



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Rytis Rainys

**IFS IR „READSOFT“ SISTEMŲ INTEGRAVIMO GALIMYBIŲ
TYRIMAS**

Baigiamasis magistro projektas

Vadovas
doc. Rita Butkienė

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

**IFS IR „READSOFT“ SISTEMŲ INTEGRAVIMO GALIMYBIŲ
TYRIMAS**

Baigiamasis magistro projektas
Informacinių sistemų inžinerijos studijų programa (kodas 621E15001)

Vadovė

doc. Rita Butkienė
2016-05-20

Recenzentas

lekt. dr. Jaroslav Karpovič
2016-05-20

Projektą atliko

Rytis Rainys
2016-05-20



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

Rytis Rainys

(Studento vardas, pavardė)

Informacinių sistemų inžinerijos studijų programa, 621E15001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“
AKADEMINIO SAŽINGUMO DEKLARACIJA

20 16 m. gegužės 20 d.
Kaunas

Patvirtinu, kad mano, **Ryčio Rainio**, baigiamasis projektas tema „IFS“ IR „READSOFT“ SISTEMŲ INTEGRAVIMO GALIMYBIŲ TYRIMAS“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nėra viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Rytis Rainys. „IFS“ ir „ReadSoft“ sistemų integravimo galimybių tyrimas. Magistro baigiamasis projektas / vadovė doc. Rita Butkienė; Kauno technologijos universitetas, Informatikos fakultetas.
Mokslo kryptis ir sritis: Informatikos inžinerija, technologijos mokslai
Reikšminiai žodžiai: *ReadSoft, IFS, integracija, Kitron, ESB*
Kaunas, 2016. 65 p.

SANTRAUKA

Šiame darbe nagrinėjamos integracijos galimybės tarp automatizuotos elektroninių sąskaitų nuskaitymo sistemos „ReadSoft“ ir verslo valdymo sistemos „IFS“. Integracijos poreikis atsirado nustačius tyrimo objekte, „Kitron“ grupėje, iškilusią problemą - visiškai automatizuoto elektroninių sąskaitų apdorojimo proceso nebuvimą. Tyrimo tikslas yra, išsiaiškinti įmonės sąskaitų apdorojimo galimybes, pateikti sprendimą, kaip būtų galima efektyviau išnaudoti turimus įmonės IT resursus elektroninių sąskaitų apdorojimo procesams paspartinti.

Tyrimo metu buvo išanalizuotos „Kitron“ grupėje naudojamos sistemos, tada palyginti metodai, kuriais galima susieti „IFS APP 8“ ir „ReadSoft“ sistemas. Pateiktos rekomendacijos, kada geriausia įmonei naudotis viena ar kita integracija.

Išanalizavus įdiegiamų sistemų privalumus ir trūkumus buvo priimtas sprendimas naudotis „Kitron“ grupėje naudojama integracine platforma „Kitron Integration Bus“, kuri yra paremta „Enterprise Service Bus“ (ESB) principais. Sprendimas buvo priimtas palyginus rinkoje esančias integracines sistemas pagal svarbiausius kriterijus: paprastumą, administravimo galimybes, produkto bendruomenes, palaikymą, funkcionalumą, lankstumą, plečiamumą, integracinius metodus, licencijavimą ir kainą.

Sprendžiant visiškai automatizuoto elektroninių sąskaitų apdorojimo proceso nebuvimo problemą „ReadSoft“ sistema buvo įdiegta į „Kitron“ grupės IT aplinką ir pritaikyta labiausiai tiriamai įmonei tinkanti integracija - „Kitron Integration Bus“. Įdiegus ir integravus sistemas, jos buvo išbandytos testavimo būdu ir perkeltos į gamybinę aplinką.

Ekspерimento rezultatai parodė, kad visi elektroninių sąskaitų duomenys buvo teisingai perduoti, tik kai kuriais atvejais buvo reikalingas buhalterio patikslinimas, kuriuose laukuose yra nurodyti apdorjami sąskaitos duomenys, ir sąskaitos buvo apdorojimos du kartus greičiau.

Rainys Rytis. “IFS“ AND „READSOFT“ SYSTEMS INTEGRATION CAPABILITIES
RESEARCH

: Master's thesis in Information Systems Engineering / supervisor prof. Rita Butkienė. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics Engineering, Technology Science

Key words: *ReadSoft, IFS, integration, Kitron, ESB*

Kaunas, 2016. 65 p.

SUMMARY

In this work we research integration capabilities between automatic invoice handling system “ReadSoft” and Enterprise Resource Planning system “IFS”. This research was initiated after analysis of problematic topic- absence of fully automated invoice handling system in research object- “Kitron group”. The purpose of this research is analyze current invoice handling processes and propose a solution, how could you use current company IT resources to speed up current invoice handling process.

In this research we analyzed systems of “Kitron” group, then researched integration methods and after comparison we decided which one is best suited for integration between “IFS APP 8” and “ReadSoft” systems. After that recommendations were formed when it is best for the companies to use one or another integration method.

After analysis of integration methods, it was decided to use “Kitron group” used integration platform “Kitron Integration Bus”, which is created based on “Enterprise Service Bus” (ESB) principals. This decision was made after comparison of integration systems offered in the global market, based on these criteria's: ease of use, maintainability and monitoring, community, enterprise support, functionality, flexibility, extensibility, connectors, cost and licensing.

After designing of integration solution, “ReadSoft” was implemented in “Kitron group” IT environment and integrated with internal integration platform- “Kitron Integration Bus”. After all installations and integration was complete, testing was performed and whole solution was transferred to production environment.

Experiment was conducted and it showed that all invoice data was transferred from “ReadSoft” to “IFS” correctly, just in some cases there was a need for accountant input for recognizing data fields, but other than that invoice handling process was faster than previous method more than two times.

TURINYS

Lentelių sąrašas	8
Paveikslų sąrašas	9
Terminų ir santrumpų žodynas	10
Įvadas	11
1. Probleminės srities analizė	13
1.1. Analizės tikslas	13
1.2. Tyrimo objektas, sritis ir problema	13
1.3. Tiriamų objektų „IFS APP 8“, „ReadSoft“ ir „Kitron Integration Bus“ analizė	13
1.3.1. „IFS APP 8“ sistema	13
1.3.2. „ReadSoft“ sistema	14
1.3.3. „Kitron Integration Bus“ integracinė platforma	16
1.3.4. Esamos situacijos sąskaitų apdorojimo procesas	19
1.4. „ReadSoft“ ir „IFS“ sistemų naudotojų analizė	21
1.5. Esamų integracinių sprendimų metodų analizė (Lietuvos ir tarptautiniu mastu)	21
1.5.1. Oracle Service Bus	23
1.5.2. Mule ESB	23
1.5.3. Fuse ESB	24
1.5.4. Talend ESB	24
1.5.5. Microsoft BizTalk	24
1.5.6. TIBCO ActiveMatrix BusinessWorks 6	25
1.5.7. IBM Sterling B2B Integrator	25
1.5.8. WebMethods Integration	25
1.5.9. Esamų sistemų lyginimoji analizė	26
1.6. Siekiamo sprendimo apibrėžimas	28
1.7. Analizės išvados	29
2. Integracijos reikalavimų specifikacija ir projektas, formalus aprašas	30
2.1. Reikalavimų specifikacija	30
2.2. Dalykinės srities modelis	32
2.3. Formalus sprendimo aprašas	34
2.4. Reikalavimų apibendrinimas	35
3. ReadSOFT diegimo ir realizacijos projektas	36
3.1. Programos	36
3.2. Periferinė įranga ir infrastruktūra	38
3.3. Servisai	40
3.4. Konfigūraciniai failai	40
4. Sprendimo realizacija ir testavimas	42

4.1. Sprendimo realizacijos ir veikimo aprašas	42
4.2. Testavimo modelis, duomenys, rezultatai.....	45
5. Eksperimentinis Integracinio sprendimo tyrimas	48
5.1. Eksperimento planas	48
5.2. Eksperimento rezultatai	48
5.3. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas	50
5.4. Sprendimo taikymo rekomendacijos.....	50
6. Rezultatų apibendrinimas ir išvados	51
7. Bibliografija	52
8. PRIEDAI.....	54
8.1. Kitron Integration Bus adapteris XML failams nuskaityti	54
8.2. <i>ReadSoft sugeneruojamo XML failo pavyzdys</i>	56
8.3. Apdorojamos sąskaitos pavyzdys	58
8.4. „ReadSoft“ sistemos sugeneruotas <i>XML dokumentas</i>	59
8.5. PowerShell kopijavimo kodas.....	60

LENTELIŲ SĄRAŠAS

1 lentelė. Licencijuotų ir atviro kodo integracinių produktų palyginimas.....	26
2 lentelė. Esamų sprendimų palyginimas	28
3 lentelė . „ReadSoft“ įskiepai	37
4 lentelė. „ReadSoft“ sistemos naujajamos duomenų bazės.....	38
5 lentelė. „ReadSoft“ serverių sąrašas	38
6 lentelė. „ReadSoft“ serveryje veikiantys servisai	40
7 lentelė. „ReadSoft“ konfigūracinių failų sąrašas.....	41
8 lentelė. „ReadSoft“ sistemai sukuri servisiniai vartotojai	41
9 lentelė. „ReadSoft“ nuskaitytų sąskaitų sąrašas.....	43
10 lentelė. Eksperimento pirmos dalies rezultatai	46
11 lentelė. Senojo sąskaitų apdorojimo proceso rezultatai.....	48
12 lentelė. Naujojo sąskaitų apdorojimo proceso rezultatai.....	49

PAVEIKSLŲ SĄRAŠAS

1 pav. „IFS Applications“ moduliai	14
2 pav. „Kitron“ elektroninių sąskaitų gavimo dėžutės.....	15
3 pav. „ScanInfo“ įskiepio langas.....	15
4 pav. „Kitron Integration Bus“ koncepcija	17
5 pav. KIB duomenų perdavimo eilė.....	17
6 pav. KIB adapterio veikimo principas.....	18
7 pav. Buhalterio PA prieš proceso automatizavimą	19
8 pav. Neautomatizuoto proceso sekų diagrama	19
9 pav. Neautomatizuoto proceso veiklos diagrama	20
10 pav. Produktai pagal integracijos sudėtingumą	22
11 pav. Integracijos reikalavimų diagrama	30
12 pav. Buhalterio PA po proceso automatizavimo.....	31
13 pav. Automatizuoto sąskaitos apdorojimo proceso sekų diagrama	31
14 pav. Pradinio duomenų srauto struktūra.....	32
15 pav. KIB integracijos komponentų diagrama.....	37
16 pav. Integracinio sprendimo diegimo diagrama	39
17 pav. KIB „EsbConfigManager“ klasių diagrama	32
18 pav. KIB „DeliveryScheduler“ klasių diagrama	33
19 pav. KIB „DeliveryRequestProcessor“ sekų diagrama	33
20 pav. KIB „DeliveryScheduler“ sekų diagrama	34
21 pav. Automatizuoto proceso veiklos diagrama	20
22 pav. Gaunama elektroninė sąskaitą į ReadSoft palaikomą pašto dėžutę	42
23 pav. „ReadSoft“ sistemoje atvaizduojamas gautos sąskaitos	43
24 pav. „ReadSoft“ sistemos automatinis duomenų atpažinimas	43
25 pav. „IFS External Supplier Invoices“ modulio sąskaitų atrinkimo langas	44
26 pav. Gautos sąskaitos IFS sistemoje iš KIB.....	44
27 pav. „IFS“ sistemoje sukuriamos apmokėjimo sąskaitos	45
28 pav. Suformuota sąskaita „IFS“ sistemoje kuri laukia apmokėjimo patvirtinimo	45
29 pav. Apdorojamos sąskaitos testinis pavyzdys.....	Error! Bookmark not defined.
30 pav. Pirmimis „ReadSoft Invoices“ diegimo langas	Error! Bookmark not defined.
31 pav. Papildomi diegimo moduliai.....	Error! Bookmark not defined.
32 pav. „ReadSoft Invoices 5-6 R2“ diegimo pradžios langas	Error! Bookmark not defined.
33 pav. „ReadSoft Invoices“ licenzijos sutarties sutikimas	Error! Bookmark not defined.
34 pav. „ReadSoft Invoices“ serverio tipo parinkimas	Error! Bookmark not defined.
35 pav. „ReadSoft Invoices“ duomenų bazės prisijungimo duomenys.....	Error! Bookmark not defined.
36 pav. „ReadSoft Invoices“ sistemos kalbos pasirinkimas.....	Error! Bookmark not defined.
37 pav. „ReadSoft Invoices“ skaitytuvo jungties metodo pasirinkimas	Error! Bookmark not defined.
38 pav. „ReadSoft Invoices“ komponentų diegimo pasirinkimas	Error! Bookmark not defined.
39 pav. „ReadSoft Invoices“ diegimo patvirtinimo langas	Error! Bookmark not defined.
40 pav. „ReadSoft Invoices“ diegimo vedlio pabaiga.....	Error! Bookmark not defined.
41 pav. „ReadSoft ScanInfo“ diegimo vietos pasirinkimas	Error! Bookmark not defined.
42 pav. „ReadSoft ScanInfo“ diegimo langas	Error! Bookmark not defined.
43 pav. „ReadSoft InvoiceRules“ diegimo vietos pasirinkimas.....	Error! Bookmark not defined.
44 pav. „ReadSoft InvoiceRules“ diegimo langas.....	Error! Bookmark not defined.
45 pav. „Kitron Validations“ diegimas terminale, administratoriaus teisėmis	Error! Bookmark not defined.
46 pav. „Kitron Validations“ .dll failų registracija per terminalą.....	Error! Bookmark not defined.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

- API- (angl. *Application Programming Interface*)- tai sąsaja kurią suteikia kompiuterinė sistema, biblioteka, ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsikeistų su ja duomenimis. [3]
- BPEL- (angl. *Business Process Execution Language*) verslo procesų valdymo kalba.
- CRM- (angl. *Customer Relationship Management*)- būdas analizuojant ir panaudojant rinkodaros duomenų bankų duomenis bei įdarbinant komunikacines technologijas sukurti bendrą įmonės praktiką ir metodus, kurie maksimaliai padidintų kiekvieno individualaus kliento ilgalaikę vertę įmonei.
- EMS- (angl. *Electronics Manufacturing Solutions*)- elektronikos gamybos produktai.
- ERP- (angl. *Enterprise Resource Planning*)- tai verslo valdymo sistema, dažniausiai integruotų sistemų paketas.
- ESB- (angl. *Enterprise Service Bus*)- programinės architektūros modelis naudojamas kuriant ir diegiant duomenų perdavimo ryšį tarp informacinių sistemų.
- HTTP- (angl. *HyperText Transfer Protocol*)- protokolas kuris pateikia standartinį būdą HTML puslapių skelbimui ir skaitymui. [4]
- IMAP- (angl. *Internet Message Access Protocol*) interneto pranešimų prieigos protokolas.
- KIB- (*Kitron Integration Bus*)- Integracinė platforma paremta ESB architektūros modeliu.
- OFM- (*Oracle Fusion Middleware*)- „Oracle“ kompanijos sukurtas integracinis paketas.
- PA- panaudojimo atvejis.
- PDF- (angl. *Portable Document Format*)- tai atviro standarto formatas, skirtas elektroniniam dvimačiam dokumentui atvaizduoti. [2]
- PVM- pridėtinės vertės mokestis.
- SCA- (angl. *Service Component Architecture*) servisų komponentų architektūra.
- SOA- (angl. *Service-Oriented Architecture*)- tai į programinius servišus orientuota programų sistemos architektūra.
- SOAP- (angl. *Simple Object Access Protocol*) paprasto objekto prieigos protokolas.
- XML- (angl. *Extensive Markup Language*)- tai „World Wide Web Consortium“ rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba. [1]

IVADAS

Šis darbas priklauso informacinių sistemų inžinerijos specializacijos sričiai. Darbe bus analizuojamos dvi informacinės sistemos, kurios bus įdiegtos įmonėje UAB „Kitron“. Pirmoji sistema, kuri bus įdiegta, yra „ReadSoft“. „ReadSoft“ yra „Lexmark“ kompanijai priklausantis produktas, skirtas automatizuoti elektroninių sąskaitų apmokėjimo procesą. Pasaulyje yra daug įmonių, kurios naudoja šią sistemą, pavyzdžiui, Europos branduolinių tyrimų organizacija „CERN“, kurios metinė finansinė apyvarta viršija milijardą Šveicarijos frankų. Antroji sistema, kuri bus įdiegta, yra „IFS APP 8“. „IFS“ teikia sprendimus finansų ir resursų valdymo srityje. Sistema sudaryta iš daugybės modulių, kuriuos reikia įsigyti atskirai, tačiau modulinis funkcionalumas leidžia įmonėms įsigyti tik tai, ko joms reikia. „IFS APP 8“ nėra pati naujausia programos versija šiuo metu rinkoje, tačiau „IFS“ modulių funkcionalumas yra toks pats, kaip ir naujausioje programos versijoje.

Tyrimo metu bus analizuojama, kokiais metodais galima integruoti minėtas sistemas ir geriausios integracijos prototipas bus realizuotas įmonės aplinkoje. Šiuo metu buhalterijos ir pirkimų skyrių darbuotojams kyla sąskaitų apdorojimo problemų, nes sąskaitų, gaunamų į elektroninio pašto dėžutę, kiekis yra didesnis negu 120 per dieną, ir tos sąskaitos privalo būti patikrintos pirkimo ir buhalterijos skyrių „IFS“ sistemoje (ar tiekėjas yra įvestas į „IFS“ sistemą, ar kaina atitinka užsakymo kainą, ar pirkimas buvo patvirtintas specialisto, kuris yra atsakingas už pirkimus). Taip pat „IFS“ sistemoje sąskaitą reikia užregistruoti ir persiųsti buhalterijai, kad būtų atliktos banko operacijos. Taigi, atliekant paprastus skaitymo ir įrašymo veiksmus, apdorojant tokį didelį kiekį sąskaitų šiuo metodu yra neefektyviai gaišamas specialisto darbo laikas. Atliekant tyrimą bus įdiegiama „ReadSoft“ sistema UAB „Kitron“ IT infrastruktūroje ir integruojama su verslo valdymo sistema „IFS“. Tai turėtų sumažinti įmonės darbuotojų sugaištą laiką, skiriamą sąskaitoms apdoroti bei sumažinti akies klaidų skaičių, atsiradusį apdorojant sąskaitas.

Darbo problematika ir aktualumas

Šiuo metu įmonės darbuotojai kasdien gauna daugybę sąskaitų iš įvairių tiekėjų, kurias reikia apdoroti „IFS“ sistemoje – nustatyti tiekėją, patikrinti, ar tiekėjas yra įregistruotas sistemoje, jei jo nėra, reikia pereiti prie naujo tiekėjo įregistravimo proceso, patikrinti, ar sąskaitos suma atitinka sutartą užsakymo sumą, ar užsakymas buvo patvirtintas, ar sąskaitos apmokėjimo terminas atitinka sutartus terminus ir kitus parametrus. Yra daugybė punktų, kuriuos turi įvykdyti darbuotojas norėdamas apmokėti gautą sąskaitą. O apdorojimo klaida gali tapti išaugusių sąskaitų kainų, laiku neapmokėtų sąskaitų, švaistomų pinigų ir darbuotojų laiko ar kitų problemų, kurios menkina įmonės įvaizdį bei mažina pelną, priežastimi.

Darbo tikslas ir uždaviniai

Darbo tikslas yra padidinti įmonės turimų sistemų efektyvumą, sumažinant buhalterio darbo, susijusio su elektroninių sąskaitų apdorojimu, krūvį. Šiam tikslui pasiekti reikia išspręsti uždavinius:

1. Išanalizuoti, kaip šiuo metu yra apdorojamos gautos elektroninės sąskaitos.
2. Išanalizuoti, kaip galima įdiegti „ReadSoft“ ir „IFS“ sistemas.
3. Sukurti projektą, kuris leistų perduoti duomenis tarp įdiegtų sistemų.
4. Realizuoti integracijos prototipą įmonės aplinkoje.
5. Atlikti eksperimentinį tyrimą ir išsiaiškinti, ar pagreitėjo elektroninių sąskaitų apdorojimo procesas.

Darbo rezultatai ir jų svarba

Šio darbo metu tikimasi sukurti efektyvesnę sąskaitų apdorojimo sistemą, kurią naudojant sąskaitos bus apdorojamos dvigubai greičiau negu dabar, kai buhalterio darbas atliekamas ranka. Įmonei labai svarbu sukurti efektyvesnę sąskaitų apdorojimo sistemą, nes šios įmonės, kaip ir kiekvienos gamybos įmonės, tikslas yra optimizuoti veiklos procesus ir kiek įmanoma labiau išnaudoti

ribotus resursus. Todėl šio darbo rezultatai padarys didelę įtaką įmonės sąskaitų apdorojimo laikui ir kainai ne tik Lietuvoje, bet ir kitose pasaulio šalyse.

Darbo struktūra

Šio darbo pirmajame skyriuje aprašomas šio darbo tikslas, tyrimo objektas ir probleminė sritis. Svarbiausi yra 1.3.- 1.6. poskyriai, kuriuose analizuojama „Kitron“ grupės įmonės turimos sistemos, probleminės srities vartotojai, ieškomas geriausias būdas įmonei integruoti „IFS“ ir „ReadSoft“ sistemas, išrinkti geriausi produktai rinkoje ir jie palyginti. Tada yra nustatyta geriausiai įmonei tinkama integracija, suformuluotas siekiamas sprendimo apibrėžimas ir padarytos analizės išvados.

Antrajame skyriuje yra nustatomi funkciniai ir nefunkciniai reikalavimai pasirinktai integracinei sistemai ir aprašomi dalykinės srities naudotojų sąsajos modeliai. Tada suformuluojamas formalus sprendimo aprašas ir pateikiamas reikalavimų apibendrinimas.

Trečiajame skyriuje yra aprašomas „ReadSoft“ diegimo bei integracijos tarp „Kitron Integration Bus“ projektas. Jame numatomos naudojamos programos, periferinė įranga ir infrastruktūra, duomenų bazės, servisai, konfigūraciniai failai, sisteminiai vartotojai, diegimo planas.

Ketvirtajame skyriuje yra aprašoma, kaip vyko visa sprendimo realizacija ir kaip buvo testuojama įdiegta sistema.

Penktajame skyriuje aprašomas atliktas eksperimentinis tyrimas, ar atlikti pakeitimai pasiekė iškeltą tikslą. 5.2. poskyryje aprašomi gauti rezultatai ir 5.3. poskyryje analizuojami gauti rezultatai bei penktojo skyriaus pabaigoje pateiktos sprendimo taikymo rekomendacijos.

Šeštajame skyriuje apibendrinami gauti rezultatai ir pateikiamos išvados.

Septintajame skyriuje yra nurodomas naudotos literatūros sąrašas.

Aštuntajame skyriuje pateikiamas sukurtas „Kitron Integration Bus“ adapteris „ReadSoft“ XML failams nuskaityti. Taip pat aprašomi „ReadSoft“ sistemos sugeneruojami XML failų pavyzdžiai bei pateikiamas apdorojamos elektroninės sąskaitos pavyzdys.

1. PROBLEMINĖS SRITIES ANALIZĖ

1.1. Analizės tikslas

Šios analizės tikslas yra išsiaiškinti esamus „Kitron“ grupės įmonės sąskaitų apdorojimo procesus ir pateikti sprendimą, kaip būtų galima efektyviau išnaudoti turimus įmonės IT resursus dabartiniams apdorojimo procesams paspartinti.

1.2. Tyrimo objektas, sritis ir problema

„Kitron“ įmonių grupė – Skandinavijos elektronikos gamybos paslaugų tiekėja (*EMS*), kurios padaliniai įkurti ne tik Europoje, bet ir JAV.

„Kitron“ įmonių grupės klientai – tai didelės tarptautinės, pasaulyje pripažintos kompanijos Azijoje, kurioms tiekiami sudėtingi elektronikos produktai. „Kitron“ sukaupe daugiau kaip 50 metų patirtį. UAB „Kitron“ įsikūrė 2001 metais, išaugo į didžiulę, su daugiau nei 380 darbuotojų įmonę, kurios padaliniai yra įsikūrę Kaune bei Užliedžiuose. Bendrai Kauno padaliniai užima 15000 m² plotą. „Kitron“ grupė teikia įvairias paslaugas:

- naujų produktų kūrimo, vystymo,
- industrializacijos,
- pirkimų ir tiekimo,
- gamybos,
- logistikos ir distribucijos,
- dizaino pakeitimo ir remonto.

Atlikto tyrimo metu išaiškėjo įmonėje esanti problema – reikalingas ilgas ir kruopštus žmogaus darbas norint apdoroti ir suvesti į sistemą gautą elektroninę sąskaitą. Gaunamų sąskaitų kiekis kiekvieną mėnesį vis didėja, todėl reikalingas sprendimas, kaip būtų įmanoma sąskaitas apdoroti efektyviau, kad kuo mažiau į sąskaitų apdorojimo procesą būtų įtrauktas įmonės darbuotojas ir būtų išvengta akies klaidų.

Tyrimo objektu buvo pasirinktas „Kitron“ grupėje naudojamų verslo valdymo sistemos „IFS APP 8“ ir dokumentų apdorojimo sistemos „ReadSoft“ integravimas. Pasirinkto tyrimo sritis - metodai, įrankiai ir technologijos programinėms integracijoms atlikti.

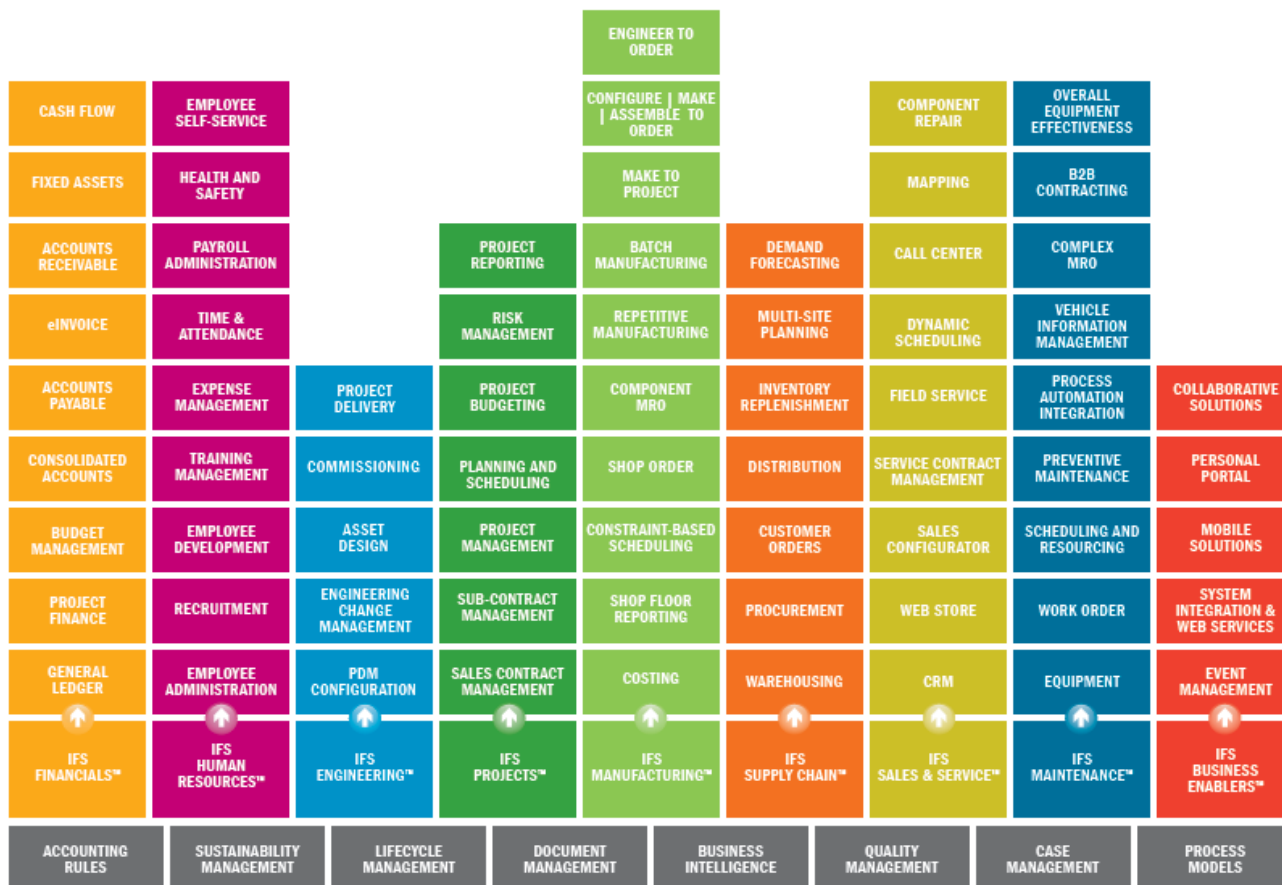
1.3. Tiriamų objektų „IFS APP 8“, „ReadSoft“ ir „Kitron Integration Bus“ analizė

Šiame skyriuje analizuojamos diegiamos informacinės sistemos ir galimi šių sistemų integraciniai metodai.

1.3.1. „IFS APP 8“ sistema

„IFS AB“ (*Industrial and Financial Systems*) yra globalios programinės įrangos kompanija, kuri yra įsikūrusi Linköping mieste Švedijoje ir turi daugiau nei 50 padalinių visame pasaulyje. „IFS“ suteikia komponentais paremtą išplėstą *ERP* (angl. *Enterprise Resource Planning*) sistemą, sukurtą pagal *SOA* (angl. *Service-Oriented Architecture*) technologiją. [5] „IFS“ pagrindinis produktas yra „IFS Applications“ verslo valdymo sistema, kuri susideda iš tarpusavyje integruotų modulių.

„IFS Applications“ apima finansus, žmogiškuosius išteklius, kokybės kontrolę, dokumentų tvarkymą, santykių su klientais valdymą (angl. *Customer Relationship Management, CRM*), verslo analitiką (angl. *Business Intelligence, BI*) ir kitas pagrindines funkcijas, skirtas aptarnauti verslo valdymo poreikius. [6]. 1 paveikslėlyje pateiktas „IFS“ modulių sąrašas. Kiekvieno modulio aprašymą anglų kalba galima rasti gamintojo svetainėje [7].



1 pav. „IFS Applications“ moduliai

Šiuo metu yra išleista naujausia programos versija „IFS APP 8“.

Darbiui aktualiausias „IFS“ sistemos modulis yra „External Supplier Invoices“, kuris yra atsakingas už gautų sąskaitos duomenų apdorojimą. Pasinaudojus modulyje esančia paieška ir jos filtrais, galima surasti reikiamas sąskaitas. Taip pat per šį modulį atliekami patvirtinimai sąskaitoms apmokėti.

Prieš automatizuojant sistemą, buhalteris turėjo atidaryti elektroninę sąskaitą ir suvesti jos duomenis į „IFS“ sistemą. Ne visada sąskaitos laukai buvo nuskaitomi ir nebūdavo įmanoma nukopijuoti sąskaitoje esančių duomenų. Taip atsirasdavo galimybė suklysti įrašant duomenis: sąskaitos datą, užsakymo numerį, sąskaitos apmokėjimo datą, sąskaitos numerį, sumą, tiekėją, PVM dydį ir kitus laukus.

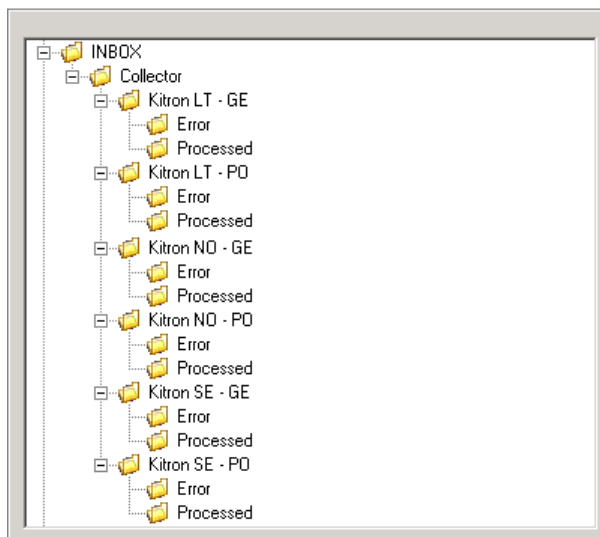
1.3.2. „ReadSoft“ sistema

„ReadSoft sistema sudaryta iš kelių įskiepių (angl. *plugins*) [8]: „COLLECTOR“, „ScanInfo“, „InvoiceRules“, „KitronValidations“ ir „ImageInXML“. Šie įrankiai yra vienas su kitu susieti, atlieka tam tikrą funkciją ir sudaro bendrą „ReadSoft“ automatizuotą elektroninių sąskaitų apdorojimo sprendimą. Elektroninės sąskaitos yra nuskaitomos ir kai gaunama to paties tiekėjo sąskaita, sistema mokosi, kurioje vietoje buvo tam tikri laukai, kurie yra suvedami į sistemą. Kuo daugiau sąskaitų yra apdorojama tuo tiksliau yra parenkami sąskaitos duomenys.

1.3.2.1. „COLLECTOR“ įskiepis

„COLLECTOR“ yra „ReadSoft“ sistemos įskiepis, kuris automatiškai apdoroja sąskaitas, gautas elektroniniu paštu. Įskiepis naudojamas „Interpret“ modulio ir yra konfigūruojamas „ReadSoft Manager“ programa. Visi nustatymai yra atskiroje duomenų bazėje, kuri yra aprašyta COLLECTOR.ini faile. Jį galima rasti „ReadSoft“ įrašymo direktorijoje.

„Kitron“ aplinkoje yra viena elektroninio pašto konfigūracija su keliomis el. pašto paskyromis, kurias nuolat nuskaito „COLLECTOR“ įskiepis, naudojantis *IMAP* protokolą.

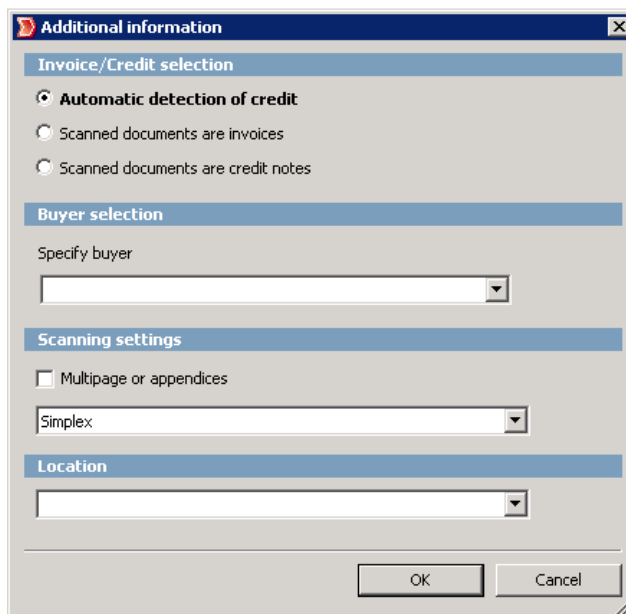


2 pav. „Kitron“ elektroninių sąskaitų gavimo dėžutės

1.3.2.2. „ScanInfo“ įskiepis

„ScanInfo“ yra „ReadSoft“ sistemos įskiepis, naudojamas sąskaitų metaduomenims nustatyti bei reguliuoti programinio skaitytuvo nustatymus. Šis įskiepis naudojamas „Scan“ ir „Interpret“ modulių bei konfigūruojamas „ReadSoft Manager“ programa. Visi įskiepio nustatymai yra ScanInfoConfiguration.xml faile, kurį galima rasti „ReadSoft“ įrašymo direktorijoje.

„Kitron“ aplinkoje „ScanInfo“ yra naudojamas nustatyti pirkėją, šalį ir jos padalinį, lapo nustatymus (pvz.: viengubas, dvigubas lapas) bei leisti nustatyti rankinį sąskaitos duomenų parinkimą.



3 pav. „ScanInfo“ įskiepio langas

1.3.2.3. „InvoiceRules“ įskiepis

„InvoiceRules“ yra „ReadSoft“ sistemos įskiepis, naudojamas tam tikriems veiksams pagal tam tikrus požymius atlikti. Šis įskiepis naudojamas visuose „ReadSoft“ moduluose, išskyrus „Optimize“. Šis įskiepis konfigūruojamas „ReadSoft Manager“ programa, nustatymai yra InvoiceRules.xml faile, kurį galima rasti „ReadSoft“ įrašymo direktorijoje.

„Kitron“ sistemoje „InvoiceRules“ atlieka kelis veiksmus:

- ištrina viską, išskyrus pirmus septynis simbolius iš užsakymo numerio,
- ant kreditinių sąskaitų prideda minuso simbolį prie PVM ir sumos be PVM,
- nustato * simbolį kaip pagrindinį Švedijos pirkimų sąskaitų nurodymų numerį,
- ištrina `NO` ir `MVA` iš PVM registracijos numerių Norvegiškose sąskaitose.

1.3.2.4. „Kitron Validations“ įskiepis

„Kitron Validations“ yra įskiepis, sukurtas „ReadSoft Norway“ kompanijos, suskaičiuoti PVM dydį ir bendrą sumą. Įskiepis aktyvus „Verify“ modulyje ir konfigūruojamas „ReadSoft Manager“ programa. Visi nustatymai yra Kitron_Validations.xml faile, „ReadSoft“ įrašymo direktorijoje.

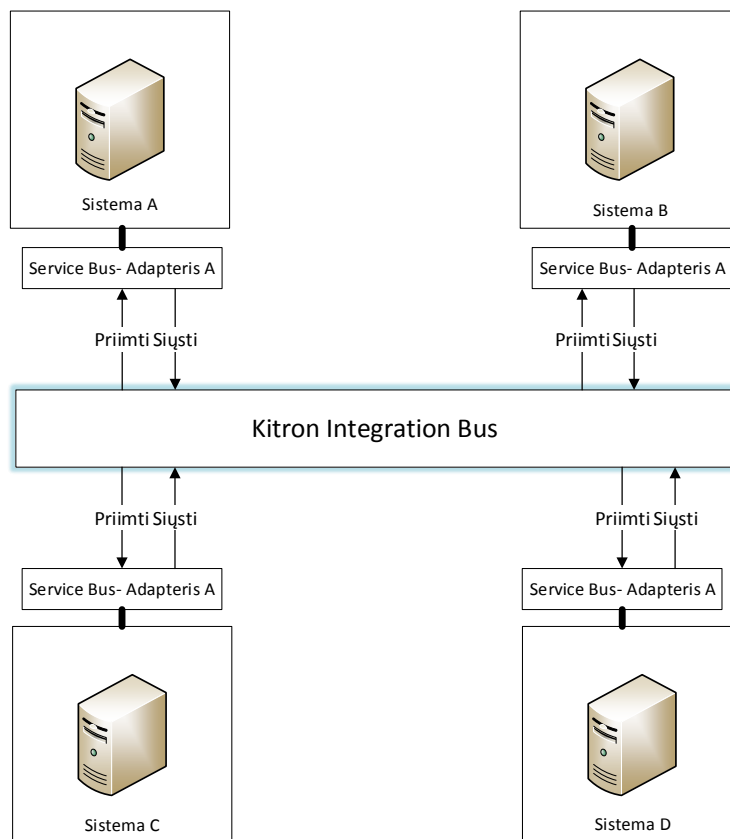
1.3.2.5. „ImageInXML“ įskiepis

„ImageInXML“ yra „ReadSoft“ sistemos įskiepis naudojamas sukurti BASE64 kodavimo eilutę iš sąskaitos atvaizdo failo ir pridėti tą eilutę prie sukuriama XML failo. Įskiepis aktyvus „Transfer“ modulyje ir konfigūruojamas naudojant „ReadSoft Manager“ programą. Visi nustatymai yra patalpinti ImageInXML.xml faile, kurį galima rasti „ReadSoft“ įrašymo direktorijoje.

1.3.3. „Kitron Integration Bus“ integracinė platforma

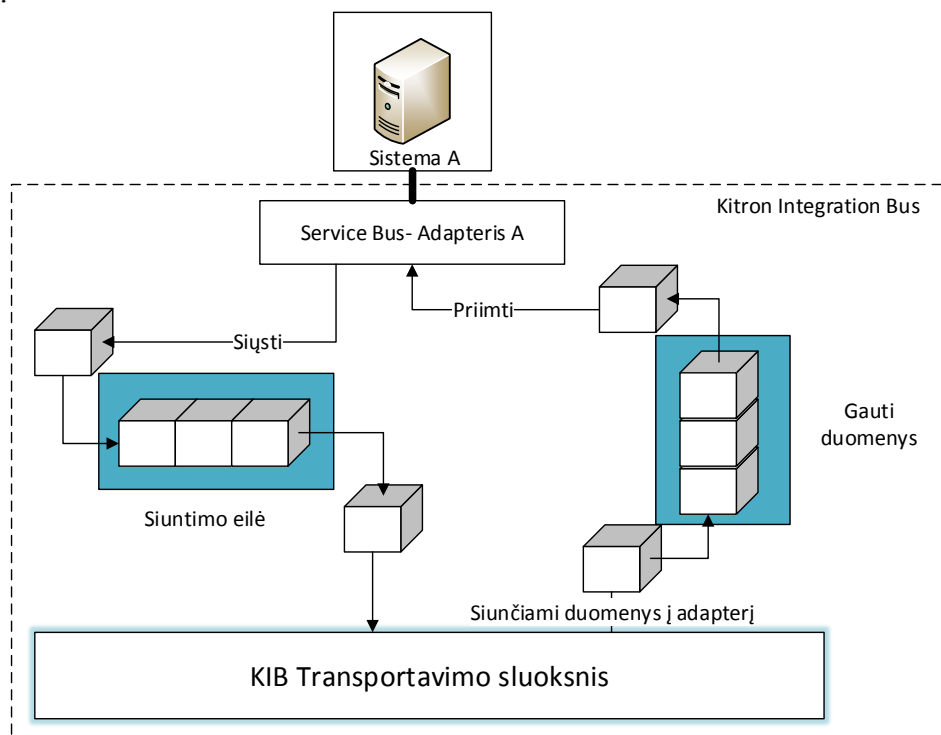
„Kitron Integration Bus“ (KIB)- integracinis sluoksnis įmonės IT aplinkoje, kuris yra sukurtas ESB (angl. Enterprise Service Bus) principais [9]. Ši platforma veikia „Altova“ serveryje „Kitron“ IT infrastruktūroje. Tai nėra pilna ESB platforma, tik jos elementai pritaikyti pagal „Kitron“ grupės poreikius pranešimams tarp turimų informacinių sistemų apdoroti.

KIB integracija yra realizuojama virtualioje, ne periferinėje IT sistemoje. Jos tikslas - realiu laiku perduoti informaciją tarp sistemų naudojantis „Publish and Subscribe“ žinučių perdavimo sistema, kuri pavaizduota 4 paveikslėlyje. Duomenų savininkas visada bus duomenis teikianti sistema. Kiekviena duomenų reikalaujanti sistema gali būti sukonfigūruota pagal tam tikras taisykles ir logiką. Bet kokia sistema gali reikalauti duomenų.



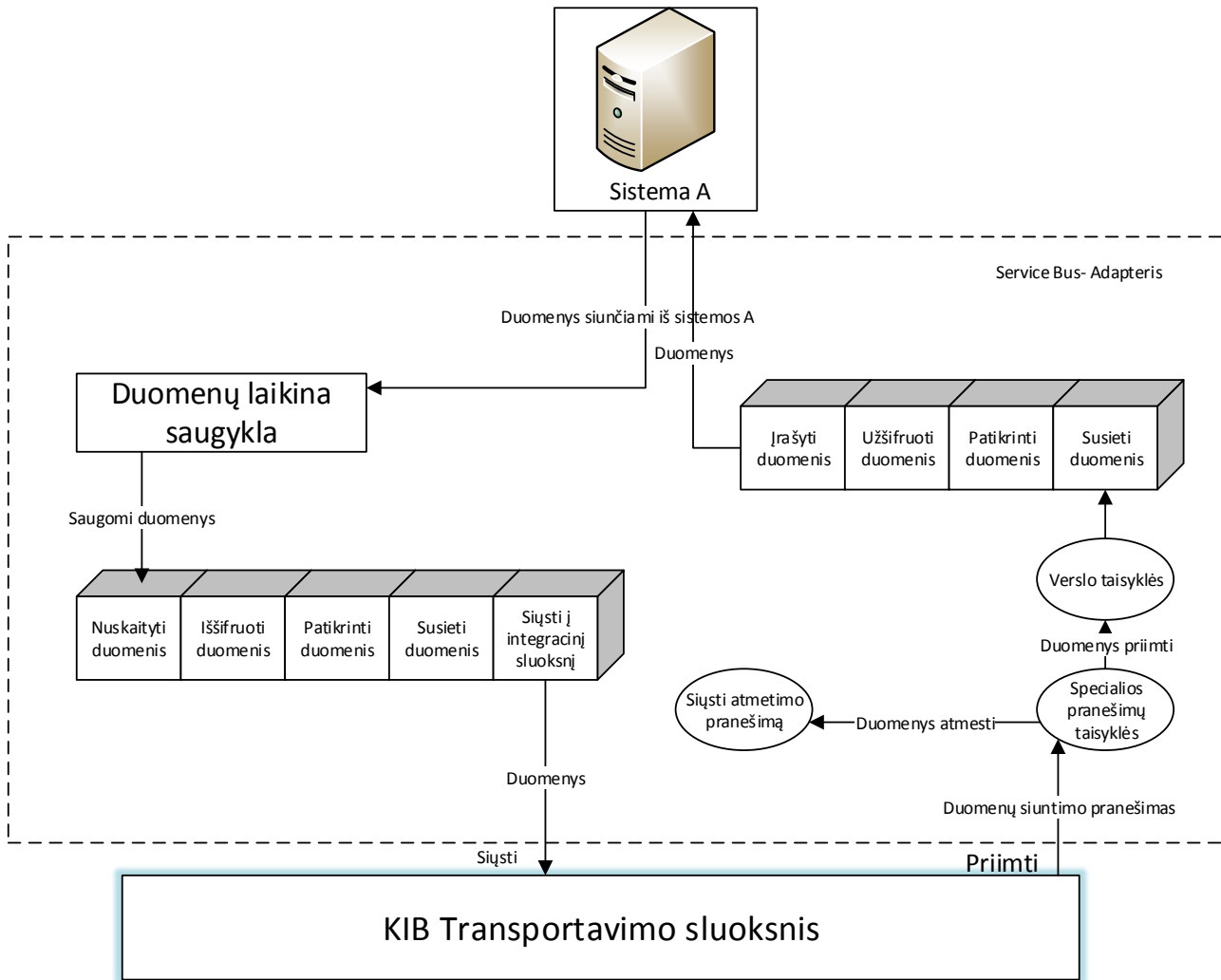
4 pav. „Kitron Integration Bus“ koncepcija

Tokią architektūrą geriausia naudoti norint minimizuoti nuotolius, jeigu viena iš sistemų būtų atjungta. *KIB* užtikrina, kad visos žinutės nebūtų ištrintos iš užsakymo eilės, kol visos žinutės nebus gražintos užsakovams su atsakymais, nes integracijos transportavimo sluoksnyje yra saugoma duomenų eilė, kuri užtikrina, visų užklausų įvykdymą. Veikimo principas pavaizduotas 5 paveikslėlyje.



5 pav. *KIB* duomenų perdavimo eilė

Taip užtikrinamas užklauso duomenų išsaugojimas, jei dingus internetui arba sutrikus sistemai buvo sutrukdytas stabilus sistemos veikimas, o pašalinus problemą, visi duomenys nukeliamas užsakiusioms sistemoms. Tai ypač svarbu įmonei, nes dėl padalinių geografinės vietos atsiranda didelis tinklo delsos laikas (angl. *latency*). Taip pat ši architektūra parankesnė, jeigu norima panaudoti tą patį duomenų adapterį įvairioms sistemoms – nebūtina kaskart kurti naujus adapterius. Adapterio veikimas pavaizduotas 6 paveikslėlyje. Naudojantis *KIB* yra paprasta plėsti arba mažinti sistemą. Be to, ši platforma turi apdorotų duomenų atsekamumą. Taip sukuriama galimybė nustatyti, kurioje vietoje informacija galėjo būti perduota neteisingai arba nesuveikė kažkuris adapteris. Adapterio kodo pavyzdį *XML* failams iš „ReadSoft“ sistemos skaityti galima rasti šio darbo priede 8.2 punkte.



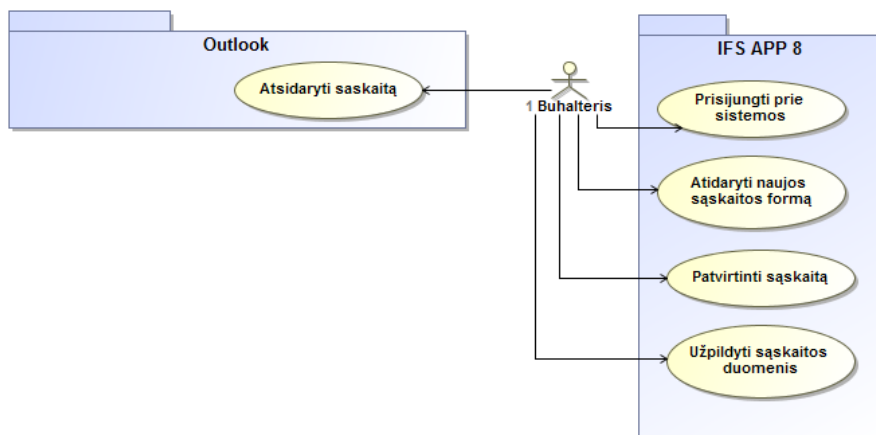
6 pav. KIB adapterio veikimo principas

6-8 paveikslėliuose yra pavaizduotas bendras integracijos platformos veikimo principas. Integruojama sistema siunčia užklausą (žinutę) į *KIB*, kuri atskiria, kuriam adapteriui ši žinutė yra skirta. Žinutė nusiunčiama į užklauso eilę. Atėjus užklauso vykdymo laikui, užklauso yra pritaikomi adapterio veikimo principai. Tada duomenys palyginami, patikrinami, ar jie yra teisingi, užšifruojami ir užrašomi gaunančios sistemos suprantamu formatu bei perduodami pranešimą gaunančiai sistemai. Gaunanti sistema atlieka užklausą ir atsiunčia duomenis atgal adapteriui apdoroti, kuriame duomenys perskaitomi, iššifruojami, patikrinami, palyginami ir išsiunčiami *KIB* užklauso davusiai sistemai, kur vyksta toks pats iššifravimo, palyginimo procesas.

Toks problemos sprendimas leidžia atsekti perlaidų duomenis, užtikrina veiklos kokybę. Sistema veikia paprastai, todėl užtenka vidinių įmonės IT kompetencijų integracijai tarp sistemų įgyvendinti ir duomenims tarp tiriamųjų sistemų perduoti. Taip pat ši platforma jau yra įdiegta „Kitron“ IT infrastruktūroje ir nereikia papildomo laiko ar kaštų integracijai įgyvendinti.

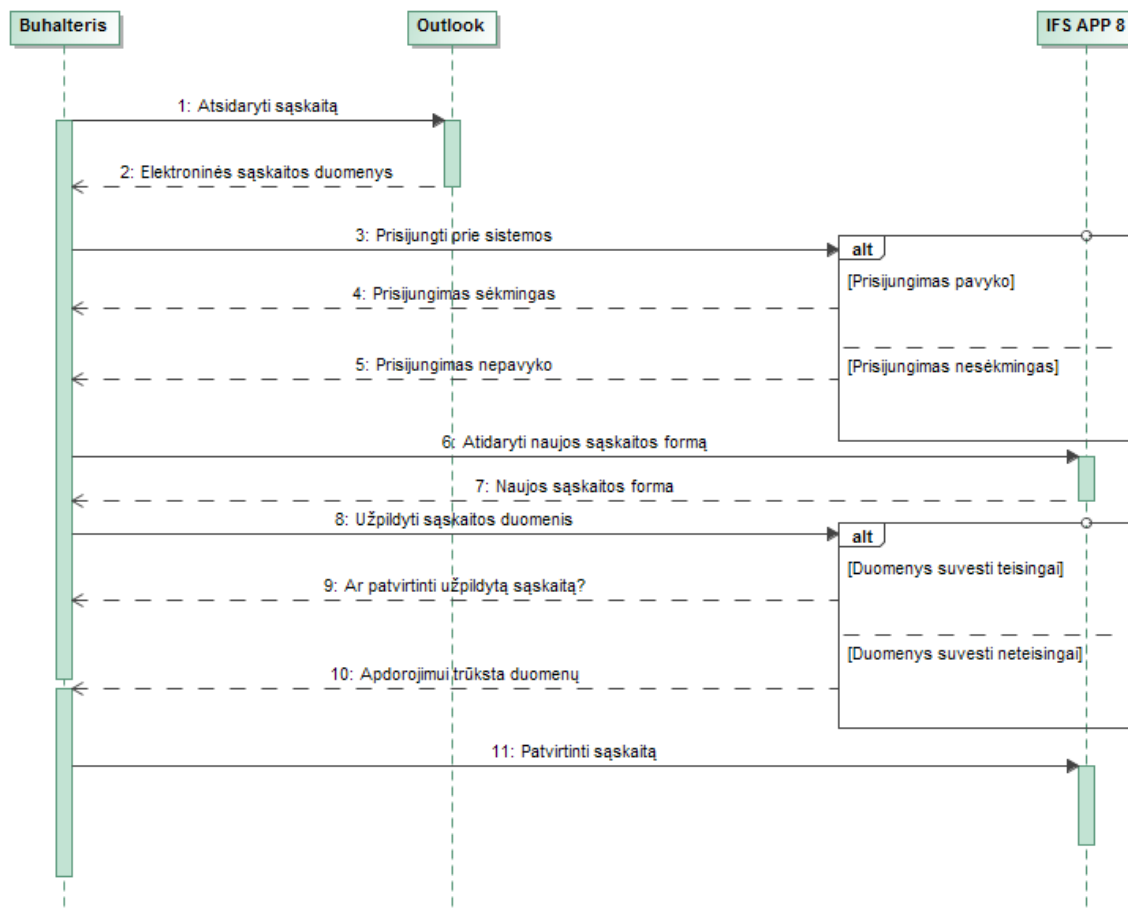
1.3.4. Esamos situacijos sąskaitų apdorojimo procesas

Buvo stebimas dabartinis sąskaitų apdorojimo procesas, kaip yra šiuo metu apmokamos gautos elektroninės sąskaitos. Proceso metu nustatyti sąskaitų apdorojimo atvejai, kurie pavaizduoti 7 paveikslėlyje.



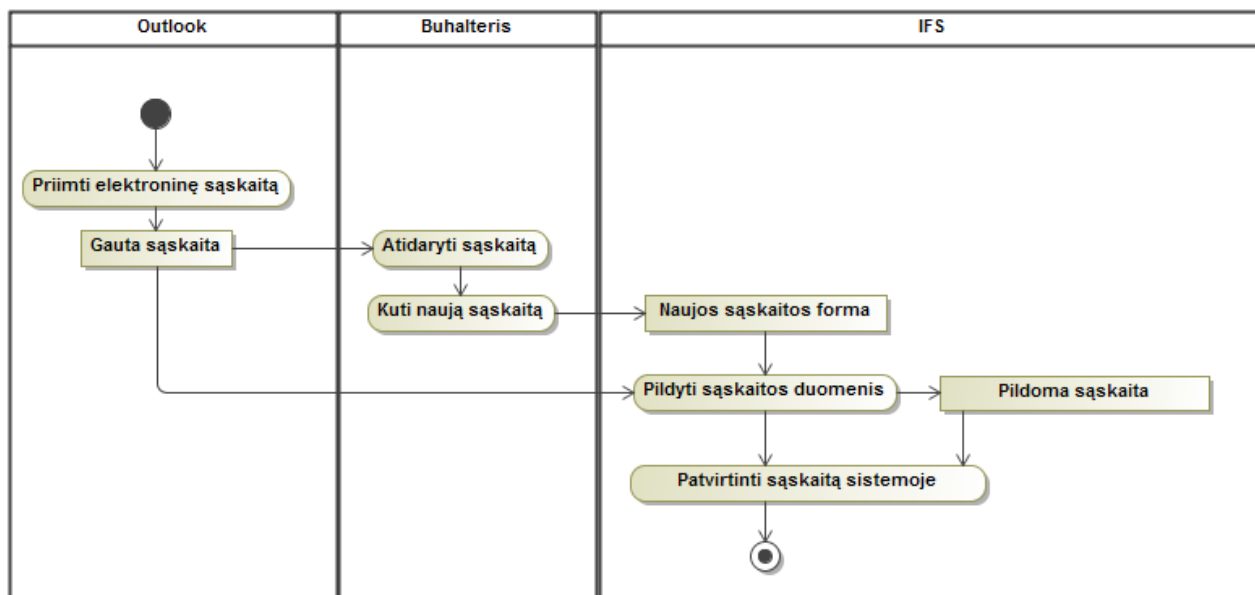
7 pav. Buhalterio atlikamos funkcijos prieš proceso automatizavimą

8 paveikslėlyje pavaizduota sekų diagrama, nurodanti neautomatizuoto sąskaitų apdorojimo procesą. Pavaizduota, kaip buhalteris, atsidaręs elektroninę sąskaitą, privalo ranka kopijuoti duomenis ir įkelti juos į „IFS“ sistemą. Šis procesas nėra efektyvus ir lemia daug akies klaidų, ypač jeigu duomenys yra įvedami išsiblaškiusio arba pavargusio žmogaus.



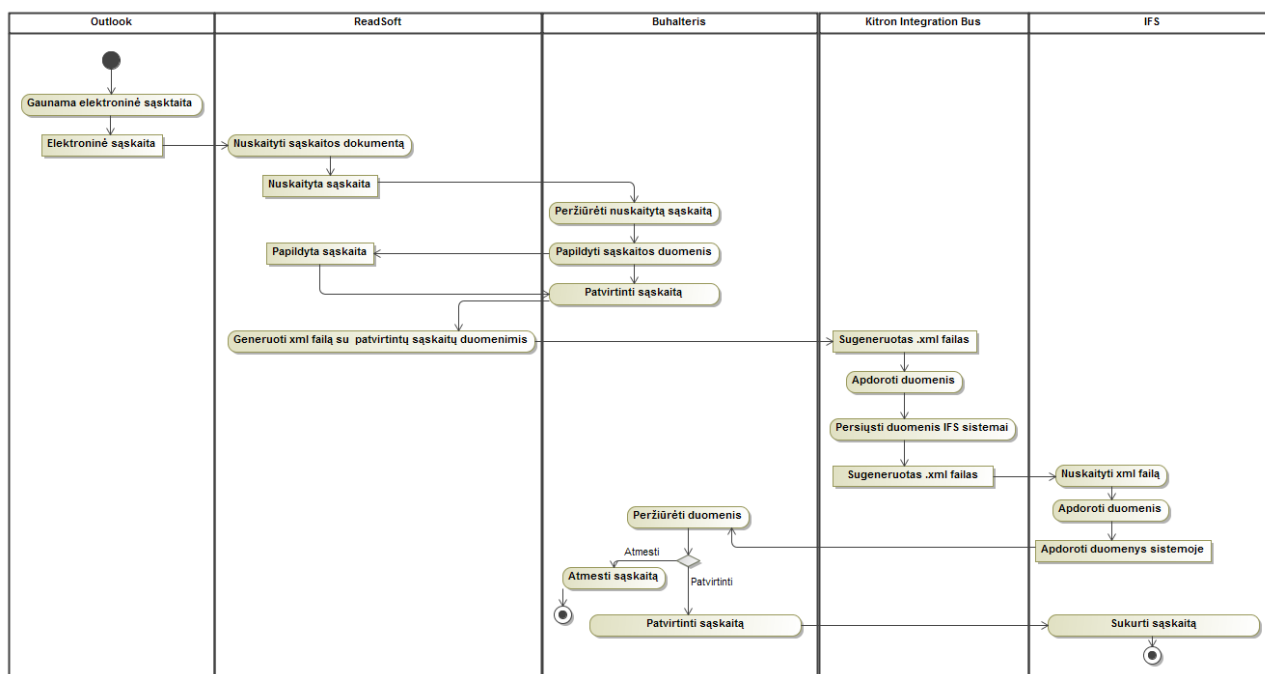
8 pav. Esamos proceso veiksmų sekos

9 paveikslėlyje pavaizduota proceso veiklos diagrama. Buhalteris „Outlook“ sistemoje atsidaro sąskaitos nuotrauką. Tada „IFS“ sistemoje, naujos sąskaitos formoje yra įvedami sąskaitos duomenys iš „Outlook“ sistemoje atidarytos sąskaitos ir, jeigu visi duomenys užpildyti teisingai, sąskaitą galima išsaugoti „IFS“ sistemoje.



9 pav. Neautomatizuoto proceso veiklos diagrama

10 paveikslėlyje pavaizduota automatizuoto proceso veiklos diagramą. Skirtingai nei senajame procese, šioje diagramoje atsiranda „ReadSoft“ ir „Kitron Integration Bus“ takeliai. Pagrindiniai objektai yra elektroninė sąskaita, nuskaityti sąskaitos duomenys, XML formate esantys duomenys, kuriais manipuliuoja integruojamos sistemos bei integracija tarp jų.



10 pav. Automatizuoto proceso veiklos diagrama

1.4. „ReadSoft“ ir „IFS“ sistemų naudotojų analizė

„ReadSoft“ ir „IFS“ naudotojai yra finansų skyriuje dirbantys buhalteriai ir finansų analitikai, kurie kasdien analizuoja ir apdoroja elektronines sąskaitas. Šie asmenys yra atsakingi už įmonės finansų valdymą: atlieka pervedimus tiekėjams, rašomo sąskaitas-faktūras, vykdo vidinį įmonės finansų auditą, apskaičiuoja atlyginimus, pildo ataskaitas apie komandiruočių pinigus, veda įmonės balansą, teikia ataskaitas valstybinėms institucijoms apie įmonės finansinę atskaitomybę. Finansų skyriaus darbuotojai negali vykdyti didelių piniginių operacijų ir tvirtinti balanso be finansų skyriaus vadovo patvirtinimo. Šių darbuotojų kompetenciją atspindi įgytas aukštasis ekonomikos ir finansų specialisto išsilavinimas bei ilgalaikis darbo stažas šioje srityje. Atlikdami daugumą savo veiklų jie naudojami „IFS APP 8“ sistema, nes ten yra visi įmonės finansiniai duomenys, susiję su numatomais pirkimais, atliktais pardavimais ir visa buhalterinė apskaita.

Šio darbo metu, išanalizavus elektroninių sąskaitų apdorojimo rutiną, buvo nustatyta, jog sąskaitų apmokėjimo procesas yra iš dalies automatizuotas. Į verslo valdymo sistemą „IFS APP 8“ įvedus pavadinimą, kuris išrašinėja sąskaitas įmonei, jo pagrindiniai duomenys yra atvaizduojami sąskaitų apmokėjimo modulyje, tačiau visą kitą sąskaitos informaciją buhalteris įveda ranka. Tai yra papildomas darbas buhalteriu, kuris yra atsakinga už pervedimus. Suprantama, kad apdorojant didelį kiekį sąskaitų, darbuotojas gali jausti stresą dėl didelio krūvio ir galimos klaidos įvedant duomenis, kas lemia ilgesnį sąskaitos apdorojimo laiką ieškant klaidos bei neapmokėtas arba permokėtas sąskaitas ir finansinius nuostolius, kurie menkina įmonės įvaizdį.

1.5. Esamų integracinių sprendimų metodų analizė (Lietuvos ir tarptautiniu mastu)

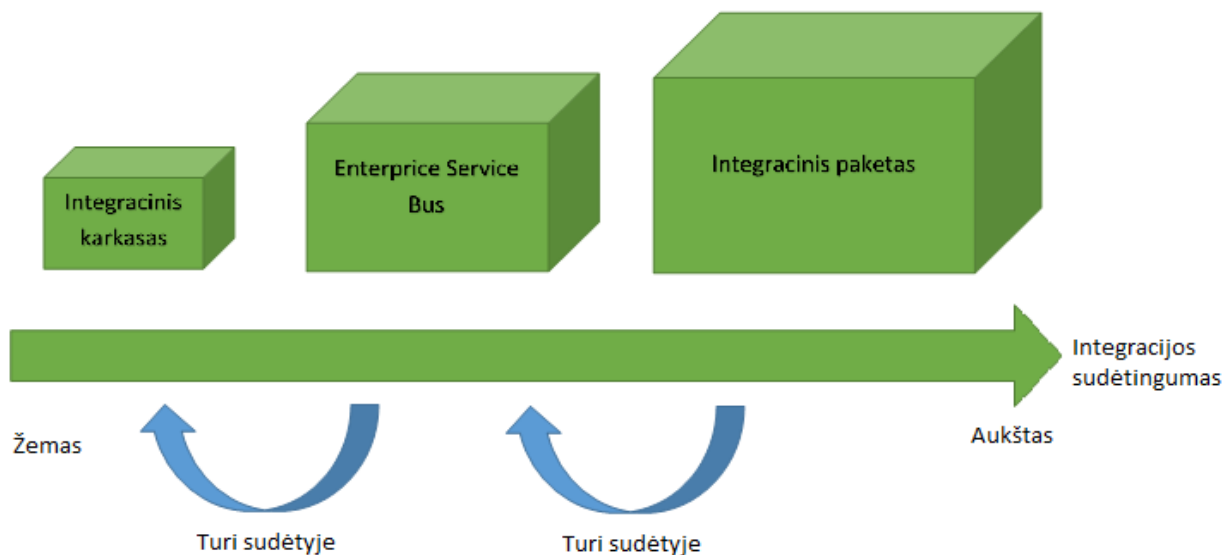
Ieškant sprendimų, kaip būtų įmanoma integruoti tiriamas sistemas, buvo kreiptasi į „IFS“ ir „ReadSoft“ kompanijų vadybininkus ir klausinama, ar yra sukurta integracija, kuri užtikrintų automatinį duomenų apsikeitimą tarp sistemų. Galbūt informacinių sistemų projektuotojų jau yra sukūrę geriausias sprendimus. Deja, vadybininkai nerado klientų, kurie analizuojamų sistemų savo IT infrastruktūroje, todėl buvo eita kitu keliu ir ieškoma tarpinės grandies, užtikrinančios saugų, greitą, funkcionalų, patikimą bei atsekamą duomenų perdavimą tarp sistemų.

Bendraujant su „ReadSoft“ sistemos inžinieriais buvo nustatyta, jog mažiausia nusiskundimų iš klientų, kurie naudoja XML failų integraciją. Ši integracija yra paprasčiau taikoma ir daugybę sistemų palaiko duomenų apsikeitimus XML formatu. Šiai integracijai, kaip alternatyvą, galima pasirinkti duomenų bazių integraciją, tačiau „ReadSoft“ sistemos inžinieriai perspėja apie problemas, kurios kyla, kai yra atnaujinamos duomenų bazių versijos, ir maži pakeitimai kartais sustabdo duomenų apsikeitimą tarp „ReadSoft“ sistemos ir integruojamos duomenų bazės. Taip pat pastebėta, jog „ReadSoft“ sistemos atnaujinimai nėra tokie dažni kaip duomenų bazių serverių atnaujinimai, ir ateityje, iškilus problemoms, jos nebus išsprendžiamos per kelias dienas.

Atsižvelgiant į „ReadSoft“ sistemų kūrėjų rekomendacijas bei IT specialistų kompetencijas, „Kitron“ grupėje buvo nuspręsta naudotis XML failų duomenų apsikeitimo integracija.

„Enterprise Service Bus“ (ESB) yra apibūdinamas kaip įrankis, palaikantis programinę integraciją. Tačiau, kas yra ESB? Kada yra geriau naudoti integracinius paketus? Kuris produktas yra geriausias tiriamoms sistemoms? Tinkamo produkto pasirinkimas yra pirmas žingsnis į projekto sėkmę.

Daugybės kompanijų produktų savyje turi žodį „Enterprise Service Bus“. Tačiau nėra visiškai apibrėžto termino apibūdinti šį įrankį. ESB terminas turėtų būti aiškiai apibrėžtas prieš jį naudojant. Kai kuriais atvejais ESB yra apibūdinamas kaip programinė įranga, kuri padeda programuotojui integruoti informacines sistemas ir suteikia reikiamą infrastruktūrą, kad būtų galima įdiegti duomenų maršrutizavimą, interpretavimą ir kitas integracines funkcijas. Pagal integracijos sudėtingumą ESB dažniausiai įsiterpia tarp integracinio karkaso bei paketo [10] kaip integracinė sistema, kuri parodyta 10 paveikslėlyje.



11 pav. Produktai pagal integracijos sudėtingumą

Integracinis karkasas

Karkasas (angl. *framework*) padeda diegti įmonių integracines struktūras „Enterprise Integration Patterns“ [9] (*EIP*), tokias kaip „Splitter“ arba „Content Based Router“, kad būtų įmanoma standartizuotai integruoti sistemas. Naudojant standartinius *API* sistemų integravimui yra ženkliai sumažinamos pastangos diegiant integracinę sistemą ir sistemos kodas yra paprasčiau suprantamas kitų programuotojų. Karkasas puikiai tinka integruoti skirtingas sistemas su skirtingais protokolais, technologijas ir konceptus, tokius kaip galutiniai taškai (angl. *endpoints*), kūrėjas (angl. *producer*), vartotojas (angl. *consumer*) ir *EIP* yra naudojami sukuriant integracinę logiką. Netgi netiesiogiai palaikomas testų automatizavimas naudojant tuos pačius konceptus. Karkasas susideda iš kelių paprastų bibliotekų ir yra suderinamas su bet kokia kūrimo aplinka, netgi klasikiniu teksto redaktoriumi.

Žinomi karkasų pavyzdžiai yra „Apache Camel“ ir „Spring“ integracijos „Java“ bei „NServiceBus“ *.NET* aplinkoje.

Naudojantis karkasais, programuotojų komanda yra daugiau ar mažiau atsakinga už projekto sėkmę, nes nėra įmonių, kurios techniškai palaikytų įmonės viduje kurtą integraciją. Naudojimasis įrankiais taip pat yra dalinai prieinamas ir nebūtinai tinkamas svarbioms įmonės sistemoms integruoti.

Enterprise Service Bus

ESB ir integraciniai paketai dažnai yra geresnis pasirinkimas nei karkasai. Kaip ir karkasai, *ESB* yra naudojami sistemoms integruoti. *ESB* yra sukurtas karkasų pagrindu, tačiau tai yra žymiai funkcionalesnis produktas. Priešingai nei karkasas, be bazinio integracinio funkcionalumo, *ESB* pateikia stiprius įrankius ir palaikymą diegiant, administruojant ir stebint integracinius procesus. Taip pat *ESB* gali būti panaudoti grafiniai redaktoriai, kurie yra naudojami diegiant įvairius integracinius scenarijus. Integracinė logika gali būti kuriama tiesiog „mėtant“ funkcionalumą (angl. *drag and drop*), o atitinkamas kodas yra generuojamas automatiškai. Komercinis palaikymas užbaigia pilną *ESB* paketą. Pagrindinis *ESB* pranašumas prieš karkasus yra geresni įrankiai, kurie sumažina integracijos kūrimo kainą ir sudėtingumą. Naudojant *ESB*, integracinės problemos gali būti sprendžiamos aukštesniame abstrakcijų lygmenyje.

Integracinis paketas

Integraciniai paketai siūlo visas *ESB* funkcijas bei papildomas funkcijas, tokias kaip verslo procesų valdymas, verslo aktyvumo stebėjimas, duomenų valdymo arba saugojimo vietos (pvz. debesyje) valdymas. Jeigu neužtenka *ESB* funkcionalumo ir reikalingas šis papildomas paketo

funkcionalumas, yra rekomenduotina naudotis integraciniu paketu. Visa integracija gali būti atlikta naudojant vieną programinę įrangą.

Tiek informacijos turėtų pakakti, norint nustatyti, kokio lygio integraciją norima diegti įmonėje.

Beveik kiekvienas licencijuotos sistemos gamintojas, toks kaip „Microsoft“, „IBM“ ar „Oracle“ siūlo sprendimus beveik kiekvienai įmanomai funkcijai atlikti. Žiūrint į atviro kodo alternatyvas verta paminėti „Unified Platform of Talend“ platformą, kadangi ji siūlo pilną integracinį paketą. Be kelių alternatyvų, kurios yra susitelkusios ties *ESB*, bene geriausios atviro kodo sistemos yra „Mule ESB“ ir „Fuse ESB“.

1.5.1. Oracle Service Bus

„Oracle Service Bus“ yra „Oracle Fusion Middleware“ (*OFM*) integracinio paketo dalis. Kaip ir paketui pridera, jame yra daugybė kitų produktų kaip „SOA Suite“, „Coherence“, „Complex Event Processing“, „Enterprise Messaging Service ir daugybė kitų plataus funkcionalumo produktų.

Pagal paketo aprašymus reiktų žiūrėti, kokio funkcionalumo šis paketas neturi. Įrankiai yra stabilūs ir funkcionalūs. Daugybė produktų turi grafines sąsajas, jei yra reikalinga, teikiamas ir aukščiausio lygio palaikymas. Savaiame suprantama, toks funkcionalumas ateina su atitinkama kaina - sudėtingi ir kokybiški produktai neturėtų būti nuvertinami. Taip pat reikia atkreipti dėmesį į dideles licencijavimo, techninės pagalbos kainas ir neaiškius apmokestinimo modelius (kainos yra derinamos su kiekviena įmone atskirai).

OFM yra grįsta „Java EE“, *BPEL*, *SOAP* bei *SCA* standartais. Visi pakete esantys produktai per ilgą laiką buvo sukurti „Oracle“ kompanijos. Tačiau kuriant buvo naudojamos skirtingos platformos, todėl skirtingi produktai gali reikalauti skirtingų įrankių. Sistemų diegimas įmonėje yra gana painus ir gali trukti daugiau nei kelias dienas, netgi paprastas programos įdiegimas į vartotojo kompiuterį. Produktai įsisavina gana daug kompiuterio resursų, todėl reikalavimai infrastruktūrai, kur bus diegiama sistema, yra ganėtinai aukšti.

Jeigu palygintume „Oracle Fusion Middleware“ ir „IBM WebSphere“ paketų turinius, galima teigti, kad jie yra beveik vienodi, tačiau tik „IBM“ turi tris skirtingas *ESB* savo pakete: „Message Broker“, „ESB“ ir „DataPower SOA Appliances“. Taip pat „Tibco“, „Microsoft“ ir „SAP“ yra svarbūs *ESB* ir integracinių paketų rinkoje. [10] Apibendrinant galima teigti, kad integraciniai paketai suteikia visišką funkcionalumą ir techninį palaikymą. Deja, viso funkcionalumo ar palaikymo nebus įmanoma panaudoti daugumoje IT projektų, todėl verta apsvarstyti atviro kodo sistemas.

1.5.2. Mule ESB

„Mule ESB“ yra vienas iš pirmųjų sėkmingų atviro kodo *ESB* integracinių sistemų. Sistema turi daug bendrų savybių su anksčiau minėtomis atviro kodo *ESB* sistemomis, tarp kurių yra „vieno paspaudimo“ įdiegimas ir intuityvūs „Eclipse“ įrankiai, paremti programavimo platforma. Dažniausiai atviro kodo *ESB* yra labai paprasti ir išplėtoti sprendimai. Atskirai nuo nemokamos atviro kodo versijos yra ir įmonėms skirta komercinė versija. Komercinė versija siūlo papildomą funkcionalumą bei aptarnavimą produktui.

Reikėtų paminėti, kad, jei sistema yra atviro kodo, tai dar nereiškia, kad tai yra nemokamas produktas, nes net ir atviro kodo programinės įrangos pardavėjai privalo uždirbti pinigų ir negali kurti produktų ar siūlyti aptarnavimo nemokamai. Atviro kodo programinė įranga yra kur kas lengviau įperkama klientams, nes kainos nėra paremtos neaiškiais kainoraščiais kaip dauguma licencijuotų produktų. Vis dėlto atviro kodo versija gali būti naudojama (net ir gamyboje) be jokių licencijavimo kaštų, tačiau atviro kodo versija yra tinkama pabandyti arba norint įsitikinti jos veiksmingumu tam, kad vėliau būtų galima pakeisti ją į įmonių komercinę versiją su papildomomis savybėmis ir aptarnavimu.

Pavadinimas „Mule ESB“ yra pilnas *ESB*. Lyginant su karkasais kaip „Apache Camel“ arba „Spring integration“, svarbūs pranašumai yra grafiniai redaktoriai, skirti veiksmingai įgyvendinti integracijos scenarijus. Taip pat prieinamos jungtys produktams „B2B“, „SAP“ bei „Salesforce“. Bet

„Mule ESB“ trūksta tam tikrų funkcionalumų, pavyzdžiui, *ESB* turi būti suderintas su kitų pardavėjų produktais. Dar vienas neigiamas aspektas, kurį turi „Mule ESB“, yra maža bendruomenė, apribotas licencijavimo modelis ir ribotas priėjimas prie atviro kodo. Šiais aspektais konkurentai turi ženklių pranašumą [11].

1.5.3. Fuse ESB

„Fuse ESB“ yra grynas *ESB* kaip ir „Mule ESB“ tik be pilno integracinio paketo. Jis pagrįstas „de facto“ standartais tokiose integracijos aplinkose kaip „Apache CXF“ ir „Apache Camel“. To rezultatas - didelė bendruomenė ir sistema yra aprašyta nuo pačių „pamatų“. Kaip kūrimo aplinka remiasi „Eclipse“ programavimo platforma ir yra labai intuityvi.

„Fuse ESB“ buvo dalis „FuseSource“, bet „FuseSource“ neseniai buvo įsigytas „Red Hat“ ir dabar priklauso „JBoss“ padaliniui. „Fuse ESB“ yra įtrauktas į plėtojamų produktų kategoriją padalinyje ir toliau bus palaikomas. „Fuse ESB“ bus integruota į „JBoss“ komercinę *SOA* platformą visai kaip ir neseniai įsigytas „BPM“ (angl. Business Process Management) sprendimas „Polymita“. Vis dėlto produktui yra labai ilgas kelias iki suvienyto paketo, nes integracija su „FuseSource“ ir „Polymita“ užtruks dar keletą mėnesių ir su „JBoss ESB“ „Switchyard“, ir „Fuse ESB“ dabar trys *ESB* produktai turi būti sujungti į vieną Rasti šaltinį. Kiti atviro kodo pardavėjai jau dabar yra pasiekę geresnių rezultatų. [12]

1.5.4. Talend ESB

„Talend ESB“ yra dalis komplekto iš „Talend“. „Talend ESB“ gali būti naudojama nepriklausomai arba kombinuota su kitais komponentais iš „Talend unified platform“. Visi komponentai yra atviro kodo ir laisvai prieinami. Komercinė versija siūlo papildomus privalumus ir aptarnavimą. Skirtumas prieš licencijuotus produktus yra tas, kad visi daliniai komponentai yra paremti tuo pačiu kodu ir visur naudojami tokie patys įrankiai. Panaudojimas skirtingais atvejais tokiuose kaip *ESB*, *BPM*, *ETL*, *MDM* ir t.t. yra daromas laisvai ir nėra atskiras integracijos projektas.

Visi įrankiai iš „Talend“ komplekto yra programuoti su programavimo platforma „Eclipse“. Išlikęs „Eclipse“ intuityvumas. „Talend“ siūlo grafinę sąsają visiems produktams ir naudoja „zero-coding“ stilių, kas leidžia efektyviai įgyvendinti integracijos scenarijų. Žinoma, vis tiek galima parašyti nestandartinę logiką savo projektui, pavyzdžiui, per „Java“ klases arba kitas programavimo kalbas.

Taip kaip „Fuse ESB“, „Talend ESB“ yra paremtas keliais „de facto“ standartais tokiose integracijos aplinkose kaip „Apache Camel“, „Apache CXF“, „Apache Karaf“ ir „Apache Zookeeper“. Be prieinamų jungčių, kurios skirtos „Apache Camel“ technologijoms „JMS“, „HTTP“, arba „FTP“, yra daug kitų prieinamų adapterių, pavyzdžiui, „Alfresco“, „Jasper“, „SAP“, „Salesforce“ arba patobulintose sistemose. Daugiau nei 500 jungčių yra įtrauktos į produktą pagal standartą. Dar vienas privalumas yra toks, kad „Talend IDE“ turi aukštesnius techninės įrangos reikalavimus nei jų konkurentai. Nederėtų įrašyti „Talend“ į per silpną nešiojamą kompiuterį. Dar vienas trūkumas yra trūkstamas „SOA“ valdymo funkcionalumas, tačiau sistemos gamintojai planuoja tai ištaisyti per kitą versijos atnaujinimą. [13]

1.5.5. Microsoft BizTalk

Daugumai žmonių žinoma kompanija „Microsoft“ išleido savo integracinį paketą, kuris automatizuoja verslo procesus, leidžiant įvairioms sistemoms perduoti informaciją vienai su kita, naudojantis adapteriais. Ši platforma startavo 2000 metų gruodį ir iki šiol yra išleistas devynios versijos (naujausia „BizTalk Server 2016“ bus išleista 2016 metų pabaigoje). Naujausia šiuo metu yra „Microsoft BizTalk 2013 R2“ versija, turinti daug funkcionalumo pradedant nuo palaikomų duomenų apsikeitimo formatų iki darbų stebėjimo sistemų, kurios automatiškai praneša administratoriui apie serveryje įvykusias integracijos problemas. BizTalk serveris programuojamas naudojantis „Microsoft Visual Studio“ programų kūrimo įrankiu. [10]

Ši platforma puikiai tiktų integracijai tarp dviejų sistemų, dėl duomenų apsikeitimo formatų palaikymo ateityje būtų įmanoma integruoti ir daugiau informacinių sistemų. Bei būtų užtikrintas aukščiausios kokybės techninis palaikymas „Microsoft“ kompanijos specialistų. Tačiau per didelė šio produkto kaina, kuri, priklausomai nuo norimo funkcionalumo, prasideda nuo 10000 iki 30000 JAV dolerių per metus. [11]

1.5.6. TIBCO ActiveMatrix BusinessWorks 6

Kompanija „TIBCO“ yra sukūrusi savo produktą duomenims tarp įvairių sistemų perduoti. „ActiveMatrix BusinessWorks 6“ leidžia įmonėms greitai ir lengvai integruoti skirtingų platformų sistemas, pavyzdžiui, debesų aplikacijas, vietinės įmonės sistemas, mobilias programėles, verslo klientų sistemas, socialinius tinklus ir daiktų interneto įrenginius.

Ši integracija patraukli įmonėms, nes ja paprasta naudotis, programuotojai gali naudotis grafinės sąsajos galimybėmis, kurios veikia „Eclipse IDE“ platformoje. Anot gamintojų, taip yra paprasčiau prijungti naujas sistemas, nes programuotojams nereikia analizuoti ir mokytis integruojamų sistemų aplikacijų programavimo sąsajos *API* (angl. *Application Programming Interface*), ir šis įrankis turi daugybę palaikomų sistemų apjungimo formatų. Taip pat šios programinės įrangos funkcionalumas leidžia automatizuoti testavimo procesą, užtikrinti integracijų saugumą, patikimumą perduodant duomenis, naudojantis verslo taisyklėmis ir automatizuotu integracijos gyvavimo trukmės valdymu.

Integracija suteikia atvirą programavimo sąsają, todėl programuotojai gali pasinaudoti savo Java programavimo įrankiais, sukurti jungtis ir komponentus su savo projektais. Šie komponentai gali būti pabendrinti ir panaudoti kitų programuotojų, kad būtų įmanoma greičiau įsisavinti naujas technologijas, valdant kūrimo kaštus. [12]

Daugiau informacijos apie šį produktą bei nemokamai jį išbandyti galima gamintojo internetiniame puslapyje.

1.5.7. IBM Sterling B2B Integrator

„IBM Sterling B2B Integrator“ yra gerai žinomas „IBM“ kompanijos produktas informacinėms sistemoms integruoti. Produktas yra labiau orientuotas į „IBM“ kompanijos kurtus produktus, tačiau puikiai susidoroja ir su kitomis informacinėmis sistemomis. Sistema sukurta *ESB* platformos pagrindu ir užtikrina realaus laiko komunikaciją tarp integruojamų sistemų. [13] Ši sistema tęsia „IBM WebSphere“ integracinės sistemos funkcionalumą, kuris bus nutrauktas 2018 metais. [14]

„Kitron“ įmonei „Microsoft BizTalk“ sistemos kaina yra per didelė, ji svyruoja nuo 8 iki 20 tūkstančių eurų priklausomai nuo techninio palaikymo ir reikalaujamų savybių. Taip pat darbuotojams trūksta IT kompetencijų, nėra dirbusių su šia sistema. Minėtos problemos trukdo įdiegti sistemą „Kitron“ IT infrastruktūroje. . [15]

1.5.8. WebMethods Integration

Kompanija „WebMethods“ susikūrė 1996 metais, kurios pirmasis produktas buvo „WebMethods Integration Server“ - programinė įranga, kuri leido integruoti sistemas *HTTP* ir *XML* standartais. Ši programinė įranga buvo nuolat tobulinama ir ja naudojosi tokios kompanijos kaip „DHL“, „Dell“, „Hewlett-Packard“. Vėliau kompanija tapo partneriais su „SAP AG“ - gerai žinomos ERP sistemos „SAP“, kurias naudoja didelės įmonės ir bankai. Ši partnerystė sukūrė į „SAP“ sistemą orientuotą integracinį produktą „SAP Business Connector“, kuris kompanijai atnešė didelį pelną, vėliau investuotą į tolimesnį integracinių sistemų tobulinimą.

Kaip ir kitos analizuojamos sistemos „WebMethods Integration“ siūlo *ESB* platformą, kuri visiškai palaiko sistemų taškas-taškas integracijas ir duomenų apdorojimo adapteriai gali būti lengvai pertvarkomi ar panaudojami pajungiant kitas sistemas. [16] Ši sistema patraukia todėl, kad kompanija WebMethods būsimiems klientams įdiegia visą sprendimą jų IT infrastruktūroje ir 30 dienų vykdo pilną palaikymą pilotiniams integraciniams bandymams. [17]

1.5.9. Esamų sistemų lyginamoji analizė

Kad būtų nuspręsta, kuris produktas yra tinkamiausias įmonei, integracinės sistemos bus vertinamos pagal šiuos kriterijus:

Kaina: kokia yra galutinė produkto kaina įtraukus palaikymą, papildomus produktus norint įsidiegti sistemą, jungtys ir kt.

Licencijavimas: koks yra naudojamas licencijavimo modelis? Kas nutinka, kai pasikeičia reikalavimai (daugiau kompiuterių, fizinių branduolių serveryje, perkėlimas į virtualius serverius ir pan.)? Ar atnaujinimai yra nemokami? Ar yra įmanoma pasendinti versiją? Ar kaina yra numatoma, ar licencijavimo kaina iš vis yra suprantama?

Jungtys: ar yra visi adapteriai visoms integruojamoms sistemoms? Kaip lengvai galima susikurti adapterį?

Plečiamumas: ar įmanoma išplėsti produktą? Ar produktas ir jo moduliai yra pritaikyti kažkokiam standartui?

Lankstumas: ar galima pakeisti funkcionalumą, kad būtų galima pritaikyti jį individualiems poreikiams?

Funkcionalumas: ar visas norimas funkcionalumas yra pateiktas?

Palaikymas: ar yra suteikiamas programinės įrangos palaikymas („darbo valandos“, „24/7“, pagalba įmonėje, telefonu, el. paštu, kt.)? Ar gali būti užtikrintos pagalbos lygio sutartys (angl. *Service Level Agreement*)? Ar teikiamas palaikymas norima kalba?

Bendruomenės: ar yra aktyvūs forumai, straipsniai, video mokymai internete? Ar produktas yra palaikomas kelių kompanijų?

Administravimas: ar yra sudėtinga administruoti produktą? Ar yra vartotojo sąsaja ir stebėjimo servisai procesams valdyti?

Paprastumas: kaip sudėtinga įdiegti produktą? Kiek tam reikia įrankių? Ar intuityvi valdymo aplinka?

Pirmoje lentelėje palyginami privalumai ir trūkumai tarp atviro kodo ir licencijuotų *ESB* sistemų. Privalumai pažymėti žalia spalva, trūkumai - raudona spalva, o vidutiniški vertinamai pažymėti geltona spalva.

1 lentelė. Licencijuotų ir atviro kodo integracinių produktų palyginimas

Kriterijus	Licencijuota sistema	Atviro kodo sistema
Paprastumas	Labai sudėtingas diegimas, gali priteikti konsultantų. Reikalinga daug įrankių	Vieno paspaudimo diegimo vedliai, galima naudotis po kelių akimirų, vienoda platforma
Administravimas	Funkcionalūs įrankiai (administravimui ir stebėjimui). Nebūtinas kodo išmanymas, programiniai pakeitimai vyksta per vartotojo sąsają	Mažiau funkcionalūs įrankiai (administravimui ir stebėjimui, gali prireikti integruoti kitus įrankius, kurtų kitų kompanijų). Nebūtinas kodo išmanymas, programiniai pakeitimai vyksta per vartotojo sąsają
Bendruomenės	Reikia pirkti palaikymą, nėra forumų (Nėra aktyvios bendruomenės, kuri padėtų)	Pagrįsta atviro kodo projektais, dažnai turi savo bendruomenes
Palaikymas	24/7 techninis palaikymas, serviso lygio sutartys pagal poreikius, galima diegti tūkstančiuose serverių	24/7 techninis palaikymas, mažiau garantijų nei licencijuotose sistemose, reikia ieškotis vietinių resursų konsultacijoms
Funkcionalumas	Integraciniai funkcionalumai bei daugybė papildomo funkcionalumo	Integraciniai funkcionalumai ir mažiau papildomo funkcionalumo

Lankstumas	Pakeitimo scenarijus 1: išskirti pakeitimo reikalavimą (angl. <i>Change Request</i>), tada ilgai laukti ir susimokėti. Pakeitimo scenarijus 2: daug sumokėti ir gauti pakeitimus greitai.	Atviraime kode gali pats keisti ką nori, svarbu turėti kompetenciją atlikti pakeitimus
Plečiamumas	Galima pasidaryti pačiam (sudėtinga) arba sumokėti	Standartais pagrįsta, viskas pagal nutylėjimą yra standartiška ir lengva praplėsti
Jungtys	Verslo produktų ir technologijų adapteriai	Verslo produktų ir technologijų adapteriai
Licencijavimas	Sudėtinga kainodara, reikia mokėti už viską (atnaujinimai, migracijos į virtualias mašinas ir pan.)	Mėnesiniai mokėjimai, atnaujinimai licencijuojami pagal senas sutartis, numatomos licencijavimo kainos, versijos pasendinimas yra įmanomas
Kaina	20 000 - X	Vidutinė (palyginus su licencijuotomis sistemomis)

Pagal šią lentelę galima daryti išvadą, kad licencijuotos sistemos siūlo didesnę funkcionalumą ir geresnę techninę palaikymą. Bet visada reikia prieš kuriant sistemą apsvaistyti, ar visa tai yra reikalinga? Reikia pagalvoti apie pastangas, sudėtingumą ir kainą, kuri yra labai didelė. Atviro kodo produktai laimi paprastesniu valdymu, didesniu lankstumu, plečiamumu ir kainuoja žymiai mažiau.

Iš visos analizės galima spręsti, kad nėra paties geriausio įrankio spręsti integracines problemas. Pirmiausia turi būti padarytas sprendimas, ar karkasas yra pakankamas sprendimas integracijoms. Reikia atkreipti dėmesį, didžioji dalis programinio kodo turės būti parašyta paties programuotojo ir administravimo bei kūrimo įrankiai gali tapti didžiule problema. Kitu atveju *ESB* yra geresnis pasirinkimas, jei ateityje įmonei reikėtų kažkokio papildomo funkcionalumo ir pradėti naudoti *ESB* nuo pačios įmonės sistemų integracijos pradžios.

Jeigu naudojamas *ESB* integracinis paketas, turi būti nuspręsta, kokio tipo sistemą naudoti - licencijuotą ar atviro kodo sistemą. Licencijuotos sistemos suteikia visą reikiamą funkcionalumą ir visapusišką techninį aptarnavimą. Vis dėlto licencijuotos sistemos yra brangios ir jas yra gana sudėtinga įdiegti ir valdyti. Pasirinktos atviro kodo sistemos kainuoja mažiau, jas yra paprasčiau naudoti ir yra gana lanksčios pakeitimams. Kai pasirenkama, kokio tipo sistema bus naudojama, reikia susidaryti funkcionalumo sąrašą, ko norima iš integracinės sistemos, ir nustatyti integracines sistemas, tinkančias toms funkcijoms atlikti. Reikia įsitikinti, kad turimos IT kompetencijos yra pakankamos įdiegti sistemą įmonės IT aplinkoje bei vėliau jas prižiūrėti. Kitaip reiks nuolatinės programinės įrangos specialisto konsultacijos, kuri gali būti mokama, jei nepavyktų greitai susisiekti su sistemos gamintojais dėl konsultacijos klaidoms ištaisyti.

Panaudojus pirmoje lentelėje pateiktus kriterijus, norint tinkamai įvertinti įrankio tinkamumą įmonės IT infrastruktūrai bei funkcionalumo užtikrinimui ir palyginus geriausias atviro kodo bei licencijuotas integracines sistemas, nustatyta, kad geriausia licencijuota integracinė sistema yra „MicrosoftBizTalk“ pagal produktų suderinamumą ir „Kitron“ grupėje naudojamą specialų Microsoft licencijavimo modelį pasiūlytą būtent „Kitron“ grupei. Iš geriausių atviro kodo sistemų buvo pasirinkta „Mule ESB“ ir „Kitron Integration Bus“. „Mule ESB“ buvo pasirinkta dėl savo paprastumo diegiant ir administruojant bei lanksčios kainų politikos. O „Kitron Integration Bus“, nes tai jau sukurtas ir pačių „Kitron“ IT specialistų gerai žinomas įrankis, kuris nereikalauja daugiau papildomų investicijų ir laiko.

2 lentelė. Esamų sprendimų palyginimas

Palyginimo kriterijus	Kitron Integration BUS	Microsoft BizTalk	Mule ESB
Paprastumas	Nėra grafinės sąsajos, vidutinis diegimo sudėtingumas	Sunku įdiegti į sistemą, paprastas ir funkcionalus valdymas	Paprasta įdiegti į sistemą, paprastas ir funkcionalus valdymas
Administravimas	Vidutinio sudėtingumo, valdoma per „Visual Studio“	Vidutinio sudėtingumo, valdoma per kelis įrankius	Paprastas administravimas, valdoma per kelis įrankius
Bendruomenės	+	Didelė bendruomenė	Maža bendruomenė
Palaikymas	Lokali kompetencija	Techninis palaikymas pagal sutartį	Techninis palaikymas mokomojoje versijoje
Funkcionalumas	Dalinis ESB	Pilnas paketas	Pilnas ESB
Lankstumas	Galima keisti kodą turint tam kompetencijų, visiškai prieinama prie kodo	Lanksti sistema, negalimas prieėjimas prie kodo	Ribotas prieėjimas prie atviro kodo
Plečiamumas	Galima plėsti sistemą, turint reikiamų kompetencijų	Nebūtina plėsti nes sistema pritaikyta daugybei atvejų	Vidutiniškai sudėtinga praplėsti sistemą dėl ribotos prieigos prie kodo ir pagalbos iš gamintojų
Jungtys	100+	500+	500+
Licencijavimas	-	Metinis	Metinis
Kaina	0	12 000 Eur	6800 Eur

Išanalizavus integracines sistemas buvo nuspręsta naudoti „Kitron Integration Bus“ integracine platforma „ReadSoft“ ir „IFS“ sistemas integruoti. Sprendimas buvo atliktas palyginus lokalias kompetencijas, ateities planus (ar bus artimiausiu metu diegiamos naujos sistemos, kurios turi būti integruotos) bei diegimo ir palaikymo kaštų nebuvimas.

1.6. Siekiamo sprendimo apibrėžimas

Šio darbo siekiamas sprendimas yra integracija, kuri leistų apsikeisti duomenimis tarp tiriamųjų objektų. Reikia sistemos, kuri sugebėtų automatiškai nuskaityti gautus laiškus į elektroninę pašto dėžutę ir paimtų elektroninės sąskaitos failą *PDF* formatu. Tada, nuskaitytus failus, sukurtų *XML* formato failą, kuris būtų perduotas į integracijai, kuri sujungs elektroninių sąskaitų apdorojimo ir verslo valdymo sistemą, kurioje yra patalpinamos ir apmokamos visos sąskaitos. Integracija turi perduoti duomenis gautus *XML* formatu jų neprarandant ar kaip kitaip neiškraipant. Perduotus sąskaitos duomenis turi atpažinti verslo valdymo sistema „IFS“ ir juos automatiškai įvesti į sistemą. Šis sprendimas turėtų sutrumpinti sąskaitų apdorojimo laiką ir žmogiškųjų klaidų kiekį jas užpildant.

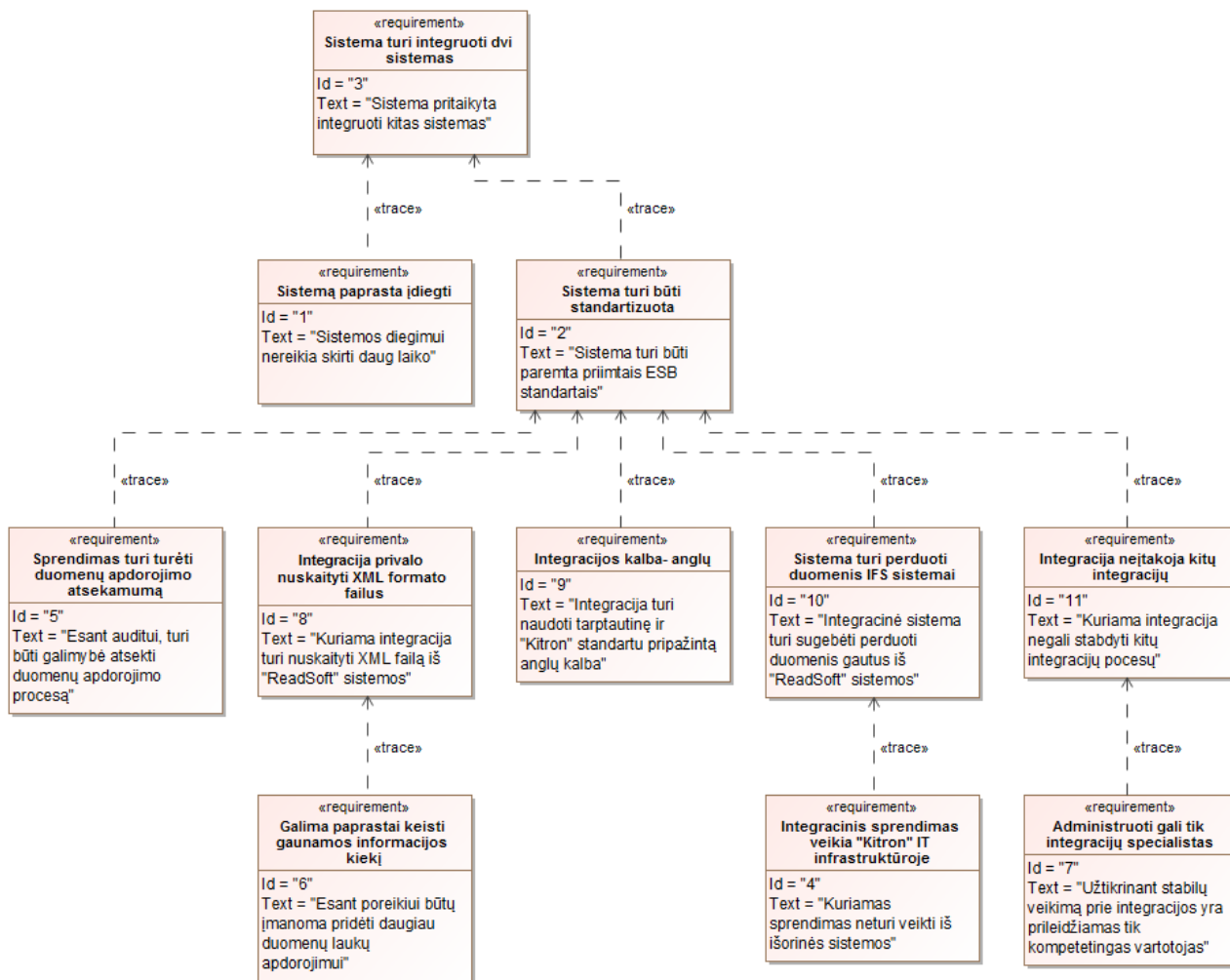
1.7. Analizės išvados

1. Išanalizavus „Kitron“ grupės elektroninių sąskaitų apmokėjimo procesą nustatyta, kad jis yra iš dalies automatizuotas ir yra reikalauja kruopštaus buhalterio darbo ranka. Todėl nuspręsta visiškai automatizuoti elektroninių sąskaitų apmokėjimo procesą įdiegus automatizuotą elektroninių sąskaitų apdorojimo įrankį „ReadSoft“.
2. Išanalizavus „Kitron“ grupėje naudojamas „ReadSoft“ ir „IFS APP 8“ informacines sistemas nustatyta, kad reikia integracinio sluoksnio, kuris užtikrintų apsikeitimą duomenimis tarp sistemų.
3. Pasikonsultavus su „ReadSoft“ sistemos gamintojais, nustatyta, kad geriausias metodas apsikeisti duomenimis tarp šių sistemų yra su *XML* failais, nes klientai, kurie turi šio tipo integraciją mažiausiai kreipiasi į „ReadSoft“ sistemos gamintojus dėl integracijos problemų.
4. Integracijai atlikti buvo ieškomi sprendimai, kurie leistų apdoroti *XML* duomenis ir perduoti juos į kitą sistemą. Buvo pasirinktos penkios integracinės sistemos ir jos išanalizuotos.
5. Pasikonsultavus su „Kitron“ IT skyrių vadovais, buvo nuspręsta iš penkių analizuotų sistemų naudotis „Kitron Integration Bus“ intragraciniu sluoksniu, kuris sukurtas ESB principais kaip ir visi kiti rinkoje rasti universalūs sprendimai. Taip pat pasirinkti šią integraciją paskatino rinkoje esančių sistemų kainos bei nenoras diegtis papildomas sistemas, kurios reikalaus papildomų kaštų bei IT kompetencijų sprendimams palaikyti. Kadangi *KIB* jau yra įdiegtas „Kitron“ IT infrastruktūroje, reikia sukurti atitinkamą adapterį duomenims apdoroti ir perduoti tarp dviejų sistemų.

2. INTEGRACIJOS REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS, FORMALUS APRAŠAS

2.1. Reikalavimų specifikacija

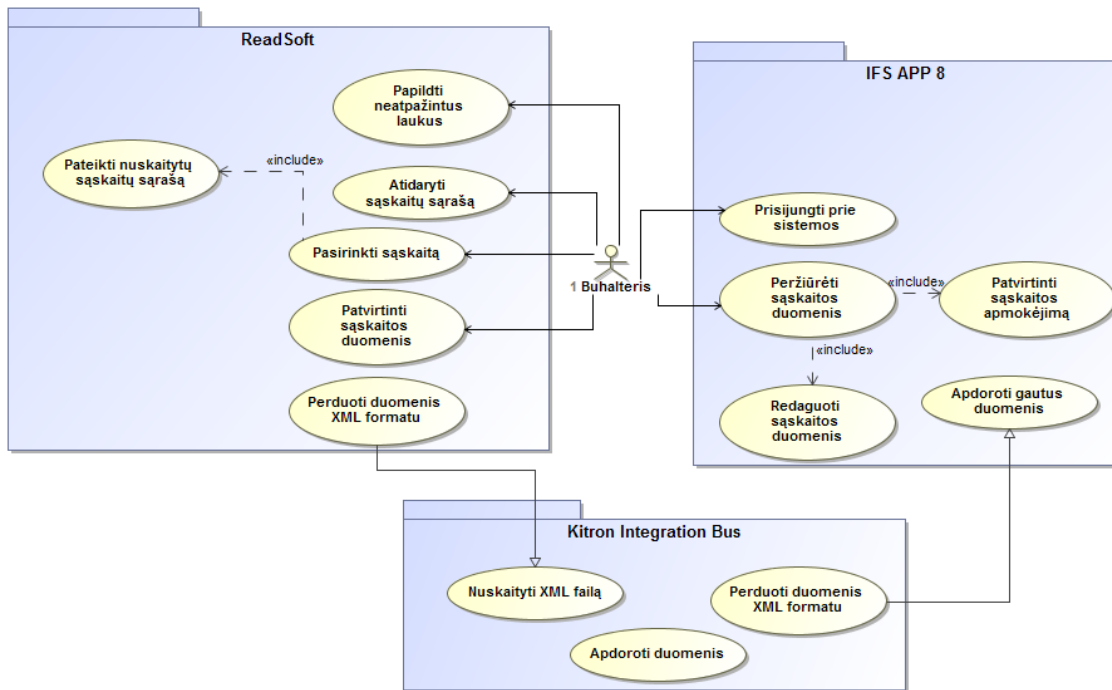
Reikalavimai kuriamai integracijai turi tenkinti 1.5. skyriuje keltus kriterijus integracinei sistemai. Kadangi šiame darbe bus panaudota jau sukurta integracinė platforma, reikalavimai bus keliami bendrai integracinių sistemų sąsajai. 11 paveikslėlyje pavaizduoti reikalavimai integraciniam sprendimui.



12 pav. Integracijos reikalavimų diagrama

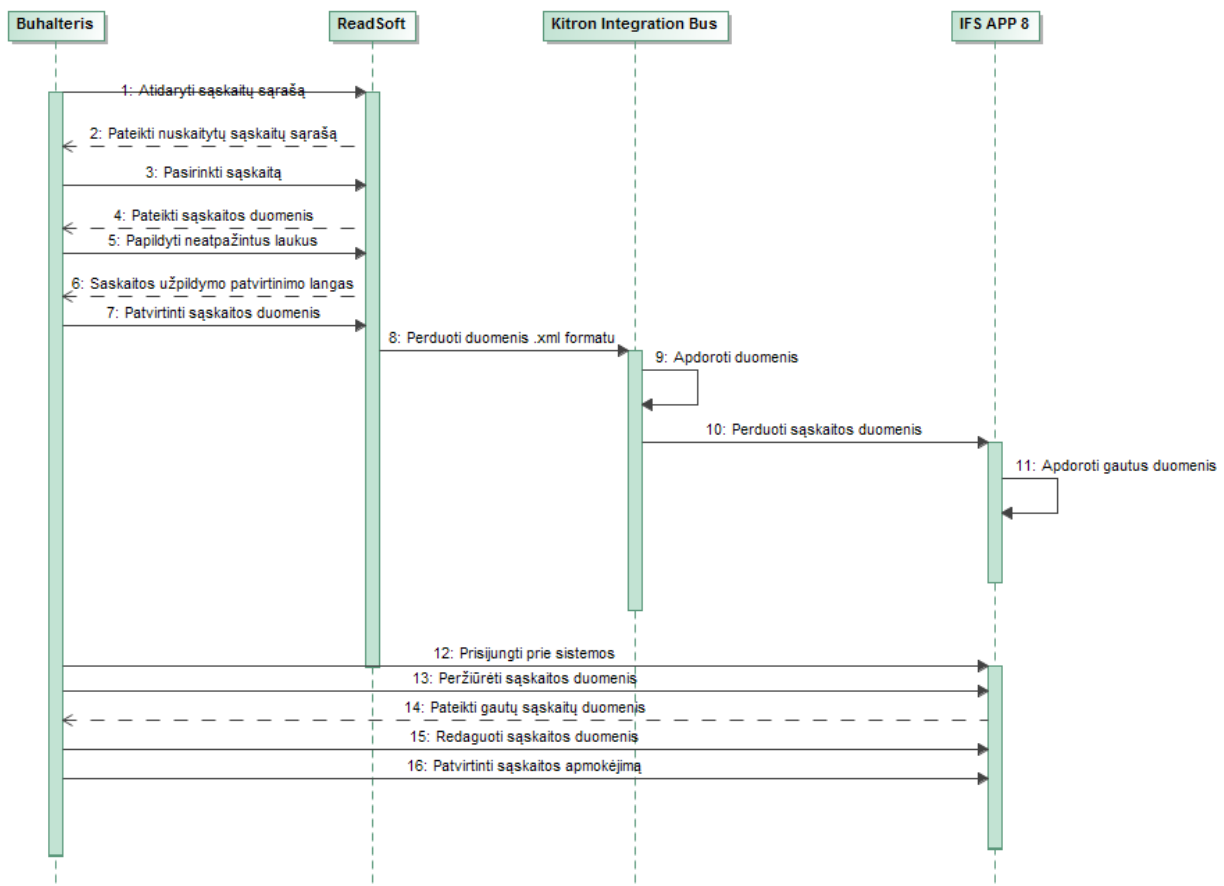
Pagrindinis reikalavimas kuriamai sistemai yra automatizuotas elektroninių sąskaitų apdorojimas. 12 paveikslėlyje pateikta panaudos atvejų diagrama, kurioje matoma, kokie yra planuojami buhalterio veiklos atvejai automatizavus sąskaitų apdorojimo procesą.

Automatizavus sąskaitų apmokėjimo procesą buhalteriu nebereikia naudoti „Outlook“ sistema ir peržiūrėti elektronines sąskaitas, o tiesiog peržiūrėti, ar teisingai suvestos sąskaitos ir jas patvirtinti „IFS“ sistemoje.



13 pav. Buhalterio veiksmai po proceso automatizavimo

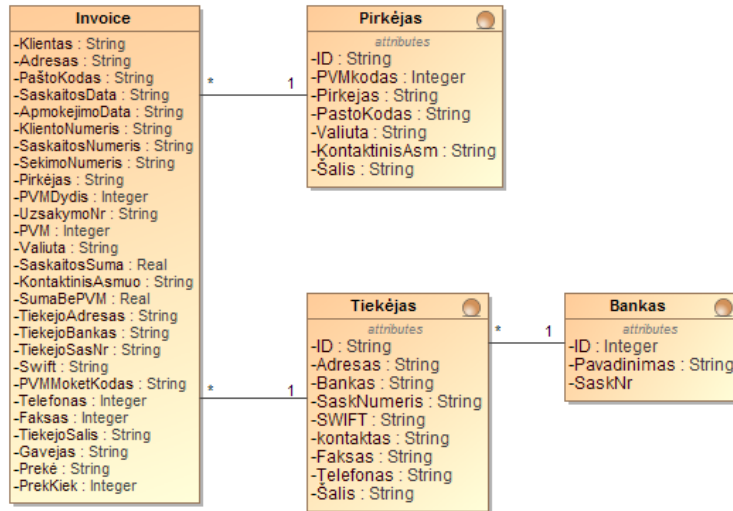
13 paveikslėlyje esanti sekų diagrama realizuoja 12 paveikslėlio panaudos atvejus ir parodo, koks yra siekiamas automatizuotas procesas, kuriame duomenys yra nuskaitomi ir perduodami automatiškai pradedant elektroniniu paštu baigiant „IFS“ sistema.



14 pav. Automatizuoto sąskaitos apdoravimo proceso sekų diagrama

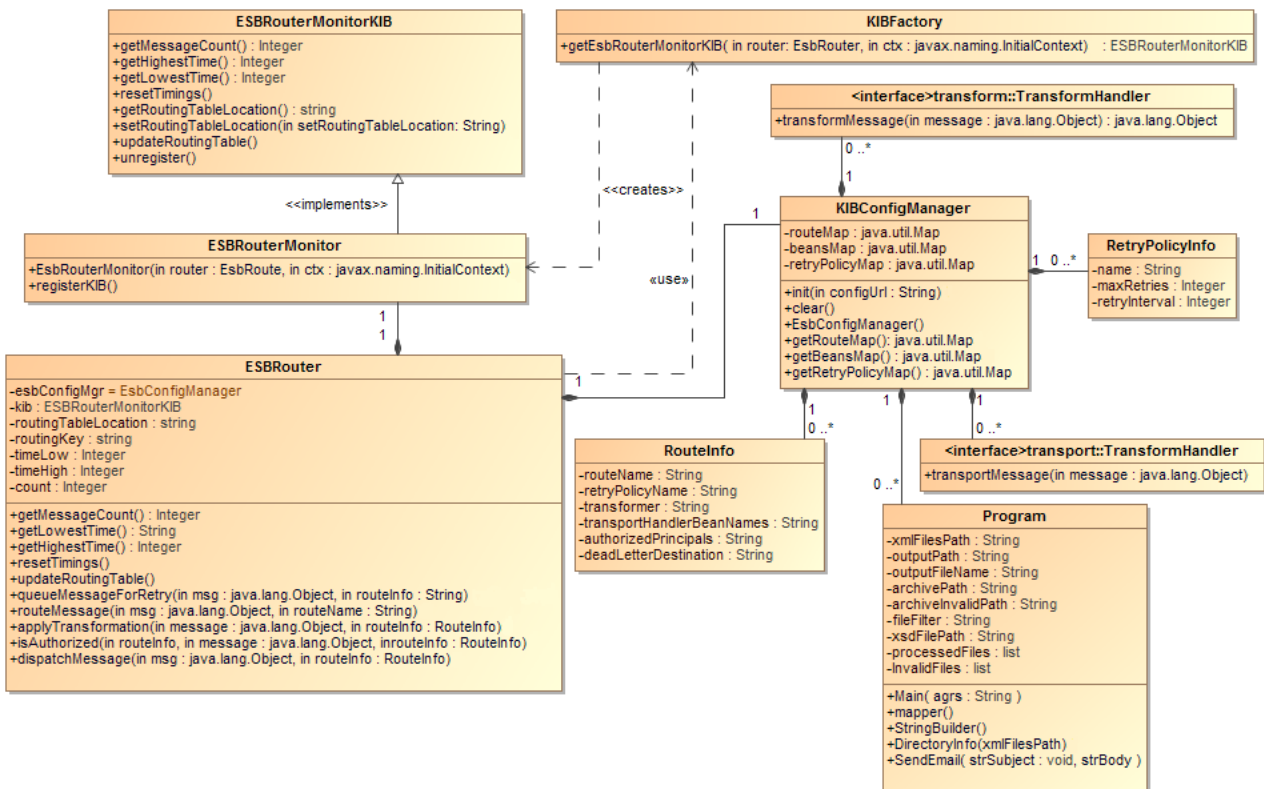
2.2. Dalykinės srities modelis

Atliekant tiriamąjį darbą nebuvo kuriama tiesioginė integracija tarp sistemų, o tik ieškomas geriausias sprendimas jas integruoti perduodant sąskaitos duomenis, daroma prielaida, kad tiek „ReadSoft“ tiek „IFS“ sistemos turi turėti panašią klasę „Invoice“, kurioje yra sudėti pagrindiniai sąskaitos duomenys ir daugiausia dėmesio bus kreipiama į šių duomenų perkėlimą. 14 paveikslėlyje pavaizduota pradinio duomenų srauto struktūra.

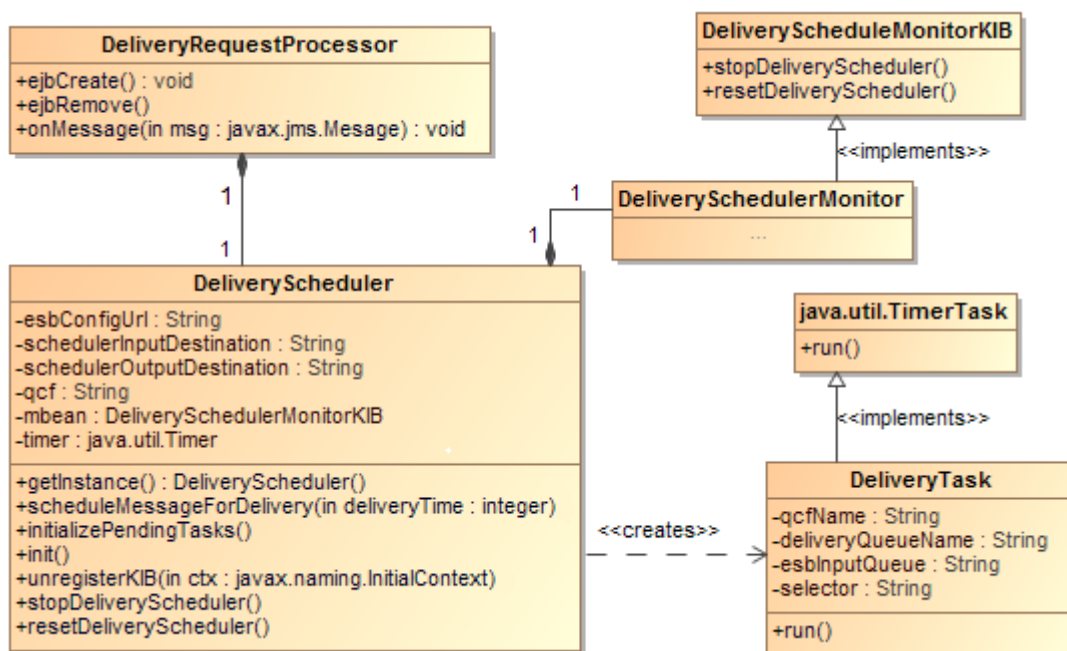


15 pav. Pradinio duomenų srauto struktūra

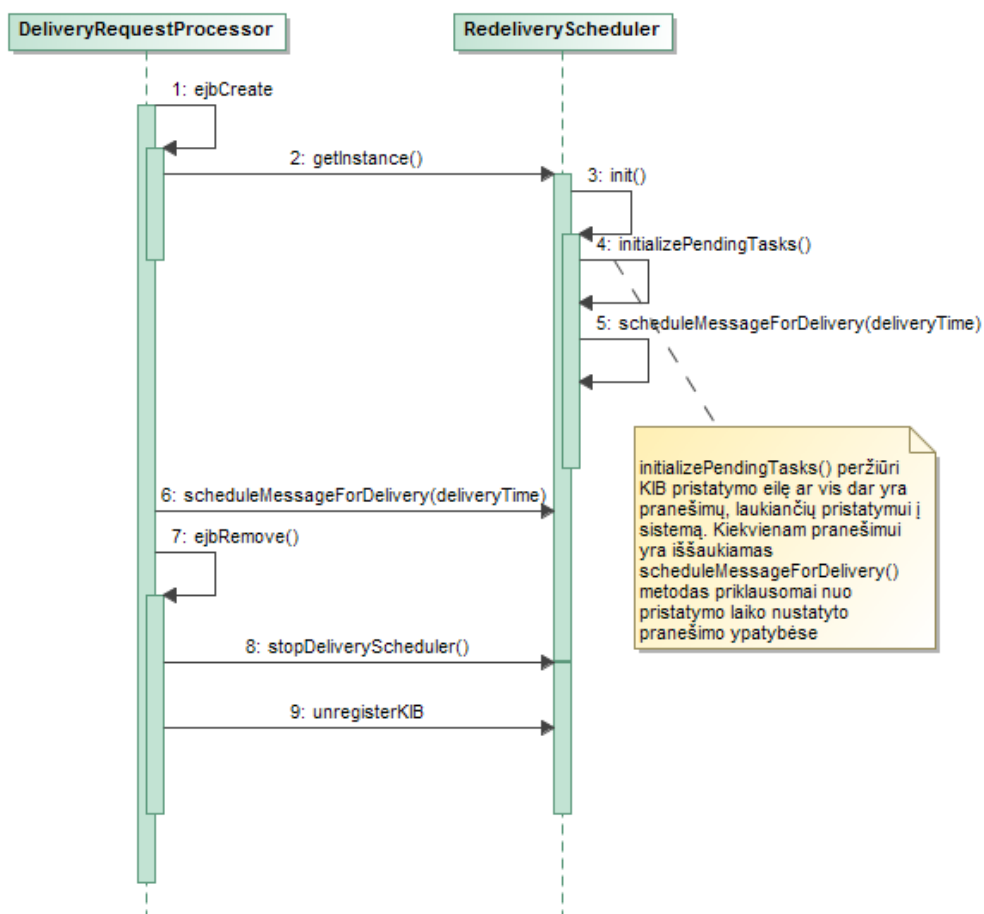
17-18 paveikslėlius pateiktos KIB klasės, kurios parodo sistemos vidinę struktūrą ir galimas atlikti operacijas. Ši integracinė platforma yra grįsta atvirojo kodo sistema parašyta „Java“ platforma.



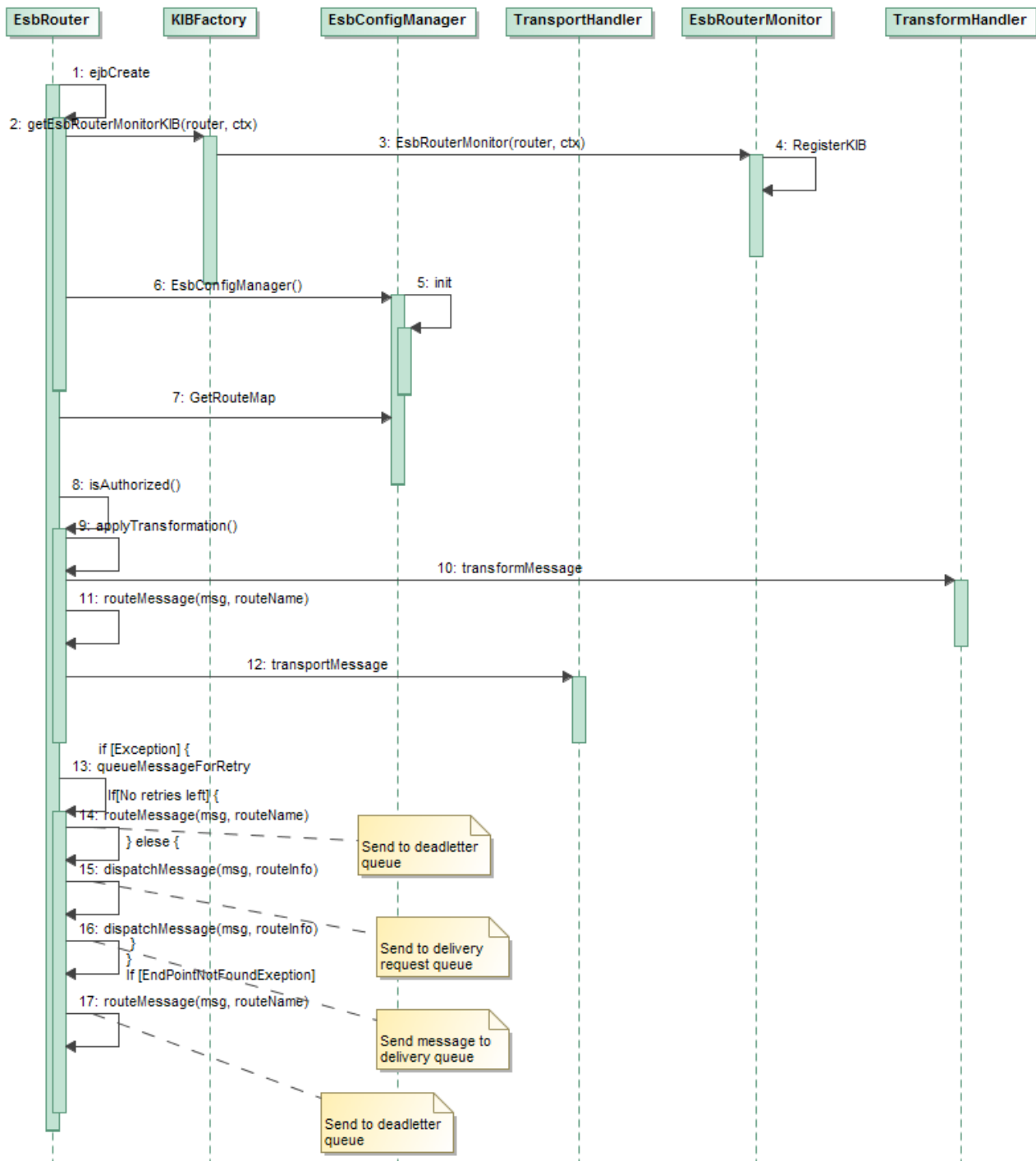
16 pav. KIB „EsbConfigManager“ klasių diagrama



17 pav. KIB „DeliveryScheduler“ klasių diagrama



18 pav. KIB „DeliveryRequestProcessor“ sekų diagrama



19 pav. KIB „DeliveryScheduler“ sekų diagrama

19-20 paveikslėliuose pavaizduoti KIB duomenų užklausų ir siuntimo sekos.

2.3. Formalus sprendimo aprašas

Šiame darbe analizuojamas galimas dviejų sistemų integravimas. Sistemos, anot gamintojų, dar nebuvo sujungtos. Šis sukurtas sprendimas leis dviem sistemoms, kurios yra skirtinguose duomenų centruose, skirtingose šalyse, apsikeisti duomenimis užtikrinant kokybišką bei greitą veikimą.

Techniniame projekte automatizuota sąskaitų apdorojimo sistema „ReadSoft“ bus įdiegta į „Evry“ Kongsberg duomenų centre esantį serverį, kuris yra toje pačioje fizinėje vietoje, kur yra „Kitron“ grupės pašto ir integraciniai serveriai. Tada sistemai bus sukurtos atskiros pašto dėžutės, kurios bus nuskaitomos „ReadSoft“ sistemos ir automatiškai paaimamos apdoroti ten nusiųstos sąskaitos. Apdorojus sąskaitą, jos duomenis XML formatu sistema perduoda katalogą, kurį nuolatos

nuskato automatizuotas procesas, kuris padaro atsiųsto *XML* failo kopiją ir paskui tą kopiją persiunčia į katalogą, kurį nuolatos nuskaito „Kitron Integration Bus“ integracijos servisas. *KIB* informacijos perdavimo duomenis saugo atskirame kataloge, o sąskaitos duomenis tiesiogiai perkelia į „IFS“ sistemą, panaudojant jau sukurtą adapterį, kuris integruoja kitą apjungtą sistemą.

„IFS“ sistemoje veikia automatizuota taisyklė, kuri perduoda tiekėjų sąrašą *XML* formatu į integracinį serverį vos tik sistemoje yra užregistruojamas naujas tiekėjas. Integracinis serveris gautą klientų sąrašą *XML* formatu perduoda „ReadSoft“ sistemai suprantamu pavidalu.

Kiekvienoje šalyje galima apdoroti popierines sąskaitas kaip elektronines. Atsisiuntus „ReadSoft“ klientinę programą, sąskaita gali būti nuskaityta ir sąskaitos paveikslėlis bus persiustas į katalogą, kurį nuskaito „ReadSoft“ sistema. Bei nuskaityti duomenys *XML* formatu perkelti į *SQL* serverį, kuris perduoda apdorotų sąskaitų duomenų istorijas pagrindiniam „ReadSoft“ serveriui. Taip pat šis *SQL* serveris saugo visas atliktas „ReadSoft“ operacijas.

Sistemos veikimo projekte vartotojas, įsijungęs „ReadSoft Invoices“ programą, galės peržiūrėti gautas sąskaitas bei papildyti nenuskaitytus duomenis. Jeigu visi sąskaitos duomenys yra nuskaityti be buhalterio įsikišimo, sąskaita iškart yra siunčiama į integracijos nuskaitomą katalogą ir duomenys automatiškai įkeliami į „IFS“ sistemą. „IFS“ sistemoje duomenys yra automatiškai patikrinami ir laukiama buhalterio patvirtinimo sąskaitų sukūrimui sistemoje.

2.4. Reikalavimų apibendrinimas

Reikalavimai integracinei platformai yra parengti pagal 1.5.9. pateiktus kriterijus. Svarbiausi funkciniai reikalavimai integraciniam sprendimui įgyvendinti - „ReadSoft“ sistema privalo automatiškai nuskaityti elektronines sąskaitas ir perduoti duomenis *XML* formatu, „Kitron Integration Bus“ privalo nuskaityti *XML* faile esančius duomenis ir juos perduoti „IFS“ sistemai.

3. READSOFT DIEGIMO IR REALIZACIJOS PROJEKTAS

16 paveikslėlyje esanti diegimo schema parodo bendrą vaizdą, kaip sistema bus įdiegta „Kitron“ grupės IT infrastruktūroje. Visas sąskaitos apdorojimo procesas prasidės nuo į el. paštą atkeliavusios elektroninės sąskaitos, kuri bus nusiųsta į „ReadSoft“ el. pašto dėžutę per „Kitron“ „Exchange“ pašto serverį. Iš pašto serverio gauta sąskaita *PDF* formatu yra perkeliama į „ReadSoft“ sąskaitų serverį KBGMPE04, kuris fiksuoja visus įvykius, kada buvo gauta sąskaita. Visi užregistruoti įvykiai yra perduodami į KBGMSQ01 serverį, kuriame bus laikomi sąskaitų gavimo įvykių duomenys.

Gautas popierines sąskaitas reikia paversti elektroninėmis, pasinaudojus skaitytuvu ir apdorojant su „ReadSoft“ programine įranga. Programinė įranga nuskaityto sąskaitos informaciją pagal nuskaitytus duomenis arba rankiniu būdu nustatytus laukus, priskirtus prie sistemoje esančių laukų (pavyzdžiui, lapo kampe toje pačioje vietoje yra sąskaitos numeris ir sistema visada tame lauke nuskaitytus duomenis interpretuos kaip sąskaitos numerį). Nuskaitytos sąskaitos duomenys *XML* formatu yra perduodami į KBGMSQ01 *SQL* serverį, iš kurio duomenys yra perduodami „Readsoft“ sąskaitų serveriui tolimesniam apdorojimui. Lygiagrečiai perduodamiems nuskaitytos sąskaitos duomenimis yra išsaugomas sąskaitos grafinis atvaizdas *TIF* formatu [20] tinkle bendriname kataloge „\\kit-fs01\felles\EDI\ReadSoft\Settings\Images“ prie kurio prieina „ReadSoft Invoice“ serveris.

„ReadSoft Invoice“ serveris pastoviai gauna naujausią sąrašą su tiekėjais iš „IFS“ sistemos. Šis sąrašas yra perkeliamas iš duomenų centro Linköping mieste Švedijoje į duomenų centrą Kongsberg mieste Norvegijoje, kuriame yra „Readsoft“ serveris. Nuskaitytą sąrašą „ReadSoft“ sistema perkelia apdorotus duomenis *XML* formatu į tinkle pabendrintą katalogą „\\kit-fs01\felles\EDI\ReadSoft\Transfer\prod\Readsoft_out“. Iš šio katalogo apdorotus duomenis perima automatizuotas servisas, kuris sukuria apdorotą dokumento kopiją tinkle pabendrintame dokumente „\\kit-fs01\felles\EDI\ReadSoft\Backup\prod“ bei perkelia duomenis į tinkle pabendrintą katalogą „\\kit-fs01\felles\EDI\ReadSoft\Transfer\prod“, iš kurio duomenis perima papildomas serveris Linköping duomenų centro serveris, kuris įkelia *XML* formate esančius sąskaitos duomenis į „IFS“ sistemą. Taip pat šis serveris perkelia įrašus apie perkėlimo duomenis tinkle pabendrintame kataloge „\\kit-fs01\felles\EDI\ReadSoft\Transfer\prod\response“.

3.1. Programos

15 paveikslėlyje pavaizduota *KIB* integracijos komponentų diagrama. Kadangi nežinome, iš kokių komponentų yra sukurta „ReadSoft“ ir „IFS APP 8“, tad juos paliekame nedetalizuotus. *KIB* pagrindiniai paketai yra „KIBTaskRouter“ ir „KIBConfigManager“, kurie yra atsakingi už duomenis, gaunamus iš adapterio - jų patikrinimą ir persiuntimą į duomenis užsakančią sistemą. Mano indėlis šioje sistemoje yra *XML* duomenų nuskaitymo adapteris, kuris schemeje nurodytas kaip „Program“ komponentas, kuris peruda *XML* dokumente saugotus sąskaitos duomenis „KIBConfigManager“ komponentui. Šiuo atveju adapteris turi tik nuskaityti duomenis, nes duomenų srautas atgal į „ReadSoft“ sistemą negrįžta. Kaip ir 6 paveikslėlyje čia yra pavaizduota, kaip adapteris nuskaityto duomenis iš duomenis teikiančio šaltinio (šiuo atveju *XML* dokumentas), tada duomenys yra patikrinami, ar jie yra teisingi (naudojama *XML* validator klasė, kuri tikrina *XML* schemą ar ji atitinka sąskaitos šabloną). Tada duomenys yra pakeičiami taip, kad juos suprastų „IFS“ sistema ir duomenys yra siunčiami į integracinį sluoksnį, kurį kontroliuoja *KIBConfigManager* komponentas. Tada suformuota užklausa yra perduodama „KIBTransportHandler“ komponentui ir duodamas pranešimas „IFS“ sistemos adapteriui, kad yra siunčiami duomenys, jie yra nuskaityti, patikrinami ir perduodami į „IFS“ sistemos atitinkamus modulius.

4 lentelė. „ReadSoft“ sistemos naujojamoms duomenų bazėms

DB pavadinimas	Serveris	DB versija	Vartotojas
INVOICES_Upgrade	KBGMSQ01	Microsoft SQL Server 2012 Standart	Readsoft
COLLECTOR_Upgrade	KBGMSQ01	Microsoft SQL Server 2012 Standart	Collector

3.2. Periferinė įranga ir infrastruktūra

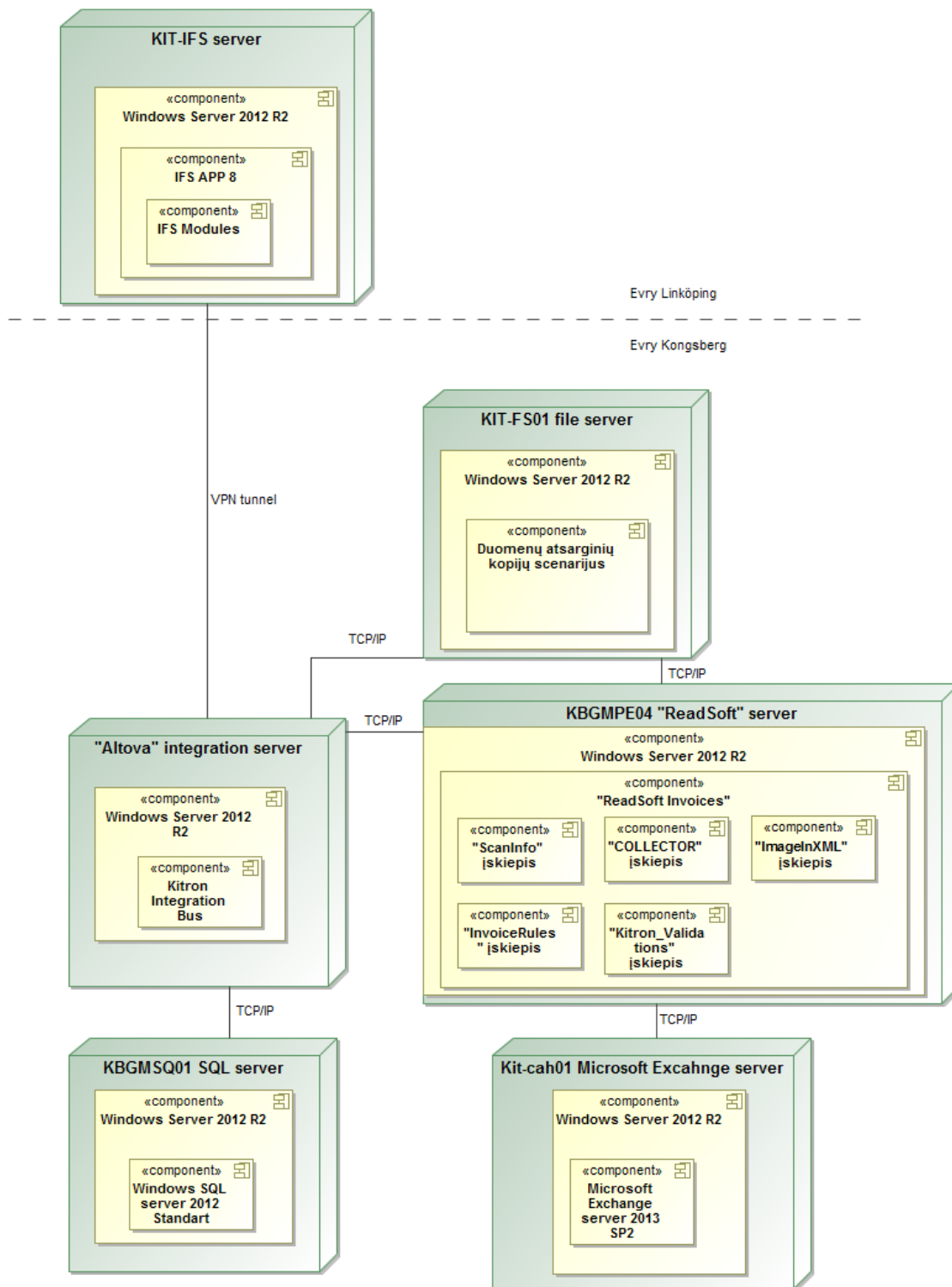
5 lentelėje pavaizduota „ReadSoft“ sistemai reikalingi serveriai, jų pavadinimai, IP adresai, operacinės sistemos bei jų aprašymai. Taip pat šioje lentelėje yra pateikti automatiniai skaitytuvai, kuriais galima apdoroti popierines sąskaitas ir nusiųsti jų elektronines versijas į „ReadSoft“ sistemą.

5 lentelė. „ReadSoft“ serverių sąrašas

Serveris / Skaitytuvas	Pavadinimas	IP	Operacinė sistema	Aprašymas
Programų serveris	KBGMPE04.kitron.local	10.1.86.30	Microsoft Windows Server 2012 R2 Standard	Programų serveris leidžia INVOICES servisus.
Duomenų bazės serveris	KBGMSQ01	10.1.86.26	Microsoft Windows Server 2012 R2 Standard, Microsoft Windows SQL Server 2012 Standart	Duomenų bazės serveris turi INVOICES duomenų bazę.
Licenzinis serveris	KBGMPE04.kitron.local	10.1.86.30	Microsoft Windows Server 2012 R2 Standard	FLEXIm licenzijos serveris prieigai prie sistemos valdyti
Failų serveris	Kit-fs01.kitron.local	10.1.86.39	Microsoft Windows Server 2012 R2 Standard	Failų serveris laiko visus konfigūracinius failus
Naudojami skaitytuvai	arec0045.sonec.no	172.30.16.76	-	Vieta: Arendal, NO Skaitytuvas: Kodak i1320 USB
	jkpw0058.sonec.no	172.30.57.140	-	Vieta: Jönköping, SWE Skaitytuvas: Kodak i1320 USB

21 paveikslėlyje matome integracinio sprendimo diegimo diagramą. Sistema apima dviejų šalių serverius, kurie yra sujungti tiesioginiu VPN (angl. *Virtual Private Network*) tuneliu. „Evry“ Kongsberg aplinkoje bus įdiegti „ReadSoft“, „Microsoft Exchange 2013“, „Microsoft SQL Server 2012“ bei failų saugyklos serveriai. „Evry“ Linköping duomenų centre yra įdiegta „IFS“ sistema su

visais „Kitron“ grupės įsigytais moduliais. „IFS“ sistema turi jau sukurtą adapterį, per kurį vyksta komunikacija su KIB. Šiame darbe negalima pateikti „IFS“ sistemos adapterio, nes be IFS analitiko ir „Kitron“ integracijų skyriaus vadovo šie duomenys negali būti atskleisti, nes jie iš dalies nusako IFS sisteminių duomenų struktūrą, o ji yra apsaugota įmonės privatumo politika. „IFS“ serveriai veikia ant „Windows Server 2012 R2“ operacinės sistemos. Šioje schemoje nėra 5 lentelėje aprašytų skaitytuvų, nes jie nėra privalomi ir gali būti panaudoti bet kuriame „Kitron“ grupės padalinyje.



21 pav. Integracinio sprendimo diegimo diagrama

3.3. Servisai

Pagrindinis servisas yra adapteris, kuris nuskaityto *XML* duomenis ir juos perduoda transportavimo sluoksniui. Adapterio pavyzdys yra aprašytas priedo 8.2. papunktyje. 6 lentelėje matomas reikalingų servisų sąrašą „ReadSoft“ sistemai elektroniniam paštui nuskaityti. Servisai gali būti dviejų tipų - duomenis interpretuojantys ir duomenis perkeliantys. Jie yra atsakingi už elektroninių pašto dėžučių nuskaitymą ir sąskaitos duomenų perdavimą į „ReadSoft“ serverį. Darbo aprašymas yra šifruojamas neužrašant trijų pirmų padalinio vietos raidžių, tada pažymima, ar servisas atsakingas už bendrines sąskaitas (angl. *General Expenses*), ar pirkimo sąskaitas (angl. *Purchase Orders*) ir ar sąskaita siunčiama elektroniniu paštu ar įkelta naudojantis skaitytuvu.

6 lentelė. „ReadSoft“ serveryje veikiančios servisos

Serveris	Modulis	Darbo aprašymas	Vartotojas
KBGMPE04	Interpret	GBR_GE_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	GBR_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	GBR_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Interpret	GBR_PO_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	GBR_PO_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	GBR_PO_Prod	_readsoftservice@kitron.local
KBGMPE04	Interpret	NOR_GE_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	NOR_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	NOR_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Interpret	NOR_PO_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	NOR_PO_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	NOR_PO_Prod	_readsoftservice@kitron.local
KBGMPE04	Interpret	SWE_GE_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	SWE_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	SWE_GE_Prod	_readsoftservice@kitron.local
KBGMPE04	Interpret	SWE_PO_Mail	_readsoftservice@kitron.local
KBGMPE04	Interpret	SWE_PO_Prod	_readsoftservice@kitron.local
KBGMPE04	Transfer	SWE_PO_Prod	_readsoftservice@kitron.local

3.4. Konfigūraciniai failai

Visi sisteminiai nustatymai ir pakeitimai turi būti atliekami konfigūraciniuose failuose kuriais naudojasi „ReadSoft“ sistema ir jos įskiepai

7 lentelė. „ReadSoft“ konfigūracinių failų sąrašas

Konfigūracijos pavadinimas	Vieta	Komentarai
eilocal.ini	C:\Users\Public\ReadSoft\INVOICES	Vietinis INVOICES konfigūracinis failas
eiglobal.ini	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	Globalus INVOICES konfigūracinis failas
COLLECTOR.ini	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	COLLECTOR konfigūracinis failas
ScanInfoConfiguration.xml	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	ScanInfo konfigūracinis failas
InvoiceRulesConfiguration.xml	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	InvoiceRules konfigūracinis failas
Kitron_Validations.xml	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	Kitron Validations konfigūracinis failas
ImageInXML.xml	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	InvoiceInXML konfigūracinis failas
License.dat	\\KBGMPE04.kitron.local\e\$\Programs\Readsoft\INVOICES\GlobalPath	INVOICES konfigūracinis failas

Tam, kad būtų galima naudotis „Microsoft Exchange“, reikalingi servisiniai vartotojai su tam tikromis prieigos teisėmis prie elektroninio pašto. Servisiniai vartotojai ir jų prieigos prie pašto serverio pateiktos 8 lentelėje.

8 lentelė. „ReadSoft“ sistemai sukuri servisiniai vartotojai

Server	Port	Username
Kit-cah01.kitron.local	993	kitron\invoices.norway
Kit-cah01.kitron.local	993	kitron\invoices.lithuania
Kit-cah01.kitron.local	993	kitron\invkitr

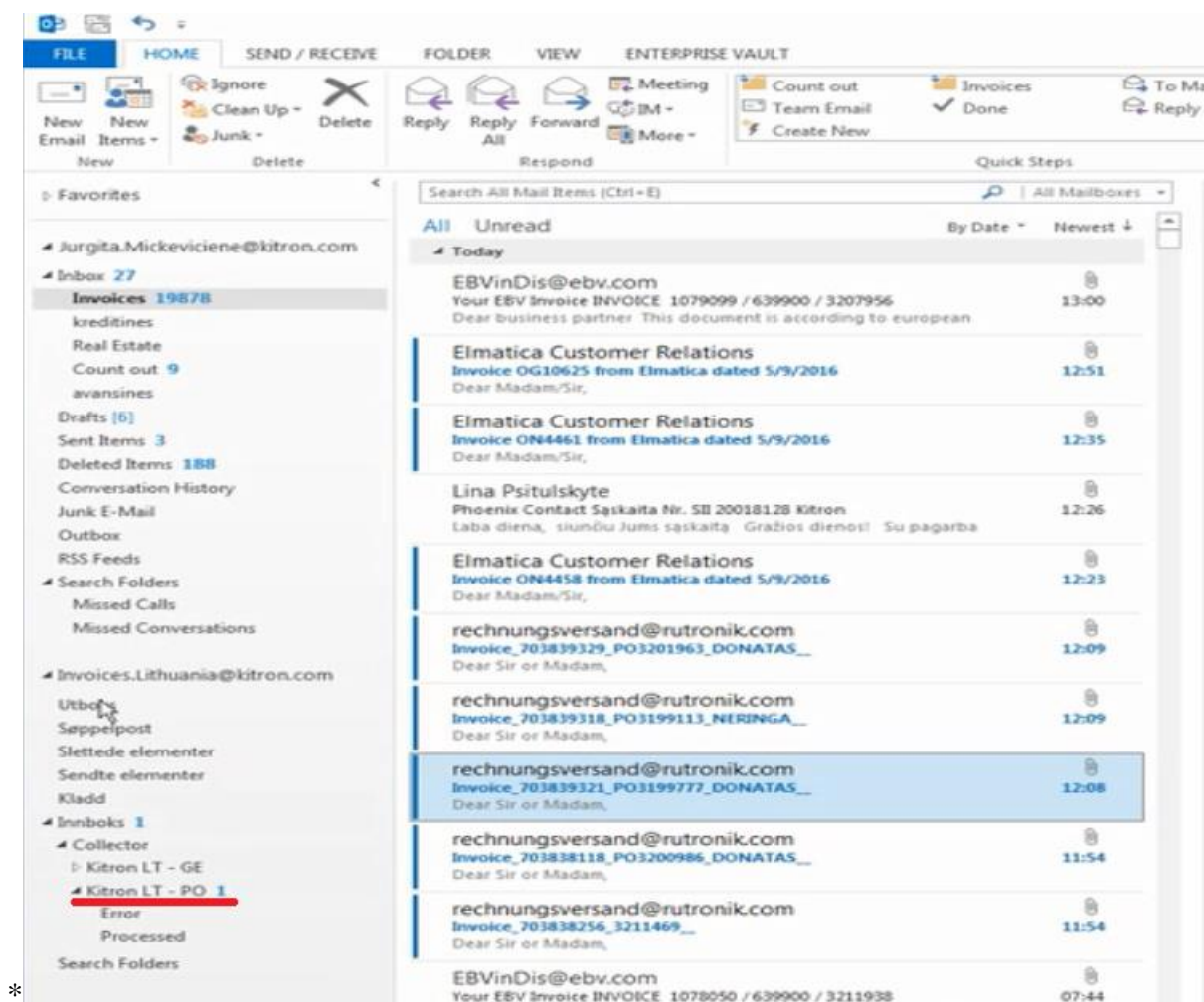
4. SPRENDIMO REALIZACIJA IR TESTAVIMAS

4.1. Sprendimo realizacijos ir veikimo aprašas

Sprendimas buvo realizuotas pagal trečiame skyriuje aprašytą projektą. Visa programinė įranga įdiegta į numatytuosius serverius ir kompiuterius. Sukurti automatizuoti scenarijai (angl. *scripts*) „Windows PowerShell“ įrankiu XML failams perkelti. Taip pat sukurti duomenų apdorojimo adapteriai „Kitron Integration Bus“ integracijoje, kurie nuskaitydavo į integracijos katalogą įkeltus XML formato failus.

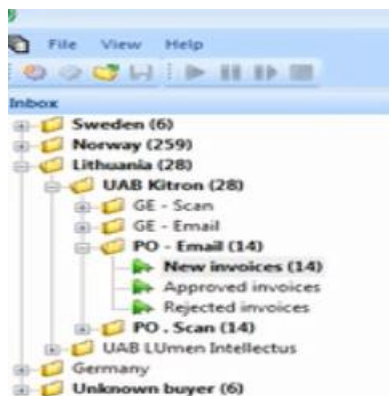
Realizuotas sprendimas buhalterio darbo aplinkoje, išsiaiškinus, ar įdiegta „ReadSoft“ sistema bei pasirinktas integracijos metodas automatizuoja gaunamų elektroninių sąskaitų apdorojimą.

22 paveikslėlyje matoma elektroninė sąskaita, kuri atsiųsta į „ReadSoft“ nuskaitomą elektroninį paštą kitronlt-po@kitron.com. Sistema automatiškai nuskaitydavo laiško turinį, adresatą, kuris siuntė laišką ir prisegtą paveiksluką su įmonei siunčiama sąskaita.



22 pav. Gaunama elektroninė sąskaitą į ReadSoft palaikomą pašto dėžutę

Sėkmingai nuskaičius duomenis, sistema juos susikelia į tam tikrus katalogus, priklausomai iš kurios šalies el. pašto dėžutės yra perkeltami duomenys, kaip pavaizduota 23 paveikslėlyje. Vos tik sąskaita yra gauta į „ReadSoft“ sistemą, ji yra perkeliama į naujų sąskaitų grupę (angl. *New invoices*), kur yra apdorojama. Priklausomai nuo to, ar buvo rastas sąskaitos tiekėjas, ar visi duomenys buvo nuskaityti, sąskaita iš naujų sąskaitų grupės yra perkeliama į patvirtintų sąskaitų (angl. *Approved invoices*) arba atmestų sąskaitų (angl. *Rejected invoices*) grupes.



23 pav. „ReadSoft“ sistemoje atvaizduojamas gautos sąskaitos

Nuskaitytos ir patvirtintos sąskaitos yra atvaizduojamos „ReadSoft“ sistemoje sąrašo principu, lentelės pavidalu, kaip parodyta antroje lentelėje, ir sąskaitos paveikslukas yra apdorojamas sistemoje, kaip parodyta 31 paveikslėlyje. Reikia atkreipti dėmesį, kad sąskaitos *ID* yra generuojamas visoms „Kitron“ grupės sąskaitoms, ir sąskaitos *ID* ne visada eis iš eilės, nes gali būti apdorojamos kelios sąskaitos vienu metu. Taip pat reikia atkreipti dėmesį, kad sąskaitos paveikslėlio vardas yra pakeičiamas, jis užšifruojamas.

ID	Supplier	Supplier Number	Buyer	Buyer Number	Profile	Source	Email sent by	Scan time	Image file
249817	Fintronic Oy	12626KAU	UAB Kitron	32	LTU_PO	Email	„Fintronic Oy“ <fintronic@fintronic.fi>	09/05/2016 12:25:52	160509120943_0001
249819	Fintronic Oy	12626KAU	UAB Kitron	32	LTU_PO	Email	„Fintronic Oy“ <fintronic@fintronic.fi>	09/05/2016 12:25:59	160509120943_0002
249821	Fintronic Oy	12626KAU	UAB Kitron	32	LTU_PO	Email	„Fintronic Oy“ <fintronic@fintronic.fi>	09/05/2016 12:26:01	160509120943_0003
249823	Fintronic Oy	12626KAU	UAB Kitron	32	LTU_PO	Email	„Fintronic Oy“ <fintronic@fintronic.fi>	09/05/2016 12:25:03	160509120943_0003

9 lentelė. „ReadSoft“ nuskaitytų sąskaitų sąrašas

Jei sistema nerada duomenų paveikslėlyje, šalia būtinų duomenų sistemoje yra pažymimas geltonas trikampis su viduje esančiu šauktuku. Tada buhalteriu reikia pažymėti lauką paveikslėlyje, kur yra duomenys. Kitą kartą sistema automatiškai ieškos tų duomenų nurodytame lauke.

Jeigu duomenys yra nuskaityti teisingai, laukas sistemoje yra užpildomas ir šalia lauko atsiranda žalias apskritimas.



24 pav. „ReadSoft“ sistemos automatinis duomenų atpažinimas

Visi iš „ReadSoft“ sistemos XML formatu suformuoti ir išsiųsti duomenys patenka į „IFS“ sistemą. „IFS“ sistemoje „External Supplier Invoices“ modulis yra atsakingas už gautų sąskaitos

duomenų apdorojimą ir, pasinaudojus sistemoje esančia paieška ir jos filtrais, galima surasti reikiamas sąskaitas.

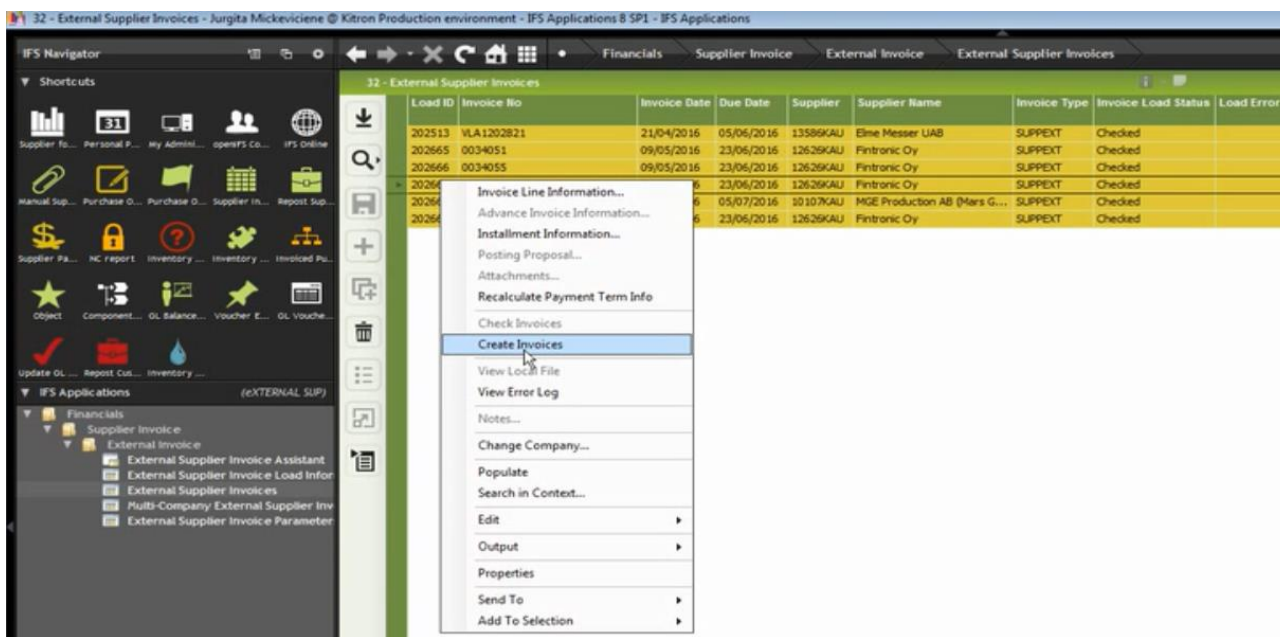
25 pav. „IFS External Supplier Invoices“ modulio sąskaitų atrinkimo langas

26 paveikslėlyje matomos visos su „Invoice Load Status- Checked“ statuso sąskaitas. Šios sąskaitos yra gautos iš „ReadSoft“ sistemos ir patikrintos pačios „IFS“ sistemos, ar visi duomenys yra teisingi, taip yra išvengiama duomenų nesutapimų.

Load ID	Invoice No	Invoice Date	Due Date	Supplier	Supplier Name	Invoice Type	Invoice Load Status	Load Error
202513	VLA1202821	21/04/2016	05/06/2016	13586KAU	Elme Messer LIAB	SUPPEXT	Checked	
202665	0034051	09/05/2016	23/06/2016	12626KAU	Fintronic Oy	SUPPEXT	Checked	
202666	0034055	09/05/2016	23/06/2016	12626KAU	Fintronic Oy	SUPPEXT	Checked	
202667	0034056	09/05/2016	23/06/2016	12626KAU	Fintronic Oy	SUPPEXT	Checked	
202668	7318	06/05/2016	05/07/2016	10107KAU	MGE Production AB (Mars G...	SUPPEXT	Checked	
202669	0034057	09/05/2016	23/06/2016	12626KAU	Fintronic Oy	SUPPEXT	Checked	

26 pav. Gautos sąskaitos IFS sistemoje iš KIB

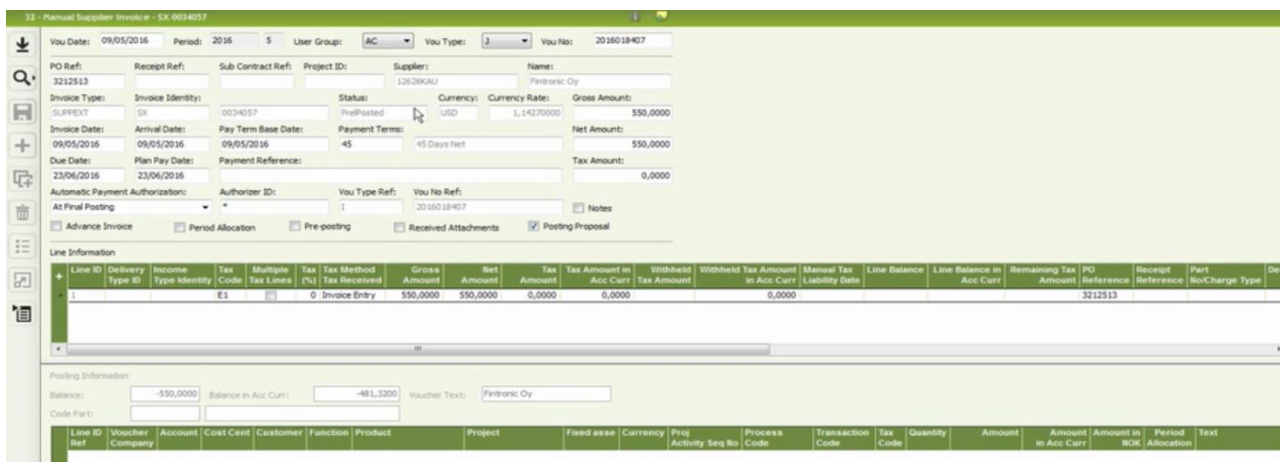
27 paveikslėlyje pavaizduotos pažymimos, patikrintos sąskaitos ir paspaudus dešinį pelės klavišą atsiradusiame meniu yra sukuriamos atskiros sąskaitos sistemoje.



27 pav. „IFS“ sistemoje sukuriamos apmokėjimo sąskaitos

Sukūrus atskiras sąskaitas, prieš tai patikrinti duomenys yra automatiškai perkeliami sistemoje, kaip pavaizduota 28 paveikslėlyje.

Kaip buvo minėta 1.3.1. punkte buhalterio darbas „IFS“ sistemoje, anksčiau buhalteris turėdavo arba ranka įrašyti duomenis, esančius šiame lange, neišvengiamai atsirasdavo žmogiškųjų klaidų surašant duomenis.



28 pav. Suformuota sąskaita „IFS“ sistemoje kuri laukia apmokėjimo patvirtinimo

Atlikus buhalteriu numatytuosius veiksmus, pradėdant „ReadSoft“ sistema baigiant „IFS“ sistema, duomenys buvo automatiškai nuskaitomi, perkeliama ir apdorojami. Patikrinus sąskaitoje esančius duomenis ir duomenis, kurie yra įvesti į „IFS“ sistemą, nebuvo rasta jokių pakitimų, tai yra duomenys nebuvo iškraipyti nuskaitymo ar perdavimo metu.

4.2. Testavimo modelis, duomenys, rezultatai

Testavimo tikslas - išsiaiškinti ar įdiegta „ReadSoft“ sistema ir panaudotas integracinis metodas tinkamai perduoda duomenis IFS sistemai.

Testuojant bus stebima „ReadSoft“ sistemos veikimas apdorojant sąskaitą ir bus stebima, ar visi duomenys teisingai perduoti „IFS APP 8“ sistemai, ar visi laukai buvo atpažinti ir ar sąskaita yra paruošta apmokėti.

Testavimui bus naudojamos 150 skirtingų iš tiekėjų gautų sąskaitų ir fiksuojamas rezultatus. Stebimi testiniai atvejai, pagal kuriuos bus sprendžiama, ar testai buvo sėkmingi, yra tokie:

- Kiek sąskaitų buvo nuskaityta, joms atkeliavus į elektroninį paštą?
- Kiek sąskaitų buvo visiškai nuskaityta „ReadSoft“ sistemos ir sukurti XML duomenų failai?
- Kiek XML sąskaitų buvo perduota į „IFS“ sistemą sėkmingai, neiškraipant duomenų?

Jeigu sąskaita buvo nuskaityta atkeliavus į elektroninį paštą teisingai ir visi jos duomenys teisingai atvaizduoti „IFS“ sistemoje, galiam daryti prielaidą, kad sistema veikia. Testavimą laikysime pavykšiu, jeigu bus tinkamai apdorota daugiau nei 99% sąskaitų.

Iš testavimui naudotų 100 sąskaitų visos buvo teisingai apdorotos ir perduotos „IFS“ sistemai. Testavimo metu 36 kartus reikėjo buhalterio įsikišimo „ReadSoft“ sistemoje.

28 kartus sistema neatpažino lauko, kuriame buvo sąskaitos duomenys. Sistema nuskaitė tik nurodžius lauką, kuriame jis buvo. 2 kartus sąskaitos apdorojimas buvo sustabdytas, nes sąskaitos tiekėjas nebuvo įtrauktas į „IFS“ sistemą. 1 kartą sąskaitos apdorojimas buvo sustabdytas „IFS“ sistemoje, nes buvo neteisingai suvestas tiekėjas sistemoje (nenurodyta, kokia valiuta yra atsiskaitoma tiekėjui). 5 kartus buvo prašoma patikslinti duomenis, nes sistema nenustatė, ar simbolis sąskaitoje yra didžioji O raidė, ar tai yra skaičius nulis.

Visi duomenys buvo perkelti į XML failą, kurį iš karto pasiėmė apdoroti „Kitron Integration Bus“ integracinis sluoksnis ir per ne daugiau kaip 30 sekundžių laikotarpį duomenis įsikeldavo „IFS APP 8“ sistema. Sistema tinkamai sudėliojo elektroninės sąskaitos duomenis ir paruošė sąskaitą sukūrimui sistemoje.

Gauti duomenys pateikti 10 lentelėje.

10 lentelė. Eksperimento pirmos dalies rezultatai

Testavimo kriterijus	Rezultatas
Nuskaitytų sąskaitų skaičius	150/150
Teisingai nuskaityti duomenys ir sukurtas failas	150/150
Neiškraipytų gautų sąskaitų „IFS“ sistemoje	150/150

Nuskaičius 150 sąskaitų „ReadSoft“ programa vis dažniau nebereikalaudavo buhalterio nurodyti duomenų laukų, nes sistema įsimena to paties tiekėjo laukus, kuriuose yra duomenys.

Buvo rasta problema, kuri yra pranešta „ReadSoft“ techninio palaikymo komandai, kad jeigu ateina sąskaita su keliais užsakymo numeriais (kelios sąskaitos sudėtos į vieną), sistema paimdavo tiktais vieną iš kelių sąskaitos numerių ir neleisdavo įvesti kitų sąskaitos numerių. Po kurio laiko „ReadSoft“ techninio palaikymo komanda atsiuntė programos pataisymą, kuri iš dalies išspendė problemą ir leido ranka nukopijuoti ir perduoti kelis užsakymo numerius viename dokumente. Kita dažna problema sistemoje - nuskaitymo metu buvo ne taip interpretuojamas skaičius nulis ir didžioji raidė O, todėl dažnai sistema reikalaudavo buhalterio patikrinti, ar teisingai yra nuskaityti duomenys. Ši problema nekyla, jei atsiųstos sąskaitos atvaizdo kokybė yra aukšta ir nėra susiliejusių raidžių.

Testuojant integracinio sluoksnio apkrovas, vienu metu buvo įkelta šimtas XML failų su sąskaitų duomenimis (realybėje failų skaičius vienu metu apdorojimo kataloge neviršydavo 6 failų) ir nors „Kitron Integration Bus“ užtruko porą minučių failus apdoroti, tačiau visi duomenys į „IFS“ sistemą nukeliavo teisingai ir nebuvo aptikta duomenų praradimo. Testinės sąskaitos pavyzdį bei sugeneruojamos sąskaitos XML failo šablonas su duomenimis yra pateiktas darbo priede 8.3 ir 8.4 skyriuose.

Kadangi daugiau nei 99% sąskaitų buvo apdorotos teisingai, galiam daryti prielaidą, kad sistemos buvo įdiegtos ir suintegruotos teisingai.

5. EKSPERIMENTINIS INTEGRACINIO SPRENDIMO TYRIMAS

5.1. Eksperimento planas

Eksperimente bus palygintas naujos sistemos našumas su senąja sistema. Tai bus daroma stebint finansų skyriaus penkių dienų darbą įmonėje UAB „Kitron“ dirbant su naująja sistema, ir palyginus senosios ir naujosios sistemos veikimą. Stebimi ir vertinami kriterijai, pagal kuriuos bus sprendžiama, ar naujoji sistema paspartino sąskaitų apdorojimo procesą, sumažino klaidų skaičių jas apdorojant, sumažino nenuskaitytų arba neteisingai nuskaitytų sąskaitų skaičių yra šie:

- Kiek sąskaitų buvo nuskaityta, joms atkeliavus į elektroninį paštą?
- Kiek sąskaitų buvo visiškai nuskaitytos „ReadSoft“ sistemos ir sukurti XML duomenų failai?
- Kiek XML sąskaitų buvo perduota į „IFS“ sistemą sėkmingai, neiškraipant duomenų?
- Klaidų skaičius.
- Laiko sąnaudos, skirtos sąskaitų apdorojimui.

Eksperimentą laikomas pavykusiu, jeigu bus tinkamai apdorota 99% sąskaitų, nebus padaryta nė vienos klaidos perduodant duomenis tarp sistemų ir laiko sąnaudos, skirtos sąskaitos apdorojimui bus bent 2 kartus mažesnės nei senosios sistemos.

5.2. Eksperimento rezultatai

Tyrimo metu, per 5 darbo dienas įmonės UAB „Kitron“ finansų skyrius gavo 205 sąskaitas.

Pirmą dieną gautos 35 sąskaitos, antrą 41, trečią 50, ketvirtą 32, penktą 47. Visos sąskaitos buvo teisingai apdorotos ir perduotos „IFS“ sistemai.

Eksperimento metu 51 kartą reikėjo buhalterio įsikišimo „ReadSoft“ sistemoje.

37 kartus sistema neatpažino lauko, kuriame buvo sąskaitos duomenys. Sistema nuskaitė, tik nurodžius lauką, kuriame jis buvo. 4 kartus sąskaitos apdorojimas buvo sustabdytas, nes sąskaitos tiekėjas nebuvo įtrauktas į „IFS“ sistemą. 2 kartus sąskaitos apdorojimas buvo sustabdytas „IFS“ sistemoje, nes buvo neteisingai suvestas tiekėjas sistemoje (nenurodyta, kokia valiuta yra atsiskaitoma tiekėjui). 8 kartus buvo prašoma patikslinti duomenis, nes sistema nenustatė, ar simbolis sąskaitoje yra didžioji O raidė, ar tai yra skaičius nulis.

Visi duomenys buvo perkelti į XML failą, kurį iš karto pasiėmė apdoroti „Kitron Integration Bus“ integracinis sluoksnis ir per ne daugiau kaip 30 sekundžių laikotarpį duomenis įsikeldavo „IFS APP 8“ sistema. Sistema tinkamai sudėliojo elektroninės sąskaitos duomenis ir parengė sąskaitą sukūrimui sistemoje. Gauti duomenys pateikti 11 lentelėje.

11 lentelė Senojo sąskaitų apdorojimo proceso rezultatai

Kriterijus	Rezultatas pagal senąjį procesą				
	1 diena	2 diena	3 diena	4 diena	5 diena
Nuskaitytų sąskaitų skaičius	35/35	41/41	50/50	32/32	47/47
	Iš viso: 205/205				
Teisingai nuskaityti duomenys ir sukurtas failas	-	-	-	-	-
	-				
Neiškraipytų gautų sąskaitų „IFS“ sistemoje	34/35	39/41	48/50	32/32	45/47
	Iš viso: 205/205				
Klaidų skaičius	1/35	2/41	2/50	0/32	2/47
	7/205				
Laiko sąnaudos nuo laiško atėjimo į elektroninio pašto dėžutę ir apdorojimo pabaigos visiems laiškam	156 min.	165 min.	223 min.	122 min.	196 min.
Vidutinis laikas vienai sąskaitai apdoroti [s]	267	241	268	229	250
	Iš viso: 862 minutės				

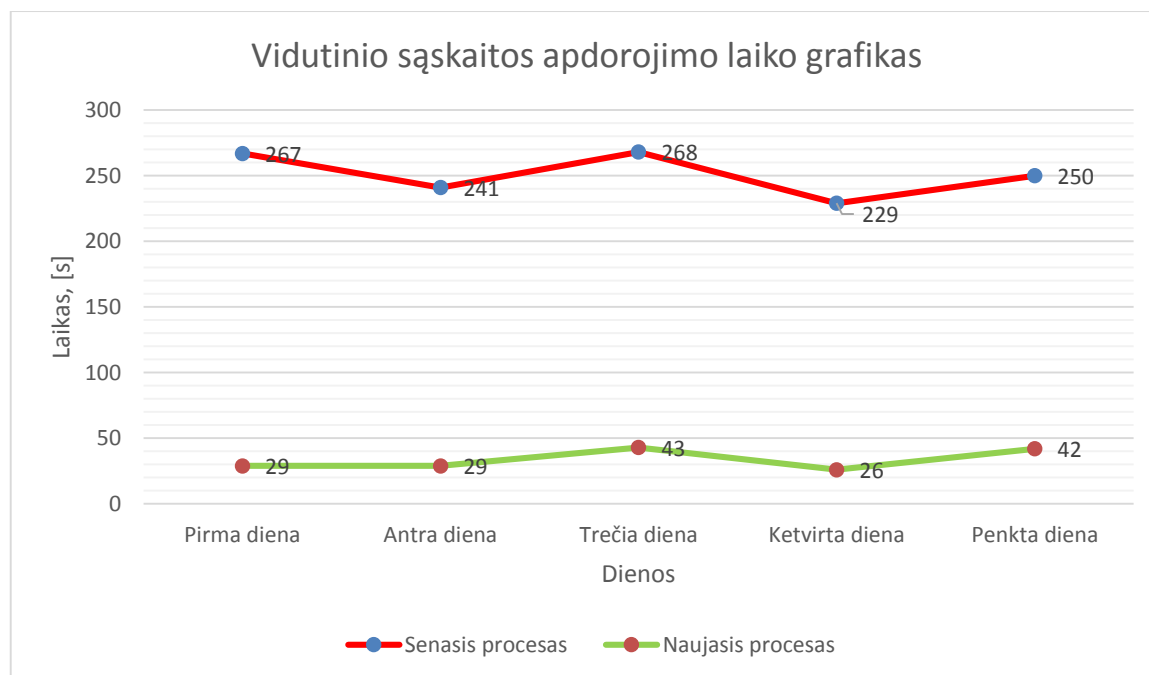
Atlikus stebėjimą pagal senąjį sąskaitų apdorojimo procesą buvo pateiktas toks pat kiekis duomenų naujajai sąskaitų apdorojimo sistemai. Kad būtų patogiau atlikti matavimus, naujojoje sistemoje registruojamas toks pat skaičius sąskaitų per dieną kaip ir senojoje sistemoje, nors visomis dienomis gautų sąskaitų kiekis buvo didesnis. Gauti rezultatai pateikti 12 lentelėje.

12 lentelė. Naujojo sąskaitų apdorojimo proceso rezultatai

Kriterijus	Rezultatas pagal naująjį procesą				
	1 diena	2 diena	3 diena	4 diena	5 diena
Nuskaitytų sąskaitų skaičius	35/35	41/41	50/50	32/32	47/47
	Iš viso: 205/205				
Teisingai nuskaityti duomenys ir sukurtas failas	-	-	-	-	-
	-				
Neiškraipytų gautų sąskaitų „IFS“ sistemoje	35/35	41/41	50/50	32/32	47/47
	Iš viso: 205/205				
Klaidų skaičius	0/35	0/41	0/50	0/32	0/47
	0/205				
Laiko sąnaudos nuo laiško atėjimo į elektroninio pašto dėžutę ir apdorojimo pabaigos visiems laiškam	17 min.	20 min.	36 min.	14 min.	33 min.
Vidutinis laikas vienai sąskaitai apdoroti [s]	29	29	43	26	42
	Iš viso: 120 minučių = 2 valandos				

Atlikus matavimus, matoma, kad sistema nuskaityė visas pateiktas sąskaitas, sukūrė reikalingus XML duomenų failus ir nepadarė nė vienos klaidos palyginus su senuoju procesu (0/205 prieš 7/205). Imant sąskaitos apdorojimo laiko vidurkį, naujasis sąskaitų apdorojimo procesas sąskaitos apdorojimui suapvalinus vidutiniškai užtrunka 34 sekundes, kai senasis sąskaitų apdorojimo procesas vidutiniškai trunka 251 sekundes.

Pagal gautus duomenis nubraižytoje diagramoje matomi procesų rezultatai.



29 pav. Vidutinio sąskaitos apdorojimo laiko grafikas

Vertinant tyrimo antros dalies rezultatus, nustatyta, kad daugiau nei 99% sąskaitų buvo apdorotos teisingai, sistema nepadarė nė vienos klaidos, o laiko sąnaudos buvo daugiau nei 7 kartus mažesnės. Daroma prielaida, kad eksperimentas pavyko, naujam procesui keliami kriterijai yra tenkinami.

5.3. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas

„ReadSoft“ elektroninių sąskaitų automatizavimas veikia be klaidų, nes kuo daugiau sąskaitų sistema apdorojo, tuo tiksliau buvo nustatomi sąskaitoje esantys laukai su apdorojamais duomenimis. Pamėginus įkelti sąskaitą, kurios savininko nėra „IFS“ sistemoje, „ReadSoft“ programinė įranga neleido tęsti sąskaitos apdorojimo iki kol nebuvo gautas naujas *XML* dokumentas su į „IFS“ sistemą įtrauktu nauju tiekėju. Išskirtiniais atvejais, kai atsiųstos sąskaitos kokybė buvo prasta, sistema negalėdavo nustatyti simbolio reikšmės tarp didžiosios O raidės ir skaičiaus nulio.

Analizuojant „Kitron Integration Bus“ veikimą nerasta spragų, visi procesai veikė kaip priklauso ir nereikėjo iš naujo paleisti sistemos dėl integracijos problemų. Buvo testuojama apkrova, kai buvo specialiai įkeliami 100 *XML* failų vienu metu, bet sistema teisingai perduodavo *XML* duomenis „IFS“ sistemai.

„IFS“ sistema visus apdorotus *XML* duomenų failus bei sąskaitų nuotraukas įsikeldavo tinkamai ir beveik niekada nekildavo problemų dėl duomenų įvedimo. Palyginus su senąja sistema, kurioje buvo padarytos trys akies klaidos, naujoji sistema nepadarė nė vienos klaidos, dėl to galime teigti, kad sistemos klaidų skaičiaus kriterijus keliamus reikalavimus atitinka. Analizuojant laiko sąnaudas, senojoje sistemoje didelė laiko dalis priklausė nuo atėjusių sąskaitų skaičiaus. Naujoji sistema, nepriklausomai nuo atėjusių sąskaitų skaičiaus, dirba vienodai greitai, priešingai nei žmogus. Rezultatų lentelėje matome, kad žmogus nuo sąskaitos atidarymo iki jos apdorojimo pradžios užtrunka nuo 1 minutės iki valandos vienai sąskaitą parengti apdoroti, o naujoji sistema tai padaro per 30 sekundžių, nepriklausomai nuo atėjusių sąskaitų skaičiaus. Dėl šio rezultato galima teigti, kad sistema dirba ženkliai greičiau ir laiko sąnaudų kriterijus yra tenkinamas.

5.4. Sprendimo taikymo rekomendacijos

Integruoti tiriamas sistemas yra geriausia *ESB* platformos pagrindu, nes taip yra užtikrinamas greitas, kokybiškas ir saugus duomenų perdavimas tarp dviejų sistemų. Taip pat patartina integruojamas sistemas laikyti tos pačios *IT* infrastuktūros vietoje dėl duomenų persiuntimo greičio, nes, pavyzdžiui, apdorojant sąskaitą ir siunčiant *XML* duomenis iš Kinijos į Švediją buvo labai lėtas interneto ryšys, kas lėmė ilgesnį sąskaitų apdorojimo laiką.

Apdorojant sąskaitas „ReadSoft“ sistema nepatartina siųsti kelių užsakymų numerių vienoje sąskaitoje, nes sistema nesupranta, kuris numeris yra teisingas ir visada klausia buhalterio nurodyti tikrąjį sąskaitos numerį, nors visi numeriai yra teisingi.

6. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

Stebint „Kitron“ grupės sąskaitų apdorojimo procesą, buvo suformuluotas šio darbo tikslas - palengvinti sąskaitų apdorojimo procesą, pagerinti sąskaitų apdorojimo laiką. Kad būtų pasiektas šis tikslas, išanalizuota įmonėje naudojama verslo valdymo sistema „IFS APP 8“, automatizuota sąskaitų apdorojimo sistema „ReadSoft“ ir „Kitron“ grupėje naudojama integracinė platforma „Kitron Integration Bus“. Atlikus šių produktų analizę, buvo padarytos išvados:

1. Šiuo metu pasaulyje nėra integracijos tarp „IFS APP 8“ ir „ReadSoft“ sistemų.
2. Reikia integracinio sprendimo, kuris perduotų duomenis sukurtus „ReadSoft“ sistemos XML formatu ir juos perduoti į „IFS APP 8“ sistemą.
3. Atlikus rinkoje esančių produktų analizę ir juos palyginus pagal iškeltus kriterijus su „Kitron“ grupės turima integracine platforma, nuspręsta sistemų integracijai naudoti „Kitron Integration Bus“ platformą.
4. Buvo suprojektuotas automatizuoto sąskaitų apdorojimo proceso sprendimas pagal sistemai keliamus reikalavimus ir jis įdiegtas „Kitron“ grupės aplinkoje.
5. Ištestavus įdiegtą sprendimą buvo nustatyta, kad sukurtas sprendimas veikia teisingai.
6. Atliktas eksperimentas parodė, kad pritaikytas sprendimas automatizavo sąskaitos apdorojimo procesą ir buvo pasiektas šio darbo tikslas.

Šiame darbe nagrinėtą sistemą automatizuotą sąskaitų apdorojimo sistemą „ReadSoft“ rekomenduoju visoms įmonėms, kurios nori automatizuoti sąskaitų apdorojimo procesą, taip palengvinant darbą įmonės buhalteriams.

7. BIBLIOGRAFIJA

- [1] W3Schools, „W3Schools,“ W3Schools, 2016. [Tinkle]. Pasiukiama: http://www.w3schools.com/xml/xml_whatism.asp. [Kreiptasi 10 gegužės 2016].
- [2] A. Inc, „Adobe.com,“ Adobe Inc, 2016. [Tinkle]. Pasiukiama: <http://www.adobe.com/lt/>. [Kreiptasi 10 gegužės 2016].
- [3] V. Beal, „Webopedia,“ Quinstreet Enterprise, 2016. [Tinkle]. Available: <http://www.webopedia.com/TERM/A/API.html>. [Kreiptasi 10 gegužės 2016].
- [4] W3C, „W3C Protokoliai,“ W3C, 2016. [Tinkle]. Pasiukiama: <https://www.w3.org/Protocols/>. [Kreiptasi 10 gegužė 2016].
- [5] IFS, „Apie IFS sistema,“ IFS, 2016. [Tinkle]. Pasiukiama: <http://www.ifsworld.com/en/about-ifs/>. [Kreiptasi 10 gegužė 2016].
- [6] IFS, „IFS korporacija,“ IFS, 2016. [Tinkle]. Pasiukiama: <http://www.ifsworld.com/en/news/events/world-conference-2013/>. [Kreiptasi 10 gegužė 2016].
- [7] IFS, „IFS moduliai,“ IFS, 2016. [Tinkle]. Pasiukiama: <http://www.ifsworld.com/en/solutions/ifs-applications/>. [Kreiptasi 10 gegužės 2016].
- [8] Lexmark, „ReadSoft invoices,“ Lexmark, 2016. [Tinkle]. [Kreiptasi 10 gegužės 2016].
- [9] J. Ryan, „Computerworld,“ 2011. [Tinkle]. Pasiukiama: <http://www.computerworld.com/article/2510364/infrastructure-management/rethinking-the-esb--building-a-simple--secure--scalable-service-bus-with-a.html>. [Kreiptasi 10 gegužės 2016].
- [10] Microsoft, „Microsoft BizTalk,“ Microsoft, 2016. [Tinkle]. [Kreiptasi 10 gegužės 2016].
- [11] Microsoft, „BizTalk Server 2013 Licensing Datasheet and FAQ,“ Microsoft, 2015. [Tinkle]. Pasiukiama: http://download.microsoft.com/download/0/4/7/0479d5e5-021b-408e-8bca-7d95af76a9a6/biztalk_server_2013_r2_licensing_datasheet_and_faq.pdf. [Kreiptasi 10 gegužės 2016].
- [12] T. S. Inc., „TIBCO ActiveMatrix BusinessWorks 6 aprašymas,“ TIBCO Software Inc., 2016. [Tinkle]. Pasiukiama: <http://www.tibco.com/products/automation/application-integration/activematrix-businessworks?capabilities>. [Kreiptasi 10 gegužė 2016].
- [13] IBM, „IBM ESB aprašymas,“ IBM, 2016. [Tinkle]. Pasiukiama: <ftp://public.dhe.ibm.com/software/integration/wsesh/library/WSW11278-USEN-00.pdf>. [Kreiptasi 10 gegužės 2016].
- [14] IBM, „IBM WebSphere software,“ IBM, 2016. [Tinkle]. Pasiukiama: <http://www.ibm.com/software/websphere>. [Kreiptasi 10 gegužė 2016].
- [15] IBM, „IBM Sterling B2B Integration,“ IBM, 2016. [Tinkle]. Pasiukiama: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=SWGE_ZZ_MR_USEN&htmlfid=ZZS03039USEN&attachment=ZZS03039USEN.PDF. [Kreiptasi 10 gegužė 2016].
- [16] „Adtmag,“ 1105 Media Inc, 2016. [Tinkle]. Pasiukiama: <http://adtmag.com/Articles/2005/10/01/Special-Report-Enterprise-Service-Bus-Synchronicity-in-the-Enterprise.aspx?Page=2>. [Kreiptasi 10 gegužės 2016].
- [17] WebMethods, „WebMethods Integration Overview,“ WebMethods, 2016. [Tinkle]. Pasiukiama: http://www.softwareag.com/corporate/products/webmethods_integration/integration/overview/default.asp. [Kreiptasi 10 gegužė 2016].
- [18] MuleSoft, „MuleSoft aprašymas,“ MuleSoft, 2016. [Tinkle]. Pasiukiama: <https://www.mulesoft.com/platform/soa/mule-esb-open-source-esb>. [Kreiptasi 10 gegužės 2016].

- [19] JBOSS, „JBoSS ESB,“ JBOSS, 2016. [Tinkle]. Available: <http://jbossesb.jboss.org/>. [Kreiptasi 10 gegužēs 2016].
- [20] Murray, James D. , Van Ryper, William, Encyclopedia of Graphics File Formats, 2nd Edition, O'Reilly & Associates, 1996-05-08.

8. PRIEDAI

8.1. Kitron Integration Bus adapteris XML failams nuskaityti

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.IO;
using System.Text;
using System.Xml.Schema;
using IFS_Readsoft_NewSupplier.FileSystem;
using IFS_Readsoft_NewSupplier.Mapping;
using IFS_Readsoft_NewSupplier.Xml;
using NLog;

namespace IFS_Readsoft_NewSupplier
{
    static class Program
    {
        private static readonly Logger Log = LogManager.GetCurrentClassLogger();

        static void Main(string[] args)
        {
            var xmlFilePath = ConfigurationManager.AppSettings["xmlFilePath"];
            var outputPath = ConfigurationManager.AppSettings["outputPath"];
            var outputFileName = ConfigurationManager.AppSettings["outputFileName"];
            var archivePath = ConfigurationManager.AppSettings["archivePath"];
            var archiveInvalidPath = ConfigurationManager.AppSettings["archiveInvalidPath"];
            var fileFilter = ConfigurationManager.AppSettings["fileFilter"];
            var xsdFilePath = AppDomain.CurrentDomain.BaseDirectory +
            ConfigurationManager.AppSettings["XsdFileRelativePath"];

            var processedFiles = new List<FileInfo>();
            var invalidFiles = new List<FileInfo>();

            Log.Debug("Scanning directory for XML files: {0} ({1})", xmlFilePath, fileFilter);
            Log.Debug("Directory for output files: {0}", outputPath);
            Log.Debug("Directory for archived files: {0}", archivePath);
            Log.Debug("XSD for validation: {0}", xsdFilePath);
            Log.Debug("Output file name: {0}", outputFileName);

            var directoryInfo = new DirectoryInfo(xmlFilePath);
            Log.Debug("Found {0} files to be processed.",
            directoryInfo.GetFiles(fileFilter).Length);

            var mapper = new Mapper();
            Log.Debug("Header: {0}", mapper.GetHeader());

            var outputFileBuilder = new StringBuilder();
            outputFileBuilder.AppendLine(mapper.GetHeader());

            try
            {
```

```

foreach (var file in directoryInfo.GetFiles(fileFilter))
{
    Log.Info("Processing file: {0}", file.Name);

    try
    {
        XMLValidator.ValidateXmlWithXsd(file.FullName, xsdFilePath);

        Log.Debug("Validated with XSD: OK");

        var mappedString = mapper.PerformMapping(file.FullName, Log);
        Log.Debug("Mapped output: {0}", mappedString);

        outputFileBuilder.AppendLine(mappedString);

        processedFiles.Add(file);
    }
    catch (XmlSchemaValidationException)
    {
        invalidFiles.Add(file);
    }
}

if (processedFiles.Count > 0 || invalidFiles.Count > 0)
{
    using (var outfile = new StreamWriter(outputPath + outputFileName))
    {
        foreach (var file in processedFiles)
        {
            FileTools.MoveAndOverwrite(file.FullName, archivePath + file.Name);
            Log.Debug("Moved file {0} to {1}", file.FullName, archivePath +
file.Name);
        }

        foreach (var file in invalidFiles)
        {
            FileTools.MoveAndOverwrite(file.FullName, archiveInvalidPath +
file.Name);
            Log.Debug("Moved invalid file {0} to {1}", file.FullName,
archiveInvalidPath + file.Name);
        }

        outfile.Write(outputFileBuilder.ToString());
        Log.Debug("Written outfile to {0}", (outputPath + outputFileName));

        var mailBody = String.Format("Processed {0} valid and {1} invalid xml files
from IFS", processedFiles.Count, invalidFiles.Count);
        SendEmail("ReadSoft New Supplier: Success", mailBody);
        Log.Info(mailBody);
    }
}
else
{

```

```

        var mailBody = String.Format("No files to process.");
        SendEmail("ReadSoft New Supplier: Success", mailBody);
        Log.Info(mailBody);
    }
}
catch (Exception exception)
{
    Log.Error(exception);

    var mailBody = new StringBuilder();
    mailBody.AppendLine("Type: " + exception.GetType());
    mailBody.AppendLine("Message: " + exception.Message);
    mailBody.AppendLine("Stacktrace: " + exception.StackTrace);
    mailBody.AppendLine("InnerException: " + exception.InnerException);
    mailBody.AppendLine("ProcessedFiles:");

    foreach (var processedFile in processedFiles)
    {
        mailBody.AppendLine("\t" + processedFile.Name);
    }
    mailBody.AppendLine("Might need to double check if these files are archived and
needs to be moved back to input folder.");

    SendEmail("ReadSoft New Supplier: Error", mailBody.ToString());
}

#if DEBUG
    Console.ReadLine();
#endif
}

private static void SendEmail(String strSubject, String strBody)
{
    Console.WriteLine("Sending Email");
    var message = new System.Net.Mail.MailMessage();
    message.To.Add(ConfigurationManager.AppSettings["mailNotifyAddress"]);
    message.Subject = strSubject;
    message.From = new System.Net.Mail.MailAddress(Environment.MachineName +
".integrations@kitron.com");
    message.Body = strBody;

    var smtp = new
System.Net.Mail.SmtpClient(ConfigurationManager.AppSettings["mailSmtpServerAddress"]);
    smtp.Send(message);
}
}
}

```

8.2. ReadSoft sugeneruojamo XML failo pavyzdys

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<IFSFInvoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SellerPartyDetails>

```



```

    <SellerPartyIdentifier>5E556036850701</SellerPartyIdentifier>
  </SellerPartyDetails>
  <InvoiceRecipientContactPersonName>*</InvoiceRecipientContactPersonName>
  <BuyerPartyDetails>
    <BuyerPartyIdentifier>SE556588973901</BuyerPartyIdentifier>
  </BuyerPartyDetails>
  <InvoiceDetails>
    <InvoiceTypeCode CodeListAgencyIdentifier="SPY">INV01</InvoiceTypeCode>
    <InvoiceNumber>696683796</InvoiceNumber>
    <InvoiceDate>20140108</InvoiceDate>
    <OrderIdentifier>2229581;2229646</OrderIdentifier>
    <InvoiceTotalVatExcludedAmount
AmountCurrencyIdentifier="SEK">3992.90</InvoiceTotalVatExcludedAmount>
    <InvoiceTotalVatAmount
AmountCurrencyIdentifier="SEK">998.10</InvoiceTotalVatAmount>
    <InvoiceTotalVatIncludedAmount
AmountCurrencyIdentifier="SEK">4991.00</InvoiceTotalVatIncludedAmount>
    <VatSpecificationDetails>
      <VatRatePercent>20.00</VatRatePercent>
      <VatRateAmount>998.10</VatRateAmount>
    </VatSpecificationDetails>
    <InvoiceFreeText />
    <PaymentTermsDetails>
      <InvoiceDueDate Format="CCYYMMDD">20140207</InvoiceDueDate>
    </PaymentTermsDetails>
  </InvoiceDetails>
  <EpiDetails>
    <EpiPartyDetails>
      <EpiBeneficiaryPartyDetails>
        <EpiAccountID IdentificationSchemeName="IBAN">291-6989</EpiAccountID>
      </EpiBeneficiaryPartyDetails>
    </EpiPartyDetails>
    <EpiPaymentInstructionDetails>
      <EpiRemittanceInfoIdentifier IdentificationSchemeName="SPY" />
    </EpiPaymentInstructionDetails>
  </EpiDetails>
  <SellerIDAtBuyer>125361</SellerIDAtBuyer>
  <InvoiceAttachments>
    <FileName>SWE_PO_28N18000000.TIF</FileName>
    <DocumentType>ATS</DocumentType>
    <Attachment>GDFhasdjkhfsapodfgdfsgdsfg5s4dfg65.....
    <MimeType>TIF</MimeType>
  </InvoiceAttachments>
</IFSFinvoice>

```

8.3. Apdorojamos sąskaitos pavyzdys



INVOICE LASKU

UAB KITRON
TAIKOS PR. 151
52119 KAUNAS
LITHUANIA

Date	Invoice number
09.05.2016	0034057
Customer number	Your reference
001078	3212513
Terms of payment	Our reference
45 days net	
Address	Delivered Duty Unpaid
Jolanta Malinauskienė	
Notice time	Overdue interest
7 Days	14,00 %
Customer VATA	Order
LT106818610	26181

Code/Koodi	Description/Nimike	Qty	Price/Hinta	Cur.Val	Exchange cur/Sidkunsai	Total
LTC2050C556T11M6/F Kitron partnr.: 4022804	OPAMP/ROHS/ZERO DRIF	500 Pcs	1,100	USD	USD 1,00000	550,00
				Exchange correction: USD	1,00000	0,000
<p>VAT VAT % No VAT incl. VAT Contact person: Piyo Kalvola</p> <p>ALV 0 % 550,00 0,00</p> <p>Total VAT in EUR 0,00</p> <p>VAT 0% Intra-Community supply</p>						
Net weight	2,50Kg	Reference number		TOTAL	USD	550,00
		00107800340575				
		Due date				
		23.06.2016				

Address: Fintronic Oy
 Malmin kauppalei 8 A
 00700 HELSINKI
 Finland

Bank: Nordea Malmi
 EUR FI3210253000239429
 USD FI5410255200015807
 SWIFT: NDEAFI33

Tel: +358 9 25127770
 Fax: +358 9 8797770
 VAT number: FI19702555

8.4. „ReadSoft“ sistemas sugeneruotas XML dokumentas

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<IFSFinvoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SellerPartyDetails>
    <SellerPartyIdentifier></SellerPartyIdentifier>
  </SellerPartyDetails>
  <InvoiceRecipientContactPersonName>*</InvoiceRecipientContactPersonName>
  <BuyerPartyDetails>
    <BuyerPartyIdentifier></BuyerPartyIdentifier>
  </BuyerPartyDetails>
  <InvoiceDetails>
    <InvoiceTypeCode CodeListAgencyIdentifier="SPY"></InvoiceTypeCode>
    <InvoiceNumber></InvoiceNumber>
    <InvoiceDate></InvoiceDate>
    <OrderIdentifier></OrderIdentifier>
    <InvoiceTotalVatExcludedAmount
AmountCurrencyIdentifier="SEK"></InvoiceTotalVatExcludedAmount>
    <InvoiceTotalVatAmount
AmountCurrencyIdentifier="SEK"></InvoiceTotalVatAmount>
    <InvoiceTotalVatIncludedAmount
AmountCurrencyIdentifier="SEK"></InvoiceTotalVatIncludedAmount>
    <VatSpecificationDetails>
      <VatRatePercent></VatRatePercent>
      <VatRateAmount></VatRateAmount>
    </VatSpecificationDetails>
    <InvoiceFreeText />
    <PaymentTermsDetails>
      <InvoiceDueDate Format="CCYYMMDD"></InvoiceDueDate>
    </PaymentTermsDetails>
  </InvoiceDetails>
  <EpiDetails>
    <EpiPartyDetails>
      <EpiBeneficiaryPartyDetails>
        <EpiAccountID IdentificationSchemeName="IBAN"></EpiAccountID>
      </EpiBeneficiaryPartyDetails>
    </EpiPartyDetails>
    <EpiPaymentInstructionDetails>
      <EpiRemittanceInfoIdentifier IdentificationSchemeName="SPY" />
    </EpiPaymentInstructionDetails>
  </EpiDetails>
  <SellerIDAtBuyer></SellerIDAtBuyer>
  <InvoiceAttachments>
    <FileName></FileName>
    <DocumentType></DocumentType>
    <Attachment>
      <MimeType></MimeType>
    </Attachment>
  </InvoiceAttachments>
</IFSFinvoice>
```

8.5. PowerShell kopijavimo kodas

```
##*=====
##* Function:      SendMail
##* Arguments:    Arg0 = Subject.
##*                                     Arg1 = Body.
##*=====
##* Purpose:      Sends mail to desired
##*                                     mail-group. Sends various
##*                                     mails if something went wrong.
##*=====
```

```
Function SendMail {
```

```
    $emailFrom = "$env:computername@kitron.com"
    $emailTo = "contact@email.com"
    $smtpServer = "193.69.13.141"
    $smtp = new-object Net.Mail.SmtpClient($smtpServer)
    $smtp.Send($emailFrom, $emailTo, $args[0], $args[1])
```

```
}
```

```
##*=====
##* Function:      CopyFiles
##* Arguments:    Arg0= Source
##*               Arg1= Destination
##*               Arg2= Current file to move
##*               Arg3= MOVEOK log path and name
##*               Arg4= Error log path and name
##*=====
##* Purpose:      Copies file(s) from source to
##*                                     destination.
##*
##*
##*=====
```

```
Function CopyItem {
```

```
    $src = $args[0]
    $dst = $args[1]
    $currentfile = $args[2]
    $copyoklog = $args[3]
    $errorlog = $args[4]
    $filecopy = ($src + "\" + $currentfile)
    Copy-Item $filecopy $dst
```

```
    if($? -eq $false)
```

```
    {
```

```
        $datetime = (get-date).ToString()
```

```
        Write-Output "[$datetime] - Could not copy $currentfile from $src to $dst" >>
```

```
"$errorlog"
```

```
        SendMail "Readsoft file not copied!" "FAILED TO COPY FILE FROM $src TO $dst
```

```
FILE: $currentfile"
```

```
    }
```

```

else
{
    $curtime = get-date -format t
    Write-Output "[$curtime] COPY SUCCESSFULLY FROM $src TO $dst FILE:
$currentfile" >> "$copyoklog"
}
}

#*=====
#* Function:      MoveFiles
#* Arguments:    Arg0= Source path
#*               Arg1= Destination path
#*               Arg2= Current file to move
#*               Arg3= MOVEOK log path and name
#*               Arg4= Error log path and name
#*=====
#* Purpose:      Moves files from source to
#*                                     destination.
#*
#*
#*=====

Function MoveItem {
    $src = $args[0]
    $dst = $args[1]
    $currentfile = $args[2]
    $moveoklog = $args[3]
    $errorlog = $args[4]
    $filemove = ($src + "\" + $currentfile)

    Move-Item $filemove $dst -ErrorVariable "moveerror"

    if($? -eq $false)
    {
        $datetime = (get-date).Tostring()
        $mailbody = @"
Failed to moved file from $src to $dst
File: $currentfile
Error code:
$moveerror

"@
        Write-Output "[$datetime] - COULD NOT MOVE FILE FROM $src TO $dst FILE:
$currentfile" >> "$errorlog"
        SendMail "Readsoft file not moved!" "$mailbody"
    }
    else
    {
        $curtime = get-date -format t
        Write-Output "[$curtime] MOVE SUCCESSFULLY FROM $src TO $dst FILE:
$currentfile" >> "$moveoklog"
    }
}

```

```
}
```

```
##*=====
##* Function:      Pathcheck
##* Arguments:    Arg0= Path to check
##*=====
##* Purpose:      Tests a path, returns $true
##*                                     or $false.
##*
##*=====
```

```
Function Pathcheck {
    $path = $args[0]
    $result = Test-Path $path
```

```
    if ($result -eq $true) {
        Write-Host "$path is available!"
```

```
    }
```

```
    else {
        SendMail "File path are not available!" "SCRIPT WILL TERMINATE! The following
path are not available: $path "
        Write-Host "Path finnes ikke"
        Exit
    }
}
```

```
}
```

```
##*=====
##* Function:      Filecheck
##* Arguments:    Arg0= Path to check
##*=====
##* Purpose:      Tests a path, returns $true or $false.
##*=====
```

```
Function Filecheck {
    $files = $args[0]
    $result = Test-Path ($files + "\" + "*.*")
```

```
    if ($result -eq $true) {
```

```
        $getfilesindir = Get-ChildItem $rsrprtrans -name -Include *.* | Out-File ($rspath + "\" +
"temppathfiles.txt")
```

```
        $filelist = [string]::join([environment]::NewLine, (get-content ($rspath + "\" +
"temppathfiles.txt")))
    }
```

```
    $filesindir = @"
```

```
There are files present in IFS pickup folder.
Please check Readsoft solution as soon as possible!
```

```
Filepath : $files
```

```

Files      :
$filelist
"@

    SendMail "IFS is not picking up Readsoft files!" $filesindir
    Remove-Item ($rspath + "\" + "temppathfiles.txt")
    Write-Host "File(s) present in IFS Pickup Point!"

    }
}

#*=====
#*----- BODY -----
#*=====

# Current date
$dt = get-date -format d

# Readsoft Workdir
$rspath = "\\kit-fs01\felles\EDI\ReadSoft"

# Readsoft output location
$rsoutput = ($rspath + "\" + "transfer" + "\" + "prod" + "\" + "readsoft_out")

# Readsoft backup location
$rsbackup = ($rspath + "\" + "backup" + "\" + "prod" + "\")

# ERP-System pick-up point folder
$rsrptrans = ($rspath + "\" + "transfer" + "\" + "prod" + "\")

# File extension on output from Readsoft
$fileoutput = "*.xml"

# Common Error log
$errorlog = ($rspath + "\" + "logs" + "\" + "RS_ERROR_TRANSFER.log")

# CopyItem variables
$copyoklog = ($rspath + "\" + "logs" + "\" + "$dt - COPYOK.log")

# MoveItem variables
$moveoklog = ($rspath + "\" + "logs" + "\" + "$dt - MOVEOK.log")

# Check if Readsoft Workdir is available
Pathcheck $rspath

# Check if any xml files in IFs pickup point is available
Filecheck $rsrptrans

# Locate and create a list consisting of XML files to copy
$rsxmlfiles = @(Get-ChildItem $rsoutput -name -Include *.xml | Where-Object
{!(($_.PSIsContainer)})

```

```
#Count the files to tranfer and if its no files to terminate, exit script.
$filecount = $rsxmlfiles.Count
if ($filecount -eq 0) {
    Write-Host "No files to transfer!"
    Exit
}

# Wait 10 seconds to ensure that the write-file process from Readsoft is finished
start-sleep -s 10

# Copies and moves each files one by one
foreach ($xmlfile in $rsxmlfiles) {

    # Copy of XML files from Readsoft output location to Backup location
    CopyItem $rsoutput $rsbackup $xmlfile $copyoklog $errorlogg

    # Moves XML files from Readsoft output location to IFS pickup point
    MoveItem $rsoutput $rsrprtrans $xmlfile $moveoklog $errorlogg

}
```


Įmonės gamybos ir verslo valdymo sistemų integracija

