyvaujant žmonėms, kai dvejų ar daugiau partnerių iš anksto suformuluotos derybų taisyklės automatiškai suderinamos ir pritaikomos perduodamiems verslo dokumentams, tai ir vartotojus laikysime ne žmones, o verslo įmones (nepriklausomai nuo to, kiek žmonių ten dirba ar kokia jų pareiginė hierarchija).

Dauguma tokių įmonių (šiuo atveju vartotojų) tam, kad išliktų rinkoje, privalo bendradarbiauti vienos su kitomis (sąveikauti). Tai nepriklauso nuo jų tipo, dydžio ar kvalifikacijos (individuali (personalinė) įmonė, akcinė bendrovė, uždaroji akcinė bendrovė, žemės ūkio bendrovė, valstybinė ar kitokia įmonė).

Tokių įmonių tarpusavio sąveikumo ir taisyklių suderinimo sprendimuose lemiamą vaidmenį ir įtaka turi tik jų darbo pobūdis. Tai reiškia, kad priklausomai nuo to, įmonė gamina ir parduoda ar teikia paslaugas ir perka, priklausys ir derybų taisyklių turinys. Pavyzdžiui įmonės, kurios daugiau gamina produkciją/paslaugą nei perka, akcentuos apribojimus ar lengvatas tokioms įmonėms, kurios iš jų perka tą paslaugą/produktą ir atvirkščiai. Taip pat, rinkoje dominuoja ir abiejų darbo pobūdžių įmonės viename.

## 2.7.2. Vartotojų tikslai ir problemos

Vartotojai, sudarinėdami derybų taisykles, turi savų tikslų, tačiau dėl to susiduria ir su įvairiomis problemomis. Jas išnagrinėsime 1 lentelėje:

1 lentelė

Vartotojų problemos aprašymas	Vartotojai tikisi, kad....
Nepilnai ir netiksliai apibrėžiamos derybų taisyklės	Būtų sukurtos formaliai apibrėžtos derybų taisyklės
Iš anksto suformuluotos derybų taisyklės nėra pilnai automatiškai suderinamos	Iš anksto suformuluotos derybų taisyklės automatiškai susiderintų su kitomis
Iš anksto suformuluotos derybų taisyklės sunkiai pritaikomos perduodamiems verslo dokumentams	Derybų taisyklės būtų suderintos su perduodamais verslo dokumentais
Suderinant bei pritaikant suformuluotas derybų taisyklių papildomai turi dalyvauti žmogus	Suderinant derybų taisykles su kitų įmonių taisyklėmis nereikėtų žmogaus įsikišimo
Žmogus turi dalyvauti visame derybų taisyklių suderinimo procese	Atsiradus nesuderinamoms tarpusavyje derybų taisyklėms, sistema „išvestų“ pranešimą, taip suteikdama galimybę pačiam žmogui ją išspręsti

Atliekant verslo transakcijas per sąveikumo sprendimus patogų kai įmonės gali nustatyti ir naudoti formaliai apibrėžiamas derybų taisykles.

Šiame darbe bus ieškomas automatinių derybų problemos sprendimo būdas, kai dvejų ar daugiau partnerių iš anksto suformuluotos derybų taisyklės automatiškai suderinamos ir pritaikomos perduodamiems verslo dokumentams, nedalyvaujant žmonėms. Norint tai įgyvendinti bus nustatyti reikalavimai, naudojami įmonių derybų taisyklių suderinimui, bei sukurta nauja jų suderinimo metodika. Naujosios metodikos veiksmingumas bus tiriamas sukūrus programinės įrangos prototipą.

## 2.8. Esamų sprendimų analizė

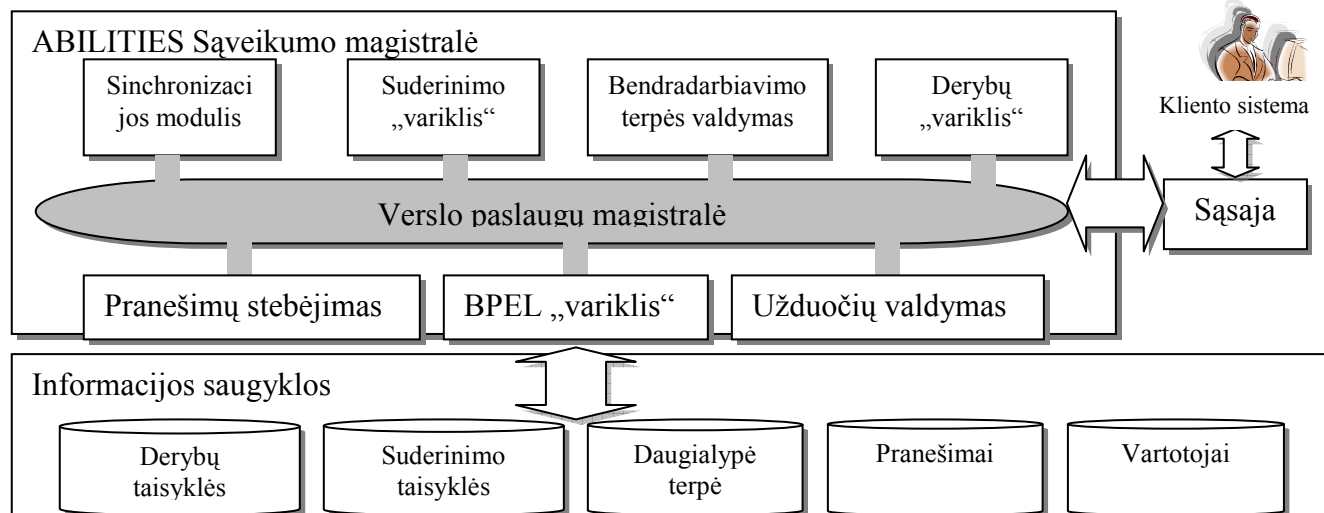
### 2.8.1. Sąveikumo sprendimai skirti SVV

Šiame skyrelyje apžvelgiamos įmonių sąveikumo sistemos. Tokios sistemos skirtos automatizuoti įmonių apsikeitimą verslo dokumentais ir sprendžia automatiško el. dokumentų standartų, kalbos, prekių pavadinimų ir t. t. suderinimo problemas. Dauguma sąveikumo sistemų niekaip neatsižvelgia į verslo partnerių taikomas derybų taisykles ir nesugeba automatiškai jų pritaikyti savo procesuose.

#### ABILITIES

Vienas iš įmonių sąveikumo sistemos variantų yra ABILITIES (*Application Bus for Interoperability In enlarged Europe SMEs*) sprendimas. Tai IT sprendimas, skirtas SVV įmonių tarpusavio bendravimo galimybėms pagerinti. Kūrėjai teigia, kad ABILITIES galima įsivaizduoti kaip magistralę, kuri jungia įmones ir per kurią galima siųsti dokumentus, pranešimus (klausimus-atsakymus), ir jais palaikyti prekybinius santykius: pradant derybomis dėl užsakymų ir baigiant atsiskaitymais. ABILITIES vienas iš tikslų – suderinti

dokumentų ir pranešimų formą, struktūrą bei kalbą. ABILITIES magistralė gali pakeisti dokumento redakciją pagal vartotojo nurodytas taisykles, suteikdama galimybes vartotojui siųsti ir gauti dokumentus jam priimtina forma [3]. Principinėje ABILITIES sąveikumo magistralės schemoje (žr. 2.2 pav.) pateikta jos struktūra bei esminės sudedamosios dalys.



2.2 pav. ABILITIES sąveikumo magistralės principinė schema [4]

ABILITIES verslo paslaugų magistralė (angl. *Enterprise Service Bus (ESB)*) apima visus pagrindinius, įgyvendinimo terpės, bruožus:

- Nukreipimą bei transformavimą (angl. *Routing and Transformation*)
- Užklausų paskirstymo „variklius“ (angl. *Distributed Query Engine*) pagrįstus SQL bei OWL (angl. *Web Ontology Language*).
- Sąsajas tarp įprastų paraiškų [3].

Jais siekiama palaikyti vieningą verslo dokumentų sąveikumo pagrindą (formatą).

ABILITIES projektas suformuluoja verslo partnerių derybų taisyklių automatinio suderinimo problemą ir pateikia vieną iš galimų tokio metodo realizaciją. Realiai sukurtas prototipas, galintis suderinti keletą tipų verslo taisykles taikomas vienam verslo dokumentų tipui. Galima teigti, jog metodas taip ir liko projektavimo ir pirmojo prototipo stadijoje, ir jo panaudoti realiose įmonių sąveikumo sistemose kol kas negalima.

## COIN

COIN (*Enterprise Collaboration and Interoperability*) – projektas skirtas asmenims (organizacijoms) norinčioms bendradarbiauti, atlikti konkrečias užduotis ar plėtoti tam tikrą veiklą, naudojantis skirtingų organizacijų tarpusavio ryšiu. COIN bendruomenės nariai skirstomi į tris skirtingas organizacijų topologijas: „mokslo“, „technologijų ir pramonės“ bei „galutinio vartotojo“. Taip galima nustatyti glaudesnius ryšius bei konkretesnius bendradarbiavimo standartus [5a].

### ***Įmonių bendradarbiavimas COIN sprendime***

Įmonių bendradarbiavimo ryšių analizė, kūrimas bei plėtojimas yra vienas iš pagrindinių COIN tikslų. Šiame sprendime teigiama, kad bendradarbiavimas gali būti klasifikuojamas. T.y. gali būti unikalus, jei bendradarbiaujančios organizacijos, ketina realizuoti tik vieną produktą arba tas bendradarbiavimas pagrįstas konkrečiu susitarimu (ar prašymu). Gali būti ribotas, jeigu ketina įgyvendinti nustatytą eilę sudėtingų produktų/paslaugų. Gali būti ilgalaikis bendradarbiavimas, kai įmonės veikla priklauso nuo kitos įmonės. Toks poreikis bendradarbiauti sparčiai formuoja bendradarbiavimo tinklus [5b]. COIN pabandė realizuoti tokį tinklą (2 lentelė) atskleidama įmonių bendradarbiavimą bei gyvavimo ciklą jame, tačiau šis sprendimas kol kas dar nėra užbaigtas.

**2 lentelė [5b]**

Pasiruošimo etapas	Formavimo etapas	Valdymo etapas	Išformavimo etapas
Registracija ir profiliavimas	Verslo galimybių ieškojimas bei kūrimas	Įmonių bendradarbiavimo valdymas	Klientų aptarnavimas
	Verslo galimybių charakterizavimas	Veiklos valdymas	
	Nauda		Įmonių įvertinimas
	Partnerių paieška ir pasirinkimas	Produktų/paslaugų valdymas	
Centralizuotų profilių bei kompetencijos paslaugos			
Bendravimo (ryšių) paslaugos			

Pasiruošimo etape, tarpusavyje galimai bendradarbiaujančios įmonės, registruojasi sistemoje, nurodo tam tikras savo charakteristikas (pvz. administracinius, procesų, paslaugų, produktų, efektyvumo ir kitus duomenis). Pliusas yra tas, kad kitos įmonės (jau užsiregistravusios) iškart gali matyti jau esamų įmonių (ar naujai įstojusių) tam tikrų duomenų pasikeitimus ar atsiradimus.

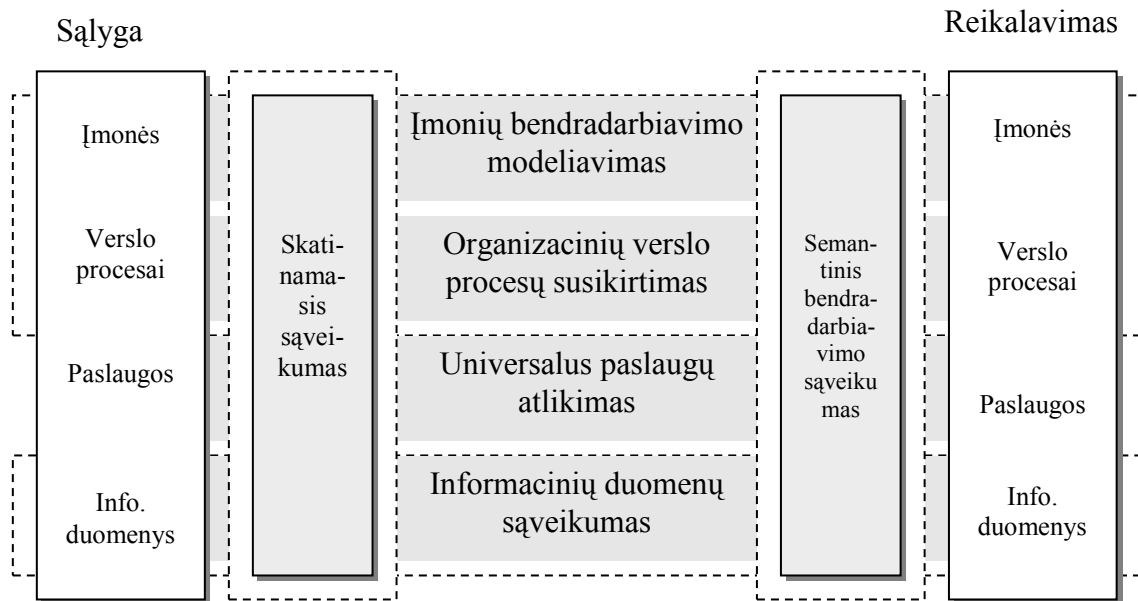
Formavimo etape ieškoma verslo plėtojimo galimybių. Analizuojamos kitos įmonės, stebimas jų efektyvumas, lojalumas ir kiti rodikliai, kurie svarbūs norint sudaryti dviem ar daugiau įmonių palankias bendradarbiavimo sutartis.

Valdymo etape dominuoja tokie veiksniai, kurie nusako bendradarbiaujančių įmonių verslo dokumentų standartus, jų derinimą, katalogu apie paslaugas ar prekes sudarymas, jų valdymas ir panašiai. Apibrėžiamas konkrečių įmonių konkretus bendradarbiavimas.

Išformavimo etape renkami ir kaupiami atsiliepimai iš visų bendradarbiavusių įmonių bei galutinių vartotojų. Taip sukaupiama duomenų bazė, kuria galima pasinaudoti sudarinėjant naujas bendradarbiavimo sutartis.

### ***Įmonių sąveikumas COIN sprendime***

COIN sprendime įmonių sąveikumo funkcionalumas bandomas realizuoti taikant įvairius IT sprendimus. Šie sprendimai padeda išspręsti daugumą spragų, kurios trukdo vykdyti efektyvų bendradarbiavimą. COIN, IT paslaugų sistema, įmonių sąveikumą tarsi padalina į penkias esmines dedamąsias (žr. 2.3 pav.). Šis metodas paimtas iš ATHENA sąveikos sistemos (angl. *ATHENA Interoperability Framework (AIF)*) [6].



2.3 pav. COIN sąveikumo pavyzdinis modelis [6]

Skatinamasis sąveikumas – COIN modelio transformacijos teikia tam tikrus pokyčius (atliekamos kompiuteriais), kurie naudojami siekiant suderinti skirtingų organizacijų standartus, integravimo ar įgyvendinimo užduotis, verslo programas bei sistemas.

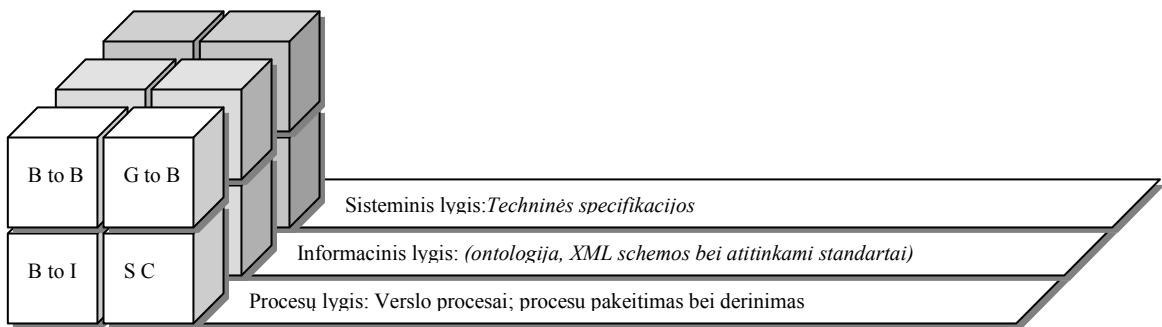
Semantinis bendradarbiavimo sąveikumas - *Athos Ontology Service* teikia funkcinės ontologijos valdymą. Ontologija yra išankstinė semantikos sąlyga, pagrįsta tarpininkavimu bei verslo dokumentų suderinimu. *Astar Semantic Annotation Service* užtikrina, kad skaitmeniniai duomenys būtų aprašyti pagal bendrą standartą, konkrečią ontologiją. *Argos Semantic Transformation Rules Service* nusako funkcinų transformacijų valdymo taisykles. *Ares Semantic Reconciliation Service* yra galutinis žingsnis užtikrinantis verslo dokumentų, egzistuojančių tarp skirtingų informacinių sistemų, suderinimą.

## GENESIS

GENESIS (*Enterprise application interoperability via Internet-Integration for SMEs, Governmental Organisations and Intermediaries in the new European Union*) – projektas, skirtas sukurti prototipinę sistemą, kuri leistų SVV įmonėms atlikti verslo operacijas interneto pagalba. Tai įgyvendinti siekiama apjungus bendradarbiaujančių įmonių programines bei komunikacines sistemas [7].

Įmonė parengia tam tikrus dokumentus. Juos, naudojantis savo esama ERP sistema arba interneto naršykle (jei neturi savo konkrečios ERP sistemos), persiunčia kitai įmonei. Šie dokumentai iš vienos įmonės į kitą keliauja per GENESIS serverį, kuriame jie yra saugomi, bei galutinį vartotoją pasiekia užtikrinant jų konfidencialumą. Be to, dokumentai, kurie atkeliauja pas vartotoją, GENESIS serveryje automatiškai transformuojami ir pritaikomi prie tos įmonės ERP sistemos (įgyvendinimui naudojama UBL, BPMN, BPEL, CCTS ir kita).

GENESIS galimybes galima atvaizduoti kaip kubą (žr. 2.4 pav.)



2.4 pav. GENESIS galimybių kubas [8]

Kaip matome iš 2.4 paveikslėlio GENESIS dėmesys nukreiptas į tam tikrus bendradarbiavimo ryšius:

- *Business to Business (B to B)* – tai yra ryšys vykstantis tarp įmonių (pvz. Užsakymai, pirkimai, sąskaitos faktūros, mokėjimai ir panašiai).
- *Business to Government (B to G)* - ir vyriausybės (pvz. PVM deklaracijos, mokesčiai, ir panašiai).
- *Business to Intermediaries (B to I)* - tai yra ryšys vykstantis tarp įmonių bei verslo partnerių, įskaitant elektroninių paslaugų tiekėjus (pvz. Sąskaitų pervedimai, draudimo organizacijos ir panašiai).
- *Supportive Companies (S C)* – Kiti ryšiai vykstantys tarp įvairių paramos operacijų (pvz. finansinių duomenų keitimosi tarp SVV, buhalteriniai dokumentai ir panašiai).



GENESIS bando spręsti problemas egzistuojančias keliose srityse (bendradarbiaujant skirtingoms įmonėms), kurios yra tarpiai tarpusavyje susijusios. Šis sprendimas padidina bendradarbiavimo naudą bei efektyvumą, kadangi padeda taupyti tiek pinigus tiek laiką. Taip pat palengvina bendradarbiavimą su tarptautiniais partneriais.

Toliau pateikiamas sąrašas pagrindinių sprendimų, kurie realizuojami GENESIS projekte:

- Kadangi BPMN yra tarptautinis standartas, jis buvo pasirinktas, kaip modeliavimo kalba, kuri apibūdina bendradarbiavimo procesų srautus bei sąveikas tarp jų.
- Atsižvelgus į tai, kad *Business to Business* modeliai yra gerai dokumentuoti UBL 2.0 kalba, ji buvo pasirinkta viešųjų procesų suderinimui.
- Modeliai iš kitų sektorių *Business to Government* ar *Business to Intermediaries* yra pagrįsti galutinio vartotojo standartais. (procesai prisitaiko prie konkrečiai realizuotų standartų, kuriuos reglamentuoja galutinis vartotojas (pvz. vyriausybė) [8].

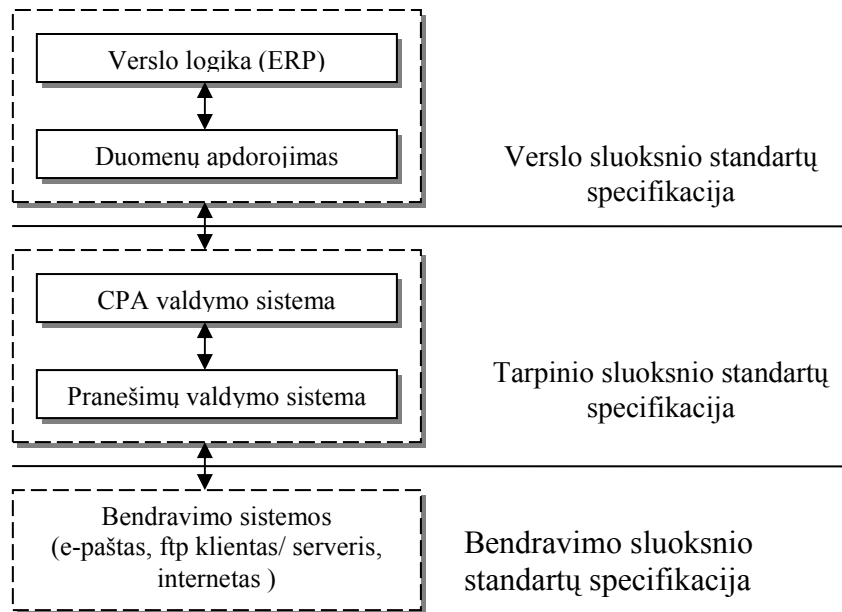
## **eBIZ**

eBIZ yra Europos organizacijų bendradarbiavimo projektas, kuriuo siekiama padidinti verslo procesų efektyvumą [9]. Pagrindiniai tikslai egzistuojantys eBIZ sprendime yra:

- Esamų, elektroninio verslo sąveikos sistemų, sprendimų analizė
- Techninės architektūros, naudojamos elektroniniame versle, plėtimas
- Skirtingų sąveikumo sistemų, kurias naudoja organizacijos, integravimas bei testavimas
- Duomenų mainai tarp bendradarbiaujančių organizacijų

eBIZ projektas pagrįstas tuo, kad bendradarbiaudamos skirtingos organizacijos duomenimis, bei įvairiais tarpusavio dokumentais, turėtų keistis elektroniniu būdu (naudojant internetą). Tačiau dėl organizacijose naudojamų skirtingų sistemų, toks bendradarbiavimas praktiškai neįmanomas. eBIZ projekto autoriai, išnagrinėję šią problemą, priėjo išvadą, jog reikia sukurti tam tikro vieningo standarto sistemą, o organizacijose egzistuojančias sistemas (ERP, RMS ir kita) suintegruoti kaip papildomus modelius. Tokiu būdu suvienodinus duomenų formatus, taptų įmanomas ir efektyvus duomenų siuntinėjimas elektroniniu būdu.

Taigi pirmiausia reikia išsiaiškinti, kokios reikia programinės įrangos norint įgyvendinti elektroninio verslo operacijas tarp dviejų organizacijų. Paveiksle (žr.2.5 pav.) parodytos pagrindinės skirtingų sistemų programinės įrangos sudedamosios dalys. Jos suskirstytos į tris sluoksnius, atitinkančius skirtingus standartus.



2.5 pav. Skirtingų sistemų programinės įrangos sudedamosios dalys [9]

Verslo sluoksnio standartų specifikacijoje sprendžiamos problemos susijusios su organizacijose esančiomis sistemomis. Dažniausiai tai yra tam tikros ERP sistemos. Jos būna specifinės, pritaikytos konkrečiai įmonei, todėl jų pakeisti iš esmės yra labai sudėtinga. Šiai problemai spręsti kuriamos įvairios sąsajos, kurios pakeičia duomenų standartus taip leisdamas skirtingų sistemų bendravimą.

Tarpinis sluoksnis, tai įrangos komponentų ir/arba sistemų, kurios pagrinde atlieka saugumo funkciją. Nebūtina, kad tokios tarpinės įrangos sistemos būtų įdiegtos, kadangi bet kokia informacija (įskaitant verslo sandorių rūšis) gali būti perduota tiesiogiai elektroniniu paštu ar internetu, tačiau šiuo atveju neužtikrinama tinkama apsauga.

Bendravimo sluoksnis tiesiog nusako, kokiomis priemonėmis yra realizuotas bendradarbiavimas.

### Kiti Sąveikumo sprendimai skirti SVV

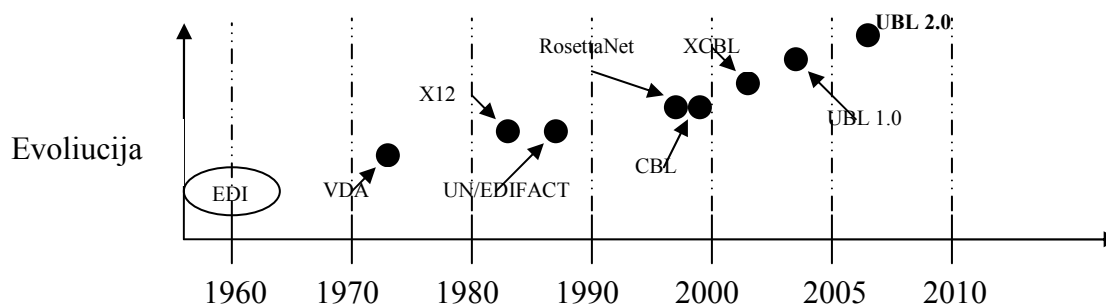


2.6 pav. Kiti sąveikumo sprendimai skirti SVV

Sąveikumo sprendimų skirtų SVV yra gana daug (žr. 2.6 pav.), tačiau dauguma jų nesprenžia įmonių verslo taisyklių automatinio suderinimo klausimų, jie nenagrinėja derybų taisyklių bei jų suderinimo klausimų. Kita problema - organizacijose naudojamų dokumentų standartai. Jų yra gana daug ir dažniausiai jie skiriasi, o tai apsunkina efektyvų jų tarpusavio suderinimą.

### 2.8.2. Sąveikumo sprendimuose naudojami dokumentų standartai

Elektroninių verslo dokumentų naudojimas daug efektyvesnis už paprastus popierinius. Taip yra todėl, kad įvairios derybos vedamos dažniausiai ne žodžiu, o verslo dokumentai siuntinėjami ne paštu. Tam realizuoti sukurta daugybė sistemų. Tačiau norint efektyviai naudotis šiomis sistemomis visi dokumentai turi būti tam tikro formato, standarto, kurių įvairovė šiomis dienomis taip pat labai didelė. Keletas iš jų pateikta verslo dokumentų standartų evoliucinėje diagramoje (žr. 2.7 pav.).



2.7 pav. Verslo dokumentų standartų evoliucija [4]

#### EDI

EDI (*Electronic Data Interchange*) skirtas "kompiuteris-kompiuteris" informacijos struktūrizuotam keitimuisi. Pagal nustatytus pranešimų standartus informacija, elektroniniu būdu, iš vieno kompiuterio keliauja į kitą naudojant minimalų žmogaus įsikišimą. EDI dokumentuose pateikti tie patys duomenys, kurie paprastai būna ir popieriniuose dokumentuose. Tačiau, EDI neapsiriboja vien verslo dokumentais, susijusiais su prekyba, bet apima ir tokias sritis, kaip įvairių standartų, nurodymų, reikalavimų ar kitokių apribojimų/lengvatų reglamentavimu bei paskelbimu.

EDI suprojektuotas taip, kad būtų nepriklausomas nuo ryšio ar programinės įrangos technologijų. Tai reiškia, kad verslo dokumentai gali būti siuntinėjami naudojant įvairiausias metodikas (ftp, e-paštas, http, AS1, AS2), svarbu kad siuntėjas ir priėmėjas sutiktų.

Yra dvi pagrindinės EDI standarto komponentės: ANSI ASC X12 (X12), kuris apibrėžia duomenų komponentių charakteristikas (standartus) bei VAN, kuris užtikrina saugų duomenų perdavimą.

VAN (*Value Added Network*) yra vienas iš populiariausių būdų perduoti elektroninius duomenis, dokumentus. Jo efektyvumas pasireiškia tuo, kad jis geba atlikti duomenų transformacijas tarp formatų (iš EDI į XML; iš EDI į EDI ir pan.). Be to VAN ne tik gauna, saugo ir perduoda pranešimus, tačiau taip pat suteikia informacijos apie auditą (iš dalies atlieka automatinio klaidų aptikimo ir taisymo ar keitimo procesą keliaujant duomenimis tarp ryšio protokolų) [4].

VAN, kaip ir įprastas internetas, naudoja savo pranešimų protokolus, siekiant užtikrinti, kad EDI dokumentai būtų perduodami saugiai. Populiariausi protokolai yra FTPS (*File Transfer Protocol Secure*), HTTPS (*Hyper Text Transport Protocol Secure*).

ANSI ASC X12 (dar žinomas tiesiog kaip X12) reglamentuoja įvairius EDI standartus, charakteristikas, jų kūrimą bei priežiūrą. X12 1979 metais buvo akredituota ANSI (*American National Standards Institute*) standartų komiteto [4].

Taigi EDI skirtas struktūrizuotam duomenų perdavimui, tarp organizacijų, elektroninėmis priemonėmis. Jis naudojamas elektroninių dokumentų perdavimui iš vienos kompiuterinės sistemos į kitą, t.y. iš vienos prekybos partnerės kitai prekybos partnerei. Tai yra daugiau nei paprastas e-mail, pavyzdžiui, organizacijos galėtų pakeisti važtaraščius ir net čekius atitinkamais EDI pranešimais. Jis taip pat susijęs su įvairiomis standartų šeimomis, įskaitant X12.

## **XML Dokumentai**

XML (angl. *Extensible Markup Language*) – išplečiama žymių kalba. Jos atsiradimą lėmė tai, jog iki 1960 metų egzistavusi SGML kalba buvo pernelyg sudėtinga, norint laisvai keisti informacija informacinėmis sistemų pagalba. Tiesa pirmaisiais metais XML taip pat buvo gana sudėtinga ir padrika kalba, kol galiausiai po kelių dešimtmečių, 1998 metais W3C konsorciumo buvo paskelbta jos naudojimo rekomendacija. Nuo to laiko XML naudojimo populiarumas tik augo [10].

XML dokumentas skiriasi nuo kitų nestandartizuotų formatų tuo, jog tai ne dvejetainiai failai, perskaitomi ir išsaugomi specialiomis programomis, o tekstinis dokumentas su standartiniais *Unicode* standarto simboliais. Tai leidžia XML dokumentus sudarinėti, bei keisti jų turinį, standartiniais tekstiniais redaktorais

XML duomenų aprašymo kalba paremta „medžio“ tipo struktūra. Šis tipas plačiai naudojamas programinėse priemonėse dėl savo galimybės aprašinėti daugumą (ne visus) duomenų tipų. Pagrindiniai elementai gali būti tiek su turiniu (t.y. kitais elementais) (angl. *container element*), tiek be jų (angl. *node element*). Kiekvienas elementas savo ruožtu gali turėti viena ar daugiau atributų [10].

Atsiradus XML užklausų kalboms, tokioms kaip XML-QL, XPath, XQuery, XQL, Quilt, ir kt., tapo įmanoma modeliuoti duomenų srautus nedetalizuojant juos realizuojančios DBVS. Dėl šios priežasties XML tapo ypatingai naudingas atviroms, susisiekančioms sistemoms.

### **ebXML**

*Electronic Business using eXtensible Markup Language* paprastai žinomas kaip elektroninis verslas, naudojant XML, arba tiesiog ebXML. Jo tikslas yra užtikrinti prieinamą, XML infrastruktūros pagrindu paremtą, saugią sąveiką visiems prekybos partneriams. Pagrindinė dedamoji čia yra XML, kuri leidžia naudoti pasaulinę elektroninio verslo informaciją.

ebXML architektūra yra standartas įgyvendinantis teorinę verslo dalį. T.y. ebXML standartai, priimti ISO bei OASIS techninio komiteto, siekia aprūpinti oficialias, XML pagrindu įgalintas sistemas. Jo architektūra sutelkta ties sąvokomis ir metodologija, leidžiančia geriau bei greičiau įgyvendinti elektroninio verslo sprendimus.

ebXML registras yra panašus į UDDI, kuriame jis leidžia organizacijoms surasti viena kitą, apibrėžti prekybos bendradarbiavimo sutartis, ir apsikeisti XML pranešimais palaikant verslo operacijas. Tikslas yra sudaryti sąlygas, kad visos šios veiklos būtų atliekamos automatiškai be žmogaus įsikišimo, per internetą. ebXML pranešimų specifikacija yra grindžiama SOAP su tam tikrais papildymais, kuri nenaudoja WSDL. ebXML įtraukia saugumą, garantuotus pranešimus ir atitinkančią verslo procesų sąveikos specifikaciją [11].

Apibendrinant galima teigti, jog ebXML yra visuotinis standartas elektroniniam verslui vykdyti. Jis įgalina visas organizacijas, esančias visame pasaulyje, bet kuriuo laiku vykdyti verslą su visomis kitomis organizacijomis naudojantis interneto pagalba. Tiesa jis pritaikytas labiau smulkaus bei vidutinio dydžio įmonėms (SVV).

### **UBL**

Vienas iš naujausių verslo dokumentų standartų yra UBL 2.0. UBL (*Universal Business Language*) apibrėžia standartinius XML verslo dokumentus bei jų specifiką ar paskirtį (pavyzdžiui nusako verslo dokumentų kilmę, vykdymo reikalavimus, apibrėžia sąskaitas faktūras, pirkimo užsakymus ir kita). UBL skirtas apjungti į vieną visumą esamas verslo, teisės, audito ar įrašų valdymo praktikas.

UBL yra standartinė elektroninė XML verslo dokumentų, tokių kaip pirkimo užsakymai ar sąskaitos faktūros, biblioteka. Ją sukūrė OASIS techninis komitetas. Pagrindiniai jų tikslai buvo:

- Tarpusavyje sujungti esamus verslo, teisės, audito ar kitokių dokumentų tvarkymo standartus (praktikas).

- Panaikinti esamus popierinius dokumentus bei sudaryti pagrindus elektroninei, mažų bei vidutinių įmonių, komercijai [12].

Naujausia UBL 2.0 specifikacija patvirtinta bei paleista viešam naudojimui 2006 m. Pabrėžiama, jog ji prieinama visiems, be jokių autorinių mokesčių. Iš viso yra 31 dokumentų standartų apibrėžiančių įvairius verslo poreikius įskaitant užsakymus, pristatymus, sąskaitas faktūras ar apmokėjimus.

Apibendrinant UBL galima teigti, jog šis standartas įmonėse gali būti naudojamas tuomet, kai joms reikia:

- Naudoti standartizuotus elektroninius verslo dokumentų mainus;
- Automatizuoti sąskaitas, užsakymus, apmokėjimus ir t.t.;
- Panaikinti popierinius dokumentus, kur nepageidaujamas pakartotinis duomenų įvedimas ir nereikalingų klaidų padarymas;
- Lengvai pritaikyti elektroninio verslo sprendimus savo aplinkoje;
- Sumažinti vystymosi metų laikotarpį;

### EDI, UBL bei ebXML tarpusavio palyginimas

EDI, UBL bei ebXML yra pagrindiniai standartai nusakantys verslo dokumentus. Dėl savo paskirties tarpusavyje jie turi daug panašumų, tačiau egzistuoja ir kiekvieno jų trūkumai bei privalumai. Pirmiausia apibrėšime visų jų struktūrą (žr.3 lentelę):

**3 lentelė**

	EDI	ebXML	UBL	
Kontekstiniai pranešimai	MIGs	Kontekstinė metodologija	UBL kontekstiniai metodai	} Apkrova (angl. <i>payload</i> )
Standartų	EDIFACT, X12	CORE komponentai	UBL biblioteka	
Verslo sutartys	Ad hoc TPA	CPP/CPA	CPP/CPA	} Infrastruktūra
Verslo procesai	CASE įrankiai	BPSS	BPSS	
Perdavimai	VAN	Pranešimų paslaugos	Pranešimų paslaugos	

Iš 3 lentelėje pateiktų duomenų galima spręsti, kad visi lyginami verslo dokumentų standartai šiek tiek skiriasi vieni nuo kitų. Kitoje lentelėje (žr. 4 lentelę) konkrečiau panagrinėsime jų skirtumus bei panašumus. Kiekvienas kriterijus įvertinamas „+“ ar „-“, ženklais, kur:

„++“ geriausiai išpildantis; „+“ gerai išpildantis,  
 „+/-“ apylygiai; „-“ blogai išpildantis,  
 „--“ blogiausiai išpildantis.

Tarkim jei EDI atitinka ++ o UBL +/-, tai reiškia kad EDI truputėlį lyginamuoju aspektu geresnis. Jeigu ženklai sutampa tai reiškia, kad tarpusavyje jie nesiskiria, o ženklų reikšmė rodo kaip išpildomas kriterijus, pagal kurį lyginama. Be to lentelės duomenys nėra oficialūs. Rezultatai tiesiog atspindi padarytos analizės rezultatus.

**4 lentelė**

	Pradinė įmonės adaptacija	Žinomumas rinkoje	Paprastumas darbuotojam	Integracija su kitom sistemom	Saugumas
EDI	+	+/-	+	+	+
ebXML	+/-	+/-	+/-	+/-	++
UBL	+/-	+	+	+/-	+/-

### **Kiti sąveikumo sprendimuose naudojami dokumentų standartai**

**RosettaNET** – yra nekomercinis konsorciumas, siekiantis nustatyti vieningus standartus egzistuojančius elektroninio verslo informacijos dalinimosi procese. Šitie standartai formuoja bendrą elektroninio verslo kalbą, rikiuodami procesus tarp aprūpinimo grandinės partnerių globaliniu pagrindu.

RosettaNet standartas pagrįstas XML. Jis apibrėžia informacijos pranešimus, t.y. verslo procesų sąsajas bei įgyvendinimo struktūros sąveikas tarp kompanijų. Šis standartas daugiausia skirtas tiekimo grandinės, gamybos, produktų ir medžiagų duomenų bei paslaugų taikymo sritims. Automatizuotas RosettaNet optimizavimo standartas (RAE) naudoja *Office Open XML* dokumento standartus.

RosettaNet kilęs iš JAV ir ten plačiai vartojamas, tačiau taip pat vertinamas ir Azijos šalyse. Atsižvelgiant į platų panaudojimą šio standarto naudojimas vis auga ir Europos šalyse.

**SPML** (*Service Provisioning Markup Language*) – verslo dokumentų standartas, pagrįstas XML struktūra, naudojamas išteklių ir teikiamų paslaugų informacijos keitimuisi tarp bendradarbiaujančių organizacijų. SPML yra atviras, OASIS sukurtas, standartas nusakantis teikiamų paslaugų integraciją bei sąveiką. Pirmoji šio standarto versija pasirodė 2003 metais. Po trijų metų buvo išleista antroji jo versija [13].

Pagrindinis SPML tikslas yra leisti organizacijoms greitai bei saugiai susikurti interneto paslaugų ir programų sąsajas su įmonės nuomojama platforma (pavyzdžiui interneto portalais, serverių programomis ar paslaugų centrais, kurie reikalingi organizacijų

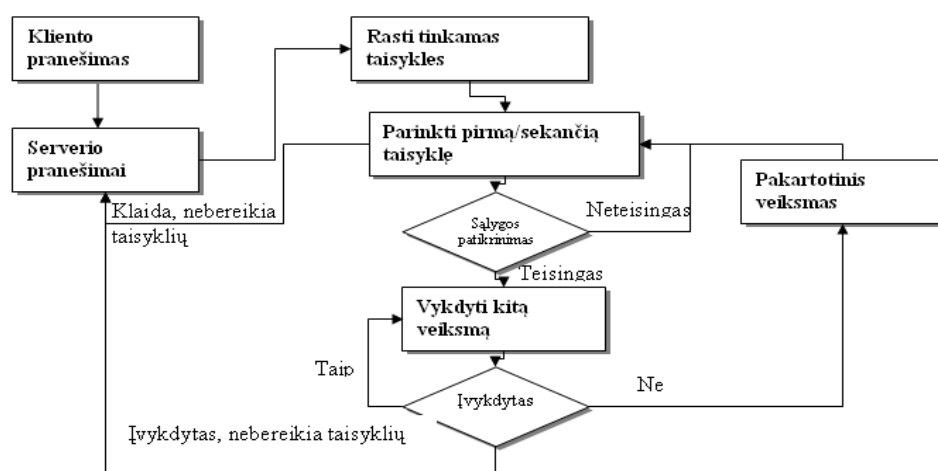
bendradarbiavimui). Tai suteikia automatizuotą vartotojo ir sistemos sąveiką, kurios pagalba vykdomas elektroninis verslas visoje IT infrastruktūroje. Tokiu atveju klientai nėra priklausomi nuo patentuotų sprendimų.

Taigi sąveikumo sprendimuose naudojamų verslo dokumentų standartų yra gana daug. Dažniausiai jų sąveika paremta XML pagrindu, kuris nusako naudojamus panešimų protokolus, kuriuos palaiko, ar saugumo standartus, kurie jame realizuoti. Tačiau vien verslo dokumentų standartų neužtenka norint sukurti efektyvią organizacijų sąveikumo sistemą. Taip yra todėl, kad skirtingos įmonės dažniausiai naudoja skirtingus standartus, kuriuos reikia kažkaip suderinti tarpusavyje. Tam reikia efektyvaus derybų taisyklių suderinimo sprendimo.

### 2.8.3. Derybų taisyklių suderinimo sprendimai

#### Rule Engine

Rule Engine yra verslo taisyklių interpretatorius realaus laiko sistemose [14]. Žemiau pateiktoje diagramoje (žr. 2.8 pav.) pavaizduota srautų diagrama, kurios atliekamus veiksmus vykdo taisyklių „variklis“. Pirmiausia taisyklių „variklis“ parenka visas derybų taisykles, kurių veiksmų pavadinimai sutampa. Šios taisyklės yra prioritetingos, kadangi taisyklių duomenų bazėje jos yra išrikiuotos. Parinkus taisyklę patikrinama jos būklė. Jei sąlyga netenkinama, tada pereinama prie kitos taisyklės. Jeigu nebėra tinkamų taisyklių, tuomet tikrinimas baigiamas išdavus signalą, kad taisyklės kartojasi ir yra netinkamos. O jeigu sąlyga patenkinama, tuomet tam tikros komandos vykdo atėjusias taisykles viena po kitos taip formuodama jų derinimą. Kai visos taisyklės yra patikrinamos, atliekami tam tikri patikrinimai, reikalingi pakeitimai bei išduodamas suderinimo rezultatas. Tačiau šis interpretatorius skiria mažai dėmesio tarp pačių taisyklių esančių probleminių situacijų sprendimui.



2.8 pav. Taisyklių „variklio“ srautų diagrama [14]



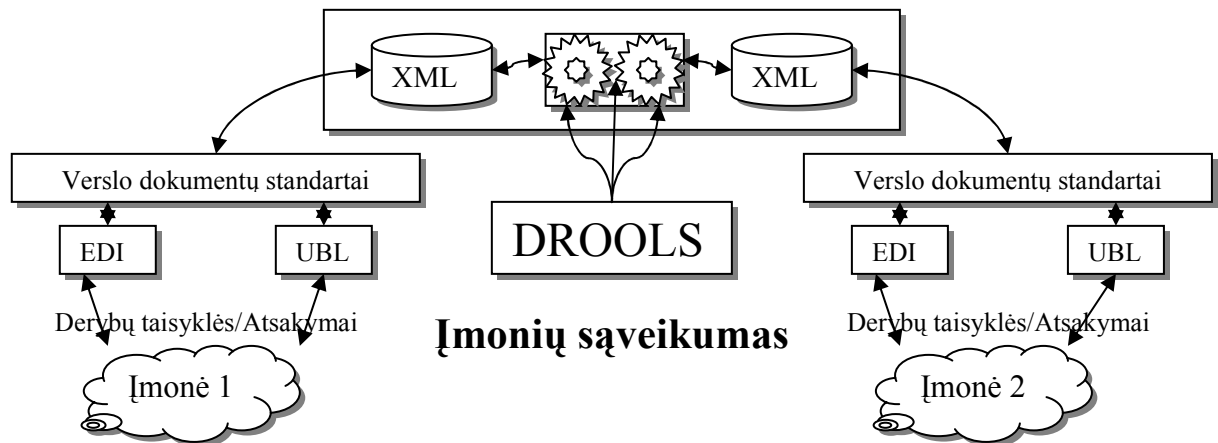
## DROOLS

DROOLS yra įrankis, realizuojantis verslo taisyklių rinkinį (angl. *Rules Engine Implementation*). Jis paremtas Charles Forgy's Rete algoritmu pritaikytu Java kalbai [15].

Iš esmės DROOLS veikia taip pat, kaip ir Forgy's originalus algoritmas, kadangi naudoja tas pačias sąvokas ir metodus. Tačiau jame yra įtraukta ir keletas naujų sąsajos sisteminių mazgų, kurie reikalingi integracijai su šia objektą orientuota kalba. Toks sąsajos pritaikymas leidžia verslo taisykles padaryti priimtinesnes (angl. *Naturals*) verslo klientams, kurie naudojami šiomis taisyklėmis.

DROOLS yra sukurtas taip, kad būtų galima vykdyti skirtingų kalbų apjungimus (angl. *Pluggeable*). Šiuo metu verslo taisyklės gali būti pateiktos tiek *Java*, tiek *Python* ar *Groovy* kalbomis. Taip pat DROOLS yra pakankamai lankstus įrankis suderinant probleminės srities semantiką su *Domain Specific Languages* (DSL) specifine kalba. DSL susideda iš XML elementų ir atributų, kurie reprezentuoja (pristato) probleminę sritį [14].

Pateiktame paveiksle (žr. 2.9 pav.) preliminariai parodyta DROOLS vieta sąveikumo sprendimuose.



2.9 pav. Drools vieta sąveikumo sprendime schema

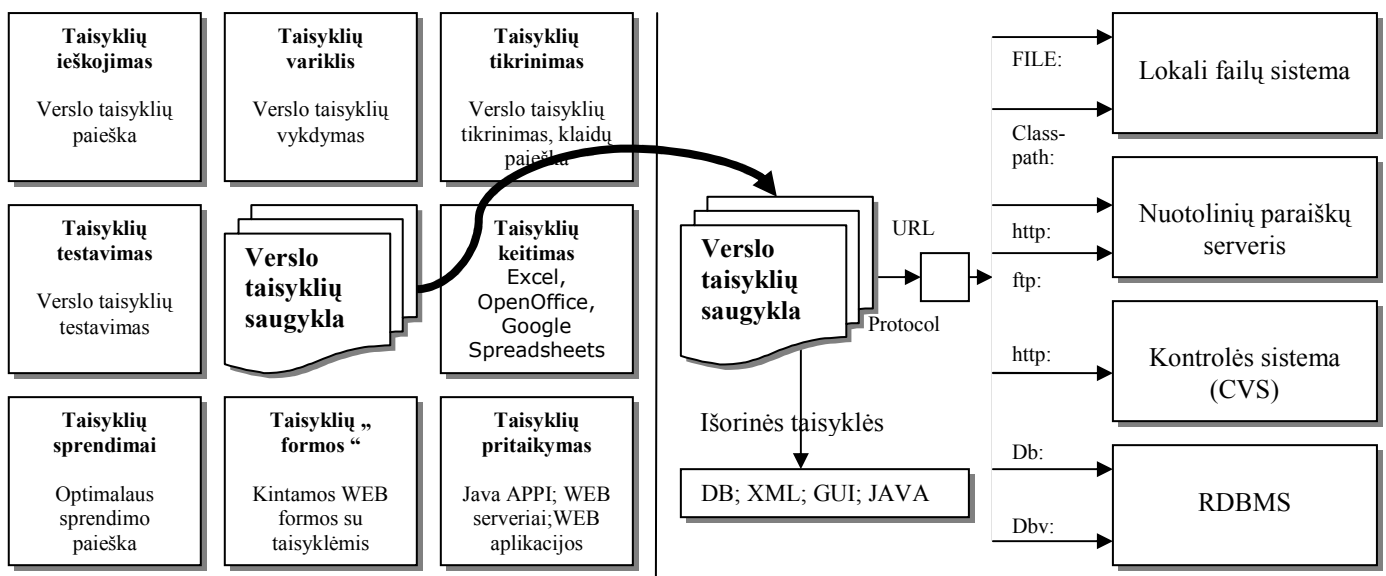
## OpenRules

OpenRules yra viso verslo taisyklių valdymo sistema (angl. *Business Rules Management System* (BRMS)), pagrįsta atviro kodo programine įranga. Ji naudoja tokias priemones, kaip "MS Excel", "Google skaičiuoklė" (angl. *Google Spreadsheets*) ar „Eclipse IDE“ [16].

OpenRules palengvina organizacijų darbą kuriant bei derinant skirtingas derybų taisykles. Jo veikimo principas (variklis) pagrįstas unikaliomis taisyklėmis, kurios yra labai efektyvios bei pakankamai patikimos. Be to visiškai suderintos su Java bei NET produktais.

Ši sistema buvo sukurta 2003 metais ir per metus tapo vienu iš populiariausių verslo taisyklių valdymo sistemų. Kiekvieną dieną OpenRules padeda klientams valdyti milijonus sandorių realioje pasaulinėje gamybos aplinkoje.

Paveikslėlyje žemiau (žr. 2.10 pav.) pateiktas OpenRules komponentės, bei centrinės dalies, verslo taisyklių saugyklos turinys.



2.10 pav. Loginis/fizinis OpenRules saugyklos organizavimas [16]

Loginėje OpenRules saugykloje laikomos įvairios verslo taisyklės. Kiekviename iš pavaizduotų blokų egzistuoja viena ar daugiau lentelių, kuriose vienaip ar kitaip apdorojamos skirtingose organizacijose egzistuojančios taisyklės. Jos tarpusavyje, panaudojus įvairius taisyklių derinimo „variklius“ ar tikrinimus, yra suderinamos, išanalizuojamos bei pratestuojamos ar nepadaryta klaidų. Atsakymai nuotoliniais serveriais vėl išsiunčiami į reikalingas organizacijas. Toks derybų taisyklių derinimas gana optimalus, tačiau kol kas ne visiškai efektyviai pritaikomas smulkaus ir vidutinio verslo įmonių poreikiams. Taip yra todėl, kad ši suderinimo sistema yra labai „masyvi“, o tai lemia didelę įsigijimo bei eksploatavimo kainą.

## **Kiti derybų taisyklių suderinimo sprendimai**

### **Open Lexicon**

OpenLexicon yra atviro kodo verslo taisyklių bei procesų valdymo sistema, grindžiama taikomųjų programų pagrindu. Ši sistema susideda iš dviejų komponentų: metaduomenų saugyklos ir verslo taisyklių variklio [17].

OpenLexicon turi internetinę formą, palengvinančią įmonių bendradarbiavimą naudojant verslo taisykles bei atliekant kitas verslo funkcijas. Be to ši sistema geba nuskaityti duomenis iš failų, atlikti informacijos paiešką bei saugoti duomenų bazės lentelėse. OpenLexicon taip pat palaiko interneto paslaugas.

Tačiau ši sistema nepasižymi efektyvumu, ypač kalbant apie taisyklių suderinimą, o be to, dėl savo „atviro kodo“ politikos, ji yra nepatikima.

### **WebSphere**

WebSphere integravimo sprendimas orientuotas į paprastą vartotoją, kuris neturi „gilių“, informacinių sistemų naudojimo, įgūdžių. Šis įrankis realizuotas IBM korporacijos remiantis *Rational Application Developer v7.0* programinės įrangos technologija. Jame realizuota keletas komponentų, tokių kaip „Verslo procesai“, „Verslo būsenų mašinos“ ar „Verslo taisyklės“. Pastarojo įrankio grafinės aplinkos pagalba, vartotojas intuityviai gali suprojektuoti bei plėtoti savo reglamentuotas verslo taisykles [18].

Bendrai paėmus, WebSphere verslo taisyklės naudojamos norint priimti sprendimus, kai tenkinama bent viena iš sąlygų:

- Reikia pakeisti tam tikrus duomenis ar rezultata realiu laiku;
- Sprendimas natūraliai įgauna sprendimų lentelės (angl. *Decision tables*) pavidalą;
- Sprendimas natūraliai įgauna tekstinę „Jei - Tai“ struktūrą.

### **SweetRules**

SweetRules yra unikali, galinga integruota priemone Web taisyklių ir ontologijų suderinimui. Ši sistema glaudžiai siejasi su RuleML (*Rule Markup/Modeling Language*) ar SWRL (*Semantic Web Rule Language*). Taip pat jis grindžiamas OWL, Jess, XSB, CommonRules, Jena, RDF bei XML standartais [19].

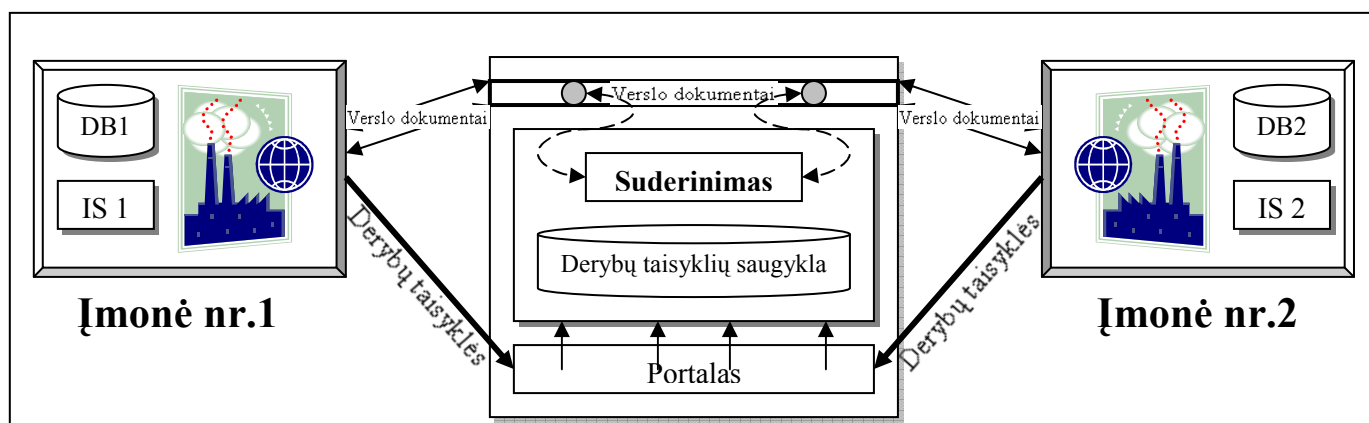
Taigi derybų taisyklių suderinimo sprendimų, literatūros šaltiniuose, galima rasti daug ir įvairių, tačiau jie nelabai atitinka šiuolaikinio verslo poreikių. Vieni jų atlieka netiksliai tas funkcijas, kurių reikia, kiti nesusieti su verslo dokumentais, dar kiti nesprenžia taisyklių nesuderinamumo klausimų ir panašiai. Esama situacija reikalauja naujų, efektyvesnių metodų ir sprendimų, kuriuos reikia sumodeliuoti bei realizuoti.

## 2.9. Siekiamas sprendimas

Šiuo darbu, išsamiai išanalizavus egzistuojančius verslo taisyklių suderinimo sprendimus, bei nustatius reikalavimus, kurie nusako kaip kuriamos ir derinamos šios taisyklės tarpusavyje, siekiama sukurti naują suderinimo metodą, padedantį maksimaliai automatizuoti verslo taisyklių suderinimą. Taip pat siekiama suvienodinti tokių taisyklių reikalavimus, standartizuoti juos, taip supaprastinant jų susiejimus.

Išanalizavus derybų taisyklių suderinimo sprendimus nustatyta, jog jų pasirinkimas yra gana didelis. Iš esmės visų jų paskirtis yra vienoda, tačiau ir vieni ir kiti turi nežymių savų plusų bei trūkumų. Įvertinus visus išnagrinėtus sąveikumo sprendimus, juose naudojamus verslo dokumentus bei jų suderinimo technologijas, galima sukurti pakankamai efektyvų derybų taisyklių suderinimo metodą, kurį būtų galima panaudoti įmonių sąveikumo sprendimuose automatinėms deryboms atlikti, ir kuri tenkintų dabartinio verslo bei organizacijų poreikius.

Pateiktame paveiksle (žr. 2.11 pav.) pateikta principinė schema, vaizduojanti vieną iš galimų variantų sprendžiant SVV tarpusavio sąveikumą naudojant verslo taisykles.



2.11 pav. Principinė SVV sąveikumo schema

Suinteresuotos įmonės siunčia savo verslo dokumentus į tam tikrą sistemą. Taip daroma todėl, kad būtų galima suvienodinti šių dokumentų standartus, kadangi kiekviena įmonė, savo viduje, gali naudoti skirtingas informacines sistemas, kas užkerta kelią tiesioginiam dokumentų siuntimui. Sistema savo ruožtu pritaiko siunčiamų verslo dokumentų standartus kitom įmonėm ir įvykdo persiuntimą.

Jeigu įmonės turi savo reglamentuotas derybų taisykles (pvz.: ieškomas verslo partneris turi atitikti ISO standartus, teikti nuolaidas perkant urchu, vystyti verslą ne mažiau kaip 5 metus, jei preke pristatinėjama daugiau kaip 5 dienas turi būti taikomos nuolaidos ir panašiai) jas per internetinį portalą ar kitą sąsają iš anksto suveda į šią sistemą. Sistema

automatiškai jas suderina, palygina tarpusavyje bei išduoda kiekvienai įmonei tokius rezultatus, kurie atitinka įvestas taisykles.

Siekiamam sprendimui įgyvendinti planuojama panaudoti derybų taisyklių suderinimo technologiją „DROOLS“. Toks pasirinkimas daromas todėl, kad šis įrankis sugeba automatiškai aptikti tam tikrus derybų taisyklių nesuderinamumus bei tai daro greičiausiai nei kiti egzistuojantys suderinimo sprendimai. Šis spartumas įgyvendinamas naudojant Rete (lot. *tinklas*) algoritmą. Tai paieškos pirmyn algoritmas, kurio esminė dedamoji - iš faktų ir jų derinių sukurtas grafas – medis.

## **2.10. Analizės išvados**

1. Atlikus išsamią esamų, nagrinėjamos problemos, sprendimų analizę nustatyta, kad egzistuoja keletas efektyvių sprendimų, kurių esminė dedamoji yra sąveikumo sistemų panaudojimas, tačiau jų pritaikymas praktikoje kol kas nėra pilnai įgyvendintas, ir jų funkcionalumas neatitinka keliamų reikalavimų. Viena to priežasčių – neefektyvus verslo taisyklių naudojimas arba jų iš viso nebuvimas.

2. Literatūros šaltinių analizė parodė, kad verslo taisyklių suderinimas SVV įmonių sąveikumo sprendimuose yra aktuali problema. Sunkumai kyla sprendimų priėmimo, pavyzdžiui užsakymo procese standartiškai būtinos vadybininkų akis į akį derybos, todėl tokias operacijas yra sunku klasifikuoti informacinių sistemų kontekste.

3. Išnagrinėjus verslo dokumentų standartus paaiškėjo, kad visame pasaulyje, priklausomai nuo regiono bei vystomos veiklos, organizacijos naudoja joms priimtinius standartus. Iš to išplaukia skirtingų organizacijų elektroninio bendradarbiavimo problema, nes šie verslo dokumentai tarpusavyje skiriasi ir nesuderina vieni su kitais.

4. Paanalizavus verslo dokumentų standartus (EDI, ebXML, UBL ir kitus) prieita prie išvados, jog UBL yra patraukliausias siūlomo metodo realizavimui įgyvendinti. Toks sprendimas priimtas remiantis tuo, jog šis standartas nuolat tobulinamas, sukurtas kitų standartų pagrindu, turi praplėtimo galimybes. Kurtas įvairių pramonės šakų atstovų.

5. Išanalizavus derybų taisyklių suderinimo sprendimus paaiškėjo, kad visų jų specifika bei veikimo principas gana ženkliai skiriasi tarpusavyje. Egzistuoja daug struktūrinių bei architektūrinių skirtumų, kurių kiekvienas turi tam tikrų privalumų bei skirtumų. Metodo realizacijai nuspręsta naudoti DROOLS įrankį. Sprendimas priimtas remiantis jo sugebėjimu automatiškai aptikti verslo taisyklių nesuderinamumus, bei tai atlikti labai greitai.

### 3. Sistemos reikalavimų specifikacija ir analizė

#### 3.1. Reikalavimų specifikacija

Siekiant iširti siūlomo, verslo taisyklių suderinimo, metodo veiksmingumą bei efektyvumą bus sukurta prototipinė informacinė sistema. Joje sukurtos atitinkamos vartotojo sąsajos, leisiančios vartotojams efektyviai ją valdyti. Reikalavimų specifikacija kuriama būtent šiai prototipinei sistemai.

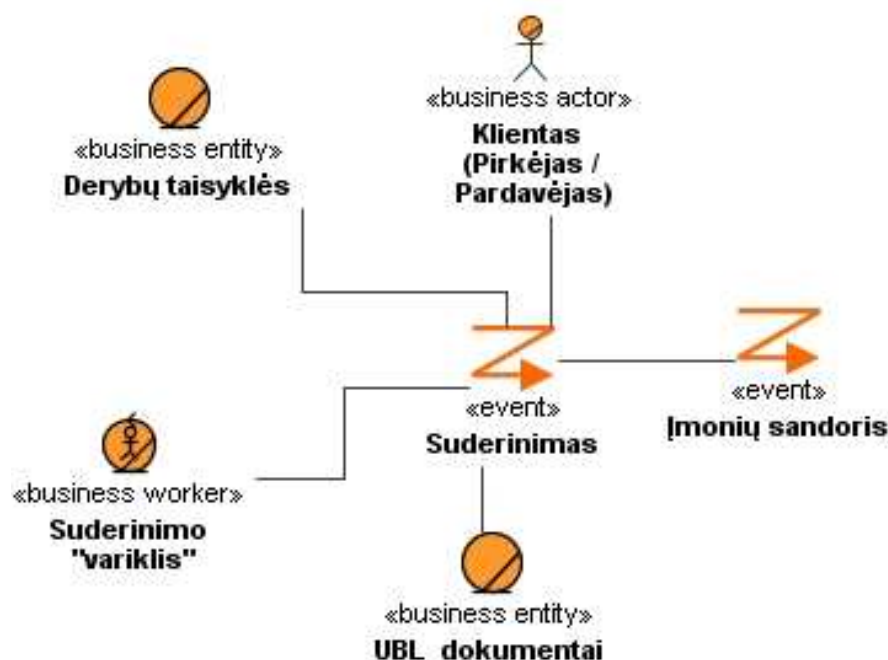
Įvedamų duomenų sintaksinė kontrolė atliekama panaudojant *JavaScript* scenarijus. Duomenų semantinė kontrolė atliekama serveryje.

Projektavimui pasirinkta *UML* kalba. Projektavimo įrankiu pasirinktas *MagicDraw UML* 16.6 paketas.

Sudarinėjant verslo dokumentus bus naudojamas UBL standartas. Toks sprendimas priimtas todėl, kad šis standartas prieinamas visiems, be jokių autorinių mokesčių, be to jis aprašo 31-ą dokumentų standartą, kurie apima įvairius verslo poreikius įskaitant užsakymus, pristatymus, sąskaitas faktūras ar apmokėjimus.

Informacinės sistemos dokumentacija ruošta *Microsoft Office Word* programa.

*Reikalavimai duomenims (veiklos konceptų modelis*



3.1 pav. Veiklos konceptų modelis

### 3.1.1. Funkciniai reikalavimai

<b>Reikalavimas #:</b>	<b>1</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>5</b>
<b>Aprašymas:</b>	Sistema turi leisti klientui (gavėjui) priimti kliento (siuntėjo) siunčiamus dokumentus				
<b>Pagrindimas:</b>	Išpildoma bendradarbiavimo sąlyga				
<b>Šaltinis:</b>	Sistemos administratorius				
<b>Tinkamumo kriterijus:</b>	Klientas (gavėjas) turi galimybę priimti dokumentus				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>			<b>5</b>

<b>Reikalavimas #:</b>	<b>2</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>4</b>
<b>Aprašymas:</b>	Sistema turi leisti siųsti užsakymo užklausa klientui				
<b>Pagrindimas:</b>	Norint pradėti sąveikumą būtina turėti klientų teikiamą prekių sąrašą.				
<b>Šaltinis:</b>	Klientas				
<b>Tinkamumo kriterijus:</b>	Klientai gali siųsti užsakymo užklausa klientui				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>			<b>4</b>

<b>Reikalavimas #:</b>	<b>3</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>2,4</b>
<b>Aprašymas:</b>	Sistema turi automatiškai pakeisti siunčiamų dokumentų turinį pritaikant derybų taisykles				
<b>Pagrindimas:</b>	Tai pagrindinis sistemos realizacijos metodas.				
<b>Šaltinis:</b>	Sistemos administratorius				
<b>Tinkamumo kriterijus:</b>	Sistema automatiškai pakeičia siunčiamų dokumentų turinį pritaikant derybų taisykles				

<b>Reikalavimas #:</b>	<b>4</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>2</b>
<b>Aprašymas:</b>	Klientui turi būti sudaryta galimybė pasirinkti reikšmes, kurie galimi būti naudojami sudarinėjant verslo taisykles				
<b>Pagrindimas:</b>	Klientas gali sukurti taisykles, kurios neatitiks dokumento reikalavimų, be to naujo dokumento kūrimas be galimų reikšmių labai sudėtingas				
<b>Šaltinis:</b>	Sistemos Klientas				
<b>Tinkamumo kriterijus:</b>	Sistema leidžia pasirinkti galimus laukus, kurių pagrindu sudaromos taisyklės				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>			<b>4</b>

<b>Reikalavimas #:</b>	<b>5</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>2</b>
<b>Aprašymas:</b>	Klientui turi būti sudaryta galimybė prioritetizuoti verslo taisykles				
<b>Pagrindimas:</b>	Verslo taisyklių prioritetizavimas yra esminė siūlomo metodo dedamoji				
<b>Šaltinis:</b>	Klientas				
<b>Tinkamumo kriterijus:</b>	Sistema leidžia klientui prioritetizuoti verslo taisykles				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>			<b>4</b>

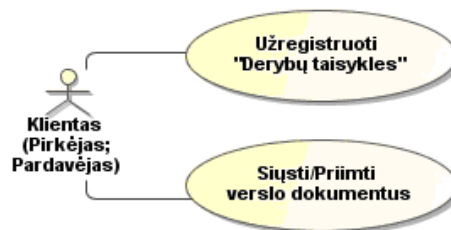
<b>Reikalavimas #:</b>	<b>6</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>1,2,3,4,5</b>
<b>Aprašymas:</b>	Sistema turi automatiškai patikrinti ar visi duomenys yra įvesti ir korektiški.				
<b>Pagrindimas:</b>	Neįvedus ar įvedus nekorektiškai kažkuriuos būtinus duomenis, sistema neduos laukiamo rezultato .				
<b>Šaltinis:</b>	Sistemos administratorius				
<b>Tinkamumo kriterijus:</b>	Sistema perspėja esant neįvestiems ar nekorektiškai įvestiems duomenims				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>	<b>4</b>		

<b>Reikalavimas #:</b>	<b>7</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>1,2,3,4,5</b>
<b>Aprašymas:</b>	Sistema turi iš duomenų bazės atrinkti ir paimti reikalingą informaciją.				
<b>Pagrindimas:</b>	Sąveikumui vykdyti reikalingos derybų taisyklių bei kitos informacijos suradimas, palyginimas bei atrinkimas.				
<b>Šaltinis:</b>	Sistemos administratorius				
<b>Tinkamumo kriterijus:</b>	Sistema automatiškai atrinka reikalingus duomenis iš duomenų bazės.				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>	<b>4</b>		

<b>Reikalavimas #:</b>	<b>8</b>	<b>Reikalavimo tipas:</b>	<b>9</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>4</b>
<b>Aprašymas:</b>	Sistema turi leisti klientui patvirtinti arba atmesti UBL_Dokumento siuntimą				
<b>Pagrindimas:</b>	Gali atsirasti netikslumų siunčiamuose dokumentuose, arba dokumentas po pakeitimo gali netenkinti siuntėjo				
<b>Šaltinis:</b>	Klientas				
<b>Tinkamumo kriterijus:</b>	Klientas (siuntėjas) turi galimybę atmesti arba patvirtinti siuntimą				
<b>Užsakovo patenkinimas:</b>	<b>3</b>	<b>Užsakovo nepatenkinimas:</b>	<b>4</b>		

- **Kompiuterizuojamų panaudojimo atvejų diagramos.**

*Kompiuterizuojamos sistemos funkcijų abstrakti diagrama.*



**3.2 pav.** Verslo taisyklių suderinimo įmonių sąveikumo sprendimuose panaudojimo atvejų diagrama

Apibendrintoje panaudojimo atvejų diagramoje (žr. 3.2 pav.) pavaizduotos pagrindinės kompiuterizuojamos funkcijos, kurias galės atlikti sistema suinteresuoti aktoriai. Iš esmės su sistema susiję dviejų rūšių aktoriai. T.y. „Klientas pirkėjas“, siunčiantis užsakymus bei „Klientas pardavėjas“, kuris gauna pirkėjo užsakymus. Šiuo atveju sistema atlieka siunčiamų dokumentų suderinimo funkciją. Taip pat vartotojai turės galimybę „Užregistruoti derybų taisykles“ kurių pagalba sistema atlieka vienokius ar kitokius veiksmus (tikrinimus), sąveikaujant įmonėms,



### Kompiuterizuojamos veiklos abstraktus veiklos modelis



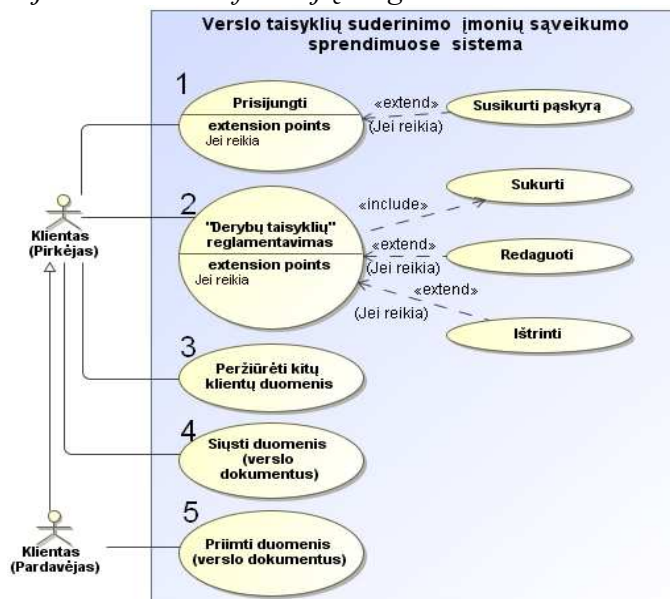
3.3 pav. Verslo taisyklių suderinimo įmonių sąveikumo sprendimuose veiklos proceso diagrama

Abstraktus veiklos proceso modelis pavaizduotas veiklos diagramoje 3.3 paveiksle kol kas sudarytas tik iš pagrindinių, sistemoje bei aplink ją egzistuojančių, veiklos procesų, nekreipiant dėmesio į smulkiąsias sudedamąsias dalis ar egzistuojančius ryšius tarp jų (detalus jų aprašymas bus pateiktas nagrinėjant sistemos elgseną 3.3.2 skyriuje). Šis abstraktus modelis padeda lengviau susidaryti bendrą pradinį suvokimą apie tai, kaip sistema susijusi su išoriniais aktoriais procesų lygmenyje.

Diagramoje matomos vertikalios juostos vaizduoja sistemą bei išorinius aktorius, kurie vykdo atitinkamas veiklas (procesus). Iš esmės „Klientą (Pirkėją“ ir „Klientą (pardavėją“ galima būtų sutapatinti tarpusavyje, kadangi tiek vienas tiek kitas gali atlikti tapačias funkcijas, tačiau norint tinkamai atskleisti pilną veiklos procesą nuo pradžios iki galo, aiškiau yra juos išskirti.

Taigi „Klientas (Pirkėjas)“ siunčia į sistemą savo reglamentuotas derybų taisykles (tam, kad paskui galėtų vykdyti sąveikumą) arba tiesiog siunčia verslo dokumentus į kitą įmonę. Kuriama sistema šiame siuntime naudojama tam, kad pritaikytų siunčiamų dokumentų formatą prie kitos įmonės informacinės sistemos. T.y. sistema suderina įmonių naudojamas, galimai skirtingas, informacines sistemas prie vienodo standarto, kuris atveria kelią į efektyvų įmonių bendradarbiavimą.

### Kompiuterizuojamos sistemos funkcijų diagrama



3.4 pav. Sistemos reikalavimų specifikacijos panaudojimo atvejų diagrama

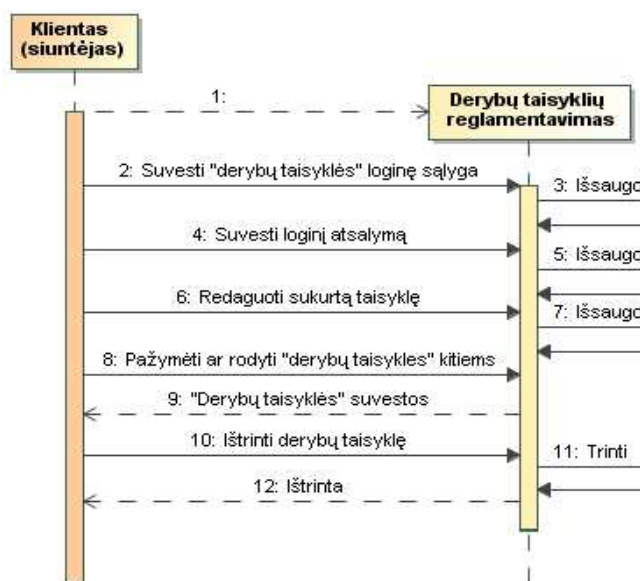
Kiekvienas, metodo panaudojimu suinteresuotas, „aktorius“ susiejamas su pagrindiniais panaudojimo atvejais („Prijungti“, „Derybų taisyklių reglamentavimas“, „Peržiūrėti kitų klientų duomenis“, „Siųsti duomenis“ bei „priimti duomenis“). Šie panaudojimo atvejai specifikuojami atitinkamai kitais panaudojimo atvejais, nusakančiais sistemos funkcionalumą, kurie apibūdina metodo išpildymą (žr. 3.4 pav.).

Sudėtingesnių (sudėtinių) panaudojimo atvejų (kurių vykdymo rezultatas nėra tik ataskaitos suformavimas) specifikacijos pateiktos lentelėse.

*Panaudojimo atvejo „Derybų taisyklių reglamentavimas“ specifikacija*

Panaudojimo atvejis „Derybų taisyklių reglamentavimas“	
<b>Aktorius</b>	Klientas(Pirkėjas), Klientas(Pardavėjas)
<b>Prieš sąlyga</b>	Klientas(Pirkėjas) prisijungęs prie sistemos;
<b>Sužadinimo sąlyga</b>	Klientai nori suvesti derybų taisykles vėlesniam dokumentų suderinimui
<b>Susiję panaudojimo atvejai</b>	„Sukurti“; „Redaguoti“; „Ištrinti“;
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Suvesti „derybų taisyklės“ loginę sąlygą	1.1. Sistema išsaugo derybų taisyklę ir priskiria ją konkrečiam agentui
2. Suvesti loginį atsakymą	2.1. Sistema išsaugo derybų taisyklės įvykdymo pasekmę
3. Pažymėti ar rodyti „derybų taisyklę“ kitiems	3.1. Sistema išsimeina pažymėjimo rezultata
4. Ištrinti „derybų taisyklę“	4.1 Sistema ištrina derybų taisyklę
<b>Po sąlyga</b>	Duomenų bazėje išsaugotos „derybų taisyklės“
<b>Alternatyvūs scenarijai</b>	
1. Redaguoti sukurtą „derybų taisyklę“	Jei sistemoje jau yra derybų taisyklė, ją galima redaguoti. <b>Po sąlyga</b> Sistema išsaugo pataisytą derybų taisyklę.

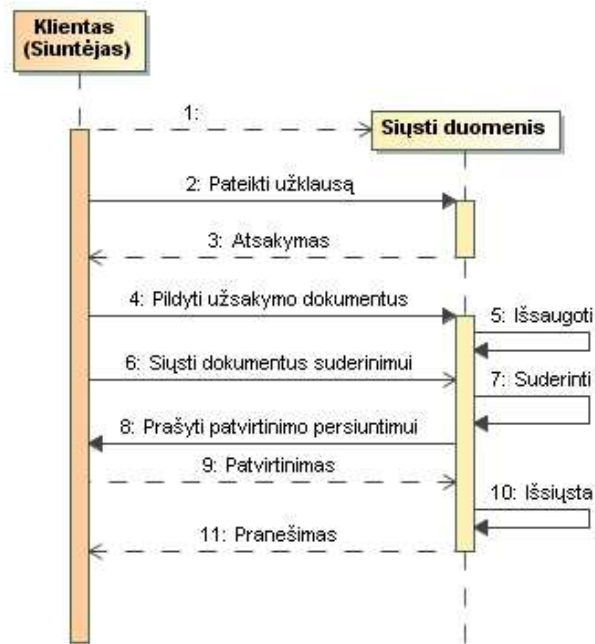
Įvykis „Pažymėti ar rodyti derybų taisyklę kitiems“ nusako galimybę aktoriui pasirinkti rodyti taisyklę ar ne. To priežastys gali būti įvairios ir priklauso nuo pačio aktoriaus poreikių ir strateginių sumetimų.



3.5 pav. Reikalavimų specifikacijos „Suvesti derybų taisykles“ sekų diagrama

Panaudojimo atvejo „Siųsti duomenis“ specifikacija

Panaudojimo atvejis „Siųsti duomenis“	
<b>Aktorius</b>	Klientas(Pirkėjas), Klientas(Pardavėjas)
<b>Prieš sąlyga</b>	Klientai prisijungęs prie sistemos; suvestos derybų taisyklės
<b>Sužadinimo sąlyga</b>	Klientas(Pirkėjas) nori sudaryti užsakymą bei pateikti jį vykdytojui;
<b>Susiję panaudojimo atvejai</b>	
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Pateikti užklausa	1.1. Klientas(Pirkėjas) siunčia užklausa klientui(pardavėjui), kad išduotų prekių katalogą
2. Pildyti užsakymo dokumentus	2.1. Klientas (Pirkėjas) pildo dokumentą (užsakymą)
3. Siųsti dokumentus suderinimui	3.1. Klientas(Pirkėjas) pirmiausia siunčia dokumentus į sistemą, kad būtų pritaikytos derybų taisyklės. Sistema atitinkamai pakeičia dokumentų turinį ir išduoda pasirinkimą <i>Persiųsti ar nepersiųsti</i> klientui(Pardavėjui)
4. Patvirtinti siuntimą	4.1. Klientas(Pirkėjas) leidžia persiųsti dokumentus
<b>Po sąlyga</b>	Užpildyti, suderinti bei išsiųsti užsakymo dokumentai
<b>Alternatyvūs scenarijai</b>	
1. Nepatvirtinti siuntimo	Klientas(Pirkėjas) neleidžia persiųsti dokumentų. <b>Po sąlyga:</b> Dokumentai neišsiųsti

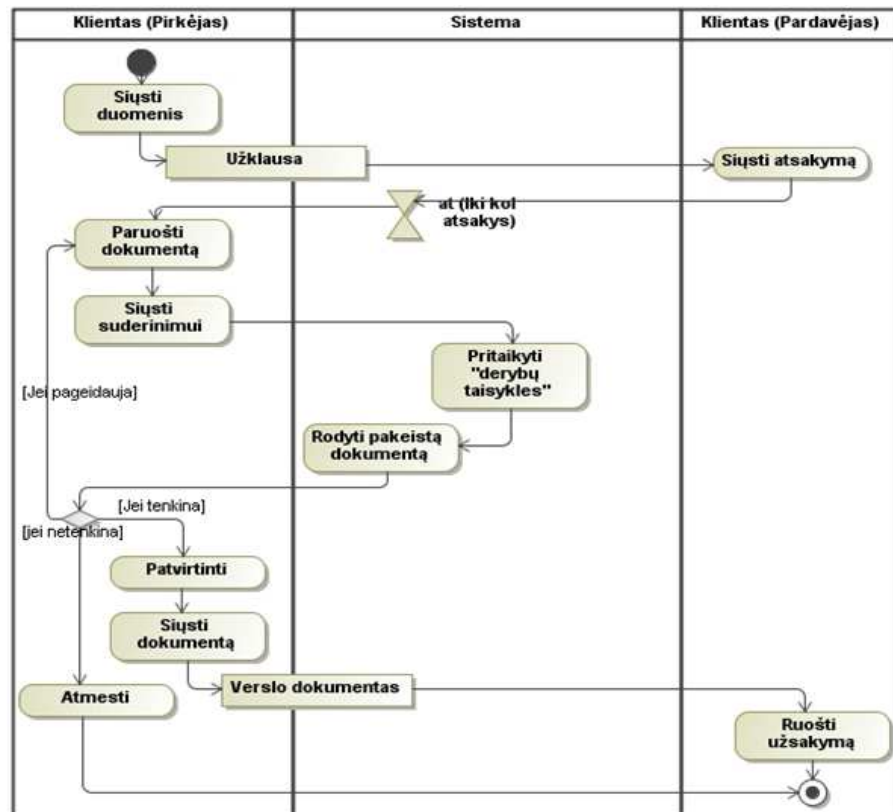


3.6 pav. Reikalavimų specifikacijos „Suvesti derybų taisyklės“ sekų diagrama

### Kompiuterizuojamos veiklos modelis

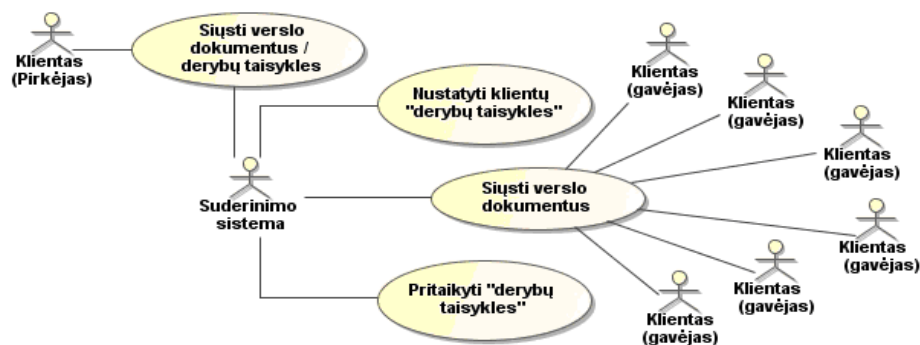
Reikalavimų specifikacijos kompiuterizuojamos veiklos modelis (žr. 3.7 pav.) nusako kokius veiksmus ir kokia tvarka galės atlikinėti klientai, tarpusavyje vykdantys sąveikumą (derybų procesą). Čia priimta, jog derybų taisyklės jau yra suvestos į sistemą ir šiuo konkrečiu atveju keičiamos nebus. Atlikus visus veiksmus, nurodytus modelyje, gaunamas galutinis užsakymo dokumentas, kuris tenkina (turėtų tenkinti) abi užsakyme dalyvaujančias „šalis“. Paminėtina, kad „Klientas(Pardavėjas)“ visame procese dalyvauja tik

du kartus: Pateikdamas atsakymą į užklausą bei priimdamas dokumentus. Dėl šios priežasties derybų proceso trukmė ženkliai sutrumpėja.



3.7 pav. Kompiuterizuojamos veiklos modelis

### Sąveikos su kitomis sistemomis panaudojimo atvejų modelis



3.8 pav. Sąveikos su kitomis sistemomis modelis

Sąveikos su kitomis sistemomis panaudojimo atvejų modelis rodo (žr. 3.8 pav.), kaip „Suderinimo sistema“ bendrauja su kitomis sistemomis. Iš esmės ji gali bendrauti su visomis skirtingose įmonėse egzistuojančiomis sistemomis, kadangi visos jos turės naudoti vienodą UBL (angl. *Universal Business Language*) dokumentų standartą.

### 3.1.2. Nefunkciniai reikalavimai

#### *Reikalavimai sistemos išvaizdai*

<b>Reikalavimas #:</b>	<b>1</b>	<b>Reikalavimo tipas:</b>	10	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4, 5
<b>Aprašymas:</b>	Neįkyri, nereikalaujanti ką nors kelis kartus tvirtinti, sąsaja.				
<b>Pagrindimas:</b>	Sistemą su neįkyria vartotojo sąsaja palankiau vertins jos vartotojai.				
<b>Šaltinis:</b>	Administratorius				
<b>Tinkamumo kriterijus:</b>	Neįkyri, neperkrauta (tik tai kas apibrėžta reikalavimuose) vartotojo sąsaja.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		

<b>Reikalavimas #:</b>	<b>2</b>	<b>Reikalavimo tipas:</b>	10	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4, 5
<b>Aprašymas:</b>	Lengvai skaitoma sąsaja bei paprastas ir nesudėtingas naudojimas.				
<b>Pagrindimas:</b>	Siekama, kad klientai kuo greičiau įsisavintų naująją sistemą.				
<b>Šaltinis:</b>	Administratorius				
<b>Tinkamumo kriterijus:</b>	Lengvai skaitoma sąsaja bei paprastas ir nesudėtingas panaudojimas.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		

<b>Reikalavimas #:</b>	<b>3</b>	<b>Reikalavimo tipas:</b>	10	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4, 5
<b>Aprašymas:</b>	Vartotojo sąsaja turi būti 3 dedamųjų: a) Skirtos dokumentams sudaryti; b) Skirtos taisyklėms sudaryti; c) Skirtos dokumentams siųsti				
<b>Pagrindimas:</b>	Skirtingi sąsajos langai leis efektyviau naudotis sistema				
<b>Šaltinis:</b>	Administratorius				
<b>Tinkamumo kriterijus:</b>	Vartotojo sąsaja realizuoja 3 skirtingus priėjimus				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>3</b>		

#### *Reikalavimai vykdymo savybėms*

<b>Reikalavimas #:</b>	<b>4</b>	<b>Reikalavimo tipas:</b>	12	<b>Įvykis/panaudojimo atvejis #:</b>	2
<b>Aprašymas:</b>	Verslo taisyklės turi būti saugomos sistemoje (po įvedimo) iki klientas pats ištrins arba klientas bus pašalintas iš sistemos				
<b>Pagrindimas:</b>	Kad nereikėtų kaskart prisijungus vesti duomenų iš naujo.				
<b>Šaltinis:</b>	Administratorius				
<b>Tinkamumo kriterijus:</b>	duomenys po įvedimo saugomi				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		

<b>Reikalavimas #:</b>	<b>5</b>	<b>Reikalavimo tipas:</b>	12	<b>Įvykis/panaudojimo atvejis #:</b>	2
<b>Aprašymas:</b>	Verslo taisyklės turi atitikti griežtus formalizavimo reikalavimus				
<b>Pagrindimas:</b>	Sistema neatpažins neformalizuotų verslo taisyklių				
<b>Šaltinis:</b>	Klientas				
<b>Tinkamumo kriterijus:</b>	Verslo taisyklės atitinka formalizavimo reikalavimus				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		

### *Reikalavimai saugumui*

<b>Reikalavimas #:</b>	<b>7</b>	<b>Reikalavimo tipas:</b>	<b>15</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>1,4,5</b>
<b>Aprašymas:</b>	Klientai su skirtingais prisijungimo vardais ir slaptažodžiais gali keisti tik savo reglamentuotus duomenis, o mato visų klientų viešai pateiktą informaciją.				
<b>Pagrindimas:</b>	Prisijungimo vardai suteikia atpažinimą bei konfidencialumą				
<b>Šaltinis:</b>	Administratorius				
<b>Tinkamumo kriterijus:</b>	Klientai su skirtingais prisijungimo vardais keičia tik savo duomenis, o mato tai ką rodo kiti				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		

### *Kultūriniai-politiniai reikalavimai*

<b>Reikalavimas #:</b>	<b>8</b>	<b>Reikalavimo tipas:</b>	<b>16</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>1,2,3,4,5,</b>
<b>Aprašymas:</b>	Sistemoje turi būti galimybė naudoti visų Europos šalių alfabetą.				
<b>Pagrindimas:</b>	Dokumentai bei visa kita informacija pildomi įvairiomis kalbomis.				
<b>Šaltinis:</b>	Klientas				
<b>Tinkamumo kriterijus:</b>	Sistemoje naudojamas visų Europos šalių alfabetas.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		

### *Teisiniai reikalavimai*

<b>Reikalavimas #:</b>	<b>9</b>	<b>Reikalavimo tipas:</b>	<b>17</b>	<b>Įvykis/panaudojimo atvejis #:</b>	<b>1,2,3,4,5</b>
<b>Aprašymas:</b>	Sistemoje saugomi klientų duomenys negali būti naudojami kitais tikslais negu tais, kurie buvo reglamentuoti dokumente, kurį jie pasirašė prieš patvirtindami paskyrą.				
<b>Pagrindimas:</b>	Klientų duomenų naudojimas ten kur jie nežino yra neleistinas poįstatyminių aktų.				
<b>Šaltinis:</b>	Administratorius; Projektų vadovas				
<b>Tinkamumo kriterijus:</b>	Sistemoje saugomi klientų duomenys naudojami kitais tikslais negu tais, kurie buvo reglamentuoti dokumente, kurį jie pasirašė prieš patvirtindami paskyrą.				
<b>Užsakovo patenkinimas:</b>	<b>5</b>	<b>Užsakovo nepatenkinimas:</b>	<b>5</b>		

## **3.2. Reikalavimų analizės apibendrinimas**

Šiame skyriuje suformuluoti funkciniai ir nefunkciniai reikalavimai kiekvienam panaudojimo atvejui, kuriuos sistemos vartotojai vykdys ją naudodami. Visi reglamentuoti reikalavimai aprašyti nurodant reikalavimo prioritetą, stabilumą, pažeidimo pasekmes, ryšius su kitais reikalavimais, reikalavimo kėlimo motyvą bei šaltinį, iš kurio kilo reikalavimas.

Taip pat pateiktos įvairios panaudojimo atvejų specifikacijos, aptarti veiklos modeliai bei nustatytos sąveikos tarp atskirų sistemos vartotojų.

## 4. Sistemos projektas

### Projekto tikslas

Projekto tikslas – suprojektuoti bei realizuoti, analizės dalyje išnagrinėtą „Verslo taisyklių suderinimo įmonių sąveikumo sprendimuose“ metodą apimančią informacinę sistemą, panaudojant *CASE* įrankius. Nustatyti, kaip turi būti realizuota vartotojo sąsaja, kokios reikalingos programinės klasės, realizuojančios tiek veiklos funkcijas, tiek vartotojo sąsają, sudaryti duomenų bazės modelį ir kt.

### Projekto gyvavimo ciklo pasirinkimas

Prieš pradėdant kurti projektą ar realizuoti norimą sistemą, pirmiausia reikia nustatyti, kokie procesai bus vykdomi, bei kokia tvarka visa tai bus atliekama. T.y. pirmiausia reikia nustatyti koks gyvavimo ciklas geriausiai tinka siekiant sukurti efektyvų bei visus suinteresuotus „aktorius“ tenkinantį produktą.

Projekto gyvavimo ciklą yra gana daug (nuoseklusis, laipsniškasis, evoliucinis, prototipinis, spiralinis, *Agile* ir daugelis kitų), todėl norint nustatyti patį tinkamiausią reikia įvertinti daugybę faktorių. Įvertinus pasaulinę informacinių sistemų (IS) kūrimo praktiką buvo nustatyta, jog pagrindiniai IS gyvavimo ciklo pasirinkimo kriterijai – kuriamos IS sistemos sudėtingumas bei neapibrėžtumas. Jeigu sistema yra neapibrėžta (iš anksto tiksliai neapibrėžti visi reikalavimai) tuomet tiksliausia būtų rinktis evoliucinį, o labai sudėtingoms ir didelėms sistemoms laipsniškąjį projekto gyvavimo ciklą.

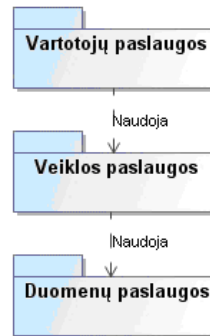
Kadangi kuriama verslo taisyklių suderinimo, įmonių sąveikumo sprendimuose, sistema nėra iš anksto išsamiai apibrėžta, nežinomi tikslūs reikalavimai (egzistuoja tik pradinė bei neišbaigta reikalavimų aibė), buvo pasirinktas evoliucinis gyvavimo ciklas. Jo pagalba bus stengiamasi sukurti prototipinį sistemos variantą, kuris aiškėjant išsamesniems reikalavimams bei pastebėtiems trūkumams, bus tobulinamas. Šis procesas kartosis keletą kartų kol bus pasiektas toks rezultatas, kurio tikisi jos kūrimu suinteresuoti „aktoriai“.

Paminėtina, jog šiuo ciklu kuriamų IS sistemų kokybė būna prastesnė, jos tolimesnis palaikymas sudėtingesnis. Taip yra todėl, kad pradėjus kurti sistemą nėra žinoma visa jos apimtis. Ji su lig kiekvienu iteracijos pasikartojimu vis didėja. Dažniausiai vietoj to, kad perdaryti sistemą, taisant jos trūkumus yra „prilipdomos“ naujos funkcijos. Dėl šių priežasčių sistemos naudingumas šiek tiek suprastėja, o jos funkcionalumo palaikymas sumažėja. Tačiau kadangi kuriama sistema turi pasižymėti ne greitumu ar sistemos palaikymo efektyvumu, o savo funkcijų naujumu bei naudingumu, šie trūkumai nėra labai aktualūs.

## 4.1. Sistemos architektūros projektas

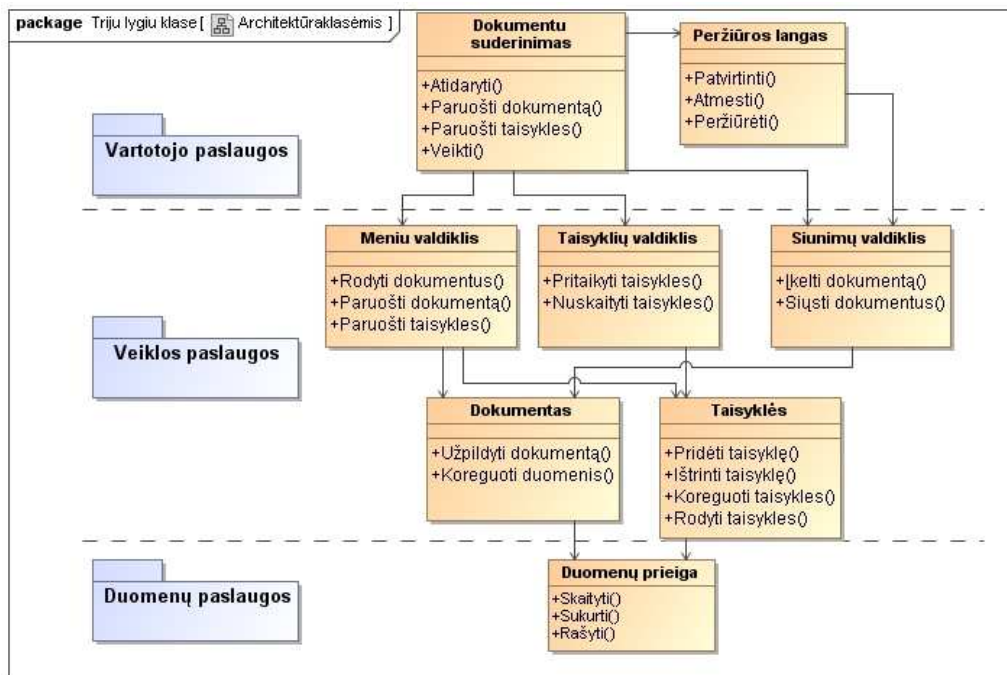
### 4.1.1. Sistemos loginė architektūra

Verslo taisyklių suderinimo įmonių sąveikumo sprendimuose metodo realizavimo informacinė sistema yra išskaidoma į atskirus posistemius (paketus): „Derybų taisyklių reglamentavimą“, „UBL\_Dokumentų sudarymą“ bei „UBL\_Dokumentų pakeitimą pritaikant derybų taisykles“. Visi šie posistemiai pasiskirsto klasikiniame trijų lygių architektūros modelyje, atskiriant vartotojo, veiklos ir duomenų paslaugas (žr. 4.1 pav.).



4.1 pav. Loginė trijų lygių architektūra

Kuriamų paketų priklausomybė ir klasės, kurios bus realizuotos atitinkamuose paketuose, pateiktos 4.2 paveiksle. Pabrėžiama, jog paveiksle vaizduojamos tik pagrindinės, iš anksto žinomos, klasės. Vėliau jų kiekis didės (jos bus detalizuojamos).

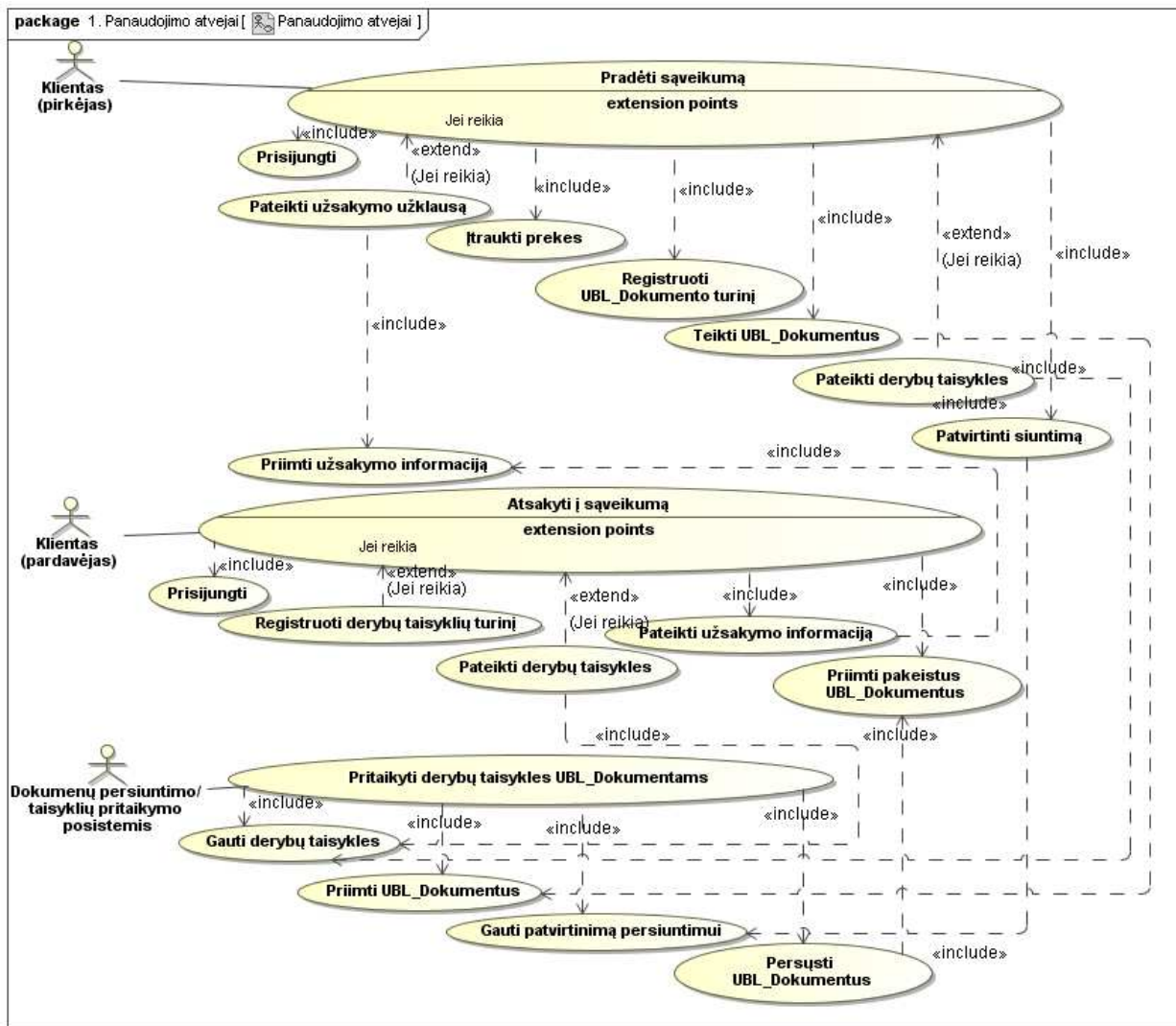


4.2 pav. Informacinės sistemos architektūra



#### 4.1.2. Panaudojimo atvejų specifikacija

Kiekvienas, metodo panaudojimu suinteresuotas, „aktorius“ susiejamas su pagrindiniais panaudojimo atvejais („Pradėti sąveikumą“, „Atsakyti į sąveikumą“, bei „Pritaikyti derybų taisykles UML\_Dokumentams“). šie panaudojimo atvejai specifikuojami atitinkamai kitais panaudojimo atvejais, nusakančiais sistemos funkcionalumą, kurie apibūdina metodo išpildymą (žr. 4.3 pav.).

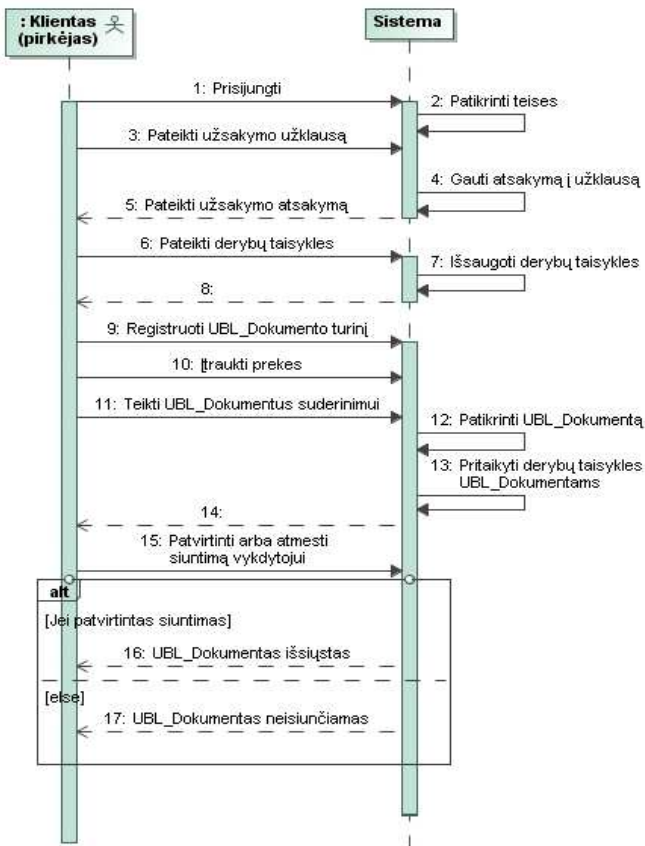


4.3 pav. Kompiuterizuojamų panaudojimo atvejų modelis

Aktorius „Klientas (Pirkėjas)“ panaudojimo atveju „Pradėti sąveikumą“ inicijuoja veiklos pradžia. Jis teikia užklausas, tvirtina dokumentų siuntimus ir panašiai. Savo ruožtu „Klientas (Pardavėjas)“ atsako į inicijuotą veiklą atitinkamais savo veiksmais. Taip pat, kaip metodo realizavimo aktorius – išskiriamas „Dokumentų persiuntimo/taisyklių pritaikymo posistemis“, kuris atlieka klientų reglamentuotus veiksmus.

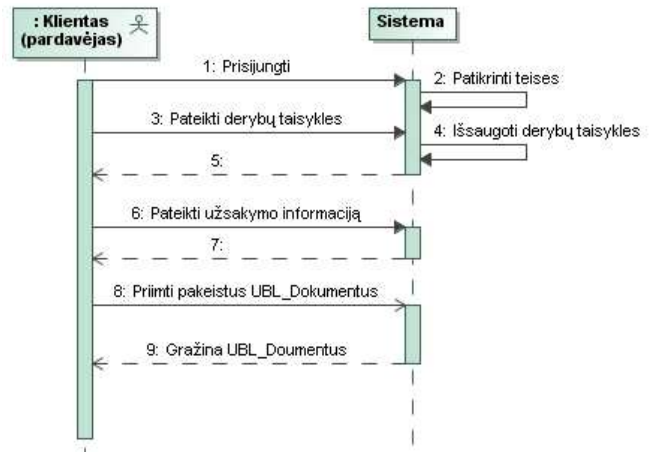
Panaudojimo atvejų „Pradėti sąveikumą“ bei „Atsakyti į sąveikumą“ specifikacijų sekos atitinkamai pateiktos 4.4, 4.5 paveiksluose.

Panaudojimo atvejo „Pradėti sąveikumą“ scenarijus



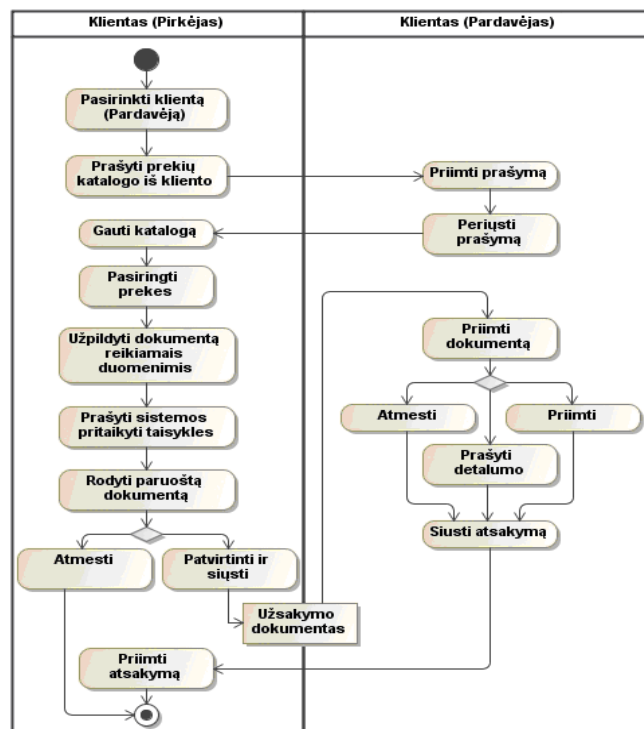
4.4 pav. Panaudojimo atvejo „Pradėti sąveikumą“ sekų diagrama

• Panaudojimo atvejo „Atsakyti į sąveikumą“ scenarijus



4.5 pav. Panaudojimo atvejo „Atsakyti į sąveikumą“ sekų diagrama

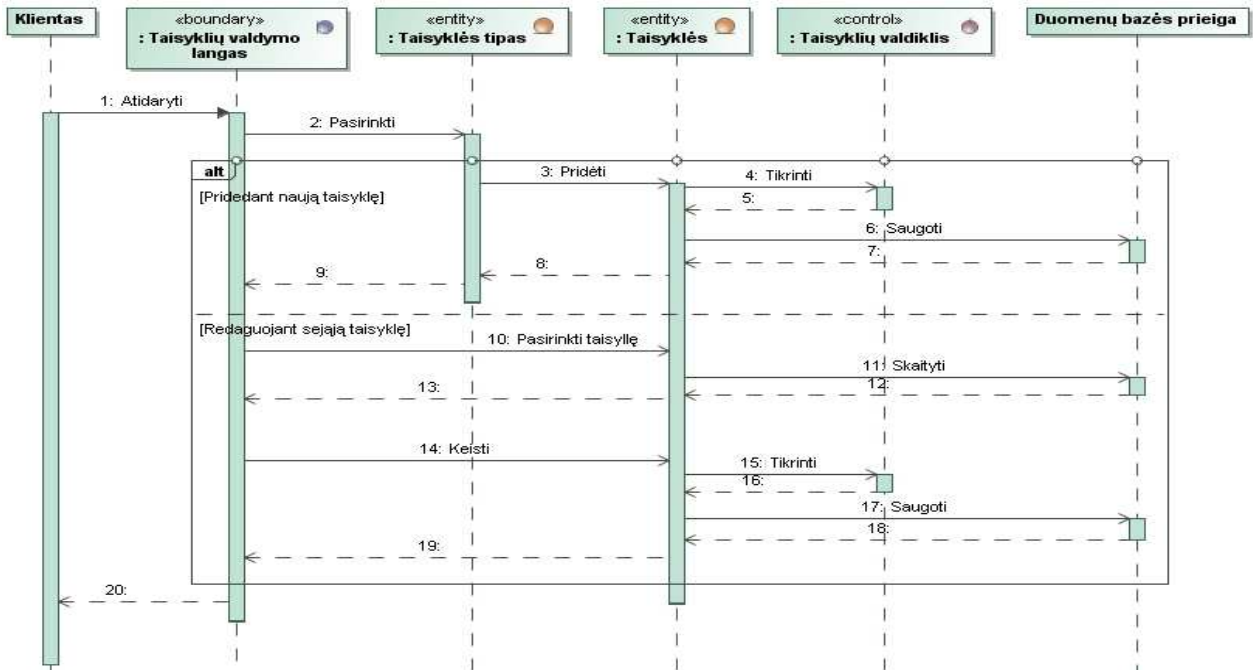
Veiklos diagramoje (žr. 4.6 pav.) parodyta, kaip susisieja panaudojimo atvejai „Pradėti sąveikumą“ bei „atsakyti į sąveikumą“ tarpusavyje.



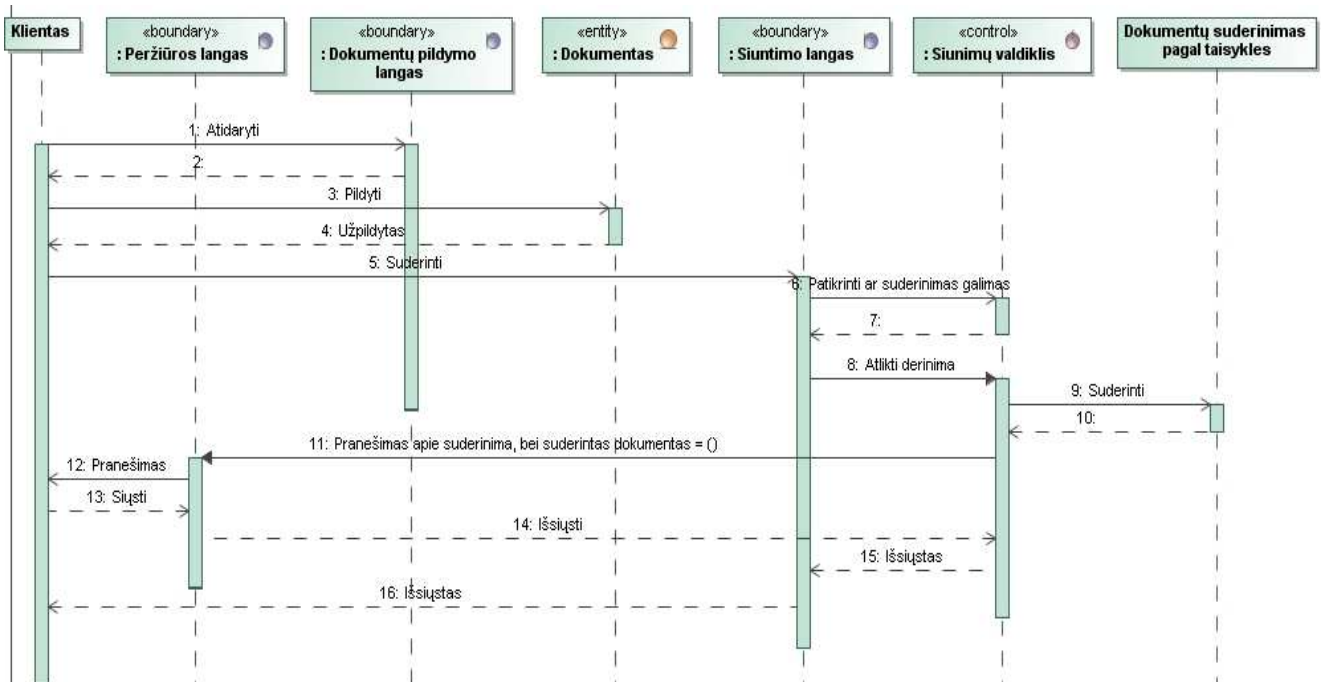
4.6 pav. Detali Klientų (pirkėjo ir pardavėjo) veiklos diagrama

## Panaudojimo atvejų sekų diagramos

Sudarytos pagrindinių panaudojimo atvejų sekų diagramos, kurios rodo bendravimą tarp klasių, realizuojančių panaudojimo atvejus.

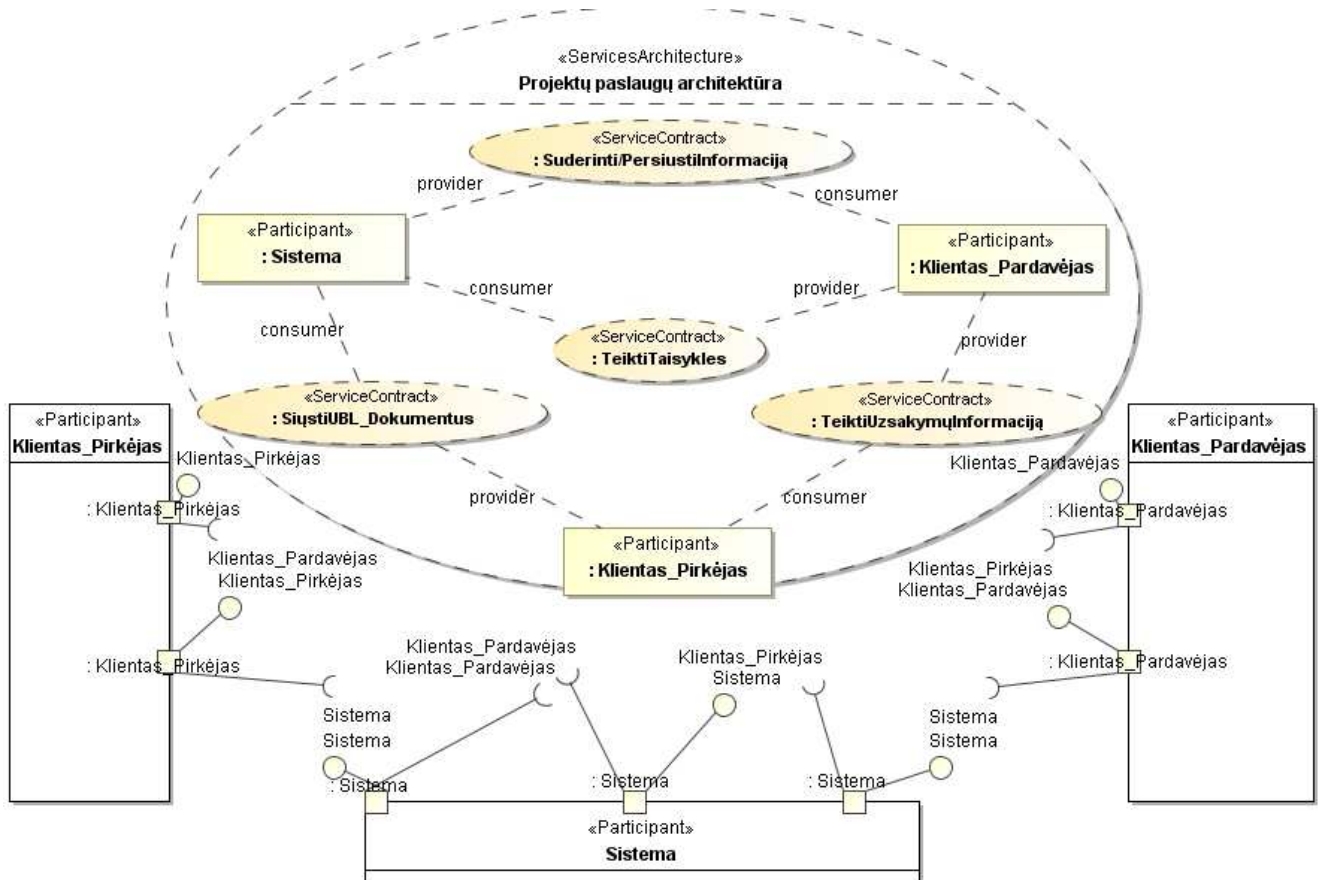


4.7 pav. Derybų taisyklių reglamentavimo sekų diagrama



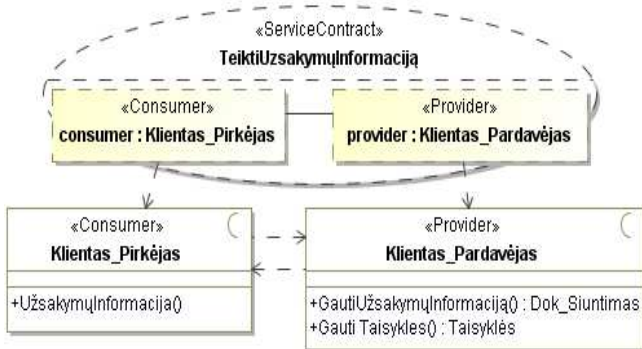
4.8 pav. UBL\_Dokumentų paruošimo bei siuntimo sekų diagrama

### 4.1.3. Veiklos paslaugu architektūra

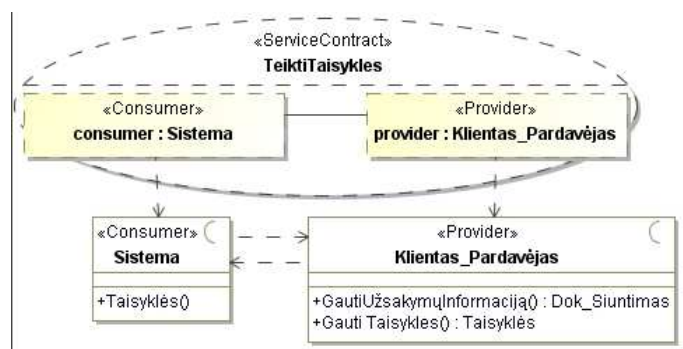


4.9 pav. Paslaugu architekturos modelis

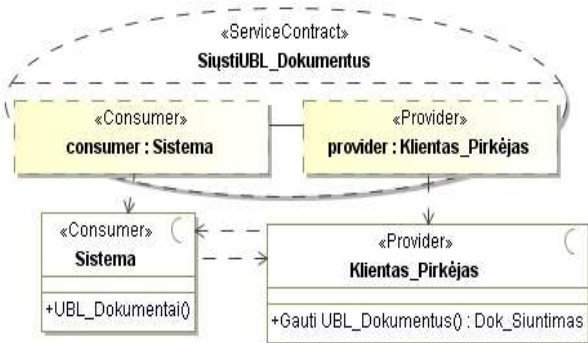
Atitinkamos paslaugas:



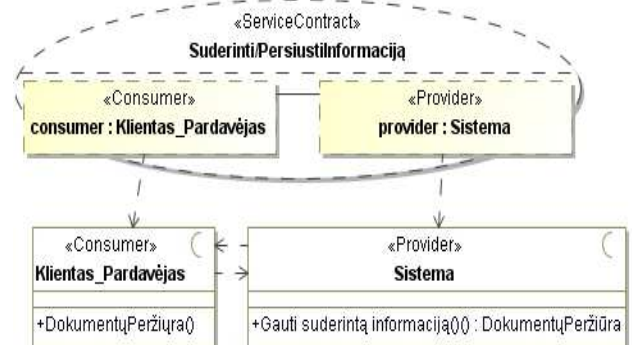
4.10 pav. Paslaugas „TeiktiUzsakymuInformacija“ modelis



4.11 pav. Paslaugas „TeiktiTaisykles“ modelis



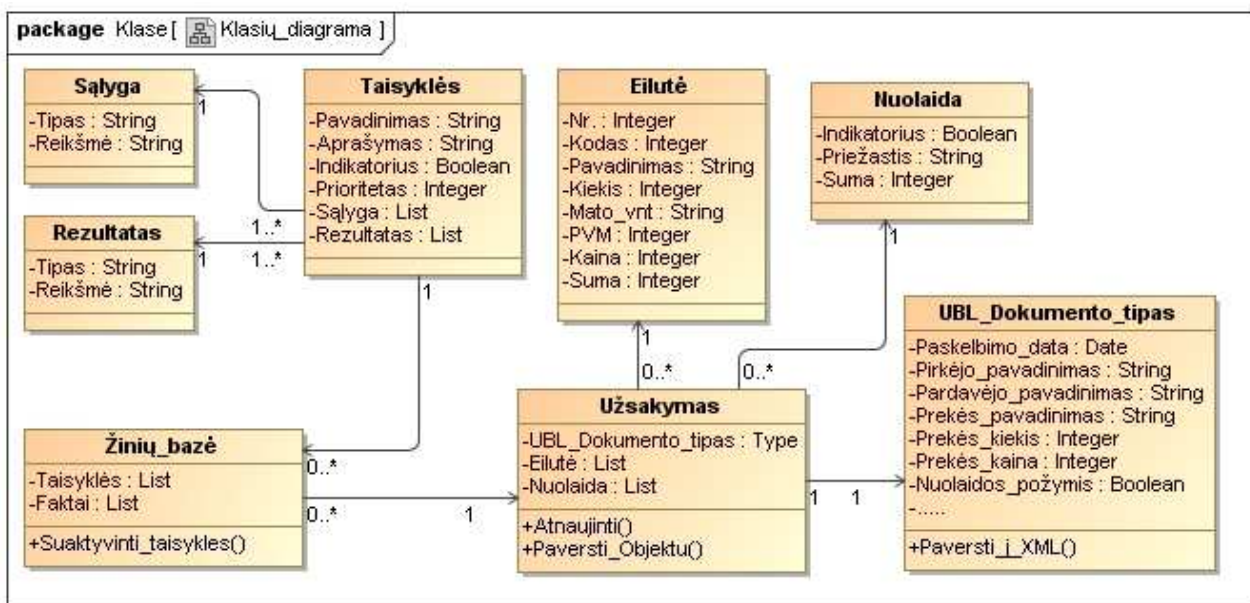
4.12 pav. Paslaugas „SiustiUBL\_Dokumentus“ modelis



4.13 pav. Paslaugas „SuderintiPersiustiInformacija“ modelis

## 4.2. Detalus projektas

Šioje dalyje pateikiama principinė (supaprastinta) UML klasių diagrama. Pateiktame pavyzdyje (žr. 4.14 pav.) atsispindi pagrindinės sistemos funkcijos. Tai yra derybų taisyklių ruošimas, dokumentų ruošimas, jų suderinimas bei siuntimas. Visi atributai bei metodai yra unikalūs, todėl atsiskleidžia sistemos (idėjos) autentiškumas bei naujumas. Panagrinėjus šią diagramą galima pastebėti, jog pritaikant taisykles UBL dokumentui, reikšmės nėra keičiamos tiesiogiai, t.y. dokumento XML kodas paverčiamas į objektą (naudojamas konstruktorius), toliau atliekami reikalingi pakeitimai ir galiausiai objektas vėl paverčiamas į XML struktūros UBL dokumentą tačiau jau su pakeistomis reikšmėmis.



4.14 pav. UML klasių diagrama

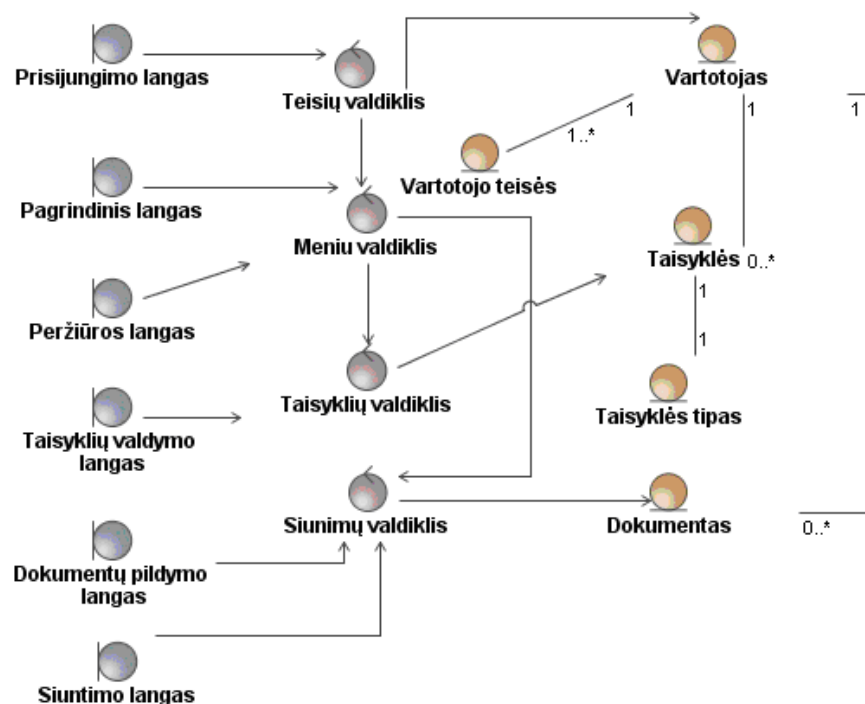
### 4.2.1. Panaudojimo atvejų realizacijos – analizės klasių diagramos

Analizės klasių diagramos parodo, kokias ribines (vartotojo sąsajos), valdymo (programines) ir esybių (duomenų) klases reikia sukurti, norint realizuoti panaudojimo atvejus. Čia pateikiamas analizės klasių modelis (žr. 4.15 pav.), rodantis panaudojimo atvejų, išanalizuotų 3.1.2. punkte, realizavimą. Kiti panaudojimo atvejai neanalizuoti, nes jų veiklos rezultatas yra ataskaitų generavimas. Šiame etape gautos klasės, vėliau bus papildomos, kad būtų galima realizuoti ir visus panaudojimo atvejus.

Norint naudotis sistema pirmiausiai reikia prisijungti. Atidaromas prisijungimo langas, kuriame įvedus prisijungimo vardą ir slaptažodį prisijungimo valdiklis juos patikrina ir, jei įvesti duomenys yra teisingi, yra pateikiamas pagrindinis langas, kuriame yra meniu.

Pasirinkus meniu punktą „Peržiūra“, meniu valdiklis kreipiasi į meniu valdiklį, kuris iškviečia metodą, suformuojantį taisyklių, vartotojų bei esamų dokumentų ataskaitas. Pasirinkus taisyklių valdymo langą taisyklių valdiklis suteikia galimybę pildyti naujas derybų taisykles, jas redaguoti ar pašalinti. Pildant derybų taisykles suteikiama galimybė pasirinkti iš galimų taisyklės šablonų, kas palengvina darbą. Dokumentų pildymo langas suteikia galimybę pildyti UBL\_Dokumentus, taip sudarant užsakymus. Siuntimų valdiklis dokumentą siunčia į posistemį, kur jis yra apdorojamas.

Įvedus visus duomenis bei juos atitinkamai pakeitus pritaikant derybų taisykles gaunamas galutinis UBL\_Dokumentas, užsakymas, kuris yra išsaugomas, t.y. sukuriama esybės „UBL\_Dokumentas“ egzempliorius, bei išsiunčiamas vykdytojui (Klientui pardavėjui).

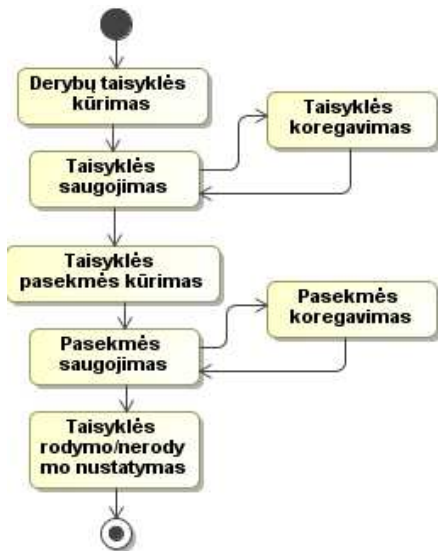


4.15 pav. Analizės klasių modelis

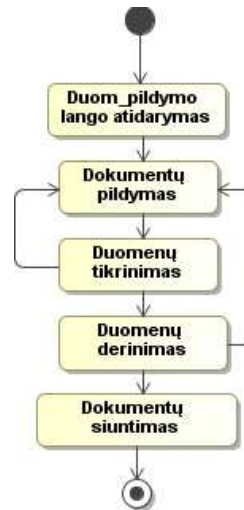
### 4.3. Sistemos elgsenos modelis

#### 4.3.1. Sistemos ir jos elementų būsenų modeliai

Būsenų diagramos (žr. 4.16, 4.17 pav.) parodo kokias stadijas reikia pereiti, kol gaunamas galutinis rezultatas. Pateiktuose pavyzdžiuose rodoma į kokias būsenas patenkama reglamentuojant taisykles bei siunčiant dokumentus.



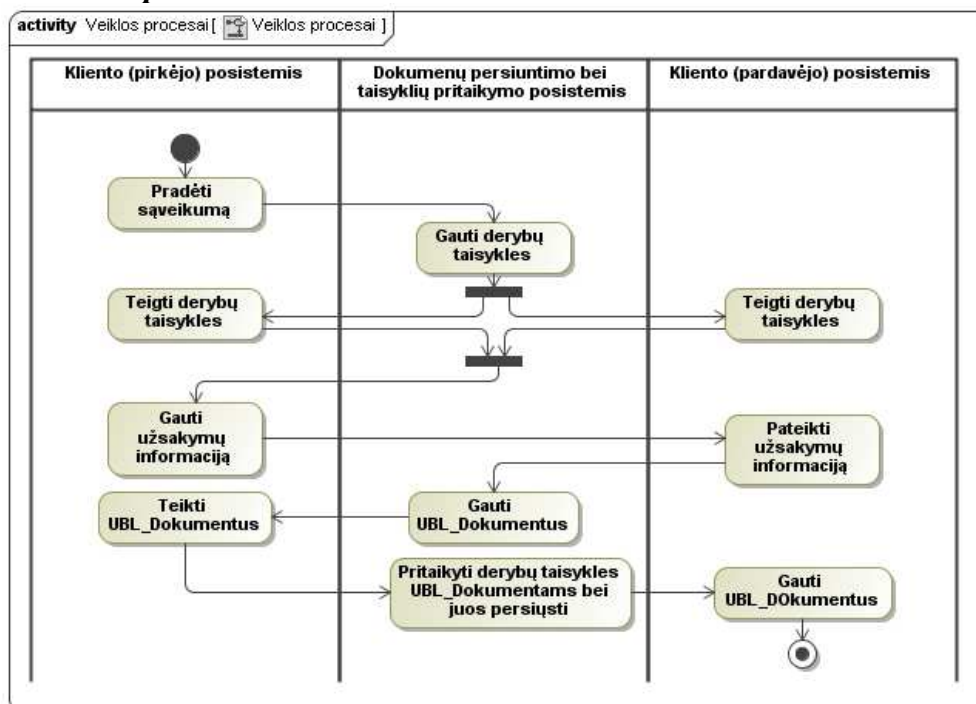
4.16 pav. Derybų taisyklių reglamentavimo būsenų diagrama



4.17 pav. UBL\_Dokumentų paruošimo bei siuntimo būsenų diagrama

#### 4.3.2. Veiklos proceso modeliai

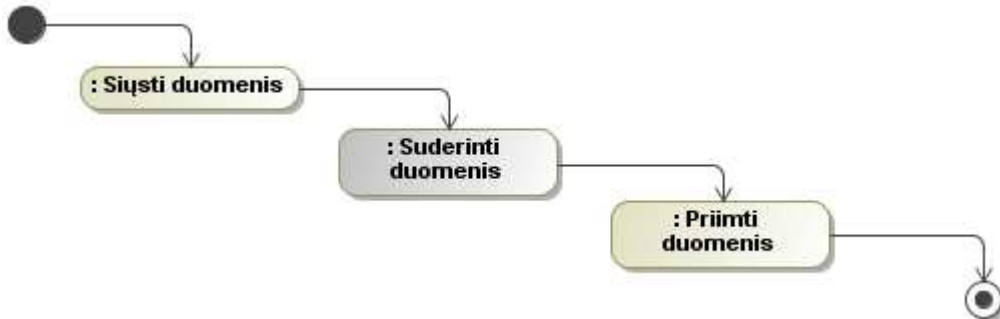
##### *Pirminis veiklos proceso modelis*



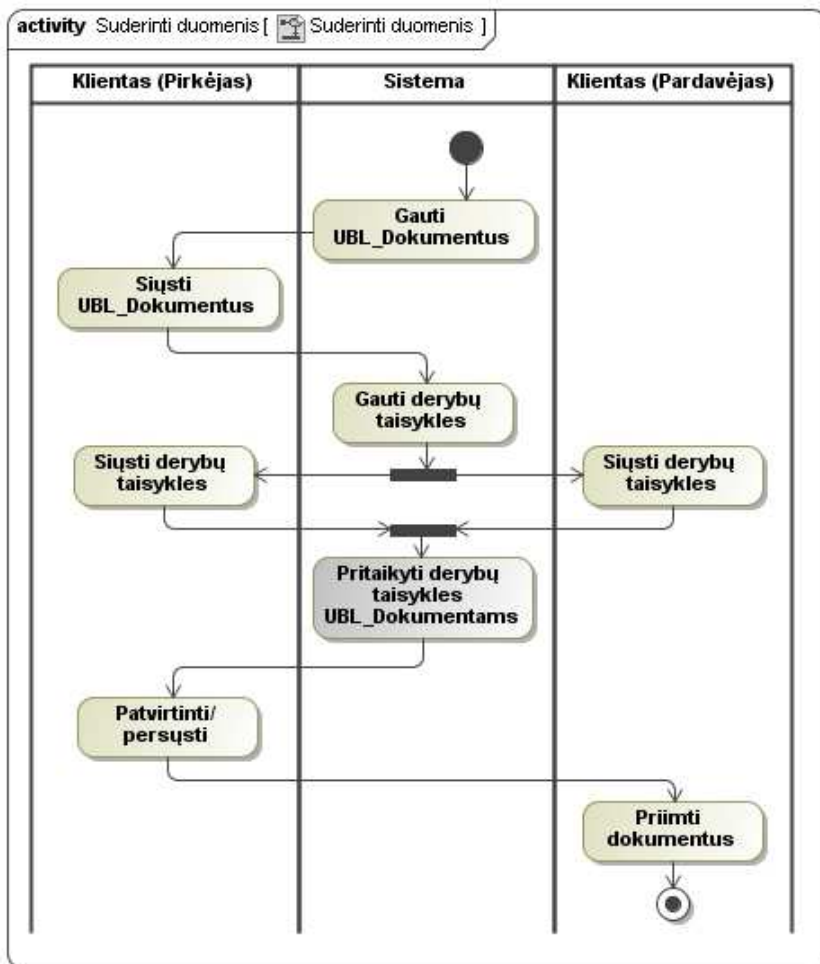
4.18 pav. Pirminis veiklos proceso modelis

### Aukščiausio, antro bei trečio lygių veiklos procesai

Aukščiausio, antro bei trečio lygių veiklos procesai – procesai, kurie yra detalizuojami. Šie procesai vaizduojami kita (tamsesne) spalva.



4.19 pav. Aukščiausio lygio veiklos procesas



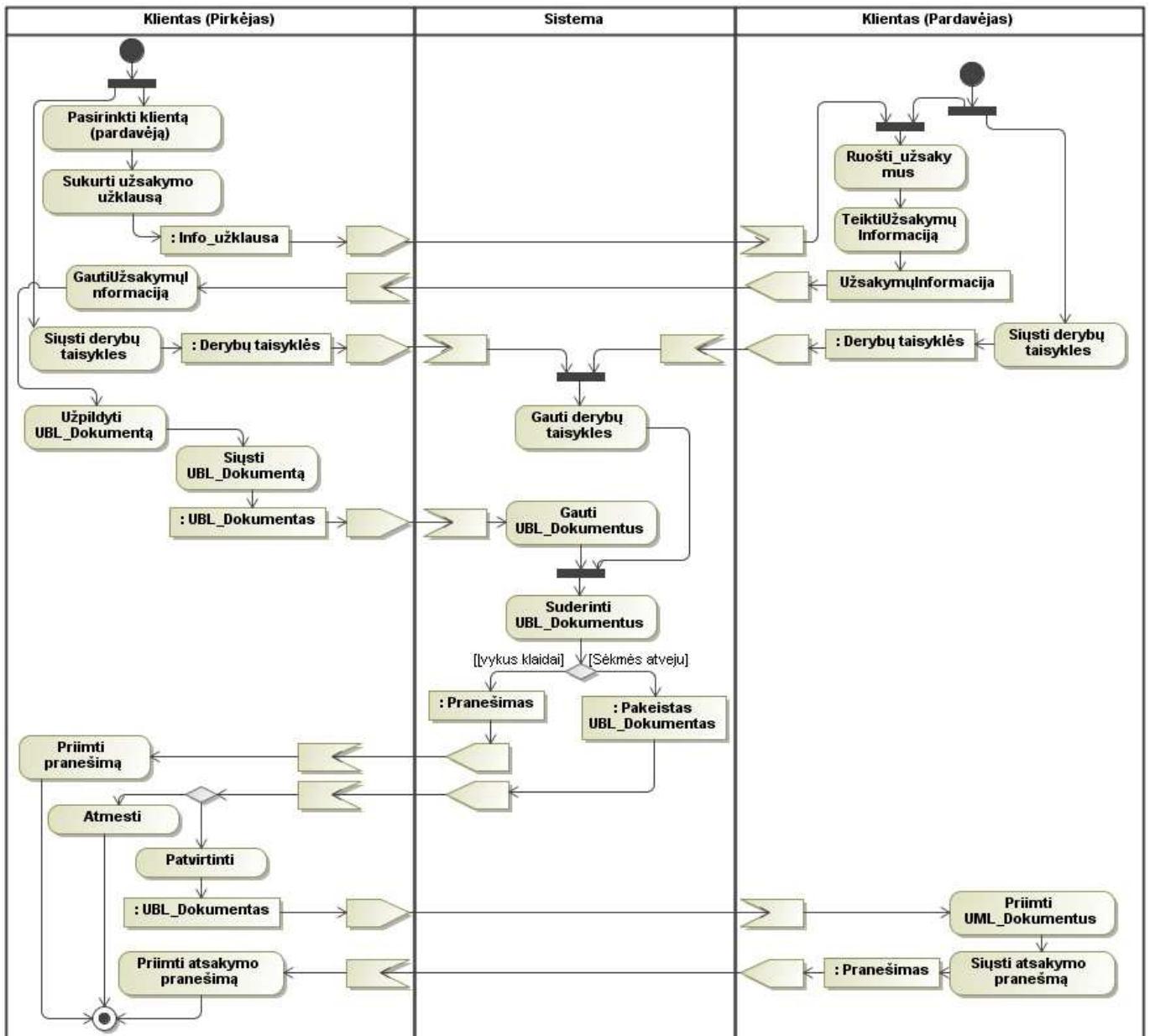
4.20 pav. Antro lygio veiklos procesas



4.21 pav. Trečio lygio veiklos procesas



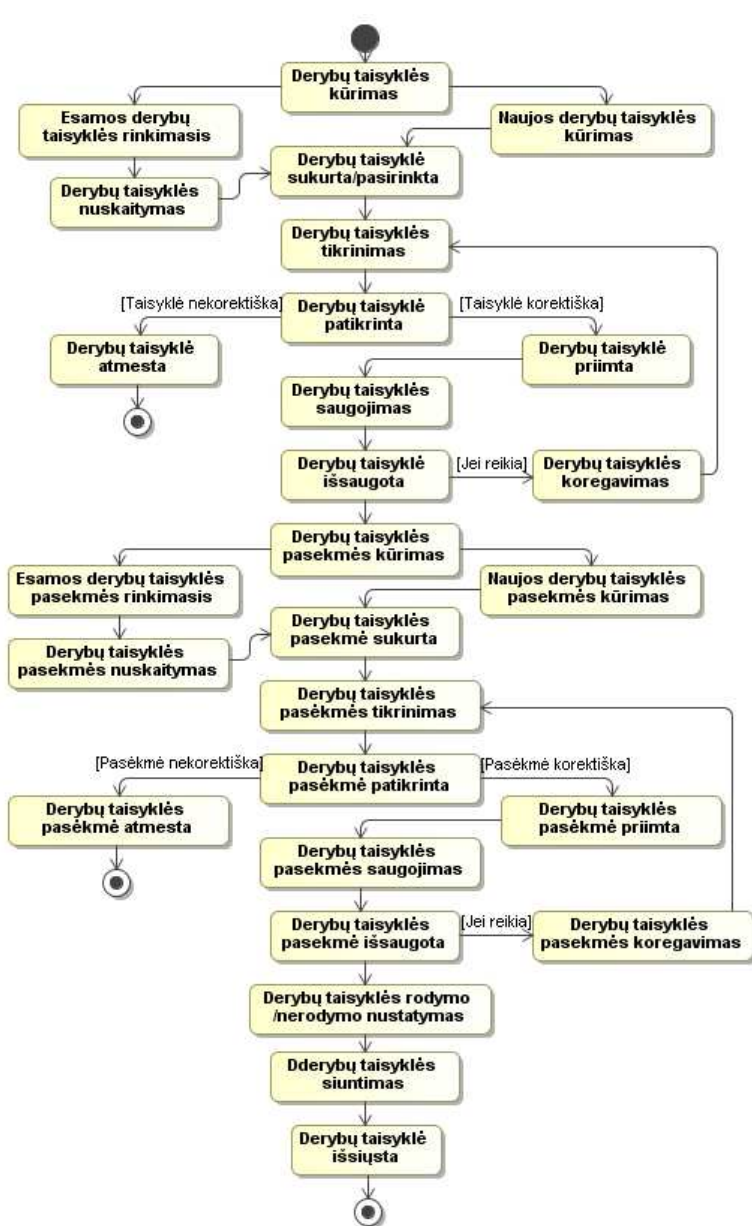
### 4.3.3. Veiklos transakcijų modelis



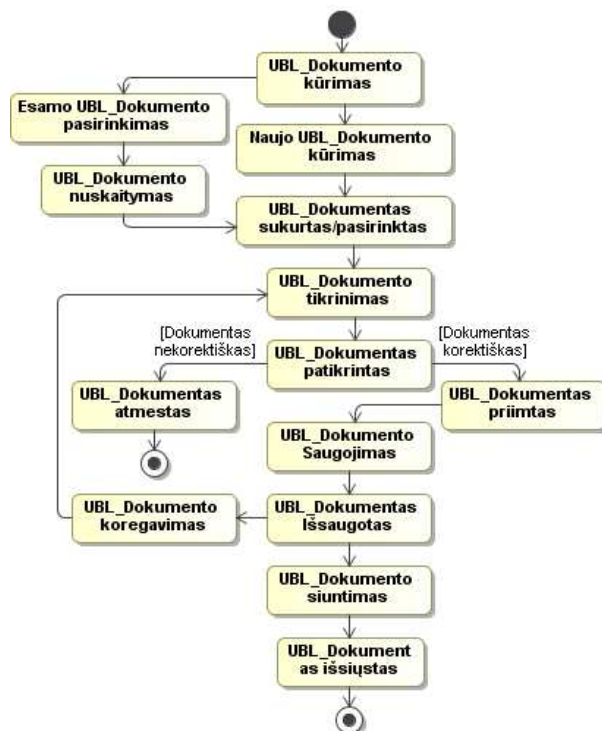
4.22 pav. Veiklos transakcijų modelis

### 4.3.4. Veiklos esybių būsenų diagramos

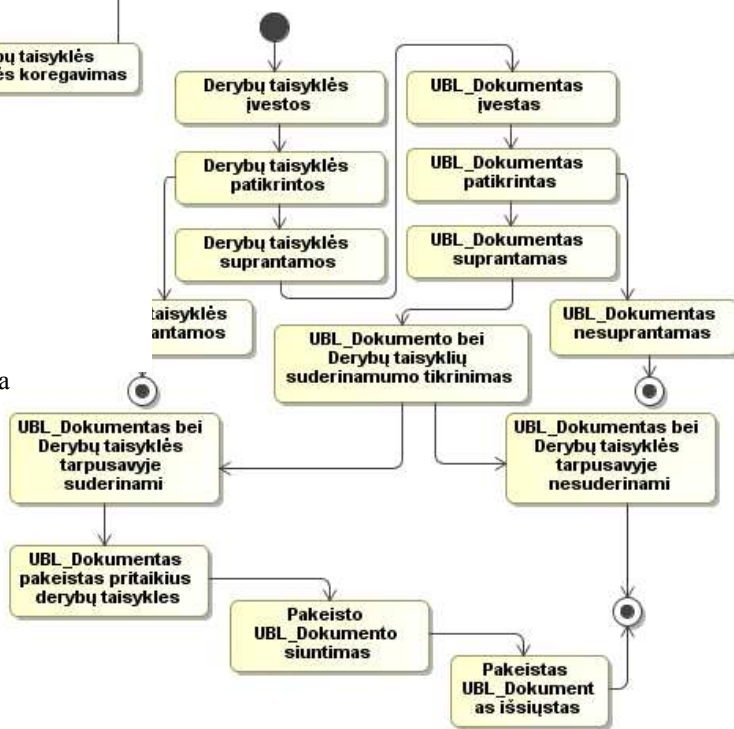
Toliau pateiktos atitinkamai 4.23, 4.24, 4.25 diagramos, nusakančios į kokias būsenas patenkame kurdami taisykles, UBL\_Dokumentus bei tikrindami/siūsdami duomenis.



4.23 pav. Derybų taisyklių kūrimo būsenų diagrama



4.24 pav. UML\_Dokumentų kūrimo būsenų diagrama



4.25 pav. Duomenų tikrinimo bei siuntimo būsenų diagrama

## 5. Realizacija

### 5.1. Verslo taisyklių suderinimo sprendimo savybių analizė bei veikimo aprašas

#### 5.1.1. Užsakymo dokumentai

Tai vienas iš užsakymo procese naudojamų UBL tipo dokumentų (UBL dokumentų tipų detalus aprašymas pateiktas 2.8.2 skyriuje „dokumentų standartai“). Užsakymo dokumentas (UBL-Order) apibūdina konkretų užsakymą įskaitant pirkėją, pardavėją, vežėją, užsakomas prekes, kainas, kiekius ir panašiai. Būsimame paveiksle (žr. 5.1 pav.) pateiktas UBL-Order dokumento grafinis vaizdas. Šiame paveiksle pateiktas dokumentas nėra pilnas pagal savo galimybes, t.y. jame naudojami tik tie laukai, be kurių būtų neįmanomas užsakymo, tarp dviejų įmonių, sudarymas (pilnas UBL-Order dokumentas turi daug papildomų laukų, kurie atliekamo tyrimo esmės atskleidimui neturi įtakos).

**Užsakymas**

Nr.  Data:

**Pirkėjas**

Pavadinimas

Gatvė

Miestas

**Pardavėjas**

Pavadinimas

Gatvė

Miestas

**Kontaktinis asmuo**

Telefonas:

El.Paštas:

**Užsakymo eilutės** Valiuta

Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL (Lt)	Suma LTL (Lt)
<input type="text"/>	<input type="text"/>	<input type="text"/>	0	Kg	21	0	0

Nuolaidos požymis

Nuolaidos/Premijos priežastis

Suma

Pridėti nuolaida

Pridėti eilutę

Iš viso	LTL (Lt)	<input type="text" value="0"/>
PVM	LTL (Lt)	<input type="text" value="0"/>
Suma su mokesčiais	LTL (Lt)	<input type="text" value="0"/>
Nuolaidų suma	LTL (Lt)	<input type="text" value="0"/>
Nuobaudų suma	LTL (Lt)	<input type="text" value="0"/>
Bendra suma	LTL (Lt)	<input type="text" value="0"/>

5.1 pav. UBL-Order dokumento grafinis vaizdas

Kaip jau minėta anksčiau, vykdant verslo taisyklių suderinimą UBL-*Order* dokumentas nebūtinai turi būti pilnas pagal savo struktūrą. Norint atskleisti tam tikrus specifinius aspektus, kai vykdant eksperimentą (jo metu pritaikant taisykles) pakinta kai kurie „laukai“, pilnai pakanka tokio dokumento turinio, kuriame aprašytos tik užsakyme dalyvaujančios „šalys“ (pirkėjas, pardavėjas) bei jų vykdomas sandoris (prekės, kiekiai, kainos ir panašiai).

Toliau aptarsime šio UBL-*Order* dokumento XML struktūrą. Toliau pateiktame paveiksle (žr.5.2 pav.) nurodomi XML elementai, kurių pagrindu suformuojamas minėtasis dokumentas, kurio grafinis vaizdas pavaizduotas 5.1 paveiksle.

```

<?xml version="1.0" encoding="UTF-8" ?>
<ns1:Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:ns1="urn:oasis:names:specification:ubl:schema:xsd:Order-2"
  <cbc:ID>1</cbc:ID>
  <cbc:IssueDate>00-00-00</cbc:IssueDate>
  <cac:BuyerCustomerParty> 1.
  <cac:SellerSupplierParty> 2.
  <cac:AllowanceCharge>
    <cbc:Amount currencyID="LTL">0.00</cbc:Amount>
    <cbc:ChargeIndicator>True</cbc:ChargeIndicator>
    <cac:TaxCategory>
      <cbc:Percent>00</cbc:Percent>
    </cac:TaxCategory>
    <cac:TaxTotal>
      <cbc:TaxAmount currencyID="LTL">0</cbc:TaxAmount>
    </cac:TaxTotal>
  </cac:AllowanceCharge>
  <cac:TaxTotal>
    <cbc:TaxAmount currencyID="LTL">0.00</cbc:TaxAmount>
  </cac:TaxTotal>
  <cac:AnticipatedMonetaryTotal>
    <cbc:TaxExclusiveAmount currencyID="LTL">0</cbc:TaxExclusiveAmount>
    <cbc:TaxInclusiveAmount currencyID="LTL">0</cbc:TaxInclusiveAmount>
    <cbc:AllowanceTotalAmount currencyID="LTL">0</cbc:AllowanceTotalAmount>
    <cbc:ChargeTotalAmount currencyID="LTL">0</cbc:ChargeTotalAmount>
    <cbc:PayableAmount currencyID="LTL">0.0</cbc:PayableAmount>
  </cac:AnticipatedMonetaryTotal>
  <cac:OrderLine>
    <cac:LineItem>
      <cbc:ID>1</cbc:ID>
      <cbc:Quantity unitCode="Kg">0</cbc:Quantity>
      <cbc:LineExtensionAmount currencyID="LTL">0</cbc:LineExtensionAmount>
      <cac:Price>
        <cbc:PriceAmount currencyID="LTL">0</cbc:PriceAmount>
      </cac:Price>
      <cac:Item>
        <cbc:Name>Pavadinimas</cbc:Name>
        <cac:CatalogueItemIdentification>
          <cbc:ID>1</cbc:ID>
        </cac:CatalogueItemIdentification>
      </cac:Item>
    </cac:LineItem>
  </cac:OrderLine> 3.
</ns1:Order>

```

5.2 pav. XML struktūra atitinkanti UBL-*Order* dokumentą

XML dokumentas visuomet prasideda įžangine sekcija `<?xml .. ?>` kurioje nurodomas dokumento tipas (XML), XML versija bei dokumento koduotė.

Toliau galima pastebėti, jog kiekvienas elementas prasideda sutrumpinimais *cac* bei *cbc*, kurie reiškia vardų erdvę (angl. *namespace*), kuriai priklauso elementai. Jie naudojami tam, kad XML bylose esančios reikšmės, su tais pačiais vardais, nebūtų interpretuojamos skirtingai. Pateiktame pavyzdyje naudojamos:

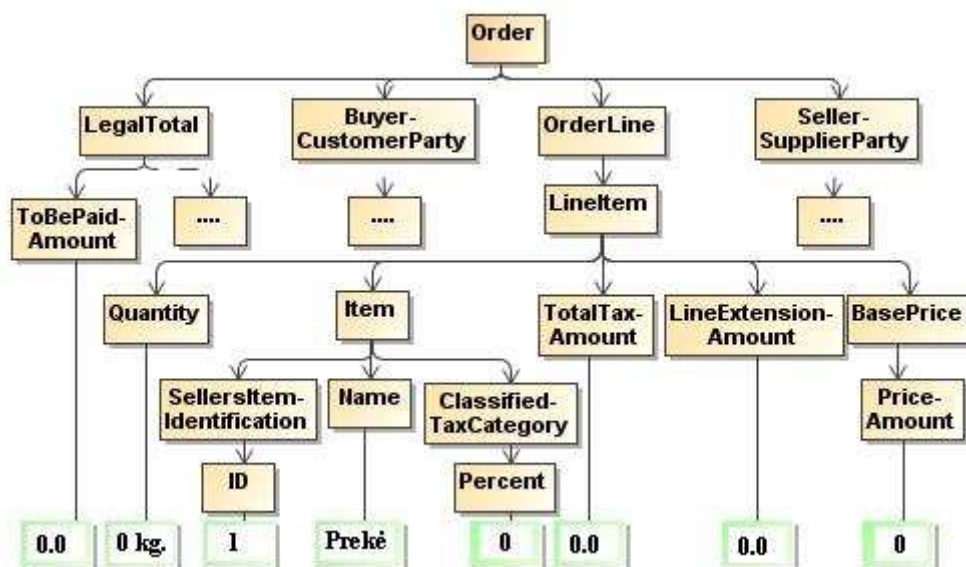
```
{ xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:ns1="urn:oasis:names:specification:ubl:schema:xsd:Order-2". }
```

vardų erdvės, todėl tarkim užrašas *cbc:IssueDate* iš tikrųjų reiškia:

```
{ urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2}IssueDate
```

Tačiau XML sintaksė neleidžia naudoti laužtinių skliaustų, todėl buvo panaudoti „priešdėliai“ reiškiantys vardų erdvę. Iš esmės šie priešdėliai gali būti bet koks raidžių derinys, tačiau dėl visuotinio suprantamumo buvo sukurti standartiniai užrašai.

Iš esmės XML dokumentas yra surūšiuotas žymėtas „medis“, kurio išraiška pasižymi tam tikromis savybėmis. Jis susideda iš mazgų, kuriuose saugomi tekstiniai duomenys (mazgai savyje gali saugoti kitus mazgus). Mazgai turi jiems priskirtą vardą bei atributus. Pabrėžtina, jog viename dokumente gali būti tik vienas šakninis (pradinis) mazgas. 5.2 nagrinėjamame pavyzdyje šakninį elementą atitinka `<Order>` elementas, o viso medžio grafinis vaizdas gali būti atvaizduojamas taip (žr.5.3 pav.):



5.3 pav. XML dokumento konceptualus medis

Atskiri XML struktūros elementai aprašo duomenis, kurie pateikiami UBL dokumente (žr. 5.2 pav.). Iš esmės, pateiktą pavyzdį galime suskirstyti į keturias dalis. Pirmu ir antru numeriu pažymėtos dalys apibūdina užsakymą vykdančius klientus (juos aptarsime vėliau), trečiu numeriu pažymėta vieta aprašo konkrečias užsakymo eilutes (gaminio pav. kiekį, kainą ir panašiai), o visa kas nėra pažymėta – bendrą struktūrą, pagalbinius laukus (tokius kaip data, bendra kaina ir t.t.).

Taigi, kaip jau buvo minėta anksčiau, pirmu ir antru numeriu, 5.2 paveiksle, pažymėti laukai atitinka informaciją, kuri susijusi su užsakymo vykdytojais, t.y. pirkėju (angl. *Buyer*) bei pardavėju (angl. *Seller*). Toliau pateiktuose paveiksluose (žr. 5.4 ir 5.5 pav.) matome šių elementų XML struktūrą. Matysime, jog visi elementai yra identiški išskyrus pradines antraštes. Taip yra todėl, kad aprašinėjant pirkėją bei pardavėją, naudojamas tas pats elementas „*Party*“. Šis elementas taip pat gali būti naudojamas aprašinėjant kitas, procese dalyvaujančias, šalis tokias kaip vežėjas, tiekėjas ir panašiai.

```
<cac:SellerSupplierParty>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>Įmonės pav.</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Gatvė</cbc:StreetName>
      <cbc:CityName>Miestas</cbc:CityName>
    </cac:PostalAddress>
    <cac:Contact>
      <cbc:Name>Vardas Pavardė</cbc:Name>
      <cbc:Telephone>+370 000000000</cbc:Telephone>
      <cbc:ElectronicMail>pas@tas.com</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:SellerSupplierParty>
```

5.4 pav. XML struktūra atitinkanti „pardavėją“

```
<cac:BuyerCustomerParty>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>Įmonės pav.</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Gatvė</cbc:StreetName>
      <cbc:CityName>Miestas</cbc:CityName>
    </cac:PostalAddress>
    <cac:Contact>
      <cbc:Name>Vardas Pavardė</cbc:Name>
      <cbc:Telephone>+370 000000000</cbc:Telephone>
      <cbc:ElectronicMail>pas@tas.com</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:BuyerCustomerParty>
```

5.5 pav. XML struktūra atitinkanti „pirkėją“

Šie laukai labai svarbūs, kadangi pagal juos identifikuojamos bei, kaip ir užsakymo eilučių atveju, sudarinėjamos pirkėjo ir pardavėjo reglamentuotos verslo taisyklės.

### 5.1.2. Taisyklės (formalizavimas, sukūrimas)

Šioje dalyje bus nagrinėjamos konkrečios taisyklės su savo struktūra, formalia išraiška bei praktiniu pritaikomumu.

Taigi verslo taisyklės – tai taisyklės atitinkančios struktūrą „Jei – Tai“. Ši struktūra, pagal savo prasmę, labai artima loginei operacijai – implikacijai, todėl tokias taisykles galima žymėti  $\beta \rightarrow \alpha$ . Kadangi šių taisyklių panaudojimo sąlyga, bendru atveju, gali būti ne viena, todėl bendrą išvadą galime išreikšti:

$$\alpha_1 \ \&\alpha_2 \ \&\alpha_3 \ \&\dots\&\alpha_n \rightarrow \beta$$

Čia  $\alpha_i$  ( $i=1 \dots n$ ) – verslo taisyklės panaudojimo sąlygos, jungiamos konjunkcijos operacija.  $\beta$  – išvada arba veiksmas, atliekamas, kai tenkinama taisyklės sąlyginė dalis [20].

Taigi galime daryti išvadą, jog šios taisyklės atitinką konstrukciją „Sąlyga-Išvada“ arba „Priežastis-Pasekmė“. Tokios konstrukcijos verslo taisyklių, sąveikaujant dviem ar daugiau įmonių, gali būti labai daug. Vienas iš pavyzdžių:

1 Pvz.:           „Jei perkamų bulvių kiekis daugiau kaip 10 kilogramu,  
                          tai bulvėms taikoma 15 litų nuolaida“

Šis pavyzdys pateiktas jį užrašant literatūrine kalba. Tai padaryti nėra labai sunku, tačiau kuriant informacinę sistemą taip daryti neišeina. Taip yra todėl, kad kompiuteriai dar nesupranta tokio užrašymo. Norint sukurti ir užrašyti taisyklę, kurią galėtų apdoroti informacinės sistemos, reikia jas formalizuoti.

Taigi pirmiausia išnagrinėkime bendrąsias taisyklės sudedamąsias dalis (jos struktūrą). Iš pirmo pavyzdžio matome, jog paprastai taisyklė susideda iš sąlygos ir rezultato. Sąlyga nusako priežastis, o rezultatas – pasekmes.

Kaip jau buvo minėta anksčiau, kiekviena sąlyga prasideda sąlygos žodeliu, dažniausiai „Jei“, o rezultatų dalį pradeda žodelis „Tai“. Pasinaudodami formalia notacija, taisyklę galime išreikšti atitinkama išraiška:

2 Pvz.:           < Verslo taisyklė > →  
                          „Jei“ < Sąlyga > „Tai“ < Rezultatas >

Pirmiausia detalai panagrinėkime sąlyginę dalį. Ji gali susidėti iš įvairių sudedamųjų dalių (elementų) priklausomai nuo dalykinės srities ir taisyklės specifikos. Be kita ko, gali atsitikti ir taip, jog norint tiksliai įvertinti vieną ar kitą parametą, reikia panaudoti ne vieną, o kelias sąlygines dalis. Todėl bendra forma, sąlygą galime išreikšti taip:

3 Pav.:           < Sąlyga > →  
                          ({< Sąlygos\_e1 > < Log\_op > } < Sąlygos\_e1 >)

Paprastieji skliaustai „( )“ šiuo atveju nurodo, kad sąlyginė dalis gali būti sudėtinė, t.y. turėti keletą skirtingų elementų, ir visi jie, priklausys vienai ir tai pačiai taisyklei. Tuo tarpu riestiniai skliaustai „{ }“ pažymi, jog turinio, esančio juose gali visai nebūti arba jis gali pasikartoti keletą kartų.

Nurodytasis loginis operatorius (*Log\_op*) kontekste gali įgyti tik tris reikšmes: „IR“; „ARBA“; „NE“. Tuo tarpu sąlygos elementas (*Sąlygos\_el*) apibūdina konkrečią vieną sąlygą su savo reikšmėmis, lyginimo operacijomis ar mato vienetais.

4 Pav.:  $\langle \text{Sąlygos\_el} \rangle \rightarrow$   
 $\langle \text{Reikšmė} \rangle \langle \text{Lyg\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle]$

Sąlygos elemento kintamasis „*Reikšmė*“ yra parametras, pagal kurį lyginama (atsižvelgiant į šią reikšmę priimami sprendimai). Ši reikšmė dažniausiai atitinka kontekstinio dokumento vieną iš parametrų (dažniausiai *string* tipo). Pateiktame, pirmame, pavyzdyje „*reikšmės*“ dalį atitiko „perkamos bulvės“.

Norint kažką su kažkuo palyginti reikalingas lyginimo operatorius (*Lyg\_op*). Atsižvelgiant į tam tikras aplinkybes gali būti panaudoti šešių tipų operatoriai: „*Nelygu*“ (!=); „*Lygu*“ (=); „*Daugiau arba lygu*“ (>=); „*Daugiau*“ (>); „*Mažiau arba lygu*“ (<=); „*Mažiau*“ (<). Pateiktame pavyzdyje buvo naudojamas operatorius „*Daugiau*“.

Turėdami ką lyginti ir žinodami kaip reikia palyginti, toliau turime nustatyti su kuo reikia palyginti. Šią dalį pažymėjome „*Reikšmė\_2*“. Paprastai ji gali būti imama iš kontekstinio dokumento arba tiesiog sugalvota taisyklės sudarinėtojo (*string* arba *integer* tipo). Pavyzdyje ši reikšmė atitiko „10“.

Galiausiai belieka apibrėžti mato vienetus (*Mato\_vnt.*), jeigu tokie reikalingi. Laužtiniai skliaustai „[ ]“ ketvirtame pavyzdyje nusako, kad informacija juose gali būti arba nebūti.

Apjungę trečią ir ketvirtą pavyzdžius gauname taisyklės sąlyginę dalį kuri formaliai atrodo taip:

5 Pvz.:  $\langle \text{Sąlyga} \rangle \rightarrow$   
 $\langle (\{ \langle \text{Reikšmė} \rangle \langle \text{Lyg\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle] \langle \text{Log\_op} \rangle \} \langle \text{Reikšmė} \rangle \langle \text{Lyg\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle]) \rangle$



Toliau seka taisyklės rezultatų dalis. Ji nusako ką reikia daryti (kokius veiksmus ar procesus atlikti) jei sąlyginė taisyklės dalis buvo tenkinama. Ji savo struktūra panaši į sąlyginę dalį. Jeigu sąlyga prasidėjo žodeliu „*Jei*“ tai rezultatų dalis prasideda žodeliu „*Tai*“.

Jeigu sąlyginės dalys galėjo būti kelios, taip ir rezultatų dalis gali būti ne viena, t.y. atitinkant kažkokiai sąlygai gali tekti pakeisti (modifikuoti) ne vieną, o kelis parametrus iškart. Taigi formaliai gauname tokia išraišką:

6 Pav.:  $\langle \text{Rezultatas} \rangle \rightarrow$   
 $(\{\langle \text{Rezultato\_el} \rangle \langle \text{Log\_op} \rangle\} \langle \text{Rezultato\_el} \rangle)$

Šiuo atveju rezultato elementas (*Rezultato\_el*) nusako konkrečius veiksmus, kuriuos reikia atlikti su konkrečiomis reikšmėmis.

7 pav.:  $\langle \text{Rezultato\_el} \rangle \rightarrow$   
 $\langle \text{Reikšmė} \rangle \langle = \rangle \langle \text{Reikšmė} \rangle \langle \text{Ar\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle]$

„*Reikšmė*“, pastaruoju atveju reiškia nebe tai pagal ką lyginsime, o tai ką keisime (koki parametą ar reikšmę modifikuosime). Ši reikšmė, kaip ir sąlygos dalyje, atitinka kontekstinio dokumento vieną iš parametų (dažniausiai *string* tipo).

Priešingai negu taisyklės sąlyginėje dalyje, rezultatų dalis turi aritmetinių operatorių (*Ar\_op*). Ji nusako kokius būtent pakeitimus reikia atlikti su nurodytomis reikšmėmis. Šis operatorius gali įgauti vieną iš keturių reikšmių „*Atimtį*“ (-); „*Sudėtį*“ (+); „*Daugybą*“ (\*) bei „*Dalybą*“ (/). Pateiktame pirmame pavyzdyje buvo naudojamas sumažinimo loginis operatorius reiškiantis atimtį.

Toliau vėl seka reikšmė (*Reikšmė\_2*), kuria reikia modifikuoti buvusį elementą bei mato vienetai (jeigu tokie reikalingi) patikslinantys naudojamų parametų reikšmes.

Galiausiai gauname tokia taisyklės rezultato formalią struktūrą:

8 Pvz.:  $\langle \text{Rezultatas} \rangle \rightarrow$   
 $\langle (\{\langle \text{Reikšmė} \rangle \langle = \rangle \langle \text{Reikšmė} \rangle \langle \text{Ar\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle]$   
 $\langle \text{Log\_op} \rangle\}$   
 $\langle \text{Reikšmė} \rangle \langle = \rangle \langle \text{Reikšmė} \rangle \langle \text{Ar\_op} \rangle \langle \text{Reikšmė\_2} \rangle [\langle \text{Mato\_vnt.} \rangle]) \rangle$

Apjungę penktą bei aštuntą pavyzdžius į vieną, gauname užrašytą verslo taisyklės formalia išraišką:

9 Pvz.: < Verslo taisyklė > ➔

„Jei“ < ({<Reikšmė>< Lyg\_op>< Reikšmė\_2> [<Mato\_vnt.>]

< Log\_op>}>

< Reikšmė>< Lyg\_op>< Reikšmė\_2> [<Mato\_vnt.>])>

„Tai“ < ({<Reikšmė><=><Reikšmė>< Ar\_op>< Reikšmė\_2> [<Mato\_vnt.>]

< Log\_op>}>

< Reikšmė ><=><Reikšmė>< Ar\_op>< Reikšmė\_2> [<Mato\_vnt.>])>

Paprastumo bei vaizdumo dėlei, stengiantis taisyklės išraišką padaryti kompaktiškesnę, riestinius skliaustus „{ }“ bei viską kas juose, pažymėsime sutartiniu ženklų  $\{\pm\}$  ir jį naudosisime vietoj viso reiškinių. Tuomet verslo taisyklė atrodo taip:

10 Pvz.: < Verslo taisyklė > ➔

„Jei“ < ( $\{\pm\}$ )>< Reikšmė>< Lyg\_op>< Reikšmė\_2> [<Mato\_vnt.>]>

„Tai“ < ( $\{\pm\}$ )><Reikšmė><=><Reikšmė><Ar\_op><Reikšmė\_2> [<Mato\_vnt.>])>

O pirmame pavyzdyje pateikta taisyklė, užrašyta literatūrine kalba, formaliai atrodo taip:

11 Pvz.: Konkreti taisyklė

*Jei prekė = bulvės IR prekės\_kiekis > 10 Kg.*

*Tai prekės\_kaina = prekės\_kaina - 15 Lt*

Dešimtame pavyzdyje gauta išraiška yra universali ir praktiškai tinkanti užrašyti visoms verslo taisyklėms, kurių tik gali prireikti sąveikaujant dviem ar daugiau įmonių.

Tam, kad taisyklės būtų įgyvendinamos, t.y. būtų galima jas pritaikyti, reikia konteksto, kuriam jos tiktų. Priešingu atveju reglamentuotos taisyklės negalėtų būti vykdomos, kadangi nebūtų duomenų, kurios suaktyvintų taisyklės sąlyginę ar rezultatų dalį. Nagrinėjamoju atveju kontekstą nusako UBL-Order dokumentai. Visos lyginamosios bei modifikuojamos taisyklių reikšmės atitinka tam tikrus dokumento laukus (parametrus). Taigi sudarinėjant naują taisyklę, atitinkamos reikšmės sukuriamos ne atsitiktinai, o parenkant vieną iš dokumente esamų laukų. Kaip pavyzdys tai galėtų būti 5.6 paveiksle pateiktos reikšmės.

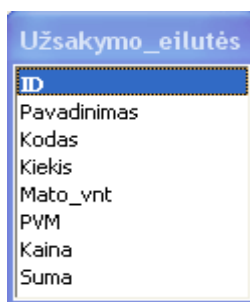
Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL (Lt)	Suma LTL (Lt)
			0	Kg	21	0	0

5.6 pav. Taisyklėse naudojamos reikšmės UBL-Order dokumente

### 5.1.3. Vartotojų problemos

Norint paimti reikšmes iš 5.6 paveiksle pavaizduoto dokumento fragmento, susiduriama su naujomis problemomis. Reikia nustatyti būdus bei priemones, nusakančias kaip tai padaryti.

Vienas iš sprendimo variantų galėtų būti duomenų bazės panaudojimas išsaugant UBL dokumentus. Sukurti lenteles, kuriose būtų indeksuojamos visos reikšmės. Tokiu atveju, sudarinėjant naują taisyklę pakaktų nurodyti lentelių bei atributų indeksus, taip priskiriant vieną ar kitą reikšmę. Pateikto 5.6 paveiksle fragmento duomenų bazės lentelė atrodytų kaip parodyta 5.7 paveiksle.



ID
Pavadinimas
Kodas
Kiekis
Mato_vnt
PVM
Kaina
Suma

5.7 pav. UBL-Order dokumento fragmento „Užsakymo\_eilutės“ duomenų bazės lentelė  
Tuomet 11 pavyzdyje pateiktą formalizuotą taisyklę galime užrašyti taip:

12 Pvz.

*Jie* **Select Užsakymo\_eilutės.Kiekis > 10 Select Užsakymo\_eilutės.Mato\_vnt**

*Tai* **Select Užsakymo\_eilutės.Kaina = Select Užsakymo\_eilutės.Kaina - 15 Select  
Užsakymo\_eilutės.Mato\_vnt**

Matome, jog toks sprendimo variantas, kai naudojama duomenų bazė, yra įgyvendinamas, tačiau iškyla naujas klausimas dėl jo racionalumo. Ar tikrai tai geriausias sprendimas? Pirmiausia atsižvelgiama į faktą, jog naudojamas UBL standarto dokumentas, kuris realizuotas panaudojant XML struktūrą. Iš to seka, kad tokį dokumentą nėra prasmės skaidyti į atskiras dalis bei juo labiau suskirstyti į skirtingas lenteles. XML pagrindu sukurtų dokumentų turinys, pagal savo prigimtį, yra saugomas kaip struktūrizuota tekstinė informacija, todėl nėra prasmės kurti naujų jo saugojimo variantų.

Kitas, racionalesnis būdas paimti reikšmes iš UBL dokumento yra XML struktūros elementų naudojimas. Pateiktame pavyzdyje (žr. 5.8 pav.) matomas UBL dokumento XML struktūros fragmentas atitinkantis pirkėjo duomenis UBL dokumente.

```

<cac:BuyerCustomerParty>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>Įmonės pav.</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Gatvė</cbc:StreetName>
      <cbc:CityName>Miestas</cbc:CityName>
    </cac:PostalAddress>
    <cac:Contact>
      <cbc:Name>Vardas Pavardė</cbc:Name>
      <cbc:Telephone>+370 000000000</cbc:Telephone>
      <cbc:ElectronicMail>pas@tas.com</cbc:ElectronicMail>
    </cac:Contact>
  </cac:Party>
</cac:BuyerCustomerParty>

```

5.8 pav. UBL dokumento XLM struktūros fragmentas

Formuojant verslo taisykles, reikšmės, atitinkančios UBL dokumento reikšmes, paimamos iš XML struktūros, atrenkant atitinkamus elementus. Kadangi XML struktūra pagrįsta „medžio“ principu, tai kiekvienas elementas turi savo „kelia“. Taigi parenkant konkretų elementą yra nurodomas „kelias“ iki jo, skirtingus „adreso“ elementus atskiriant dvitaškiu. Pavyzdžiui:

13 Pvz.      *BuyerCustomerParty:Party:Address:StreetName*; arba  
                  *BuyerCustomerParty:Party:PartyLegalEntity:CompanyID*

Pritaikius šią logiką 11 pavyzdžiui jos išraiška dabar atrodys taip:

14 Pvz.

***Jei OrderLine:LineItem:Quantity > 10 OrderLine:LineItem:Quantity unitCode***

***Tai OrderLine:BasePrice:PriceAmount = OrderLine:BasePrice:PriceAmount - 15***

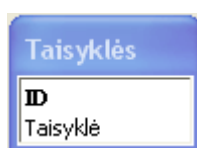
***OrderLine:LineItem:Quantity unitCode***

Viena iš didžiausių šio metodo problemų yra tai, kad nurodant „kelia“ iki konkretaus elemento, susidaro gana ilga ir sunkiai suprantama nuoroda. Klientui, pildančiam tokio tipo taisyklę, gali būti labai nepatogu, o svarbiausia neaišku, kaip tai reikia daryti. Šiai problemai spręsti galima panaudoti elementų kelių konvertavimą. Tai reiškia, kad kuriant informacinę taisyklę, kiekvieno elemento keliui priskirtas sutrumpinimas, apibūdinantis naudojamos

reikšmės esmę. Pavyzdžiui tryliktame pavyzdyje pateiktoms nuorodomis galima priskirti konkrečius pavadinimus:

15 P.vz. *BuyerCustomerParty:Party:Address:StreetName* = “Pirkėjo\_gatvė”  
 arba *BuyerCustomerParty:Party:PartyLegalEntity:CompanyID* = “Pirkėjo\_Im.kodas”

Sukūrus naują verslo taisyklę belieka ją išsaugoti. Iš pirmo žvilgsnio atrodo gana nesudėtingas uždavinys. Belieka sukurti duomenų bazės vieną lentelę, kurioje būtų saugoma taisyklė tekstiniu formatu (žr. 5.9 pav.).



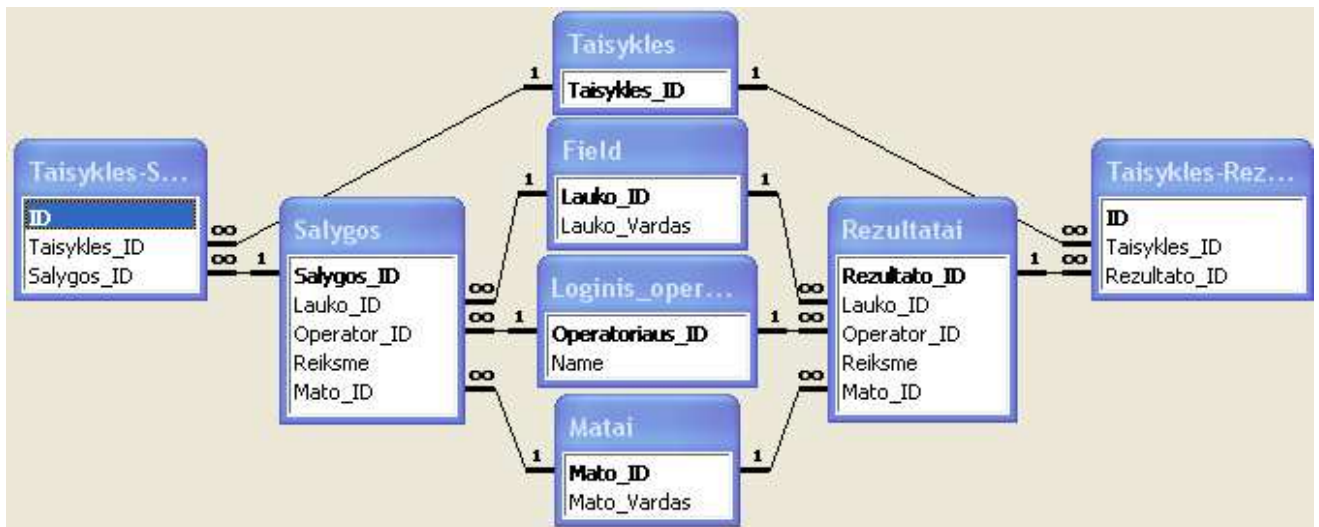
5.9 pav. Taisyklės saugojimo duomenų bazės lentelė su vienu įrašu

Akivaizdu, jog šitoks saugojimo būdas netinka, kadangi informacinei sistemai praktiškai neįmanoma nustatyti taisyklės sąlyginės ir rezultatų dalių. Iš to seka, kad norint išsaugoti taisyklę, reikia turėti bent tris duomenų bazės lenteles. Viena jų nusakytų taisyklę, kitos dvi apibūdintų jos sudedamąsias dalis (sąlyginę bei rezultatų dalį). Pateiktame paveiksle (žr. 5.10 pav.) matomas tokio sprendimo grafinis vaizdas.



5.10 pav. Taisyklės saugojimo duomenų bazės lentelėse struktūra išskyrus sąlygą ir rezultatą

Pateiktos, duomenų bazės, struktūros taip pat dar neužtenka norint išsaugoti, o svarbiausia efektyviai nuskaityti ir panaudoti taisyklėmis. Informacinė sistema, nuskaitinėdama išsaugotas taisykles, nesugebės tinkamai interpretuoti išskirtų sąlyginės bei rezultatų dalių. Minėtas dalis reikia taip pat išskaidyti. Taip atsiranda dar penkios lentelės, reikalingos reikšmių bei loginių operatorių saugojimui. Galiausiai visas lenteles apjungę į vieną bendrą struktūrą, gauname duomenų bazę (žr. 5.11 pav.), kurios pagalba galima efektyviai operuoti reglamentuotomis taisyklėmis.



5.11 pav. Taisyklės saugojimo duomenų bazės lentelėse struktūra

Taigi norint sukurti informacinę sistemą, kuri naudotų verslo taisykles, pirmiausia tenka apibrėžti kaip jos „atrodo“ ir kokias funkcijas atlieka. Kai jau turime konkrečią taisyklę, ją formalizuojame. Taip daroma tam, kad galėtume išsaugoti ir po to efektyviai panaudoti. Sukurtų taisyklių realus pritaikymas (nagrinėjamu atveju dokumentų reikšmių pakeitimas) yra taip pat gana sudėtingas uždavinys, todėl rasti optimaliausią variantą kaip tai tinkamai padaryti nėra lengva.

#### 5.1.4. Taisyklių pritaikymas

Norint pritaikyti sukurtas verslo taisykles konkreitiems dokumentams, pirmiausia reikia pasirinkti kurios iš jų yra reikalingos, kadangi duomenų bazėje jų gali būti labai daug. Jeigu visos jos būtų saugomos atskirai, t.y. pas konkrečius klientus, tai užtektų nurodyti tik tų klientų identifikavimo duomenis. Tačiau taisykles saugoti pas klientus yra neracionalu, nes tokiu atveju kiekviena jų turėtų turėti tam tikras informacines sistemas.

Praktiškesnis būdas yra saugoti taisykles bendroje duomenų bazėje, kuri patalpinta tam tikroje, nutolusioje informacinėje sistemoje. Tokiu atveju kiekvienas klientas turi savo identifikavimo numerį, pagal kurį atrenkamos jam priklausančios taisyklės. Šį numerį patys klientai nurodo pildydami UBL dokumentus.

Taigi informacinė taisyklių pritaikymo programa, norėdama atlikti tam tikrus veiksmus, susijusius su verslo taisyklėmis, pirmiausia turi nusiskaityti bei tinkamai interpretuoti UBL dokumento XML struktūros elementų turinį.

Kaip jau buvo minėta, pirmiausia nustatomas klientų (siuntėjo bei gavėjo) identifikavimo numeriai. Pagal šiuos numerius atrenkamos jiems priklausančios verslo

taisyklės. Sekančiame, 5.12 paveiksle, pateiktas XML elementas, pagal kurį atliekama identifikacija.

```
<cac:BuyerCustomerParty>
  <cac:Party>
    <cac:PartyLegalEntity>
      <cbc:CompanyID>LT 236589</cbc:CompanyID>
    </cac:PartyLegalEntity>
  </cac:Party>
</cac:BuyerCustomerParty>
<cac:SellerSupplierParty>
  <cac:Party>
    <cac:PartyLegalEntity>
      <cbc:CompanyID>LT 659845</cbc:CompanyID>
    </cac:PartyLegalEntity>
  </cac:Party>
</cac:SellerSupplierParty>
```

5.12 pav. XML elementai apibūdinantys pirkėjo bei pardavėjo identifikavimo numerius

Toliau seka, tenkinančių tam tikras sąlygas, taisyklių paieška. Paeiliui tikrinamos visos rastos taisyklės ir jeigu jos tinka konkrečiai situacijai, dedamos į sąrašą. Konkreti situacija reiškia, kad taisyklės turi būti susijusios tiek su pirkėju tiek su pardavėju, nes tarp visų užrašytų taisyklių yra tokių, kurios skirtos tam tikroms atskiroms įmonėms ar situacijoms. Pavyzdžiui 16 pavyzdyje pateikta taisyklė tinka tik įmonei „UAB Giraitė“ (šią taisyklę užrašė pardavėjas). O jeigu dokumentą siunčianti įmonė (pirkėjo įmonė) nėra minėtoji, tai tokia taisyklė bus praleista ir nevykdoma.

16 Pvz.: Taisyklė parašyta pardavėjo

Jei **BuyerCustomerParty:Name** = **UAB Giraitė**

Tai **OrderLine:LineItem:TotalPaidAmount** = **OrderLine:LineItem:TotalPaidAmount** - 10 %

Žinant visas reikalingas taisykles galima pereiti prie kito etapo, tai yra jų vykdymo. Šio etapo metu tikrinama kiekviena atrinkta taisyklė ir atsižvelgiant į jos paskirtį modifikuojamas dokumentas (tam tikros jo reikšmės). Tarkime turime 16 pavyzdyje pateiktą taisyklę ir 5.13 paveiksle pateiktą XML dokumento (dokumento fragmento) struktūrą.

```

<cac:BuyerCustomerParty>
  <cac:Party>
    <cac:PartyName>
      <cbc:Name>UAB Giraitė</cbc:Name>
    </cac:PartyName>
  </cac:Party>
</cac:BuyerCustomerParty>
-----
<cac:SellerSupplierParty>
  <cac:OrderLine>
    <cac:LineItem>
      <cbc:ID>1</cbc:ID>
      <cbc:Quantity unitCode="KG">5</cbc:Quantity>
      <cbc:LineExtensionAmount currencyID="LTL">15</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="LTL">2.7</cbc:TotalTaxAmount>
      <cac:BasePrice>
        <cbc:PriceAmount currencyID="LTL">3</cbc:PriceAmount>
      </cac:BasePrice>
      <cac:Item>
        <cbc:Name>Bulves</cbc:Name>
        <cac: SellersItemIdentification>
          <cbc:ID>1111</cbc:ID>
          <cac:ClassifiedTaxCategory>
            <cbc:Percent>18</cbc:Percent>
          </cac:ClassifiedTaxCategory>
        </cac: SellersItemIdentification>
      </cac:Item>
      <cbc:ToBePaidAmount currencyID="LTL">17.7</cbc:ToBePaidAmount>
    </cac:LineItem>
  </cac:OrderLine>
</cac:SellerSupplierParty>

```

5.13 pav. UBL-Order dokumento fragmento XML struktūra

Matome, jog pateikta taisyklė tikrina sąlygą ar pirkėjo įmonės pavadinimas yra „UAB Giraitė“. Tuo tarpu pateiktame XML dokumento struktūros fragmente matome, kad būtent taip ir yra (reikšmė apibrėžta). Iš to seka, jog taisyklės sąlyginė dalis yra tenkinama. Taigi toliau pereinam prie taisyklės rezultato dalies („*Tai*“) bei pradėdami ją įgyvendinti. Matome, jog ši taisyklės dalis nurodo bendrą kainą, kurią turi sumokėti pirkėjas, reikia sumažinti dešimčia procentų. Minėtoji kaina pavyzdyje yra lygi „17.7“ (reikšmė taip pat apibrėžta).

Pritaikius minėtą taisyklę dokumento galutinės sumos reikšmė turėtų sumažėti 10 procentų (galutinė suma turi būti 15.93). Kitame XML elemento pavyzdyje, kuris parodo pakeistą bendrąją kainą, tai ir matome (žr. 5.14 pav.).

```

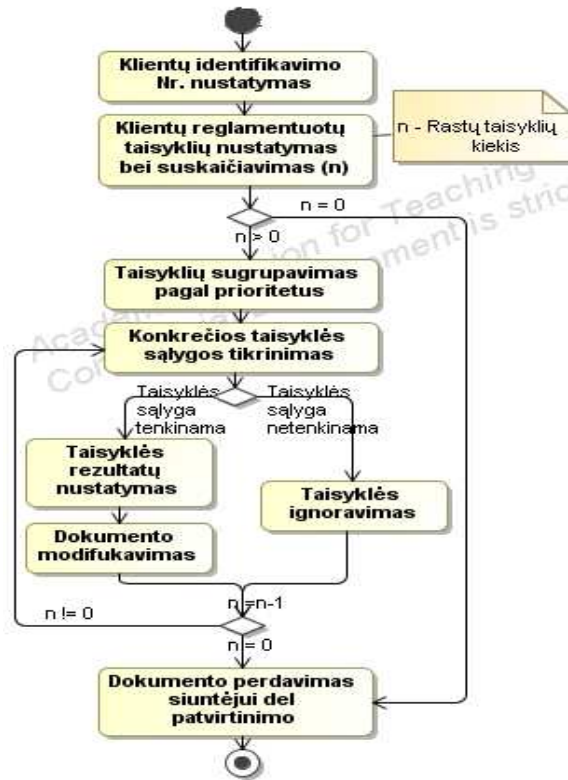
<cbc:ToBePaidAmount currencyID="LTL">15.93</cbc:ToBePaidAmount>

```

5.14 pav. UBL-Order dokumento bendros kainos XML elementas

Tokiu principu nagrinėjama ir vykdoma (jeigu ją įmanoma įvykdyti) kiekviena klientų reglamentuota verslo taisyklė. Nagrinėjamasis pavyzdys buvo labai paprastas, o be to tik vienas. Toliau pateikiamas algoritmas (žr. 5.15 pav.) grafiškai demonstruojantis taisyklių pritaikymo procesą.





5.15 pav. Taisyklių pritaikymo algoritmas

Realiose informacinėse sistemose taisyklių gali pasitaikyti labai daug ir sudėtingesnių. Tuomet iškyla nauja problema susijusi su jų tarpusavio suderinimu.

### 5.1.5. Taisyklių nesuderinamumas

Kaip jau buvo minėta 4.1.3 skyriuje, klientams suvedus savo reglamentuotas verslo taisykles atsiranda didelė problema susijusi su jų apdorojimu bei tarpusavio derinimu. Dažniausiai taip atsitinka dėl dviejų priežasčių:

- 1) Vartotojų „išsiblašymo“ sudarinėjant taisykles.
- 2) Taisyklės sąlyginės ir/arba rezultato dalies nesuderinamumo su kitomis taisyklėmis.

Pirmuoju atveju dažniausiai tai būna taisyklės, prieštaraujančios logikos principams. Taisyklės sudaromos taip, kad jų praktiškai neįmanoma įgyvendinti. Vienas iš tokių pavyzdžių galėtų būti:

16 Pvz.: *Nekorektiška taisyklė Nr.1*

*Jei pirkėjo vardas > UAB Giraitė **IR**  
prekė = "bulvės"*

*Tai prekės kiekis = prekės kiekis - Suma %*

Matome jog visiškai supainiotos reikšmės, loginiai operatoriai bei mato vienetai. Užtenka vieno netinkamo parametro, kad tokios taisyklės būtų neįmanoma įgyvendinti. Tokias taisykles sistema ignoro.

Taip pat daug netikslumų atsiranda kai taisyklės sąlyginė ir/arba rezultatų dalis yra sudėtinė. Nors reikšmės pateiktos teisingos, kiekvieną atskirą sąlyginės ar rezultatų dalies elementą galima suprasti bei pritaikyti, tačiau bendras jų visų poveikis tampa nebeaiškus. Toliau pateiktas tokios taisyklės pavyzdys:

17 Pvz.: *Nekorektiška taisyklė Nr.2*

*Jei Prekė = „Bulvės“ IR Kiekis > 100 Kg IR Kiekis < 50 Kg*

*Tai Prekės\_kiekis = Prekės\_kiekis - 10% IR Prekės\_kiekis = Prekės\_kiekis + 10%*

Matome jog tiek sąlyginė dalis, tiek rezultatų dalis yra prieštaringos savo viduje, todėl tinkamai interpretuoti ką ši taisyklė tikrina ir kokius pakeitimus iššaukia yra neįmanoma. Šitokio pobūdžio taisyklės, informacinė taisyklių suderinimo sistema, taip pat ignoruos.

Kitokio pobūdžio probleminės verslo taisyklės yra tokios, kurios prieštarauja kitoms taisyklėms. Tai reiškia, kad vienu taisyklių rezultatai prieštarauja kitų taisyklių rezultatams. Taip gali atsitikti tiek dėl vartotojų, įvedinėjančių taisyklės, „išsiblašymo“ (dėl galimai didelio taisyklių kiekio), tiek dėl pačių taisyklių specifikos. Pirmuoju atveju tai yra taisyklės, turinčios vienodas sąlygines, bet skirtingas rezultatų dalis, antruoju – taisyklės tinkančios tai pačiai įmonei. Sekančiuose pavyzdžiuose (žr. 18 ir 19 pvz.) pateikta pastarojo prieštaravimo pavyzdys. Tarkime turime dvi taisyklės:

18 Pvz.: *Taisyklė Nr.3 (Neigiamo poveikio taisyklės pavyzdys)*

*Jei Pirkėjo vardas = UAB Giraitė IR*

*Prekė = „Bulvės“*

*Tai Prekės\_kaina = Prekės\_kaina + 10%*

19 Pvz.: *Taisyklė Nr.4 (Teigiamo poveikio taisyklės pavyzdys)*

*Jei Prekė = „Bulvės“ IR*

*Bendra suma už prekę >= 1000 Lt*

*Tai Prekės\_kaina = Prekės\_kaina - 15%*

Matome, jog abi taisyklės taikomos tai pačiai įmonei „UAB Giraitė“ jeigu jos užsakymo bendra užmokesčio suma bus didesnė kaip 1000 litų. Abi šios taisyklės iššaukia priešingus dokumento pakeitimus, t.y. pirmoji taisyklė padidina kainą už bulves, o antroji ją sumažina.

Esant tokiai situacijai pasidaro nebeaišku kaip tinkamai reaguoti į tokio tipo taisyklės. Jas abi ignoruoti, abi vykdyti, o gal vykdyti vieną kažkurią. Paskutiniu atveju iškyla naujas klausimas „kurią?“. Pateikto pavyzdžio atveju duotos dvi taisyklės, kurios iššaukia priešingus pakeitimus, t.y. viena teigiamo poveikio, kita neigiamo. Tačiau tokios pat problemos egzistuoja ir taisyklėms esant vienodo poveikio. Tarkim jei abi jos būtų teigiamo ar abi neigiamo (20 ir 21 pavyzdžiai).

20 Pvz.: Taisyklė Nr.5 (Teigiamo/Neigiamo poveikio taisyklės pavyzdys)

*Jei Pirkėjo\_vardas = UAB Giraitė IR  
Prekė = „Bulvės“  
Tai Prekės\_kaina = Prekės\_kaina + /(-) 10 %*

21 Pvz.: Taisyklė Nr.6 (Teigiamo/Neigiamo poveikio taisyklės pavyzdys)

*Jei Prekė = „Bulvės“ IR  
Bendra suma už prekę  $\geq$  1000 Lt  
Tai Prekės\_kaina = Prekės\_kaina + /(-) 15 Lt*

Esamą situaciją dar pablogina skirtingų mato vienetų naudojimas. Pavyzdžiui 20-oje taisyklėje naudojami procentai (%), o 21-oje litai (Lt). Tokiu atveju tampa neįmanomas abiejų taisyklių vykdymas, nes gautas rezultatas priklausys nuo jų įvykdymo eiliškumo.

### 5.1.6. Taisyklių suderinimas

Norint išspręsti aprašytuosius nesuderinamumus, projektuojant informacinę sistemą reikia numatyti ne vieną sprendimo variantą, o kombinuoti kelias iškart. Priešingu atveju praktiškai neįmanoma užtikrinti tinkamo taisyklių funkcionalumo.

4.1.5 skyrelyje aprašyto pirmojo pobūdžio verslo taisyklių nesuderinamumo, kai problemos atsiranda dėl vartotojų klaidos, sprendimas nėra labai sudėtingas. Užtenka programiškai realizuoti tokių taisyklių užrašymo logikos tikrinimą. Tikrinimas atliekamas stebint ar įmanoma vieną taisyklės reikšmę palyginti su kita (tiek sąlyginėje tiek rezultatų dalyje). Tarkime turime taisyklę:

22 Pvz.: *Nekorektiška taisyklė Nr.6*

*Jei Pirkėjo\_vardas = 15 Kg IR  
Prekė = „Bulvės“  
Tai Prekės\_kiekis = Prekės\_kiekis - Suma Lt*

Matome jog pateiktoji taisyklė (22 pvz.) minėtojo tikrinimo „nepraeis“, kadangi pirkėjo vardo neįmanoma palyginti su skaičiumi penkiolika, juolab, kad jis reiškia kilogramus.

Taigi tokio tipo taisyklė automatiškai bus ignoruojama. Tačiau jeigu taisyklė „praeina“ tikrinimą, tuomet vykdomas jos vidinės logikos korektiškumo įvertinimas. Informacinė verslo taisyklių suderinimo sistema, prieš vykdydama konkrečią taisyklę, pirmiausia turi ją tinkamai interpretuoti. Tai reiškia, kad taisyklė turi turėti logišką sąlygą, kuri inicijuotų tam tikrų pakeitimų būtinybę, bei rezultatą, kuris sugebėtų kažką įvykdyti, t.y. modifikuoti, priskirti ar panašiai. Dažniausiai tokio tipo trūkumas, logikos nesuderinamumas taisyklės viduje, pasireiškia sudėtinėse taisyklėse kur naudojamos kelios sąlyginės ir/arba rezultatų dalys. Jeigu logikos suderinamumo nėra, tuomet tokia taisyklė taip pat yra

praleidžiama ir pereinama prie kitos. Taip daroma tol, kol patikrinamos ir pritaikomos, arba atmetamos, visos reikalingos taisyklės.

Taip pat stebima, ar reglamentuotame taisyklių sąrašė nėra įvesta taisyklių su besidubliuojančiomis sąlyginėmis dalimis. Jei atrandama sutapimų, tokios taisyklės automatiškai patenka į rizikingų taisyklių sąrašą, turint įtarimą, jog jų rezultatų dalys gali būti konfliktiškos. Tokiu atveju, iššaukiamas pranešimo signalas, reikalaujantis taisyklių administratoriaus dėmesio, dėl patvirtinimo jog taisyklės tikrai yra korektiškos.

Racionalesnis būdas minėtąjį tikrinimą, dėl vartotojų įvestos nekorektiškos informacijos, būtų atlikti dar verslo taisyklių kūrimo metu. Tokiu atveju, vartotojas būtų iškart informuojamas apie, „greičiausiai“, netyčia įvestą nekorektišką taisyklę. Taip pat serveris nebūtų apkrautas perteklinėmis taisyklėmis, kurias reikia saugoti ir dėl kurių organizuojamas papildomas korektiškumo tikrinimas. Tačiau dėl galimo didelio taisyklių kiekio, bei dėl nutolusio serverio, minėti tikrinimai gali užtrukti, kas priverstų vartotoją galimai nepagrįstai laukti.

Sudėtingesnis atvejis pastebimas tada, kai tam tikrų taisyklių sąlyginės ir/arba rezultato dalys, yra nesuderinamos kitų taisyklių atžvilgiu. Šiam nesuderinamumui taisyklių logikos tikrinimas nepadės, nes kiekviena jų atskirai yra įgyvendinamos ir teisingos. Minimas atvejis aprašytas 4.1.5 skyrelyje aptariant 18, 19, 20 ir 21 pavyzdžius. Norint likviduoti šią problemą reikia išspręsti tokius klausimus kaip: „Ar vykdyti visas taisykles, net jei jos prieštarauja viena kitai?“, „Jeigu vykdyti ne visas, tai kurias tuomet?“, „Ar vykdant kelias prieštaringas taisykles kelis kartus bus gaunamas tas pats rezultatas?“ ir panašiai. Vienareikšmio atsakymo į šiuos klausimus turbūt nerasime, kadangi viskas priklauso nuo konteksto bei konkrečių taisyklių specifikos.

Tarkime turime dvi prieštaringas taisykles, kurių viena duoda teigiamą, atskiram vartotojui, poveikį, o kita neigiamą:

23 Pvz.: Taisyklė Nr.7 (Neigiamo poveikio taisyklės pavyzdys)

*Jei Pirkėjo vardas = UAB Giraitė **IR***

*Prekė = „Bulvės“*

*Tai Prekės\_kaina = Prekės\_kaina + 10 Lt*

24 Pvz.: Taisyklė Nr.8 (Teigiamo poveikio taisyklės pavyzdys)

*Jei Prekė = „Bulvės“ **IR***

*Bendra suma už prekę  $\geq$  1000 Lt*

*Tai Prekės\_kaina = Prekės\_kaina - 15 Lt*

Šiuo pateiktu atveju galima vykdyti abi taisykles ir nesvarbu kokia tvarka, kadangi galutinių rezultatų reikšmės pateiktos litais. Tai reiškia, kad pirma įvykdžius 7-ąją taisyklę,

o paskui 8-ąją rezultatą bus toks pat kaip ir įvykdžius jas kita tarka.

$$\text{Prekės\_kaina} = \text{Prekės\_kaina} + 10 \text{ Lt} - 15 \text{ Lt} ==$$

$$\text{Prekės\_kaina} = \text{Prekės\_kaina} - 15 \text{ Lt} + 10 \text{ Lt} ==$$

$$\text{Prekės\_kaina} = \text{Prekės\_kaina} - 5 \text{ Lt}$$

Pateikto pavyzdžio sprendimas, kai nepaisant nieko vykdomos visos taisyklės, tinkamas visais atvejais jeigu skirtingos taisyklės „operuoja“ vienodais matų vienetais.

Taigi į klausimą „ar vykdyti visas prieštaringas taisykles“ lyg ir galėtume atsakyti „Taip“, nors ir egzistuoja nuostata, kad nuolaidos („nuobaudos“) nepliusuojamos. Tačiau egzistuoja situacijos kai nėra prasmės ignoruoti vieno ar kito sprendimo, jeigu įmonė atitinka jai keliamus reikalavimus (pavyzdžiui nuolaidos, gaunamos už didelį prekių kiekio poreikį bei galimybės patiems tas prekes atsivežti). Nors paminėtina, jog pirma pliusuoti nuolaidas, o tik po to jas pritaikyti, negalima, ypač jei jos išreikštos procentais. Taip yra todėl, kad tokiu atveju atsakymas skirsis nuo to, kurį gautume taisykles vykdydami nuosekliai vieną po kitos.

$$\text{Kaina} = \text{Kaina} + 10 \% + 15 \% \neq \text{Kaina} = \text{Kaina} + 25 \% \quad ^1$$

Tačiau jeigu egzistuoja bent dvi prieštaraujančios taisyklės, iš kurių bent vienoje „operuojama“ skirtingais matų vienetais, tuomet pateiktasis sprendimas nebetinka. Taip yra todėl, kad tas pačias taisykles pritaikę tam pačiam dokumentui kelis kartus, dėl jų vykdymo eiliškumo nesutapimo gausime skirtingus rezultatus.

$$\text{Kaina} = \text{Kaina} + 10 \% + 15 \text{ Lt} \neq \text{Kaina} = \text{Kaina} + 15 \text{ Lt} + 10\% \quad ^2$$

Veiksmingiausias taisyklių valdymo metodas, padedantis visuomet gauti tapatį rezultatą kelis kart bandant pritaikyti tas pačias taisykles, yra jų vykdymo pirmenybės nustatymas. Tai reiškia, kad kuriant taisyklę reikės nurodyti jos vykdymo prioritetą. Tarkime susikūreime taisykles, kurios išdėstomos kūrimo eilės tvarka (žr. 5 lentelę).

**5 Lentelė**

<i>Taisyklių sąrašas</i>	
Taisyklės numeris	Prioritetas
<i>Taisyklė Nr. 1</i>	1
<i>Taisyklė Nr. 2</i>	1
<i>Taisyklė Nr. 3</i>	1
<i>Taisyklė Nr. 4</i>	1
....	...
<i>Taisyklė Nr. n</i>	1

Matome, jog visų sukurtų taisyklių prioritetą, pagal nutylėjimą, lygus vienetai. Tai reiškia, kad vykdamas jų pritaikymą, sistema, kaskart kreipdamasi į taisyklių sąrašą, atsitiktinai

<sup>1</sup> (Reiškinys užrašytas tokia notacija dėl paprastumo. Nuosekliai jį skaitant reikia daryti prielaidą, jog vis didinama gautoji reikšmė, t.y iš pradžių kaina padidinama 10 % nuo buvusios reikšmės, o po to gautas rezultatas dar padidinamas 15 %. Matematiškai tai turėtų atrodyti taip:  $\text{Kaina} = (\text{Kaina} + \text{Kaina} * 10\%) + (\text{Kaina} + \text{Kaina} * 10\%) * 15 \%$ )

<sup>2</sup> Galioja tapati taisyklė kaip ir 1 aprašytuojū atveju

paima taisykles iš sąrašo. O tai sukelia problemas, susijusias su jų suderinamumu. Norint išspręsti šią problemą, kiekvienai taisyklei priskirkime prioritetą, kuris nulems, kokia tvarka jas vykdyti. Vartotojas, užrašęs taisyklės sąlyginę bei rezultatų dalis, taip pat turi nuspręsti, kokio svarbumo taisyklė tai bus. Kuo prioriteto laipsnis didesnis, tuo taisyklė bus vykdoma anksčiau. Tačiau tai dar nereiškia, kad ši taisyklė vartotojui yra svarbiausia (taip gali būti, bet ne visada). Prioritetais vartotojas tiesiog išskiria, koks taisyklių vykdymo eiliškumas jam yra palankiausias. Tarkime turime dviejų tenkinamų taisyklių rezultatų dalis:

25 Pvz.: *Taisyklė Nr.9 (Dviejų tenkinamų taisyklių rezultatų dalys)*

$$1) \quad \text{Tai Prekės\_kaina} = \text{Prekės\_kaina} + 15 \text{ Lt}$$

$$2) \quad \text{Tai Prekės\_kaina} = \text{Prekės\_kaina} - 15 \%$$

Visais atvejais, pirma vykdant pirmąją taisyklę, o tik po to antrąją, gautasis rezultatas bus mažesnis negu tai darant atvirkščiai. Taigi vartotojas turi pats nuspręsti, kokio rezultato jis ištiktųjų tikisi.

Didžiausias šio metodo trūkumas yra tai, kad vartotojui norint priskirti prioritetą, dėl galimai didelio taisyklių kiekio, patampa labai sunku nustatyti visas galimas pasekmes, priklausančias nuo prioriteto reikšmės. Tačiau atsižvelgiant į tai, kad taisyklių suvedinėjimas nėra dažnas procesas, pati taisyklė turi būti labai gerai įvertinama, prioriteto kiekis nėra didelis bei testavimo būdu sistema suteikia galimybę galimas reikšmes patikrinti, ši problema nėra labai aktuali.

Šeštoje lentelėje pateiktas vienas iš prioritetaizavimo pavyzdžių, kur  $m$  – galimai didžiausias prioriteto numeris,  $n$  – taisyklės numeris.

**6 Lentelė**

<i>Taisyklių sąrašas</i>	
Taisyklės numeris	Prioritetas
<i>Taisyklė Nr. 5</i>	$m$
....	....
<i>Taisyklė Nr. 10</i>	4
<i>Taisyklė Nr. 1</i>	3
<i>Taisyklė Nr. 7</i>	3
....	....
<i>Taisyklė Nr. n</i>	1

Esant galimai dideliame taisyklių kiekiui prioriteto laipsnis neišvengiamai ima kartotis. Tai vėlgi iššaukia taisyklių, turinčių tokį patį prioritetą, galimą nesuderinamumą. Naujai išryškėjusiai problemai spęsti galima panaudoti „statusų“ vartotojams priskyrimą. Lygiai kaip ir taisykles išskirstėme prioritetais, taip pat galime išskirstyti klientus, priskyre jiems tam tikrą kiekį tam tikrų statusų, pavyzdžiui „Auksinis narys“, „Sidabrinis narys“ ir t.t. Racionaliausias statusų priskyrimo būdas - atskirų klientų susiejimas su derybų vykdymo istorija, tačiau tam reikia turėti didelę duomenų bazę, kurioje saugomi buvę sandoriai.

Kadangi tokios duomenų bazės nėra (bent jau pradiniam sprendimo naudojimo etape), galima pasinaudoti keliomis, papildomomis taisyklėmis.

26 Pvz.: Priskyrimo taisyklė (Statusų priskyrimas klientams)

*Jei Bendra pirkimo suma  $\geq 1000$  Lt*

*Tai kliento statusas = Auksinis narys*

Tokiu atveju, tam tikrą statusą turintiems klientams galima nurodyti nevykdyti kai kurių prioritetų taisyklių. Pavyzdžiui klientai, kuriems priskirtas „Auksinio nario“ statusas, galima nurodyti, kad nebūtų vykdomos pirmo laipsnio prioritetą turinčios taisyklės ir panašiai. Paminėtina, jog naudojant statusų naudojimą, reikia labai atidžiai priskyrinti prioritetus taisyklės, kadangi gali atsitikti taip, jog dėl netinkamo prioriteto bus ignoruotos reikalingos taisyklės. Tačiau egzistuoja ir tokios situacijos, kai galimai didelis sandoris gali „pražūti“ dėl nereikšmingų, tarpusavyje prieštaraujančių, taisyklių, todėl šis metodas bent dalinai išsprendžia šią problemą.

Statusų kiekį apsprendžia vartotojas, kuriantis savo taisyklių sąrašą. Šis kiekis gali būti keičiamas vykdymo metu, tačiau to daryti nerekomenduojama, kadangi šiuo atveju tektų perskirstyti prioritetus sprendimų lentelėje (žr. 7 lentelė). Ši lentelė nurodo kokio prioriteto taisyklės bus naudojamos vykdant derybas su atitinkamą statusą turinčiais klientais.

**7 Lentelė**

Statusas\Prioritetas	n	6	5	4	3	2	1
Auksinis narys	?	+	+	-	-	-	-
Sidabrinis narys	?	+	+	+	+	-	-
Bronzinis narys	?	+	+	+	+	+	+
X narys	?	?	?	?	?	?	?

Tačiau matome, jog siūlomoji statusų lentelė tik dalinai išsprendžia vienodą prioritetą turinčių taisyklių nesuderinamumą. Mažiausią statusą turintiems klientams vis tiek bus taikomos visos taisyklės, taigi šį metodą reikia kombinuoti su „Taisyklių grupavimu“. Tai reiškia, kad visos taisyklės, atsižvelgiant į jų pobūdį, bus grupuojamos į dvi dalis. Tai yra į taisykles, kurios daro tam tikrą poveikį visam dokumentui, bei taisykles, kurios skirtos konkrečioms prekėms. Pavyzdžiui turime dvi taisykles pateiktas 27 pavyzdyje:

27 Pvz.: Taisyklių grupavimas (Dviejų grupių taisyklės)

1) *Jei Pirkėjo vardas = UAB Giraitė **IR***

*Prekė = „Bulvės“*

*Tai Prekės\_kaina = Prekės\_kaina + 0,5 Lt*

2) *Jei Pirkėjo\_vardas = „UAB Giraite“*

*Tai Bendra\_suma = Bendra\_suma - 15 %*

Matome, jog pirmu numeriu pažymėta taisyklė poveikį daro tik konkrečiai prekei, o antru numeriu – viso dokumento sumai. Esant tokiai situacijai, sistema visuomet pirma pritaikys

taisykles, skirtas konkrečioms prekėms, o po to visas likusias. Taip daroma todėl, kad atvirkščias variantas logiškai nekorektiškas.

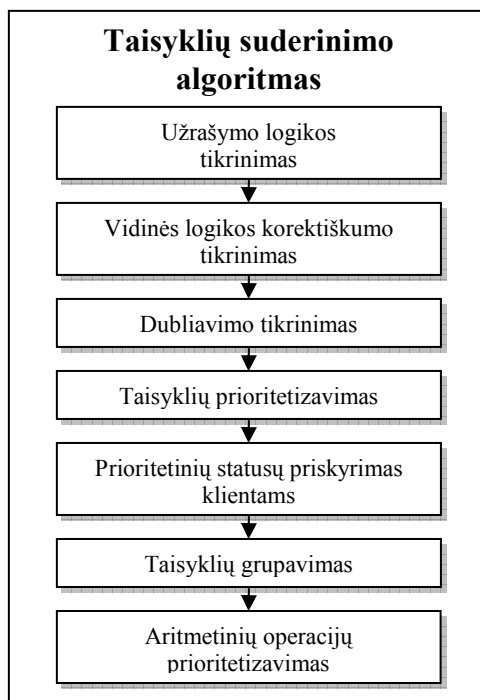
Jeigu pritaikius visus minėtus metodus vis dar pasitaiko nesuderinamumų, tuomet naudojamas paskutinis „Operacijų prioritizavimo“ metodas. Tai kraštutinis metodas, kuriame nusprendžiamas taisyklių vykdymas atsižvelgiant į jų aritmetines operacijas. Aštuntoje lentelėje, pateiktas aritmetinių operacijų galimas prioritizavimas.

**8 Lentelė**

Prioriteto grupė	Operacijos
4	-
3	/
2	+
1	*

Atsiradus dviem ar daugiau nesuderinamų taisyklių, pirmiau bus vykdoma ta, kurios aritmetinės operacijos prioritetas yra didesnis. Tačiau pabrėžiama, jog šis tikrinimas atliekamas tik tuomet, kai visi kiti metodai yra neveiksmingi.

Apibendrinant galima teigti, jog norint visiškai ar bent didžiąja dalimi išspręsti verslo taisyklių tarpusavio nesuderinamumą, galima pasinaudoti septynių „žingsnių“ algoritmu. Jo pagrindu valdomos taisyklės yra visapusiškai patikrinamos bei sugrupuojamos, kad neiššauktų konfliktinių situacijų. Pateiktoje schemeje (žr. 5.16 pav.) galima pastebėti, jog verslo taisyklių suderinimo algoritme naudojami trys skirtingi tikrinimai, bei keturi skirtingi prioritizavimai.



**5.16 pav.** Verslo taisyklių suderinimo metodo algoritmas



## 5.2. Eksperimento realizacijos aprašymas

### 5.2.1. Sprendimo reikalavimų specifikacija

Prieš pradėdant aptarinėti eksperimento realizacijos specifikaciją, pirmiausia reikia apibrėžti jog sistemos prototipas (produktas) nėra skirtas galutiniam vartotojui. Tai reiškia, jog vartotojo sąsaja bei tam tikros funkcijos nėra pilnai realizuotos ar jų išviso nėra. Eksperimento realizacijos esmė yra ištestuoti siūlomo metodo funkcionalumą bei efektyvumą, todėl buvo kuriami tik tie moduliai, kurie yra neišvengiami norint pilnai atskleisti sukurto metodo savybes.

Sprendimo realizacijos modelis (prototipinė sistema) sudaryta tik iš dviejų dalių – serverio bei kliento. Dėl šios priežasties sistema gali veikti bet kuriame šiuolaikiniame kompiuteryje. Pagrindiniai kompiuterio reikalavimai:

1) Turi veikti JBOSS taikomųjų programų serveris (angl. *Application server*) su Drools ir Drools Guvnor programomis, kadangi jo pagrindu realizuotos pagrindinės siūlomo metodo funkcijos.

2) Turi būti įdiegta MySQL, kurio pagalba sukurta duomenų bazė bei realizuotos reikalingos užklausos.

Kliento, vartotojo sąsajos, dalyje įdiegtas specialiai eksperimentui vykdyti sukurtas „vartotojo klientas“. Jis realizuotas naudojantis Java programavimo kalba, todėl veikia bet kuriame Java palaikančiame kompiuteryje, kuriame įdiegtas JRE 6 (*Java Runtime Environment*). Naudojami dokumentai realizuoti bei peržiūrėti *Microsoft Office InfoPath 2007* įrankio pagalba.

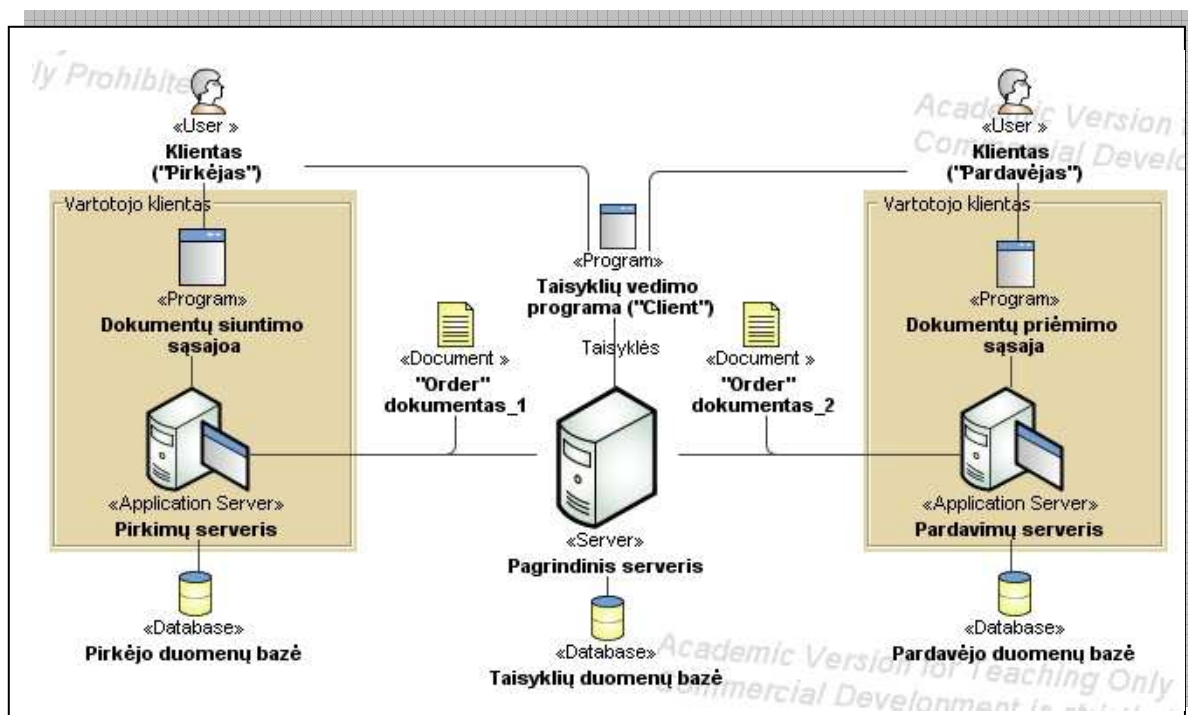
Visi realizuojamos eksperimentinės sistemos komponentai, darbo metu, įgyvendinti bei tarpusavyje apjungti naudojantis NetBeans 6.9 programine įranga.

Taigi darbas atliktas naudojantis tokia programine įranga:

- JBOSS *application server* (su *Drools* ir *Drools Guvnor* programomis)
- MySQL
- JRE 6 (*Java Runtime Environment*)
- NetBeans 6.9
- Microsoft Office InfoPath 2007

### 5.2.2. Eksperimento realizacijos koncepcija

Realizuojant sistemą buvo sukurti visi komponentai aprašyti sistemos projekto dalyje (4.1 skyrius). Apjungus visus planuotus elementus į vieną visumą gavome bendrą eksperimento realizacijos vaizdą (žr. 5.17 pav.), kuriame aiškiai matoma, kaip šie elementai sąveikauja tarpusavyje.



5.17 pav. Eksperimento realizacijos koncepcinė schema

Pirmasis žingsnis, kurį turi atlikti klientai, norintys, kad sistema atliktų veiksmus, kurių jie tikisi, „Taisyklių vedimo programos“ pagalba turi suvesti savo reglamentuotas verslo taisykles (jeigu jų nėra, tai suvedinėjimas nėra privalomas). Toliau klientas „Pirkėjas“ naudodamasis „Vartotojo kliento“ sistemos pagalba išsiunčia užpildytą „Order“ dokumentą klientui „Pardavėjui“. Išsiųstas dokumentas pirmiausia nukeliauja į „pagrindinį serverį“ kur yra apdorojamas ir tuoj pat parsiuočiama atgal su pakeistomis reikšmėmis (jeigu buvo pritaikytos verslo taisyklės ir laukai pasikeitė). Pirkėjas peržiūri pasikeitimus ir jeigu jį tenkina, pakeitimai kuriuos mato, duoda patvirtinimo signalą patvirtinantį dokumento siuntimą pardavėjui.

Toliau aptarsime kiekvieną komponentą atskirai, atkreipiant dėmesį į jų funkcionalumą bei bendrą poveikį visai sistemai ar konkreitiems procesams.

### **Taisyklių vedimo programa:**

Ši programa tai tam tikra sąsaja su pagrindiniu serveriu, skirta sukurti bei išsaugoti verslo taisykles. Tai atlikti jie gali bet kuriuo laiko momentu. Sukurtas taisykles galima modifikuoti ar reikalui esant visiškai pašalinti iš sistemos. Taip pat suteikiama galimybė pažymėti taisyklės poveikį (tai bus *nuolaidos*, *nuobaudos* ar tiesiog reikšmių *keitimo* taisyklė), bei nustatyti prioritetą.

Taigi sistemos vartotojas, pasinaudojęs taisyklių vedimo programa, gali sukurti taisykles, kurios automatiškai modifikuotų užsakymo dokumentus. Taisyklę sudaro "Sąlygos" ir "Rezultatai". Kiekviena iš jų gali būti sudėtinės. Jei sąlyga(os) tenkinama(os), vykdomi "Rezultatai", nurodyti pakeitimai.

Dokumentas gali atitikti kelių skirtingų taisyklių sąlygas. Visos jos bus taikomos. Jei tarp taisyklių, pritaikomų vienam dokumentui, atsiranda prieštaravimų, pagrindinis serveris juos aptinka bei imasi reikalingų priemonių, kad juos panaikintų (pagrindinio serverio veikimo principas aprašytas toliau).

Kitame paveiksle (žr. 5.18 pav.) pateiktas panaudotas taisyklių suvedinėjimo langas su užrašyta konkrečia taisykle.

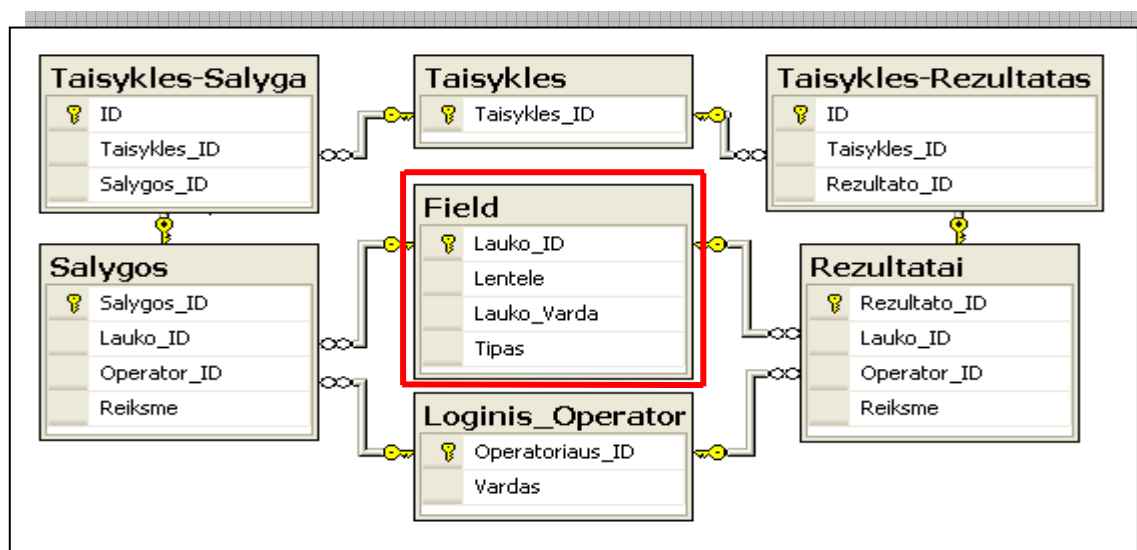
The screenshot shows a software window titled "Redaktorius" with a "Tekstas" tab. It contains a form for defining a rule. The "Pavadinimas" field is filled with "Konkreti\_Taisyklė". The "Aprašymas" field contains a descriptive text: "Taisyklė su išplėstine sąlyga. jeigu Pirkėjo įmonė yra "UAB Giraitė" ir perkamos prekės kiekis daugiau kaip 1000, tuomet kaina sumažinama 15 procentų". Below this, there are three radio buttons: "Nuolaida" (selected), "Nuobauda", and "Keitimas". To the right, "Prioritetas" is set to "2". The "JEI" (IF) section has two conditions: "Pirkėjo pavadinimas" is "yra" "UAB Giraitė" and "Prekių kiekis" is "daugiau už" "1000". The "TAI" (THEN) section has one result: "Nuolaida užsakymui" is "15 %". At the bottom, there are "OK" and "Atšaukti" buttons.

5.18 pav. Eksperimento metu realizuota taisyklė

Kuriamos naujos taisyklės, visi kintamieji (įskaitant ir lyginimo operatorius) duodami pasirinkti ir galimo sąrašo, kuris suformuojamas atsižvengiant į UBL orderį dokumentą.

### Realizacijos duomenų bazė:

Norint realizuoti 5.17 paveiksle pavaizduotą eksperimentinės realizacijos funkcionavimą, reikia sukurti duomenų bazę, aprašytą 4.1.3 skyriuje. Aprašytoji duomenų bazė svarbi netik saugant sukurtas taisykles, bet ir jas kuriant. Taip yra todėl, kad jame sukaupti loginiai kintamieji reikalingi formuojant veiklos taisyklę. Šios duomenų bazės vaizdas pateiktas 5.19 paveiksle. Jame matome jog ji susideda iš septynių duomenų bazės lentelių atsakingų už taisyklių kūrimą, rezultatų priskyrimą bei saugojimą. Realizuojant eksperimentinę sistemą buvo pridėta „Field“ lentelė, kurios nebuvo aptariant verslo taisyklių saugojimą.



5.19 pav. Taisyklių duomenų bazė

9 lentelėje „Field“ naudojama reikiamų parametru gavimui pildant taisyklių formą (formos tikslas bei veikimas buvo aptartas aprašant „*taisyklių vedimo programą*“).

9 Lentelė

ATRIBUTAS	TIPAS	PASKIRTIS
<b>FIELD</b>		
LAUKO_ID	INTEGER	Pirminis raktas
LENTELĖ	STRING	Nurodoma DB lentelė kurioje saugoma reikiama reikšmė
LAUKO_VARDAS	STRING	Nurodytos lentelės konkretaus lauko pavadinimas, tam kad būtų atfiltruojami duomenys, kurie bus patalpinti į „iškrentantį meniu.“
TIPAS	STRING	Gražinamos reikšmės tipas. Pvz.: String, integer ir t.t.

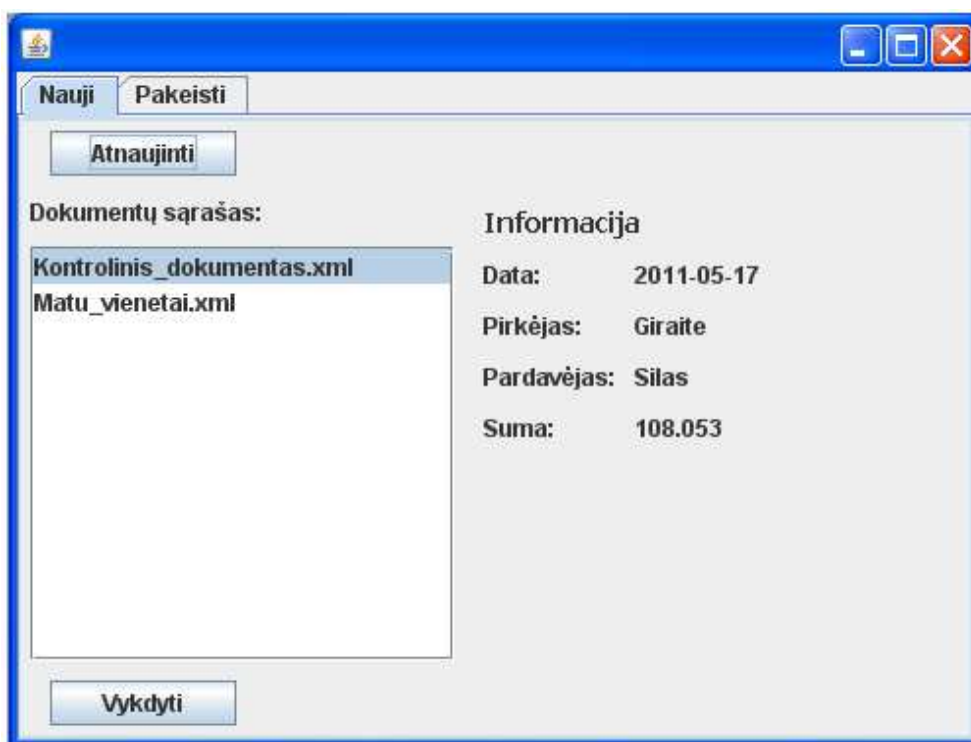
Pabrėžiama, jog dokumentai kuriami XML formatu tai duomenų bazės jiems saugoti nereikia. Būtų galima kaupti konkrečius užpildytus dokumentus, tačiau realiai naudojant sistemą klientai saugos šiuos dokumentus pas save sistemoje, o kadangi eksperimente naudosime tik vieną realų XML formatu parašytą UBL standarto „Order“ dokumentą tai jo saugojimas į realizuojamos sistemos funkcionalumą neįeina.

**Vartotojo klientas:**

Šių sistemų pagrindinės funkcijos yra leisti vartotojams siųsti bei priimti verslo dokumentus. Antruoju atveju sistema išduoda pranešimą apie gautą dokumentą ir saugo jį savo viduje kol bus patvirtintas vartotojo (labai panašu į elektroninio pašto veikimo principą). Siuntimas atliekamas pasirenkant dokumentą, kuris iš anksto paruošiamas, bei paspaudus mygtuką „Vykdėti“.

Iš esmės dokumentų kūrimas neįeina į siekiamo eksperimento realizacijos sudėtį. Norint pademonstruoti konfliktuojančių taisyklių suderinimą, pakanka turėti jau paruoštus dokumentus. Tačiau atsiranda nenumatytų situacijų kai norint parodyti vienokią ar kitokią taisyklių nesuderinamumą, prisireikia dokumento su tam tikrais duomenimis, todėl buvo sukurtas šablonas dokumentams kurti.

Naujų dokumentų pildymas paremtas UBL dokumentų principu. Tai reiškia, kad laikomasi šio standarto nuostatų bei taisyklių. Atliekant eksperimentą vartotojas turės galimybę, užpildydamas tam tikrą formą, sukurti dokumentą, kurį bus galima siųsti sistemai, kuri, jeigu reikia, pritaikytų jo paties bei kitų klientų įvestas taisykles bei persiųs gavėjui. Tačiau pabrėžiama, jog dokumentų pildymas nėra esminis punktas atliekamame eksperimente, todėl į jo realizacija nebus išsamiai gilinamasi. Toliau (žr. 5.20 pav.) pateiktas vartotojo langas, kuriame pasirenkamas norimas dokumentas, bei siunčiamas taisyklių pritaikymui bei gavėjui.



5.20 pav. Vartotojo kliento grafinė sąsaja

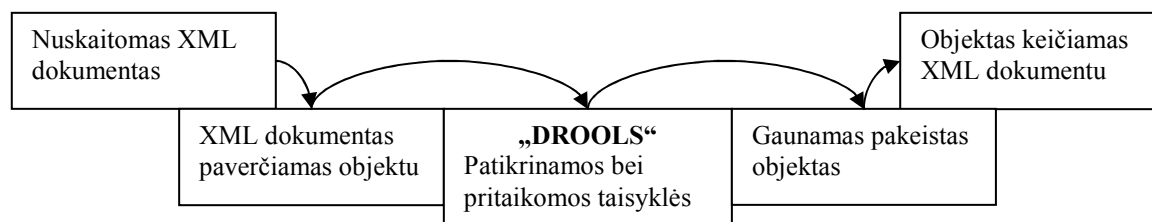
**Pagrindinis serveris:**

Tai svarbiausias eksperimento realizacijos komponentas, atsakingas už verslo taisyklių pritaikymą UBL-Order dokumentams. Kaip matome 5.17 paveiksle jis tiesiogiai siejasi su:

- 1) Taisyklių duomenų baze, iš kur gauna taisykles.
- 2) Vartotojų kliento sistemoma, iš kur atkeliauja ar į kur išsiunčiami dokumentai.

Dalį suderinimo darbo šiame serveryje atlieka „DROOLS“ įrankis. Tai priemonė, realizuojanti verslo taisyklių rinkinį (angl. *Rules Engine Implementation*) bei sugebanti aptikti ir dalinai išspręsti konfliktuojančias taisykles. Kombinuojant jį su 5.1.6 punkte aptartais metodais buvo sukurta efektyvi priemonė, derinant konfliktuojančias derybų taisykles.

Tam, kad būtų galima efektyviau bei suprantamiau naudotis „DROOLS“ įrankiu, buvo sukurtas papildomas vartotojo sąsajos klientas, kuriame įtraukta keletas naujų sąsajos sisteminių mazgų. Jie reikalingi integracijai su į objektą orientuota kalba. Taip daroma todėl, kad „DROOLS“ iš esmės su XLM dokumentais elgiasi taip, lyg tai būtų objektai. Dėl šios priežasties, pritaikant derybų taisykles, įrankis reikalauja 5.21 paveiksle pavaizduotos procesų eigos.



5.21 pav. XML dokumentų vertimo objektais procesų schema

Taigi papildomos sąsajos pritaikymas leidžia verslo taisykles padaryti priimtinesnes verslo klientams, kadangi nereikia „operuoti“ objektais.

Toliau aptarsime „pagrindinio serverio“ veikimo principą bei tai, kaip „DROOLS“ siejasi su 5.16 paveiksle pavaizduotu taisyklių suderinimo algoritmu. Pats „DROOLS“ įrankis, be papildomo konfigūravimo, geba tik aptikti bei pritaikyti užrašytas derybų taisykles. Toks veikimo principas neišsprendžia konfliktuojančių taisyklių suderinamumo klausimo. Dėl šios priežasties ir reikalingas metodas, išsprendžiantis šią problemą. Taigi kuriant verslo taisykles, sukurtoji taisyklių vedimo sąsaja įgyvendina siūlomo metodo punktus. Tai reiškia, jog atliekami visi logiškumo tikrinimai bei sužymimi taisyklių prioritetai, grupavimo sąlygos ir panašiai. Toliau visi šie duomenys, naudojant duomenų bazę, perduodami „DROOLS“ įrankiui, kuris atsižvelgdamas į gautąsias taisykles bei nurodytuosius parametrus atlieka suderinimo darbą.

DROOLS įrankis yra labai greitas dirbant su dideliu kiekiu taisyklių. Tokį jo parametą nulemia Charles Forgy's Rete algoritmas. Šio paieškos pirmyn algoritmo esmė – iš faktų ir jų derinių sukuriamas grafas – medis. Jame faktai išdėstomi ir sujungiami taip, kad pagal taisyklių prielaidų šablonus būtų vykdoma kuo efektyvesnė peržiūra/paieška. T.y. medis atsimena surastus faktų derinius ir taip sutaupo paieškos laiko (kai tų derinių dažnai ieškoma). Atsiradus naujam faktui, medis minimaliai reorganizuojamas.

Taigi klientas „Pirkėjas“ užpildytą UBL-*Order* dokumentą vartotojų kliento sistemos pagalba siunčia klientui „Pardavėjui“, tačiau šis dokumentas pirmiausia nukeliauja ne pas gavėją, o į šį „pagrindinį serverį“. Čia pirmiausia nustatomos užsakyme dalyvaujančios „šalys“. Tai atliekama panaudojus paieškos algoritmą, kuris suranda įmonių pavadinimus duomenų bazėje (pavadinimai joje atsiranda, kai įmonės reglamentuoja savo derybų taisykles). Ko ieškoti nustatoma pagal gauto UBL dokumento XML struktūros elemento „Name“ reikšmę:

```
<cac:BuyerCustomerParty>                                <cac:SellerSupplierParty>
  <cac:PartyName>                                         <cac:PartyName>
    <cbc:Name>UAB "Giraitė" </cbc:Name>                   <cbc:Name>UAB "Šilas" </cbc:Name>
  </cac:PartyName>                                       </cac:PartyName>
```

Pagal šiuos pavadinimus nustatomos jų reglamentuotos taisyklės. Toliau seka tenkinančių tam tikras sąlygas taisyklių paieška. Paeiliui tikrinamos visos rastos taisyklės ir jeigu jos tinka konkrečiai situacijai, dedamos į sąrašą. Konkreti situacija reiškia, kad taisyklės turi būti susijusios tiek su pirkėju tiek su pardavėju, nes tarp visų užrašytų taisyklių yra tokių, kurios skirtos tam tikroms atskiroms įmonėms.

Kai visos tinkamos taisyklės yra atrinktos, gautasis sąrašas modifikuojamas atsižvelgiant į 4.1.6 punkte aprašytus metodus. Kai taisyklės paruošiamos, serveris pradeda jas taikyti, tai yra jos pradedamos vykdyti. Vykdomo metu tikrinama kiekviena taisyklė ir atsižvelgiant į jos paskirtį modifikuojamas dokumentas (tam tikros jo reikšmės).


Pakeistas dokumentas siunčiamas atgal siuntėjui – galutiniam patvirtinimui. Jeigu siuntėją tenkina galutinis dokumento variantas, pritaikius taisykles, jis jį patvirtina. Po patvirtinimo dokumentas tiesiogiai nukeliauja pas gavėją.

### 5.3. Testavimo modelis bei duomenys, kontrolinis pavyzdys

Norint atlikti verslo taisyklių suderinimo, įmonių sąveikumo sprendimuose, sistemos testavimą, pirmiausia turime jam pasiruošti. Realizuota eksperimentinė sistema, norėdama įgyvendinti sukurtą metodiką, privalo gauti tam tikrus dokumentus bei duomenis.


Taigi pirmiausia reikia pasiruošti derybų taisykles, kurios turėtų įtakos sukurtam dokumentui. Mūsų sukurtasis užsakymo dokumentas turi sąlyginai daug „laukų“, pagal kuriuos galima priiminėti tam tikrus sprendimus, bei atlikinėti pakeitimus. Sukurkime bent tris taisykles. Jas užrašykime literatūrine, formalizuota bei drools taisyklių kūrimo grafine sąsaja:

#### 28 Pvz. Taisyklė reglamentuota pirkėjo (1):

Taisyklė literatūrine forma	<i>Jei perkamų prekių kiekis artimas 50-čiai kg. tai jį prilyginti šiam skiaičiui(penkesdešimčiai)</i>
Taisyklė formalizuota forma	<i>Jei prekių_kiekis &gt; 45 kg. <u>IR</u> prekių_kiekis &lt; 60 kg. Tai kiekis == 50 kg.</i>
Taisyklė grafinės sąsajos forma	


5.22 pav. Testinis kliento (pirkėjo) taisyklės pavyzdys

#### 29 Pvz. Taisyklė reglamentuota pardavėjo (2):

Taisyklė literatūrine forma	<i>Jei užsakymo suma viršija 100 Lt. tai suteikiama 10 Lt. nuolaida.</i>
Taisyklė formalizuota forma	<i>Jei užsakymo_suma &gt; 100 Lt.. Tai užsakymo_suma -10 Lt.</i>
Taisyklė grafinės sąsajos forma	

5.23 pav. Testinis kliento (pardavėjo) taisyklės pavyzdys

#### 30 Pvz. Taisyklė reglamentuota pardavėjo (3):

Taisyklė literatūrine forma	<i>Jei įmonė UAB „Giraite“ perka daugiau kaip 50 kg. tam tikros prekės, tai toms prekėms suteikiama 15 Lt nuolaida.</i>
Taisyklė formalizuota forma	<i>Jei pirkėjo_vardas ==UAB Giraite <u>IR</u> kiekis &gt;=50 kg. Tai prekes_kaina - 15Lt.</i>
Taisyklė grafinės sąsajos forma	

5.24pav. Testinis kliento (pardavėjo) taisyklės pavyzdys



Pirmoji taisyklė reglamentuota pirkėjo (žr. 5.22 pav.). Ji apsprendžia užsakomų prekių kiekius. Tokios taisyklės poreikį gali įtakoti tai, jog užsakymo dokumentus gali siųsti ne vienas įmonės asmuo. Tokiu atveju gali atsitikti taip, jog vadybininkas (ar kitas įgaliotasis asmuo) nori užsakyti prekių tiek, kiek konkrečiu momentu reikia, tačiau taisyklės reglamentuojantis atstovas, įvertinęs pristatymo galimybes, apibrėžia tokius skaičius, kurie sumažina pristatymo kaštus. Konkrečiu atveju žinoma, jog pakuotėje telpa būtent 50 kg.

Antroji taisyklė, reglamentuota pardavėjo, tikrina ar bendra suma už prekes yra didesnė už 100 litų ir jei taip, suteikia 10 litų nuolaidą (žr. 5.23 pav.). Tai tarsi pristatymo nuolaida.

Trečioji taisyklė, taip pat reglamentuota pardavėjo (žr. 5.24 pav.), suteikia nuolaidą, jei pirkėjas yra „UAB Giraitė“ bei perkamas prekės kiekis nemažesnis 50.

Toliau reikia susikurti užsakymo dokumentą, kurio pagrindu bus demonstruojami tam tikri pakeitimai, bei įrodinėjamas sistemos funkcionalumas, atitinkantis išsikeltiems reikalavimams. Kaip jau buvo minėta anksčiau, dokumentas kuriamas užpildant tam tikrą šabloną duomenimis, kuris buvo sukurtas naudojantis „Microsoft office InfoPath 2007“ programine įranga. 5.25 paveiksle pateiktas dokumento pavyzdys su duomenimis.

### Užsakymas

Nr.  Data:

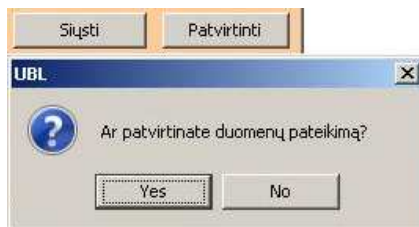
Pirkėjas				Pardavėjas			
Pavadinimas				Pavadinimas			
<input giraitė"="" type="text" value="UAB "/>				<input type="text" value="UAB " šilas"=""/>			
Gatvė				Gatvė			
<input type="text" value="Trakų"/>				<input type="text" value="Santakų"/>			
Miestas				Miestas			
<input type="text" value="Kaunas"/>				<input type="text" value="Vilnius"/>			
Kontaktinis asmuo							
Vardas Pavardė <input type="text" value="Vardenis Pavardenis"/>							
Telefonas: <input type="text" value="+37062535698"/>							
El.Paštas: <input type="text" value="Pas@tas.com"/>							

Valiuta

Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL (Lt)	Suma LTL (Lt)	
1	1111	Bulvės	47	Kg	21	1,9	89,3	
Nuolaidos požymis		Nuolaidos/Premijos priežastis					Suma	
<input type="checkbox"/>								
<input checked="" type="checkbox"/>		Pridėti nuolaida					Iš viso LTL (Lt)	89,3
<input checked="" type="checkbox"/>		Pridėti eilutę					PVM LTL (Lt)	18,753
		Suma su mokesčiais LTL (Lt)						89,3
		Nuolaidų suma LTL (Lt)						0
		Nuobaudų suma LTL (Lt)						0
		Bendra suma LTL (Lt)						108,053

5.25 pav. Užpildytas užsakymo dokumentas

Turint dokumentą bei taisykles galima pradėti eksperimentą. Pasinaudodami „Vartotojo kliento“ programa, pasirenkame dokumentą kurį reikia siųsti, bei paspaudžiame mygtuką „Siųsti“. Kitame paveiksle (žr. 5.26 pav.) pateiktas siuntimo lango patvirtinimo vaizdas.



5.26 pav. Dokumento siuntimo lango vaizdas

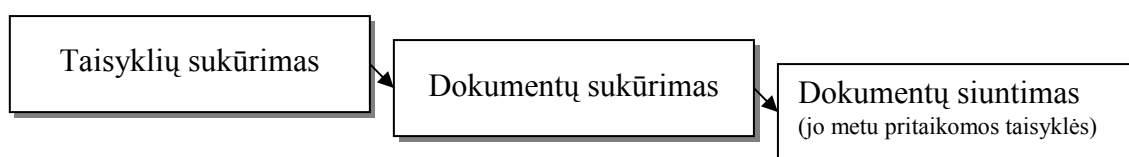
Programa atlieka tam tikrus pakeitimus po kurių grąžina dokumentą, su pakeista eilutės kainos reikšme, patvirtinimui ar dokumentas vis dar tenkina siuntėjo reikalavimus. Po patvirtinimo dokumentas nukeliauja gavėjui. Gavėjas taip pat mato jau pakeistą dokumentą. Pakeisto užsakymo eilutės vaizdas pateiktas 5.27 paveiksle:

Užsakymo eilutės				Valiuta LTL				
Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL	Suma LTL	
1	2223	Bulves	50	KG	21	1,9	95	
Nuolaidos požymis		Nuolaidos/Premijos priežastis					Suma	
<input type="checkbox"/>		Pritaikyta taisykle 3_Kiekis_daugiauLygu_50					15	
<input checked="" type="checkbox"/>		Prideti nuolaida						
<input checked="" type="checkbox"/>		Prideti eilute						
						Iš viso LTL	95	
						PVM LTL	19,95	
						Suma su mokesčiais LTL	114,95	
						Nuolaidų suma LTL	15	
						Nuobaudų suma LTL	0	
						Bendra suma LTL	99,95	

5.27 pav. Užsakymo eilutės po pakeitimo

Matome jog buvo pritaikytos tik pirmoji (1) bei trečioji (3) taisyklės. Taip yra todėl, kad realizuota eksperimentinė sistema pirmiausia įvykdo taisykles susijusias su konkrečiom užsakymo eilutėmis, o tik po to visas likusias. Šiuo konkrečiu atveju padidinus prekių kiekį iki 50 kilogramų bei pritaikius 15 litų nuolaidą, bendra užsakymo suma tapo mažesnė nei 100 litų, todėl antroji (2) taisyklė jau nebeatitiko sąlygos.

Apibendrinant galima sudaryti testavimo modelį (algoritną) pagal kurį atliekamas realizuotos sistemos funkcionalumo bei korektiškumo tikrinimas (žr. 5.28 pav.).



5.28 pav. Testavimo modelis

## 6. Eksperimentinis sistemos tyrimas

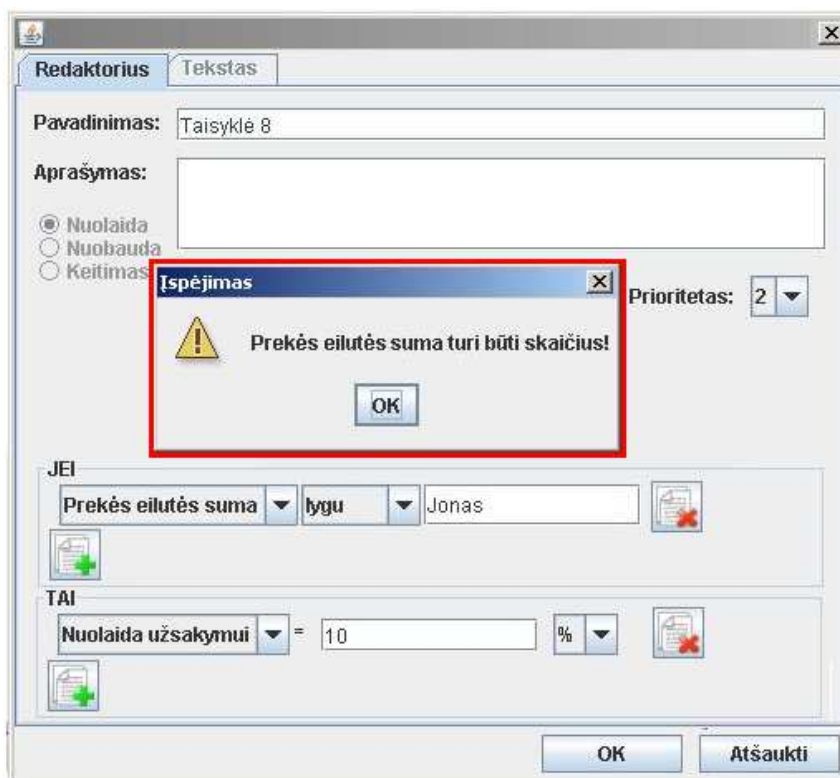
### 6.1. Eksperimento planas bei gauti rezultatai

Norint įsitikinti, jog sistema tikrai veikia taip kaip buvo tikėtasi prieš imantis ją projektuoti, privalu atlikti eksperimentinį jos tyrimą. T.y. imituoti tokias situacijas, kurios galimai galėtų sutrikdyti jos numatytąjį funkcionalumą. Tinkamiausios tokio tikrinimo situacijos yra labai glaudžiai susijusios su sistemos metodų naudojimu. Dėl šios priežasties, ištyrę kaip sistema reaguoja į 4.1.6 punkte aprašytuosius verslo taisyklių suderinimo algoritmo metodus, nustatysime ar tikrai pasiekėme užsibrėžto rezultato.

Pirmosios funkcijos, realizuotos kuriamoje sistemoje, buvo taisyklių „užrašymo logikos“ bei „vidinės logikos korektiškumo“ tikrinimai. Norint ištestuoti šias funkcijas pakanka pabandyti sistemoje užrašyti nekorektiškas taisykles ir iškart matomas pranešimas apie klaidas. Šiuo atveju buvo panaudotos tokios dvi taisyklės:

$Je\ i\ Prek\ \_suma = Jonas$ $Tai\ Bendra\_suma = Bendra\_suma - 10\ Lt$	$Je\ i\ Prek\ = „Bulv\ \_“\ IR\ Kiekis > 100\ Kg\ IR\ Kiekis < 50\ Kg$ $Tai\ Prek\ \_kaina = Prek\ \_kaina - 10\ \%$
--	---

Gautas rezultatas pateiktas 5.1 paveiksle:

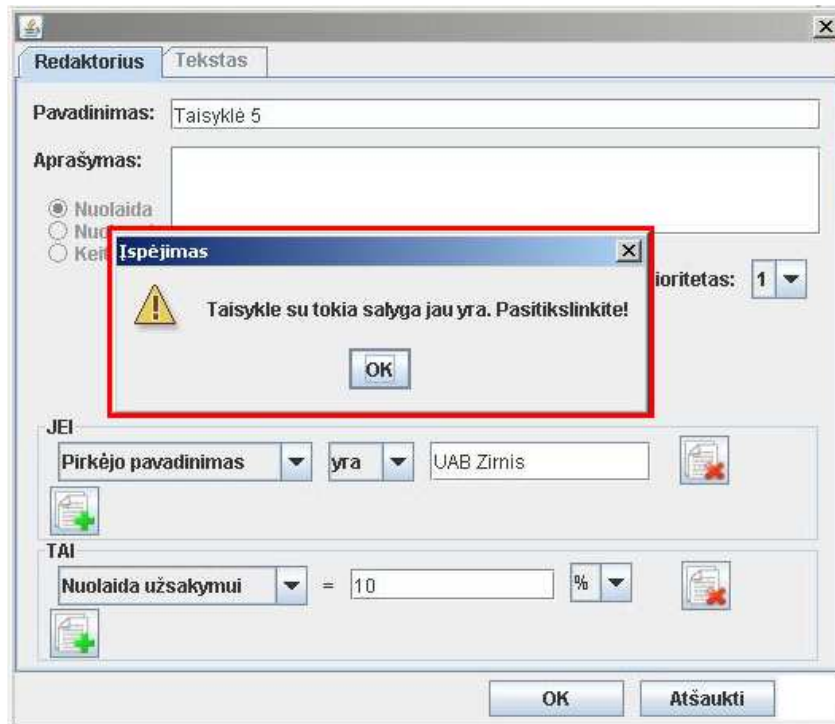


5.1 pav. Testavimo rezultatas 1

Matome, jog prie eilutės galutinės sumos parašius ne skaičių (*Integer*), o žodį (*String*) iššaukiamas pranešimas „Prekės eilutės suma turi būti skaičius“.

Taisyklių galimo dubliavimosi tikrinimas taip pat atliekamas nesudėtingai. Į sistemą suvedamos dvi taisyklės pateiktos lentelėje žemiau. „Paleidus“ sistemą, iškart buvo gautas pranešimas apie galimai nekorektiškas taisykles (žr. 5.2 pav.).

<i>Jei Pirkėjo vardas = „UAB Zirnīs „ Tai Prekės kiekis = Prekės kiekis - 10 Kg</i>	<i>Jei Pirkėjo vardas = „UAB Zirnīs“ Tai Prekės kaina = Prekės kaina - 10 %</i>
---	---



5.2 pav. Testavimo rezultatas 2

Matome, jog bandant antrą kartą įrašyti taisyklę su besikartojančia sąlygos dalimi (šiuo atveju *Jei Pirkėjo pavadinimas lygu „UAB Zirnīs“*) gavome pranešimą „Taisyklė su tokia sąlyga jau yra. Pasitikslinkite!“.

Taisyklių grupavimo šioje dalyje nebetirsime, kadangi aprašinėjant kontrolinį testavimo modelį (4.3 punktas) ištyrėme sistemos reakciją į tokio tipo metodų naudojimą. Matėme jog rezultatai tikrai atitinka išsikeltiems reikalavimams.

Metodas, kuris turėtų priskirti prioritetų laipsnius vartotojams nebuvo realizuotas, todėl situacijų, susijusių su šia funkcija netirsime, kadangi jokių rezultatų nebus gaunama.

Galiausiai lieka iširti taisyklių bei aritmetinių operacijų prioritetizavimo metodus. Šie metodai atsakingi už didžiąją dalį konfliktuojančių taisyklių, todėl siekiant tinkamai juos iširti panaudosime sudėtingesnių taisyklių, kurios tarpusavyje konfliktuotų.

<i>Jei Prekė = „Bulvės“ IR Nr.1</i> <i>Bendra suma už prekę ≤ 100 Lt</i> <i>Tai Prekės kaina = Prekės kaina - 50 %</i>	<i>Jei Prekė = „Bulvės“ IR Nr.2</i> <i>Bendra suma už prekę ≥ 50 Lt</i> <i>Tai Bendra suma už prekę = Bendra suma už prekę + 5 Lt</i>
<i>Jei Pirkėjo vardas = UAB Liepa IR Nr.3</i> <i>Prekė = „Morkos“</i> <i>Tai Prekės kaina = Prekės kaina + 10 Lt</i>	<i>Jei Prekė = „Morkos“ IR Nr.4</i> <i>Bendra suma už prekę ≥ 1000 Lt</i> <i>Tai Prekės kaina = Prekės kaina - 15 %</i>

Esant 100-ui Lt. už prekes ir vykdant taisyklę nr. 1 bus vykdoma ir taisyklė nr. 2, bet jei pirma vykdoma nr. 2 tai nr. 1 nebevykdoma. Taisyklės nr. 3 bei nr. 4 „operuoja“ skirtingais matų vienetais.

Rezultatas, gautas pirma vykdant 3-čiąją taisyklę, o po to tik 4-tąją pateiktas 5.3 paveiksle.

**Užsakymo eilutės** Valiuta LTL

Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL	Suma LTL	
4	5548	Morkos	500	KG	21	2	1.000	
Nuolaidos požymis							Suma	
<input checked="" type="checkbox"/>	Pritaikyta taisykle 4_Nuolaida_litais						10	
Nuolaidos požymis							Suma	
<input type="checkbox"/>	Pritaikyta taisykle 5_nuolaida_procentais						151.5	
Pridėti nuolaida							Iš viso LTL	1.000
Pridėti eilute							PVM LTL	210
							Suma su mokesčiais LTL	1.210
							Nuolaidų suma LTL	151,5
							Nuobaudų suma LTL	10
							Bendra suma LTL	1.068.5

5.3 pav. Testavimo rezultatas 3

Rezultatas, gautas pirma vykdant 4-tąją taisyklę, o po to tik 3-čiąją pateiktas 5.4 paveiksle.

**Užsakymo eilutės** Valiuta LTL

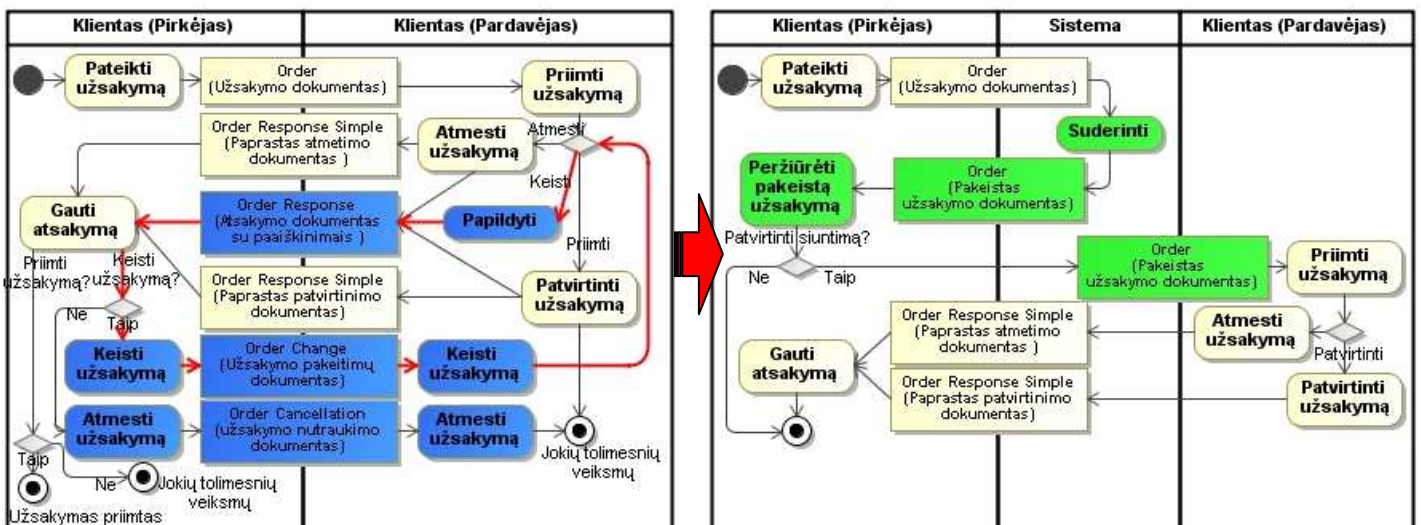
Nr.	Kodas	Pavadinimas	Kiekis	Mato vnt.	PVM [%]	Kaina LTL	Suma LTL	
4	5548	Morkos	500	KG	21	2	1.000	
Nuolaidos požymis							Suma	
<input type="checkbox"/>	Pritaikyta taisykle 5_nuolaida_procentais						150	
Nuolaidos požymis							Suma	
<input checked="" type="checkbox"/>	Pritaikyta taisykle 4_Nuolaida_litais						10	
Pridėti nuolaida							Iš viso LTL	1.000
Pridėti eilute							PVM LTL	210
							Suma su mokesčiais LTL	1.210
							Nuolaidų suma LTL	150
							Nuobaudų suma LTL	10
							Bendra suma LTL	1.070

5.4 pav. Testavimo rezultatas 4

Aritmetinių operacijų tyrimas labai komplikuoatas, nes tam reikia didelio kiekio taisyklių.

## 6.2. Sukurto metodo ir realizacijos apibendrinimas

Apibendrinant tyrimo metu suformuluotą metodą bei jo realizacijos prototipą galima teigti, jog pasiekti rezultatai gali iš esmės pagerinti SVV įmonių sąveikumą. Tiesa, norint pasiekti tokių rezultatų, būtina tinkamai naudoti verslo taisyklių suderinimo metodą, kadangi priešingu atveju rezultatai ne tik kad neduos teigiamos naudos, bet gali ir pakenkti. Bet jeigu taisyklės kuriamos apgalvotai bei atsakingai, jų naudojimas smarkiai pagreitina užsakymų sudarymą bei jų suderinimą atsižvelgiant į įmonių interesus. Metodo pagrįstumą galima įrodyti palyginus 2.5 skyrelyje pateiktą probleminę schemą (žr.2.1 pav.) su ta, kuri gaunama naudojant verslo taisykles (žr. 5.29 pav.).



5.29 pav. Verslo procesai nenaudojant ir naudojant verslo taisykles

Matome jog naudojant verslo taisykles įmonių sąveikume, galimai nebelieka pasikartojančio ciklo, reikalingo užsakymų koregavimui. Realizuotas sistemos prototipas tai padaro automatinio būdu (išimtiniais atvejais papildomi derinimai vis tiek reikalingi, todėl teigti, jog 100 procentų nebelieka ciklų, negalima). Tokį naudojamo metodo funkcionalumą apsprendžia įvairūs prioritetizavimai, grupavimai bei ignoravimai (pastarasis naudojamas atlikus įvairius tikrinimus). Schemoje mėlyna spalva išskirti procesai bei dokumentai, kurių (nevertinant išimtinų atvejų) nebelieka bendrame verslo procese, o žalia spalva išskirti elementai, kurie atsiranda papildomai.

Siekiant patikrinti sukurto metodo veiksmingumą bei korektiškumą buvo sukurta prototipinė sistemos modelio realizacija. Tam panaudota taisyklių sudarinėjimo posistemė, dokumento šablonas ir jo siuntimo posistemė bei pagrindinis suderinimo serveris. Taip pat buvo sukurta testavimo modelis, eksperimentiniai duomenys bei atliktas kontrolinis pavyzdys. Apibendrinant gautus rezultatus, galima teigti jog metodas veiksmingas, tačiau kiek efektyvus, be išsamesnių tyrimų, pasakyti gana sunku.

## 7. Išvados

Sukūrus ir įgyvendinus pasiūlytą verslo taisyklių suderinimo, smulkaus ir vidutinio verslo įmonių sąveikumo sprendimuose, metodą prieita prie šių išvadų.

1. Išsamios analizės metu nustatyta, jog priklausomai nuo regiono bei organizacijoje vykdomos veiklos, skiriasi jose naudojamų verslo dokumentų standartai. Iš to seka tokių dokumentų tarpusavio nesuderinamumas kas iššaukia neefektyvaus elektroninio bendradarbiavimo problemą. Dėl šios priežasties kyla poreikis ieškoti visuotinai priimtino standarto.

2. Automatinio verslo taisyklių suderinimo metodo panaudojimas smulkaus ir vidutinio verslo įmonių sąveikumo sprendimuose gali užtikrinti greitesnį verslo dokumentų apdorojimą, taigi ir greitesnį sandorio vyksmą.

3. Atlikta ERP sistemų įgyvendinimo ir eksploatavimo ypatumų analizė parodė, kad verslo taisyklių valdymas ERP sistemose, ypač sudarinėjant užsakymus, yra aktualus informacinių sistemų inžinerijos uždavinys. Be to, nustatyta, kad egzistuojančių sprendimų logikoje nėra realizuotas verslo taisyklių suvienodinimas, o tose kuriose yra, tai jų panaudojimas praktikoje šiuo metu reikalauja daug resursu ir yra tobulintinas.

4. Tyrimo metu nustatyta, kad naudojant specialiai sukurtą metodą, pagįstą visuotinai priimtu UBL standartu bei „DROOLS“ įrankiu, yra tinkamas būdas derybų taisyklių suderinimui įmonių sąveikumo sprendimuose įgyvendinti.

5. Pasiūlytas metodas sugeba verslo taisykles, išreikštas formalizuota forma, suderinti tarpusavyje taip, kad jos nepapultų į konfliktines situacijas. Tokį funkcionalumą apsprendžia prioritizavimai, grupavimai bei tikrinimai. Eksperimentas, atliktas su siūlomą metodą įgyvendinančia prototipine sistema, parodė, kad egzistuojant situacijoms, kuomet konfliktuoja verslo taisyklės, metodas sugeba jas suderinti ir pritaikyti reikiamiems dokumentams.

6. Eksperimentinio tyrimo rezultatai parodė siūlomo metodo privalumus bei trūkumus. Paaškėjo, jog siūlomo metodo pagrindu veikianti sistema ne tik sugeba suderinti verslo taisykles, bet ir sutaupo laiko, reikalingo sąveikaujant dviem ar daugiau įmonių. Be to, vartotojams suteikiama galimybė patiems kurti bei redaguoti verslo taisykles, kas leidžia tiesiogiai dalyvauti vystomoje verslo logikos politikoje.

## 8. Literatūra

1. Lietuvos respublikos vidaus reikalų ministerija: *Sąveikumas* [žiūrėta 2009 11 09]. Prieiga per internetą: < <http://www.vrm.lt/index.php?id=1405> >
2. *OASIS-Universal Business Language v2.1* [žiūrėta 2010 11 13]. Prieiga per internetą: <<http://docs.oasis-open.org/ubl/UBL-2.1.pdf>>, 25 September, 2010. p. 28 of 132
3. *Kiauleikis, Mindaugas; Kiauleikis, Valentinas; Guglielmina, Claudia; Tolle, Karsten; Morkevičius, Nerijus. Performance and architecture modeling of interoperability system for SME's // Business Information Systems. Vol. 7. Lecture Notes in Business Information Processing. Berlin : Springer, 2008. ISBN 978-3-540-79395-3. p. 345-456.*
4. *Application Bus for Interoperability in Enlarged Europe SMEs (ABILITIES)*, 027306. Assoc. Prof. V. Kiauleikis, Faculty of Informatics. 2006-2008.
5. *COIN Integrated Project- Enterprise Collaboration and Interoperability for Networked Enterprise* [žiūrėta 2010 01 12]. Prieiga per internetą: a) <<http://www.coin-ip.eu/>>; b) <<http://www.coin-ip.eu/research/coin-results/enterprise-collaboration/enterprise-collaboration>>;
6. *ATHENA Interoperability Framework (AIF)* [žiūrėta 2009 11 17]. Prieiga per internetą: <[http://www.txtgroup.com/newsletter/attachment/Athena\\_Paper.pdf](http://www.txtgroup.com/newsletter/attachment/Athena_Paper.pdf)>
7. *Enterprise Application Interoperability via Internet-integration for SMEs, Governmental Organisations and Intermediaries in the New European Union (GENESIS)* 027867. Prof. R. Gatautis, Faculty of Economics and Management. 2006-2008.
8. *OMG-CS European Union's GENESIS Project* [žiūrėta 2010 01 11]. Prieiga per internetą: <<http://www.bptrends.com/publicationfiles/04-09-OMG-CS-EU%27s-Genesis%20Project.doc.pdf>>
9. *eBiz-TCF: Architecture Details* [žiūrėta 2010 11 09]. Prieiga per internetą: <<http://spring.bologna.enea.it/ebiz/imple/pg.asp?lingua=en&p=296>>
10. *Extensible Markup Language (XML)* [žiūrėta 2010 11 09]. Prieiga per internetą: <<http://www.w3.org/XML/>>
11. *ebXML – Enabling a Global Electronic Market* [žiūrėta 2010 11 09]. Prieiga per internetą: < <http://www.ebxml.org/geninfo.htm> >
12. *OASIS Universal Business Language (UBL)* [žiūrėta 2010 01 10]. Prieiga per internetą: [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ubl](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl)
13. *SPML (Service Provisioning Markup Language)* [žiūrėta 2010 01 11]. Prieiga per internetą: <<http://www.networkworld.com/details/5623.html>>
14. *Rule Engine - IRODS -* [žiūrėta 2011 02 22]. Prieiga per internetą: <[https://www.irods.org/index.php/Rule\\_Engine](https://www.irods.org/index.php/Rule_Engine)>
15. *JBoss Community – „Drools“ The Business Logic integration Platform* [žiūrėta 2009 11 16]. Prieiga per internetą: <<http://www.jboss.org/drools/>>
16. *OpenRules - BDMS, Business Decision Management System* [žiūrėta 2011 02 22]. Prieiga per internetą: <<http://openrules.com/index.htm>>
17. *Open Lexicon* [žiūrėta 2009 11 17]. Prieiga per internetą: < <http://www.openlexicon.org/> >
18. *IBM Software – WebSphere* [žiūrėta 2009 11 17]. Prieiga per internetą: <<http://www-01.ibm.com/software/websphere/>>
19. *SweetRules Project Home Page* [žiūrėta 2009 11 18]. Prieiga per internetą: <<http://sweetrules.projects.semwebcentral.org/>>
20. *Žinių inžinerija ir sprendimų išvedimas – paskaitų medžiaga* [žiūrėta 2011 03 18]. Prieiga per internetą: <<ftp://isd.ktu.lt/Studentams/Butleris/Ziniu-inzinerija/Ziniu-inzinerija-ir-sprendimu-isvedimas-paskaitu-medziaga.pdf>>



21. Guglielmina, Claudia; Janavičiūtė, Audronė; Kiauleikis, Mindaugas; Kiauleikis, Valentinas; Morkevičius, Nerijus. *Performance modeling of interoperability system for SME's // Informacinės technologijos ir valdymas = Information technology and control / Kauno technologijos universitetas. ISSN 1392-124X. 2006, T. 35, nr. 4. p. 424-430.*

## Terminų ir santrumpų žodynas

- **ABILITIES** - IT sprendimas, skirtas SVV įmonių tarpusavio bendravimo galimybės pagerinti (angl. *Application Bus for Interoperability In enlarged Europe SMEs*).

- **CASE** – Kompiuterizuota instrumentinė sistema, skirta programų sistemoms kurti (angl. *Computer Aided Software Engineering System*)

- **DBVS** – duomenų bazių valdymo sistema

- **ebXML** - Praplečiamoji dokumentų aprašų kalba skirta elektroniniam verslui (angl. *Electronic Business using XML*)

- **EDI** – Elektroniniu dokumentų mainai (angl. *Electronic Data Interchange*)

- **ERP** – Verslo valdymo sistema (angl. *Enterprise Resource Planning*)

- **IS** – Informacinė sistema

- **IT** – Informacinės technologijos

- **Įmonių sąveikumas** – Veiklos sritis, kurios tikslas gerinti įmonių sąveikavimą su kitomis įmonėmis, organizacijomis ar kitais tos pačios organizacijos padaliniais tam, kad būtų galima plėsti verslą (angl. *Enterprise Interoperability*)

- **OCL** – Objektų ribojimų aprašymo kalba (angl. *Object Constraint Language*)

- **PĮ** – Programinė įranga

- **SQL** – Struktūrizuota užklausų kalba (angl. *Structured Query Language*)

- **SVV** – Smulkus ir vidutinis verslas

- **UBL** – Standartinė elektroninio verslo XML dokumentų, tokių kaip užsakymas ar sąskaitos-faktūros, biblioteka (angl. *Universal Business Language*).

- **UI** – Vartotojo sąsaja (angl. *User Interface*)

- **UML** – Unifikuota modeliavimo kalba (angl. *Unified Modelling Language*)

- **Užsakymas** – Raštu arba žodžiu užfiksuotas įsipareigojimas tam tikrų paslaugų ar produktų komercinis sandoris (angl. *Order*)

- **Verslo procesas** – Rinkinyje tarpusavyje susijusių užduočių, skirtų tam tikrai problemai spręsti (angl. *Business Process*)

- **XML** – Praplečiamoji dokumentų aprašų kalba (angl. *Extensible Markup Language*).

- **XSD** – Kalba skirta aprašyti XML dokumentų turinį (angl. *XML Schema Definition*)

## 9. Priedai

### 1 priedas: Order dokumento formos laukai

Pateiktoje lentelėje (žr. 10 lentelė) matomi visi galimi eksperimente naudojamo Order dokumento formos laukai. Atsižvelgiant į šiuos laukus galima kurti verslo taisykles. Lentelėje naudojami „B / P“ simboliai nurodo kokią įtaką pažymėtų laukų keitimas turės dokumentui. T.y „B“ – pasekmė daroma visam (bendram) dokumentui, „P“ – pasekmė daroma tik konkrečiai eilutei (prekei). Naudojami atskirimai turi įtakos metodo funkcionalumui pritaikant verslo taisykles.

10 Lentelė

B/ P	Formalus aprašas	Sutrumpinimas	Galima sąlyga/rezultatas (Jei „Suma“ -> Tai „Suma“ )
-	IssueDate	Paskelbimo data	Jei „pirkimo data = 11- 04 – 25“ ->
-	DocumentCurrencyCode	Valiuta	Jei „valiuta = LTL“ ->
----			
-	BuyerCustomerParty:Party:PartyName:Name	Pirkėjo pavadinimas	Jei „pirkėjo pavadinimas = Giraitė”
-	BuyerCustomerParty:Party:PostalAddress: StreetName	Pirkėjo gatvė	Jei „pirkėjo gatvė = Trakų“
-	BuyerCustomerParty:Party:PostalAddress: CityName	Pirkėjo miestas	Jei „pirkėjo miestas = Kaunas“
-	BuyerCustomerParty:Party:Contact: Name	Kontaktinio asmens vardas	Jei „kontaktinio asmens vardas = Jonas“
-	BuyerCustomerParty:Party:Contact: Telephone	Kontaktinio asmens tel.	
-	BuyerCustomerParty:Party:Contact: ElectronicMail	Kontaktinio asmens paštas	
----			
-	SellerSupplierParty:Party:PartyName:Name	Pardavėjo pavadinimas	Jei „pardavėjo pavadinimas = Giraitė”
-	SellerSupplierParty:Party:PostalAddress: StreetName	Pardavėjo gatvė	Jei „pardavėjo gatvė = Trakų“
-	SellerSupplierParty:Party:PostalAddress: CityName	Pardavėjo miestas	Jei „pirkėjo miestas = Kaunas“
----			
P	OrderLine:LineItem:Item:CatalogueItemIdentification:ID	Prekės kodas	Jei „prekės kodas = 2123“
P	OrderLine:LineItem:Item:Name	Prekės pavadinimas	Jei „prekės pavadinimas = bulvės“
P	OrderLine:LineItem:Quantity	Prekės kiekis	Jei „prekės kiekis = 2123“ ->
P	OrderLine:LineItem:Quantity:unitCode	Prekės mato vienetai	Jei „prekės mato vienetai = kg.“ ->
P	OrderLine:LineItem:Item:ClassifiedTaxCategory:Percent	Prekės PVM (%)	
P	OrderLine:LineItem:Price:PriceAmount	Prekės kaina	Jei „prekės kaina > 50“ ->
P	OrderLine:LineItem:LineExtensionAmount	Bendra suma už prekę	Jei „bendra suma už prekę > 500“ ->
P	OrderLine:LineItem:AllowanceCharge:ChargeIndicator	Nuolaidos požymis	
P	OrderLine:LineItem:AllowanceCharge: AllowanceChargeReason	Nuolaidos/nuobaudos priežastis	
P	OrderLine:LineItem:AllowanceCharge:Amount	Nuolaidos suma	Jei „nuobauda >20“ ->
----			
B	AnticipatedMonetaryTotal: TaxExclusiveAmount	Iš viso suma	Jei „viso suka >500“ ->
B	TaxTotal:TaxAmount	Iš viso PVM suma	Jei „viso PVM suma > 50“ ->
B	AnticipatedMonetaryTotal:TaxInclusiveAmount	Suma su mokesčiais	Jei „suma su mokesčiais >500“ ->
B	AnticipatedMonetaryTotal:AllowanceTotalAmount	Bendra nuolaidų suma	Jei „nuolaidų suma >50“ ->
B	AnticipatedMonetaryTotal:ChargeTotalAmount	Bendra nuobaudų suma	Jei „nuobaudų suma >50“ ->
B	AnticipatedMonetaryTotal:PayableAmount	Bendra suma	Jei „bendra suma > 500“ ->

## 2 priedas: Taisyklės naudojamos eksperimentiniame tyrime

//--Rodomas galimas taisyklių variantas--//

1 Pvz.: Taisyklė Nr.1 (Neigiamo poveikio taisyklės pavyzdys)

Jei Pirkėjo vardas = UAB Giraitė **IR**  
Prekė = „Bulvės“  
Tai  $Prekės\_kaina = Prekės\_kaina + 10\text{ Lt}$

1 Pvz.: Taisyklė Nr.2 (Teigiamo poveikio taisyklės pavyzdys)

Jei Prekė = „Bulvės“ **IR**  
Bendra suma už prekę  $\geq 1000\text{ Lt}$   
Tai  $Prekės\_kaina = Prekės\_kaina - 15\text{ Lt}$

---

//--Rodomas 1 eiliškumo pvz.-skirtingi vienetai--//

2 Pvz.: Taisyklė Nr.3 (Neigiamo poveikio taisyklės pavyzdys su litais)

Jei Pirkėjo vardas = UAB Giraitė **IR**  
Prekė = „Bulvės“  
Tai  $Prekės\_kaina = Prekės\_kaina + 10\text{ Lt}$

2 Pvz.: Taisyklė Nr.4 (Neigiamo poveikio taisyklės pavyzdys su procentais)

Jei Prekė = „Bulvės“ **IR**  
Bendra suma už prekę  $\geq 1000\text{ Lt}$   
Tai  $Prekės\_kaina = Prekės\_kaina - 15\%$

---

//--Rodomas 2 eiliškumo pvz.- tikrinami tie patys laukai --//

3 Pvz.: Taisyklė Nr.5 (Jei vykdoma 5 tai vykdoma ir 6)

Jei Prekė = „Bulvės“ **IR**  
Bendra suma už prekę  $\geq 100\text{ Lt}$   
Tai  $Prekės\_kaina = Prekės\_kaina - 50\%$

3 Pvz.: Taisyklė Nr.6 (Jei vykdoma 6 tai nevykdoma 5)

Jei Prekė = „Bulvės“ **IR**  
Bendra suma už prekę  $\leq 50\text{ Lt}$   
Tai  $Bendra\_suma\_už\_prekę = Bendra\_suma\_už\_prekę + 15\text{ Lt}$

---

//--Dviejų grupių taisyklės--//

4 Pvz.: Taisyklė Nr.7 (Taisyklė tik prekei-vykdoma pirmiau)

Jei Pirkėjo vardas = UAB Giraitė **IR**  
Prekė = „Bulvės“  
Tai  $Prekės\_kaina = Prekės\_kaina + 0,5\text{ Lt}$

4 Pvz.: Taisyklė Nr.8 (Taisyklė visam dokumentui)

Jei Pirkėjo vardas = „UAB Giraite“  
Tai  $Bendra\_suma = Bendra\_suma - 15\%$