



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Tomas Urbanavičius

**DUOMENŲ PERKĖLIMO TARP INFORMACINIŲ SISTEMŲ ANALIZĖ IR
ĮGYVENDINIMAS**

**ANALYSIS AND IMPLEMENTATION OF DATA MIGRATION BETWEEN
INFORMATION SYSTEMS**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 62407T104

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos mokslo kryptis

Vilnius, 2010

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU

Katedros vedėjas

(Parašas)

Genadijus Kulvietis

(Vardas, pavardė)

(Data)

Tomas Urbanavičius

**DUOMENŲ PERKĖLIMO TARP INFORMACINIŲ SISTEMŲ ANALIZĖ IR
ĮGYVENDINIMAS**

**ANALYSIS AND IMPLEMENTATION OF DATA MIGRATION BETWEEN
INFORMATION SYSTEMS**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 62407T104

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos mokslo kryptis

Vadovas dr. Mindaugas Rybokas

(Moksl. laipsnis, vardas, pavardė)

(Parašas)

(Data)

Konsultantas doc. dr. Angelė Kaulakienė

(Moksl. laipsnis, vardas, pavardė)

(Parašas)

(Data)

Konsultantas _____

(Moksl. laipsnis, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2010

Vilniaus Gedimino technikos universitetas

Fundamentinių mokslų fakultetas

Informacinių technologijų katedra

ISBN

ISSN

Egz. sk. 2

Data 2010-06-02

Informacinių technologijų studijų programos baigiamasis magistro darbas

Pavadinimas **Duomenų perkėlimo tarp informacinių sistemų analizė ir įgyvendinimas**

Autorius **Tomas Urbanavičius**

Vadovas dr. **Mindaugas Rybokas**

Kalba

☒ X

lietuvių

☐

užsienio

Anotacija

Baigiamajame magistro darbe yra tiriamas duomenų perkėlimas tarp informacinių sistemų. Šio darbo tikslas – perkelti kliento duomenis iš senos informacinės sistemos į naują.

Darbe pateikiami dažniausiai praktikoje naudojami duomenų perkėlimo būdai bei jų privalumai ir trūkumai. Išsamiai pateikti duomenų perkėlimo etapai ir kaip jie turi būti taikomi duomenų perkėlimo procese.

Praktinėje dalyje pristatytas duomenų perkėlimo įrankis, sukurtas įmonėje UAB „Exigen Services Vilnius“. Programa sukurta seniems draudimo polisų duomenims perkelti naudojant XML rinkmeną. Kūrimo proceso metu buvo sudaryta duomenų perkėlimo specifikacija, pasirinkta duomenų perkėlimo metodika ir sukurtas duomenų perkėlimo įrankis.

Darbą surado 5 dalys: duomenų perkėlimo metodika, duomenų perkėlimo etapai ir problemos, duomenų perkėlimo procesas, duomenų perkėlimo įgyvendinimas, duomenų perkėlimo specifikacija.

Darbo apimtis 63 puslapiai teksto be priedų, 25 paveikslai, 2 lentelės ir 11 bibliografinių šaltinių.

Prasminiai žodžiai: duomenų perkėlimas, XML, XSD, ACORD, duomenų tikrinimas, duomenų transformavimas.

Vilnius Gediminas Technical University
Fundamental science faculty
Information technologies department

ISBN ISSN
Copies No. 2
Date 2010-06-02

Information Technologies study programme master thesis
Title: **Analysis and implementation of data migration between information systems**
Author **Tomas Urbanavičius** Academic supervisor dr. **RMindaugas Rybokas**

Thesis language

☐

Lithuanian

☒

Foreign (English)

Annotation

A research of master's final work is data migration between information systems. The aim of this work is to migrate client data from old information system to new one.

Data migration methodologies in this work are thoroughly analyzed. Most popular data migration methodologies are analyzed, by introducing their advantages and drawbacks and how they should be applied in data migration process.

Data migration tool example is also analyzed in this work. Software tool, that was implemented in UAB „Exigen Services Lietuva“ organization, is analyzed in this master's work. This data loading tool is using XML for information transfer between export tool and data loader. Data loader also has its specification, that is also described in this job.

Thesis structure consists of 5 parts: data migration methodology, data migration phases and problems, data migration process, data migration implementation, data migration tool specification.

Thesis consists of 63 pages of text without extras, 25 pictures, 2 table and 11 bibliographical entries.

Keywords: Data migration, XML, XSD, ACORD, data validation, data transformation.

Paveikslų sąrašas

- 1 pav. Globalios duomenų perkėlimo rinkos prognozės ir perviršiai 2007-2012
- 2 pav. Informacinių sistemų projektų tipai
- 3 pav. Projektų biudžetai
- 4 pav. Duomenų perkėlimui skiriama biudžeto dalis
- 5 pav. Biudžeto pereikvojimo priežastys
- 6 pav. Biudžeto pereikvojimas procentais
- 7 pav. Prognozuojamos globalios išlaidos, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus
- 8 pav. Prognozuojamos išlaidos Šiaurės Amerikos regione, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus
- 9 pav. Prognozuojamos išlaidos Azijos ir Ramiojo vandenyno regione, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus
- 10 pav. Prognozuojamos išlaidos Vakarų Europoje, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus
- 11 pav. Perviršio mastai
- 12 pav. Perviršio priežastys
- 13 pav. Naudojamų įrankių tipai
- 14 pav. Duomenų šaltinio taikomosios programos langas
- 15 pav. Bendroji architektūrinė IPB sistemos schema
- 16 pav. XML rinkmenos ištrauka
- 17 pav. HTML kalba atvaizduotas kodas
- 18 pav. Standartinės ACORD XSD schemos ištrauka
- 19 pav. Papildytos ACORD XSD schemos ištrauka
- 20 pav. IPB duomenų bazės lentelių medis
- 21 pav. Ryšiai tarp poliso ir jo finansinius duomenis saugančių duomenų bazės lentelių
- 22 pav. Duomenų perkėlimo įrankio katalogų ir rinkmenų struktūra
- 23 pav. XML elementų ir IPB sistemos laukų priskyrimo dokumento ištrauka
- 24 pav. Klaidos XML rinkmenoje registravimas ir išvedimas į pultą
- 25 pav. Sėkmingo XML rinkmenos apdorojimo vykdymo registro išvedimas į pultą

Lentelių sąrašas

1 lentelė. Globali duomenų perkėlimo biudžeto prognozė milijonais dolerių pagal industrijos sektorius 2007-2012 m.

2 lentelė. Esminiai rinkos rodikliai

Santrumpų paaiškinimai

Santrumpa	Paaiškinimas
IS	Informacinė sistema
IPB	Draudimo sistemos pagrindas/stuburas, angl. Insurance Process Backbone. „Exigen Services“ įmonėje sukurtos draudimo sistemos sutrumpintas pavadinimas.
BLS	Verslo logikos sistema, angl. Business Logic System, skirta pagal iš anksto nustatytas taisykles tikrinimui atlikti.
XML	Paprasta, struktūrizuota kalba, skirta struktūroms tekstiniu formatu perteikti. Angl. Extensible Markup Language.
XLS	Microsoft Excel programoje naudojamas skaičiuoklių duomenų išsaugojimo formatas.
XSD	XML struktūros taisyklių visumos apibrėžimas, naudojamas apibrėžti XML rinkmenos struktūrą, angl. XML Schema Definition
ACORD	Draudimo industrijos tyrimų ir kūrimo ne pelno siekianti asociacija, užsiimanti standartų kūrimu, angl. Association for Cooperative Operations Research and Development
FORBES GLOBAL 2000	Kasmetinis žurnalo „Forbes“ sudarytas geriausių 2000 pasaulio kompanijų sąrašas. Įmonės reitinguojamos pagal keletą kriterijų: pardavimai, pelnas, pajamos ir rinkos vertė. Sąrašas pateikiamas kasmet nuo 2003 metų.
CAGR	Sudėtinio periodinio augimo dažnis, kuriuo apibūdinamos periodinės pajamos iš investicinio objekto per tam tikrą laiko tarpą, angl. Compound Annual Growth Rate.
AES	Simetrinio rakto šifravimo algoritmo standartas, priimtas naudoti JAV vyriausybės.
CSV	Paprastas tekstinio dokumento tipas, dažniausiai naudojamas duomenų bazių programinėje įrangoje, angl. Comma Separated Value

Turinys

Įvadas.....	10
1. Duomenų perkėlimo metodika	12
1.1. Duomenų perkėlimo metodai.....	12
1.1.1. Didžiojo sprogimo metodas.....	13
1.1.2. Lygiagretusis sistemų veikimas.....	14
1.1.3. Perkėlimas dalimis.....	14
1.1.4. Nulinio išjungimo laiko perkėlimas	15
2. Duomenų perkėlimo etapai ir problemos	16
4.1. Duomenų perkėlimo apibrėžimas.....	17
4.2. Esminiai faktai.....	18
4.3. Duomenų perkėlimo rinka.....	18
4.4. Rezultatai ir išvados	20
4.5. Tolesnės pasekmės	29
4.6. Problemų priežastys	29
5. Duomenų perkėlimo procesas	33
5.1. Fazė 0: Keliamų užduočių ir jų įvykdymo galimybių įvertinimas.....	33
5.2. Fazė 1: Planavimas ir projektavimas.....	34
5.3. Fazė 2: Pasikeitimų valdymas	36
5.4. Fazė 3: Duomenų perkėlimas.....	36
5.5. Fazė 4: Rezultatų analizė ir apžvalga po migravimo	37
5.6. Išteklių priskyrimas	38
5.7. Tiesinis duomenų perkėlimas.....	38
5.8. Iteracinis duomenų perkėlimas.....	39
6. Duomenų perkėlimo įgyvendinimas	39
6.1. Duomenų šaltinis – sena informacinė sistema	40
6.2. Draudimo procesų sistema – IPB.....	40
6.2.1. Bendroji sistemos architektūra	40
6.2.2. Sistemai keliami reikalavimai	42
6.2.3. Pagrindiniai sistemos tikslai	43
6.2.4. Ryšių su klientais valdymo posistemė.....	43
6.3. Duomenų perkėlimo metodika.....	44
6.4. Perkeliamų duomenų pobūdis.....	45
6.5. XML dokumentas, XML standartas	45

6.6.	Duomenų perkėlimo įrankio tikrinimas	50
7.	Duomenų perkėlimo specifikacija.....	51
7.1.	Funkcionalumas	51
7.2.	Duomenų kiekis ir nefunkciniai reikalavimai	51
7.3.	Techninis duomenų perkėlimo įrankio dizainas.....	51
7.4.	Duomenų ryšiai lentelėse	52
7.5.	Katalogų struktūra	54
7.6.	Duomenų perkėlimo procesas	57
7.7.	ACORD standarto naudojimas RPL duomenims.....	57
7.8.	Duomenų perkėlimo įrankio naudojimas	58
7.9.	Programos proceso registravimas	58
7.10.	Perkeliamų duomenų tikrinimas.....	58
7.11.	Perkeltų polisų papildomas žymėjimas IPB sistemoje.....	59
7.12.	Perkeltų polisų naudojimas IPB sistemoje	59
7.13.	Duomenų eksportavimas iš šaltinio.....	60
7.14.	Duomenų perkėlimo įrankio nuostatos ir paruošimas darbui.....	60
7.15.	Duomenų perkėlimo įrankio naudojimas	61
	Išvados.....	63
	Literatūros sąrašas	65
	Priedai	66

Ivadas

Aktualumas

Duomenų perkėlimas verslo pasaulyje yra vis dažniau aptinkamas procesas. Įvairios įmonės ir organizacijos pageidauja, kad būtų išsaugoti seni sistemos duomenys. Kadangi informacinių sistemų taikymas pasiekė tokį lygį, kai verslo procesuose stengiamasi kuo efektyviau panaudoti visas kompiuterio galimybes. Todėl dažnai ankstesnės informacinės sistemos, sukurtos verslo procesams kompiuterizuoti, tačiau nebeatitinkančios dabartinių verslo reikalavimų ir lūkesčių, yra keičiamos. Dažnas užsakovas kaip vieną iš užduočių, diegiant naują informacinę sistemą, reikalauja, kad būtų perkelti duomenys iš senos sistemos į naujesnę.

Kokybiškas ir tikslus duomenų perkėlimas dabartinėje informacinių technologijų rinkoje yra siekiamybė, todėl atsiranda žmonių, organizacijų, kurios bando formuoti duomenų perkėlimo technologijų tipus ir metodologiją. Tokios metodologijos aktualumas nuolat didėja ir reikalauja sukauptos patirties apžvalgos ir analizės.

Keliami tikslai ir uždaviniai

Darbo pagrindinis tikslas yra sėkmingai sukurti įrankį, galintį perkelti duomenis į naujai kuriamą informacinę sistemą. Šis tikslas yra sudarytas iš keleto konkrečių uždavinių:

- Pasirinkti duomenų perkėlimo būdą, atitinkantį kliento poreikius ir lūkesčius.
- Patikrinti įkeliamus duomenis, kad jie atitiktų tikslinės sistemos duomenų reikalavimus.
- Pažymėti duomenis, šiuo būdu įkeliamus į tikslinę sistemą, kad vėliau juos galima būtų atskirti nuo visų kitų.
- Sukurti galimybę pakartotinai įkelti duomenis, jei juos reikės pataisyti dėl tam tikrų neatitikčių ar tam tikrus pasirinktus duomenis pašalinti.

Mokslinis naujumas

Darbe pateikti duomenys apie duomenų perkėlimo rinką, realybėje kylančias problemas ir jų sprendimo būdus. Pateikta užsienio organizacijų duomenų perkėlimo projektų tyrimo informacija. Joje pristatomos dažniausiai kylančios problemos ir formuojama tinkamų sprendimų praktika duomenų perkėlimo projektams.

Nors tyrimo ataskaitoje pateikti pasiūlymai nėra griežti, jų aktualumas ir siūlomi sprendimai gali padėti įveikti nemažai kylančių problemų ar tiesiog būti prevencine priemone, mažinančia projekto riziką. Šiame darbe aprašytas sukurtas programinis įrankis atitinka perkėlimo dalimis metodo specifiką. Taip pat pateiktos problemos, kilusios įrankio egzistavimo laikotarpiu, ir galimi jų sprendimo būdai.

Praktinė nauda

Buvo sukurtas duomenų perkėlimo įrankis ir atliktas duomenų perkėlimas, o tai ir yra pagrindinis šio darbo tikslas. Todėl darbo praktinė nauda yra akivaizdi. Perkėlus duomenis iš senos informacinės sistemos, informacinės sistemos užsakovas ir toliau galės be didesnių problemų teikti paslaugas, kurios buvo teikiamos su senąja sistema. Esamų klientų duomenų valdymas yra esminis sėkmingo sistemos egzistavimo poreikis.

Sukurtas įrankis ir perkėlimo metu įgyta patirtis įmonėje gali būti pritaikoma kuriant naujas informacines sistemas. Įgyta patirtis galėtų būti panaudota ir kuriant naują programinį įrankį, jei perkeliama duomenys būtų visai kito pobūdžio ir nebūtų galima pritaikyti aprašytąjį įrankį.

Aprobacija

Darbas buvo aprobuotas 12-ojoje Lietuvos jaunųjų mokslininkų konferencijoje „Mokslas – Lietuvos ateitis“, (Vilnius, Lietuva, 2009 m. balandžio mėn. 13 d.), Informatikos sekcijoje.

1. Duomenų perkėlimo metodika

Duomenų perkėlimas yra būtinas, kai norima prijungti arba įdiegti naują sistemą, arba kai perkama ar programuojama nauja sistema. Reikia žinoti tik vieną, jog norint tvarkyti tos pačios rūšies duomenis, reikia atlikti panašias užduotis ir darbus. Be to, informacija iš vienos sistemos turi būti lengvai perkeliama į kitą. Tačiau tai gana retas atvejis.

Nors duomenų perkėlimas gali reikalauti gana daug laiko, bet nauda gali būti verta kainos tiems, kurie duomenis naudoja finansų, rinkodaros, prekybos ir pan. srityse, kur labai svarbu matyti praeitų laikotarpių duomenis ir pagal juos atlikti įvairias analizes bei numatyti tendencijas. Taip pat ir senos taikomosios programos turi būti prižiūrimos ir tvarkomos [11].

1.1. Duomenų perkėlimo metodai

Klausimas „Ar reikia perkelti senus duomenis?“ yra keliamas nuo to laiko, kai pirmosios įmonės pradėjo saugoti duomenis saugyklose ir vėliau nusprendė pakeisti sistemas. Žemiau išvardyti klausimai yra dažniausiai keliami prieš duomenų perkėlimą:

- Ar reikia perkelti duomenis į naują sistemą?
- Jeigu taip, ar reikia perkelti visus duomenis ar tik jų dalį?
- Jei tik dalį, tai kurią dalį? Kokiais kriterijais remiantis?
- Jei nuspręsta perkelti duomenis, tai kokio senumo duomenis reikia perkelti trijų paskutinių mėnesių, šešių paskutinių mėnesių, praeitų metų? Galbūt visus duomenis?
- Ar reikia išrinkti duomenis pagal specialius kriterijus?
- Ar mūsų pageidautini kriterijai tinkami esamai duomenų bazei?
- Ar pageidautini laukai įkeliami į naują sistemą?
- Ar duomenų perkėlimas iš senos sistemos yra įtrauktas į naujos sistemos pirkimo kainą? Jei ne, ar mums reikės kurti scenarijus, kurie automatiškai perkels duomenis į įsigytą naują sistemą.
- Jei ne, ar reikės samdyti kitą įmonę (fizinį asmenį) šiam darbui atlikti?
- Ar ši užduotis bus atliekama rankiniu būdu?

Kai yra galvojama apie duomenų perkėlimą, visi veiksniai turi būti išnagrinėti prieš darant prielaidą, kad visas duomenų rinkinys turi būti perkeltas į naują sistemą arba kad nieko nereikia perkelti į naują sistemą. Patikrinti galima pagal tai, ar tie duomenys bus naudojami, ar turės įtakos naujai sistemai, kai ji bus paleista. Planuojant duomenų perkėlimą yra svarbūs du kintamieji – duomenų apimtis (dydis) ir duomenų vertė (reikšmė).

Duomenų apimtis yra paprasčiausias kintamasis, planuojant duomenų perkėlimo procesą. Koks įrašų skaičius: 1000, 10000, 100000, 250000? Kiek per savaitę ar mėnesį tikimasi perkelti šių duomenų į naująją sistemą? Reikia patikrinti, ar yra techninės kliūtys, trukdančios perkelti tam tikrą duomenų skaičių, ar didelės duomenų bazės paveiks sistemos veikimą atliekant įprastas sistemos funkcijas, pvz., paiešką. Jei ne, tuomet 10 ar 100000 įrašų perkėlimas neturi jokios reikšmės. Jei duomenų talpykla (angl. *Volume*) užima mažai vietos, tada verta perkelti duomenis apie naudotojus, kad jie galėtų atlikti prognozes. Jei duomenų talpykla užima daug vietos, tada verta ištirti duomenų senumą ir vertę. Tai padarius galima atrinkti duomenis pagal konkrečius kriterijus.

Duomenų vertė yra kur kas svarbesnis kintamasis planuojant duomenų perkėlimo procesą. Daromos įvairios prielaidos, iš kurių sprendžiama, kokią vertę suteikia esantys duomenys. Jei naudotojai nederba su duomenimis dabartinėje sistemoje, labai tikėtina, jog jie nesinaudos senaisiais duomenimis ir naujoje sistemoje, net ir pagerinus paieškos veikimą. Perkelti tokius duomenis, kyla natūralus klausimas „Kam perkelti duomenis, kurie lems sistemos veikimą, bet niekada nebus naudojami?“ Kai yra priimtas sprendimas, kuriuos duomenis reikės perkelti, taip pat reikia nuspręsti, kurios duomenų įrašų dalys yra svarbios [11].

1.1.1. Didžiojo sprogimo metodas

Naudojant didžiojo sprogimo (angl. *Big Bang*) metodą visą laiką senoji sistema funkcionuoja kaip įprasta ir nėra išjungiamą visą duomenų perkėlimo projekto laiką. Naujoji sistema neįjungiamą nė vieno etapo metu. Paskutinę akimirką visi duomenys įkeliami į naują sistemą, po šios operacijos senoji sistema išjungiamą, o naujoji įjungiamą. Šis metodas turi 2 privalumus. Pirma, pašalinamos priklausomybės tarp sistemų, nes tam tikros senos sistemos turi visą laiką veikti su naujomis. Antra, šis metodas sumažina aparatinės įrangos ir sistemos reikalavimus, nes tik vieną kartą, kai atliekamas duomenų perkėlimas, abi sistemos veikia. Tačiau taip turėtų būti teoriškai, bet praktiškai tai sunkiai įmanoma padaryti. Duomenų mainų tarp sistemų procesas paprastai atliekamas savaitgaliais, jei didelė sistema, tokia kaip elektroninė banko sistema būtų išjungta, tarkim, 2 savaitėms, galima įsivaizduoti, kokių nuostolių patirtų įmonė. Kita problema, kad net atliekant duomenų perkėlimą savaitgalį, senosios sistemos negalima visiškai išjungti (pvz., elektroninės parduotuvės sistemos). Didžiojo sprogimo metodo trūkumas yra naujosios sistemos veikimas. Kas atsitiks, jeigu sistema veiks blogai arba neveiks išvis. Jei taip įvyks, perkėlimo darbus teks pratęsti ilgesniam nei savaitgalis laikui arba vėl įjungti seną sistemą ir atidėti duomenų perkėlimą kitam savaitgaliui. Bet kuriuo atveju bus padaryta didelė žala informacinių technologijų padalinio reputacijai, o klientas praras pinigus dėl to, kad nauja sistema nebus laiku įjungta [8].

1.1.2. Lygiagretusis sistemų veikimas

Lygiagrečiojo sistemų veikimo principas panašus kaip ir didžiojo sprogimo metodo, skirtumas tik tas, kad perkėlus duomenis senoji sistema nėra išjungžiama. Priešingai, abi sistemos veikia lygiagrečiai (vienu metu), senoji sistema naudojama tam, kad naujoji sistema būtų iki galo ištestuota prieš leidžiant ją naudotis sistemos naudotojams. Privalumas yra tas, kad naujoji sistema veiks gerai, kai duomenis pagaliau bus perkelti. Be to, senoji sistema yra išjungžiama labai trumpam laikui, kai duomenys perkeliama. Tačiau lygiagretusis veikimas yra labai brangus, ne tik dėl to, kad reikia nemažų kompiuterinės įrangos išteklių, bet ir dėl to, kad reikia nemažai žmogiškųjų išteklių, kurie prižiūrėtų senąją sistemą ir užtikrintų, kad naujoji sistema veikia taip, kaip ir buvo planuota. Lygiagretusis veikimas yra kur kas sudėtingesnis nei didžiojo sprogimo metodas, nes įeinančius į senąją sistemą duomenis taip pat reikia perkelti į naująją. Vadinasi, reikalinga tam tikros paskirties programinė įranga, kuri atliktų sinchronizaciją tarp abiejų sistemų. Kad ir kokių metodų ar būdų prireiktų perkeliant duomenis, yra trys pagrindiniai metodai: didžiojo sprogimo, lygiagretusis veikimas ir perkėlimas dalimis [8].

1.1.3. Perkėlimas dalimis

Perkėlimas dalimis veikia panašiai kaip ir lygiagretusis perkėlimas, išskyrus tai, kad nuolat yra įjungžiama naujoji sistema, užuot ją įjungus vieną kartą. Taigi gali nutikti taip, kad tam tikros transakcijos yra atliekamos senoje sistemoje, o kitos jau naujoje sistemoje. Lygiai taip pat tam tikros funkcijos gali būti atliekamos senoje sistemoje, o kitos naujoje sistemoje. Vienas iš perkėlimo dalimis privalumų tas, kad investuoti pinigai gali grįžti kur kas greičiau nei buvo tikėtasi. Taip pat reikia atkreipti dėmesį, kad kai visi duomenys bus perkelti į naują sistemą ir bus laikas išjungti seną sistemą, įvyks lygiagretusis veikimas, kuris truks labai trumpą laiką, jei išvis įvyks. Paprastai atliekant perkėlimo dalimis metodą naujoje sistemoje yra sukuriama žymės duomenų bazė, kurios nurodo programai, kurių duomenų ieškoti naujoje sistemoje. Trūkumas – duomenis naujoje sistemoje nuolat reikia taisyti ir prižiūrėti, tokiu atveju gali pablogėti sistemos veikimas, kai duomenys yra naujoje sistemoje. Kitas būdas - naudoti būsenos modelį, tokiu atveju yra sukuriama įrašas apie esamą duomenų perkėlimo būseną. Iš šio įrašo programa žino, kada jai reikia kreiptis į naują sistemą, o kada į seną prireikus tam tikrų duomenų.

Nesvarbu, kuris būdas yra naudojamas duomenims perkelti, svarbiausia, kad bet kuriuo metu galima būtų atšaukti atliktus keitinius, jei dėl kažkokių priežasčių jie neveiktų. Tai labai sunku atlikti naudojant didžiojo sprogimo arba lygiagrečiojo perkėlimo metodą, abiem atvejais duomenų rinkiniai yra perkeliama per labai trumpą laiką. Perkeliant dalimis, perkeliama palyginti mažas

duomenų skaičius tam tikru laiko momentu. Tai taip pat labai palengvina duomenų tikrinimo procedūrą, nes ji irgi gali būti atliekama dalimis. Ir jei prireiks atkurti keitinius, tai padaryti bus kur kas paprasčiau dėl duomenų rinkinio dydžio [11].

1.1.4. Nulinio išjungimo laiko perkėlimas

Tai nėra pats svarbiausias metodas, palyginti su aprašytais anksčiau. Tai antros pakopos metodas, kurį atliekant yra svarbu sumažinti sistemos išjungimo laiką iki labai mažo, artimo nuliui, perkeliant duomenis. Viskas priklauso nuo to, kokioje sistemoje bus perkeliami duomenys. Jei tai internetinė (angl. *Web-based*) arba kitokia programa, kuri turi veikti 24 valandas per parą, 7 dienas per savaitę, tokiu atveju nulinio išjungimo laiko metodas yra būtinas. Nulinio išjungimo laiko perkėlimas turi būti naudojamas, kai iškyla problemų paleidžiant naują sistemą.

Norint sėkmingai perkelti duomenis reikia atlikti keletą veiksmų [11]:

1. Duomenų perkėlimas turi būti valdomas kaip atskira projekto dalis su savo komanda. Jam turi būti skiriamas atskiras biudžetas ir ištekliai.
2. Duomenų perkėlimo komandai reikia atrasti visus duomenų šaltinius ir atlikti analizę, kaip efektyviai perkelti duomenis tarp sistemų. Tada komanda galės tiksliai numatyti, kiek reikės pinigų, išteklių ir t.t.
3. Duomenų perkėlimo komandai negalima paskirti kitų darbų, nesusijusių su duomenų perkėlimu, nebent visi darbai atlikti daug greičiau nei buvo numatyta.
4. Būtina naudoti įvairius įrankius, padedančius geriau suprasti duomenis, nustatyti ryšius tarp duomenų, juos transformuoti ir galiausiai perkelti. Duomenų transformacijos įrankius reikia naudoti, jei kiltų problemų perkeliant duomenis, kad galima būtų viską nesunkiai atšaukti.

2. Duomenų perkėlimo etapai ir problemos

Duomenų perkėlimo sritis nėra ta, kuri atrodytų visiems įdomi vien jau dėl to, kad dažnai ten kylančios problemos sudaro situacijas, dėl kurių projektas patiria nesėkmes, pereikvoja biudžetą ar laiką. Dėl tokių problemų, ir jų sprendimo dažnai projektas žlunga. Todėl tik geras vadovavimas ir tinkamas situacijos valdymas gali padėti išgyventi tokias krizes. Tačiau, kaip ir kiekvienoje srityje, jei iš anksto žinotum apie kitų patirtas problemas ir kaip jos buvo išspręstos, yra didelė tikimybė, kad kitų žmonių patirtis padėtų lengviau įveikti sudėtingas situacijas. Todėl, turėdamas tikslą sudaryti patikrintos praktikos patarimų rinkinį, Philipas Howardas inicijavo duomenų perkėlimo projektų tyrimą.

Šio tyrimo pagrindas buvo įvertinti duomenų perkėlimo rinką ir aptikti dažniausiai pasitaikančias problemas, su kuriomis susidūrė keletas įmonių, ir parengti patikrintos praktikos vadovą.

Tyrimo dalyvavo daugiau nei 700 kompanijų iš „Forbes Global 2000“ sąrašo, parengto žurnalo „Forbes“. Buvo įtrauktos įmonės, kurios vykdė projektus, kurių kiekvieno vertė – daugiau nei vienas milijonas dolerių. Vis dėlto, iš visų įmonių esančių tame sąrašė, detalius atsakymus į klausimus pateikė tik 43 organizacijos (nors jų gerokai daugiau abstrakčiai apibūdino projektus). Greičiausiai respondentų skaičius buvo ganėtinai mažas todėl, kad žmonės nelabai mėgsta kalbėti apie savo nesėkmes. Ir atliktas tyrimas rodo, kad dauguma tokių informacinių technologijų projektų nepasiekia savo užsibrėžtų tikslų - pabaigti projektą laiku ir nepereikvojant planuoto biudžeto. Reiktų pasidžiaugti, kad atsirado protingų respondentų, kurie nebijojo pasidalinti patirtomis problemomis ir išlaidomis. Nepaisant to, kad jų vykdyti projektai užtruko tris kartus ilgiau nei buvo planuota iki tol ar kad projekto kaina išaugo daugiau nei dvigubai.

Nors ir šios imties dydis buvo pakankamai nedidelis, tačiau galima teigti, jog pavyko pateikti realius duomenų perkėlimo rinkoje kylančių problemų aprašus ir įvertinimus. Visgi mažas imties dydis reiškia, kad reikėjo normalizuoti gautus rezultatus taip, kad potencialios išskirtys neturėtų per daug įtakos galutiniams rezultatams. Nors tai yra praktikoje taikomas ir teisingas duomenų apdorojimo būdas, tačiau jis gali lemti situaciją, kada kai kurie rodikliai yra nepakankamai įvertinami. Pavyzdžiui, trys bendrovės vykdė projektus, kurių kiekvieno vertė buvo daugiau nei 100 milijonų dolerių, tačiau nė vienas iš visų kitų likusių projektų neturėjo didesnio nei dvidešimties milijonų dolerių biudžeto. Sunku pasakyti, ar tie labai dideli projektai yra tipiniai ar ne, bet tenka pripažinti juos kaip mažumą. Todėl reikia laikytis vieningos nuomonės, per daug nesureikšminant gautos aibės išimčių. Analizuojant duomenis, normalizavimas buvo taikomas tik ten, kur tai atrodė reikalinga, todėl labai tikėtina, kad gauti rezultatai bus ganėtinai konservatyvūs [9].

4.1. Duomenų perkėlimo apibrėžimas

Duomenų perkėlimą galima apibrėžti pagal šio proceso charakteristikas:

- Duomenų perkėlimas yra vienkartinis procesas. Tai nereiškia, kad šis procesas, atliekamas etapais ar sustabdomas su pertraukomis negali užtrukti. Tačiau, kai jis yra atliktas, jau yra pabaigtas. Jis skiriasi nuo daugumos sistemos integrinių užduočių, kurių esmė yra pakartotinis atlikimas, neatsižvelgiant į trukmę.
- Duomenų perkėlimą sudaro savotiškas duomenų restruktūrizavimas. Tam tikrų laukų duomenys yra sujungiami, taip pat kai kuriems laukams gali būti keičiami duomenų formatai arba duomenims taikomos tam tikros duomenų apdorojimo funkcijos. Reikia pabrėžti, jei procese nebūtų duomenų restruktūrizavimo etapo, tai būtų vadinama tiesiog duomenų judėjimu.
- Duomenų perkėlimas turi vykdomų operacijų nuoseklumo darną. Todėl duomenų perkėlimą reikėtų laikyti analitiniu procesu, kadangi duomenų perkėlimas niekada nėra tiesiog duomenų paėmimas iš vietos „A“ ir padėjimas į vietą „B“.
- Duomenų perkėlimas gali, o kartais ir negali daryti sistemos konteksto pakeitimų. Pavyzdžiui, jei duomenų perkėlimas reikalingas tik dėl pasikeitusios informacinės sistemos versijos, tada kontekstas išlieka toks pats. Tačiau jei, tarkim, yra kuriama atskira, nauja informacinė klientų duomenų informacinė sistema, tada sistemos kontekstas duomenims turės būtinai keistis. Duomenų perkėlimas, atliekantis kartu ir sistemos konteksto pakeitimą, neabejotinai yra sudėtingesnis nei toks, kuris konteksto nekeičia.

Kitais žodžiais tariant, duomenų perkėlimas nereiškia tiesiog paprasto duomenų kopijavimo iš vienos sistemos į kitą. Geras pavyzdys, kad galima būtų suvokti skirtumą tarp paprasto duomenų kopijavimo iš vienos vietos į kitą ir duomenų perkėlimo, būtų kraustymasis gyventi iš Bostono į Niujorką visam laikui, lyginant emigraciją iš Bostono į Paryžių. Pirmasis atvejis būtų tiesiog gyvenamosios vietos pakeitimas, o antrasis variantas būtų: kultūros, aplinkos ir kalbos keitimas. Paprastai tariant, duomenų migravimas nėra tas pats kas paprastas duomenų kopijavimo ir įkėlimo kitur procesas. Realiam pasaulyje duomenų perkėlimas visada yra sudėtinė didesnės informacinės sistemos dalis, tiek atvejais kai tiesiog atnaujinama programinė įranga, tiek kai iš esmės keičiama informacinė sistema, taip pat įvairūs kiti variantai, kaip duomenų valdymo sisteminės dalies pasikeitimai, užsakomųjų paslaugų (*angl.* outsourcing) įtraukimas į sistemą ir t.t.

Sėkmingas duomenų perkėlimas pašalina poreikį papildomai perindeksuoti duomenis, esančius duomenų šaltinyje. Duomenų perkėlimo procese visada yra duomenų korektiškumo patikrinimo ir duomenų valymo funkcijos, kurios užtikrina, kad duomenys, kurie bus perkelti į kitą

sistemą, bus teisingi ir reikiamo formato, tam kad būtų galima juos tinkamai naudoti naujoje informacinėje sistemoje [9].

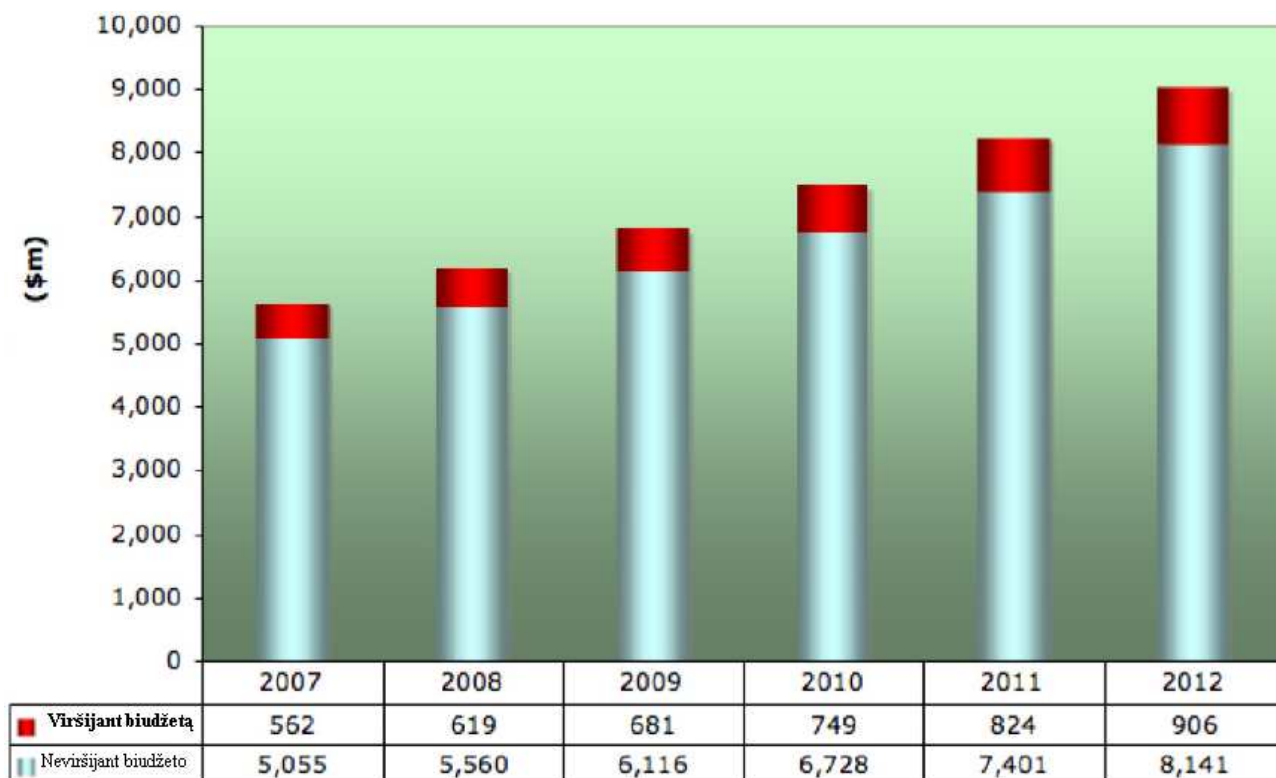
4.2. Esminiai faktai

- Iš visų tyrime dalyvavusių duomenų perkėlimo projektų, kurie laiku ir laikydamiesi suplanuoto biudžeto sėkmingai įvykdė planą, yra tik 16 %.
- 64% visų duomenų perkėlimo projektų buvo užbaigti vėliau nei planuota, ir 37% pereikvojo planuotą biudžetą, t.y. projekto kaina buvo didesnė nei planuota. Šie skaičiai atrodo prieštariniai. Priežastis, kodėl buvo pereikvoti projekto ištekliai, aptarsime vėliau.
- Iš viso duomenų perkėlimui skirto laiko vidutiniškai buvo pereikvojama 40% atvejų.
- Projekto biudžeto pereikvojimas vidutiniškai buvo 30% duomenų perkėlimo projekto biudžeto.
- Daugiau nei pusė respondentų, nurodė kad priežastis, kodėl buvo pereikvotas jų projekto biudžetas, buvo negalėjimas iš anksto tinkamai įvertinti duomenų perkėlimo projekto apimtį.
- Daugiau nei dvi trečiosios respondentų, viršijusių projektui skirtą laiką, labiausiai kaltino negebėjimą tinkamai įvertinti projekto apimtį trukmės atžvilgiu.
- Tik 1 iš 10 kompanijų naudojo duomenų profiliavimo įrankius, kad galėtų suvokti turimus duomenis. Tyrimas parodė, kad egzistuoja glaudi sąsaja tarp šio ir prieš tai pateikto fakto.
- Beveik 20% respondentų neturėjo atskiro biudžeto ar projekto vykdymo trukmės plano, skirto tik duomenų perkėlimui, bet ne visam projektui.
- Duomenų perkėlimo biudžetų paviršius buvo 562 milijonai dolerių per 2007 m., planuojant kilimą iki 906 milijonų dolerių 2012 m. Tyrimo autorius mano, kad tai yra dėl blogo planavimo [9].

4.3. Duomenų perkėlimo rinka

Duomenų perkėlimo rinka yra didelė ir auganti. Bendras duomenų perkėlimo rinkos biudžetas 2007 m. viršijo 5 milijardų dolerių ribą, toliau kylant link 8 milijardų 2012-aisiais. Reikia pabrėžti, kad prognozės yra ganėtinai konservatyvios, kadangi yra nemažai projektų, kurių biudžetas neviršija 1 milijono dolerių, taip pat yra daug kompanijų, neįtrauktų į Forbes Global 2000 sąrašą, kurios vykdo didelius informacinių sistemų projektus, kurių viena iš užduočių, yra duomenų perkėlimas. Naudojant „ilgos uodegos“ (mažesni biudžetai, bet jų yra žymiai daugiau) principą, galima manyti, jog bendros duomenų perkėlimo projektų išlaidos turėtų būti bent dvigubai didesnės

nei pirmasis 5 milijardų skaičius. Labai svarbu yra tai, kad šie skaičiai sudaromi tik iš bendrų informacinių technologijų projektų biudžetų, skirtų duomenų perkėlimui, į kuriuos nėra įtrauktos užsakovo darbuotojų išlaidos, projektų vykdymo laikas.



1 pav. Globalios duomenų perkėlimo rinkos prognozės ir perviršiai 2007-2012 m. (milijonais JAV dolerių)
 Prognozuojama, kad daugiausiai pinigų duomenų perkėlimui išleis bankai, įvairios finansinės institucijos ir komunalinių paslaugų sektorius. Tačiau yra keletas variacijų, priklausomai nuo šalies ir regiono, kuriuos mes aptarsime vėliau.

1 lentelė. Globali duomenų perkėlimo biudžeto prognozė milijonais dolerių, pagal industrijos sektorius, 2007-2012 m.

Industrijos sektorius	2007	2008	2009	2010	2011	2012
Bankai	783	862	948	1043	1147	1262
Finansų sektorius	414	456	502	552	607	668
Komunalinės paslaugos	296	325	358	394	433	476

- Pagal regionus duomenų perkėlimo projektams didžiausios investicijos prognozuojamos Šiaurės Amerikoje ir Vakarų Europoje.
- Jungtinės Amerikos Valstijos, manoma, išlaikys pirmaujančio duomenų perkėlimo projektų investuotojo poziciją. Kadangi JAV yra didžiausias šios srities investuotojas, su 1,7 milijardo dolerių 2007 m. Paskui JAV turėtų eiti Japonija, 2007 m. tam išleidusi 809 milijonus dolerių. Šis skaičius yra daugiau nei pusė viso Azijos Ramiojo vandenyno regiono

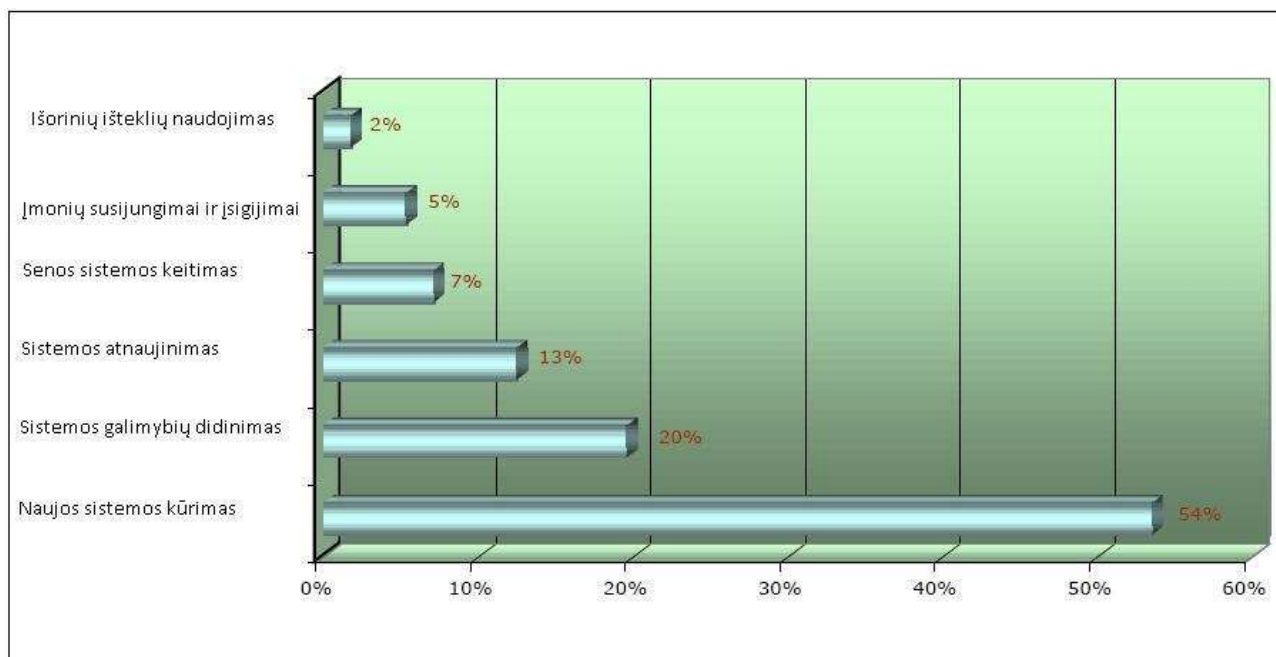
duomenų perkėlimo globalaus biudžeto. Jungtinė Karalystė (316 milijonų dolerių per 2007 m.) yra ir greičiausiai bus didžiausia šios srities rinka Europoje, po kurios eis Prancūzija, Vokietija ir Italija [9].

4.4. Rezultatai ir išvados

Remiantis tyrimo rezultatais galimi padaryti dvi aiškias išvadas. Pirma, duomenų perkėlimo rinka yra labai reikšminga: patvirtintais duomenimis bent 5 milijardai dolerių per 2007 m., tačiau remiantis nepatvirtintais duomenimis ši suma yra dvigubai didesnė. Kadangi duomenų perkėlimo metu už išleidžiamus pinigus yra perkami žmogiškieji ištekliai, paslaugos ir konsultavimas, programinė ir kompiuterinė techninė įranga, yra labai keista, kad tokio dydžio rinka dar nėra pripažinta esanti atskiru verslo sektoriumi. Tyrimas parodė, kad tokios rinkos (su visomis savo savybėmis ir standartais, gera praktika ir t.t.) suformavimas yra labai reikalingas tam, kad taikant atitinkamų disciplinų teoriją praktikoje, galima būtų sumažinti laiko ir finansinių išlaidų perviršį, o gal net išvis jį sunaikinti.

Antra, šio tyrimo apimtis buvo duomenų migravimo projektai, vykdyti įmonių, esančių Forbes Global 2000 sąraše. Daugiau nei 700 kompanijų buvo apklaustos telefonu. Tyrimo ataskaitoje buvo pateikti preliminarūs teiginiai, pagrįsti detalių diskusijų su 43 respondentais. Visi apklausti respondentai iš dalies ar visiškai buvo atsakingi už duomenų perkėlimo projektą, vykdytą jų kompanijose. Atlikto tyrimo tikslas buvo susidaryti vaizdą apie duomenų perkėlimo rinką ir atrasti esminius iššūkius patiriamus šios srities specialistų.

Atlikta studija nagrinėjo duomenų perkėlimo projektus, didesnių informacinių technologijų sistemų projektų kontekste (žr. 2 pav.), kur bendras biudžetas buvo nemažesnis nei 1 milijonas dolerių (kai kurie skaičiavo netgi šimtus milijonų). Nagrinėjami projektai daugiausia buvo vykdomi Šiaurės Amerikoje ir Europoje [9].



2 pav. Informacinių sistemų projektų tipai

Esminiai rinkos rodikliai yra pateikti 2 lentelėje.

2 lentelė. Esminiai rinkos rodikliai

Vidurkis per metus idiegiamų informacinių sistemų	Vidutinė viso projekto biudžeto dalis, skirta duomenų perkėlimui (mln. dolerių)	Procentas duomenų perkėlimo projektų, patiriančių biudžeto perviršį	Vidutinis duomenų perkėlimo projekto biudžeto perviršio kiekis
4.48	0.56	37%	30%

Išanalizavus ir sujungus rinkos pasidalijimo ir geografinio pasiskirstymo informaciją, pateiktą 2 lentelėje, buvo sukurtas didžiausių pasaulio kompanijų duomenų perkėlimo rinkos dydžio ir prognozių modelis.

Šis modelis buvo panaudotas sugeneruoti duomenų perkėlimo rinkos Šiaurės Amerikoje ir Vakarų Europoje biudžeto prognozėms. Tyrimo duomenys taip pat panaudoti modeliuojant Azijos ir Ramiojo vandenyno (APAC), Rytų Europos ir likusio pasaulio biudžetų prognozes.

Tyrimo ataskaitoje dviem tūkstančiams globalių kompanijų buvo pateiktos biudžetų prognozės ir vertinimai, visą piniginę išraišką pateikiant JAV doleriais. Pateikti skaičiai nurodo biudžete numatomą pinigų kiekį, skirtą duomenų perkėlimui projektuose, kurių bendras biudžetas yra ne mažesnis kaip 1 mln. dolerių. Nereikia nė sakyti, kad yra daug mažesnių duomenų perkėlimo projektų, vykdomų mažesnėse organizacijose. Todėl bendra duomenų perkėlimo rinka yra gerokai didesnė nei šiame tyrime nurodyti skaičiai.

Prognozių vertinimas pateiktas penkiems skirtingiems geografiniams regionams: Azija ir Ramiojo Vandenyno regionas, Vakarų Europa, Rytų Europa, Šiaurės Amerika ir likusio pasaulio šalys. Kiekvieno regiono nagrinėjami pavyzdžiai buvo kelios šalys: Azijos (Japonija, Kinija/Honkongas, Pietų Korėja), Vakarų Europos (Jungtinė Karalystė, Vokietija, Prancūzija) ir Šiaurės Amerikos (JAV, Kanada). Tyrime pateiktas tolesnis industrinių sektorių išskaidymas į 27 atskirus sektorius pagal geografinį regioną.

Prognozės buvo apskaičiuotos taikant CAGR konstantą. Savaimė suprantama, kad realybėje šios prognozės bus skirtingos tarp atskirų programinės įrangos gamintojų, tokių kaip SAP, IBM, Oracle ir t.t. Nebuvo reikalo modeliuoti šiuos pasiskirstymus tiksliai, bet yra manoma, kad per visą prognozės laikotarpį skirtumai išsilygins. Apskritai buvo paimta 10% CAGR rodiklio reikšmė visoms šio tyrimo prognozėms ir vertinimams. Ši vertė paimta iš susijusio tyrimo, nagrinėjančio verslo informacinių sistemų rinką.

Papildomai naudojami anksčiau minėti duomenų perkėlimo rinkos esminiai rodikliai, kad galima būtų parodyti, kiek pinigų vidutiniškai yra bereikalingai išleidžiama, kai pereikvojama projekto trukmė ar biudžetas.

Dažniausiai yra manoma, kad duomenų perkėlimo projektai yra skirstomi tik kiekybiškai, imant prielaidą, kad visi jie bus įvykdomi nepriekaištingai ir nepatiriant didelių problemų. Tačiau šis tyrimas parodė realią situaciją, kuri yra kiek kitokia nei tikėtasi ir tai atvaizduojantys duomenys pateikiami sekančiame skirsnyje [9].

Projekto tikslų įgyvendinimas

Tyrimo pradžioje buvo apklausti visi respondentai, apie jų patirtį vykdant duomenų perkėlimo projektus. Tik 16% iš vykdytų projektų nepatyrė laiko ar projekto biudžeto pereikvojimo. 19 % projektų neturėjo atskiro projekto vykdymo tvarkaraščio ir atskiro duomenų perkėlimui skirto biudžeto, todėl jie negali būti įtraukti į rezultatų aibę. Iš likusiųjų:

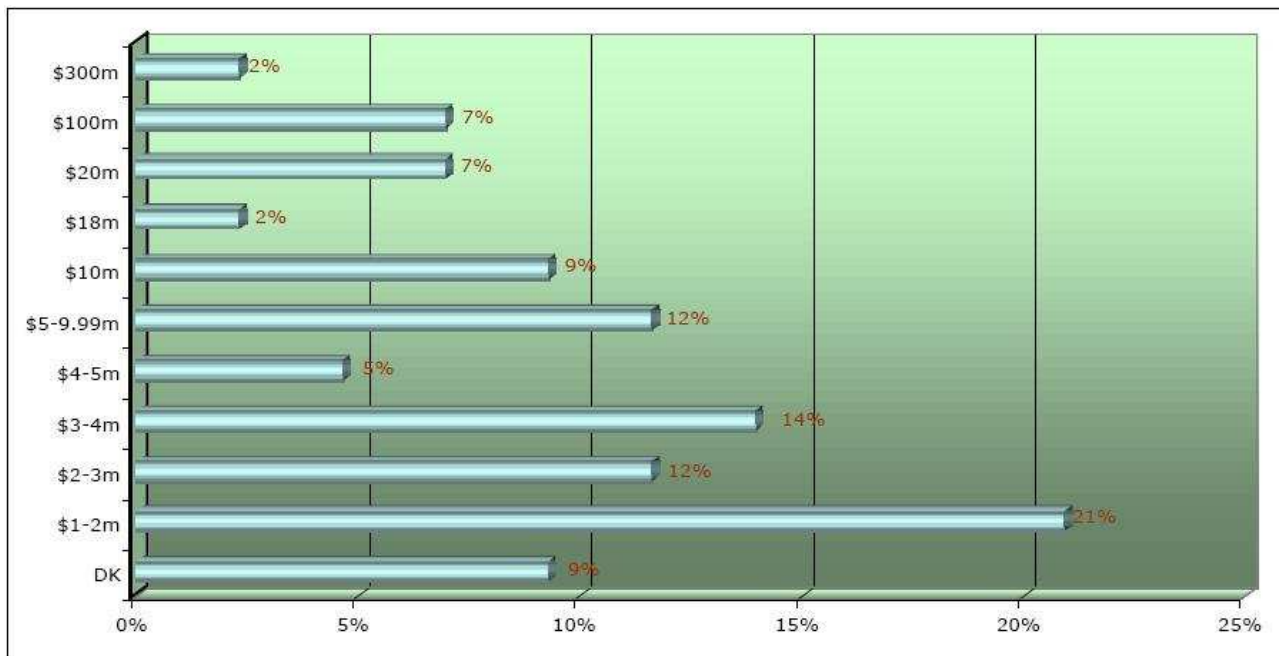
- 3 % pereikvojo biudžetą, bet įgyvendino projektą laiku;
- 46 % pereikvojo projektui skirtą laiką, bet ne biudžetą;
- 51 % pereikvojo tiek projekto biudžetą, tiek jam skirtą laiką.

Įdomu, kaip galima pereikvoti laiką, nepereikvojant biudžeto. Yra keletas variantų, dėl ko taip gali nutikti. Pati įtikinamiausia to priežastis yra projekto resursų išskaidymas ir panaudojimas kitose srityse, kurios viso projekto atžvilgiu tuo metu pasirodė esą svarbesnės. Kita priežastimi galėjo būti resursų perskirstymas, kai projektas turi tiksliai suplanuotą laikotarpį, kada jis turėtų būti pabaigtas. Jei projektas laiku neįvykdytas, spaudimas bandant pasivyti numatytą projekto tvarkaraštį, dingsta ir vėl resursai gali būti perskirstomi kitur. Tačiau tokiu atveju labiau tikėtinas projekto paskelbimas žlugusiu, negu projektui skirto laiko pereikvojimas. Nebent atsiranda objektyvių priežasčių pabaigti pradėtus darbus, viską iš naujo suderinus su klientu. Taip pat svarbu pastebėti, kad pereikvoti

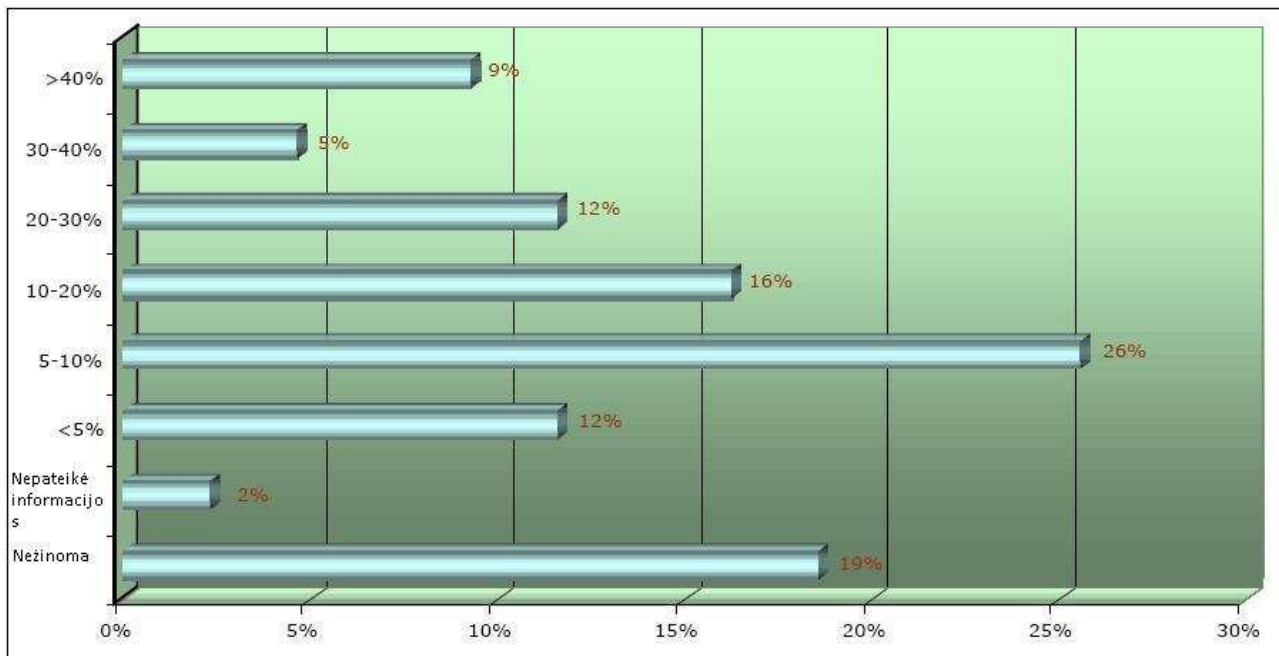
projektui skirtą laiką yra lengviau, nepereikvojant biudžeto, nei kad pereinoti biudžetą, neišnaudojant projektui skirto laiko [9].

Biudžetai ir projektų laiko planai

Tyrimo metu respondentų buvo teiraujamasi apie viso projekto biudžeto dalį, skiriamą išskirtinai duomenų perkėlimui. Gauti atsakymai pateikti 3 ir 4 paveiksluose.

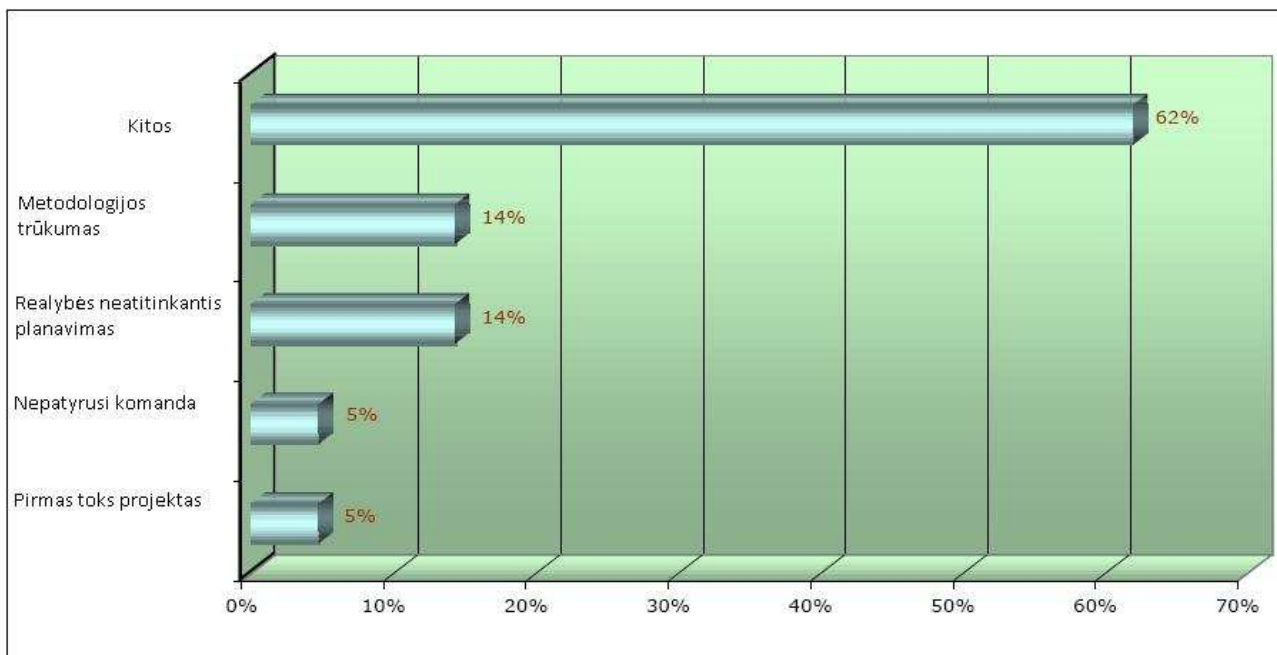


3 pav. Projektų biudžetai



4 pav. Duomenų perkėlimui skiriama biudžeto dalis

Taip pat buvo pateikta informacija apie biudžeto pereinimo priežastis. Gauti rezultatai atvaizduoti 5 paveiksle.



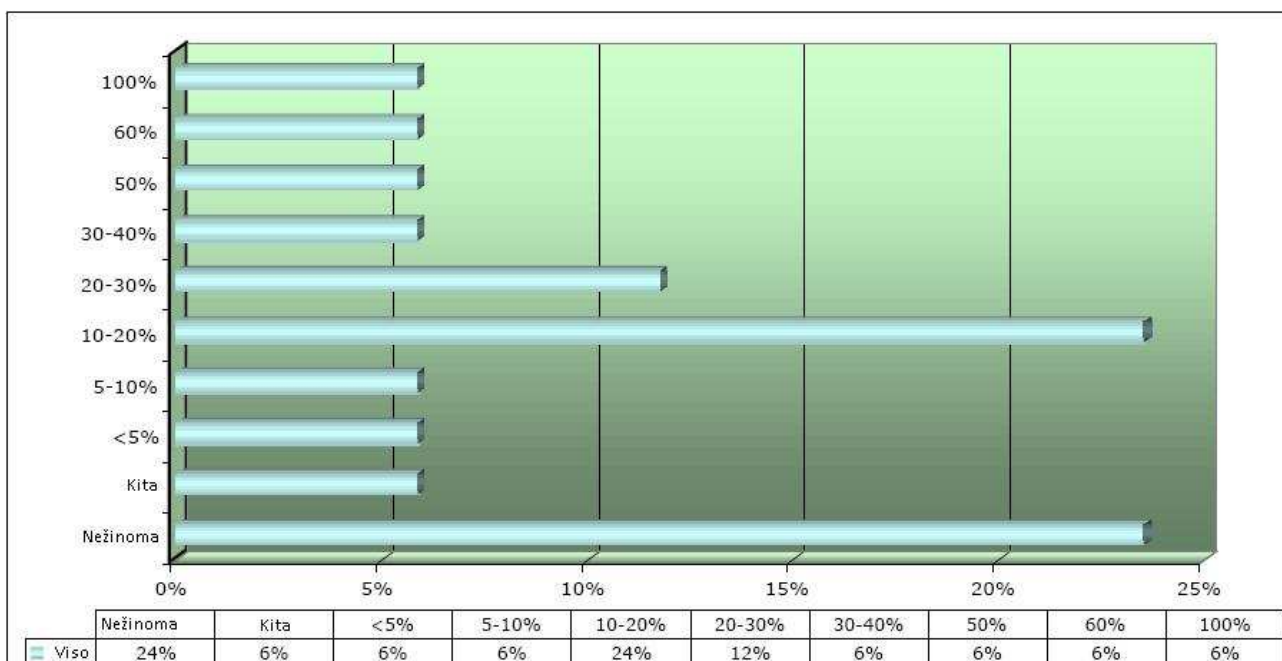
5 pav. Biudžeto pereinimo priežastys

Atsakiusieji „kita“, atsakymą patikslino, pateikdami tokias priežastis:

- 46 % problemos priežastimi nurodė pereinimą projektui skirtą laiką;
- kiti 38 % respondentų pateikė problemas, kurias galima apibendrinti, kadangi jų priežastis buvo blogas projekto apimtį vertinimas;
- keletas kompanijų paminėjo turėję problemų, kadangi neturėjo tinkamos programinės įrangos ar nežinojo, kaip ja naudotis.

Prie glaudžiai susijusių biudžeto ir projekto laiko pereinimo problemų prisidėjo labai įdomi projekto užduočių sudėtingumo vertinimo problema. Kadangi tik 14 % buvo pasirengę pagrindine problema pripažinti realybės neatitinkantį sudėtingumo vertinimą, komentarai tų, kurie pasirinko atsakymą „kita“, buvo: „neaiškūs veikimo reikalavimai“, „sudėtingumas nebuvo aiškiai aprašytas“ ir „projekto kompleksiskumas pradėjo aiškėti tik projektui prasidėjus“. Visos šios priežastys galėtų būti apibendrintos kaip prastas projekto reikalavimų ir sudėtingumo įvertinimas. Sujungus visas šias priežastis, galima teigti, kad daugiau nei pusė respondentų kaltino prastą, neatitinkantį reikalavimų ar tolumą nuo realybės projekto sudėtingumo įvertinimą, dėl ko buvo pereinamas biudžetas.

Pabaigoje respondentai buvo paklausti, kiek daug buvo viršytas biudžeto planas. Gauti rezultatai pateikti 6 paveiksle.

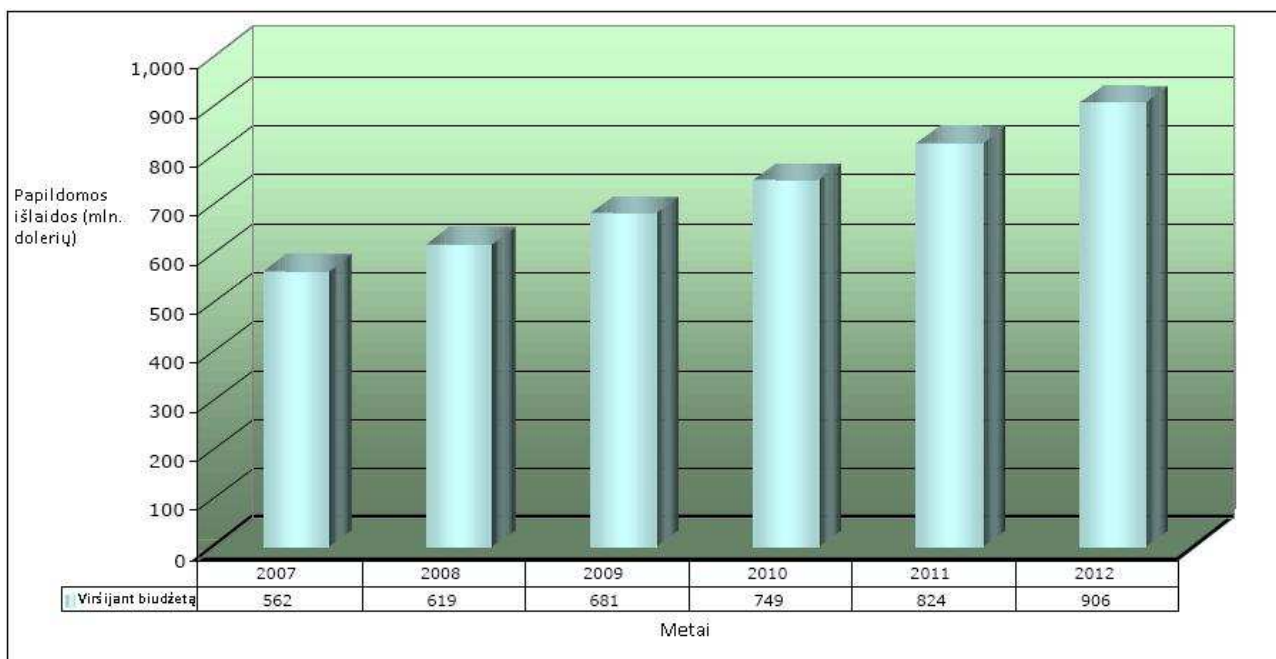


6 pav. Biudžeto pereikvojimas procentais

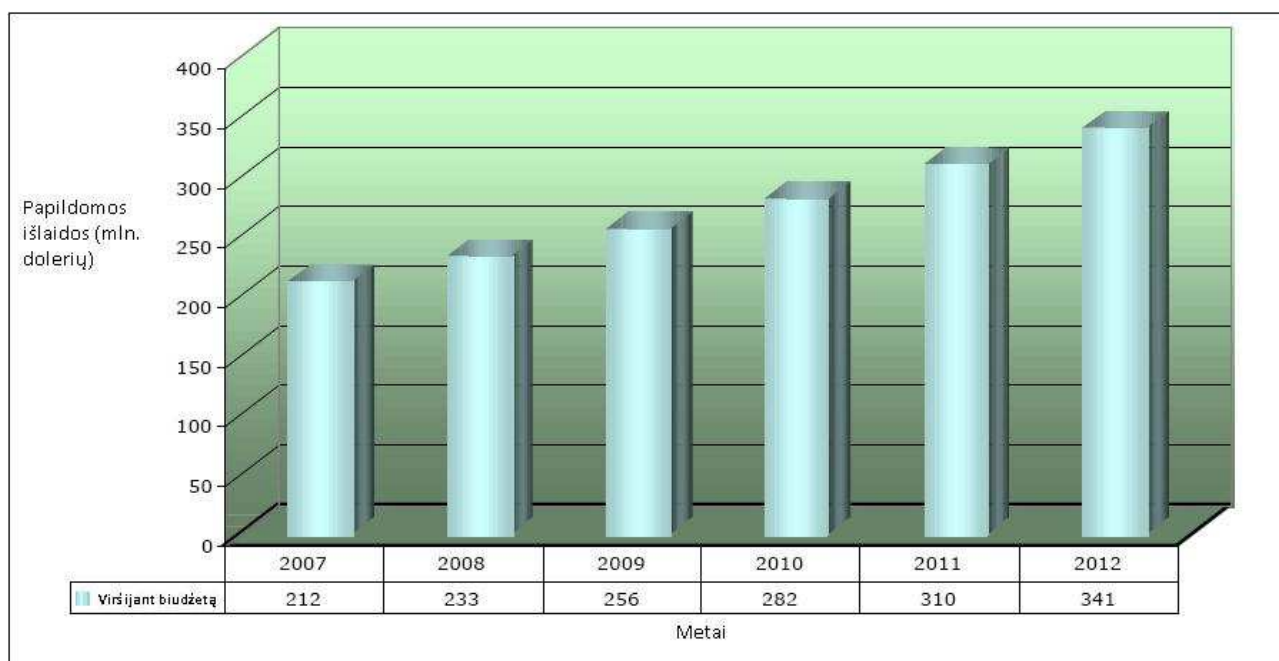
Normalizavus anksčiau pateiktą informaciją, buvo apskaičiuota, kad vidutiniškai kiekvieną kartą biudžetas buvo viršijamas 30,42 % visose kompanijose, kurios pripažino viršijusios biudžetą.

Apskritai 37% duomenų perkėlimo projektų patyrė biudžeto pereikvojimą ir vidutinis toks pereikvojimas yra daugiau nei 30 % viso duomenų perkėlimo projekto biudžeto. Jei gautus rezultatus panaudotume skaičiavimuose apie bendrą duomenų perkėlimo rinkos dydį, aprašytą ankstesniame skyriuje, galėtume pateikti vidutines šioje srityje globaliai patirtas papildomas išlaidas, kurios būtų vertinamos 562 mln. dolerių 2007 m., kylant iki 906 mln. 2012 m. (įtraukiant tik projektus iš Forbes Global 2000 sąrašo, kurių vertė didesnė nei 1 mln. dolerių). Tai parodo įmonių finansinių išteklių švaistymą, todėl jeigu pavyktų šias išlaidas sumažinti tinkamai planuojant didelio masto duomenų perkėlimo projektus, įmonėms tai būtų labai naudinga.

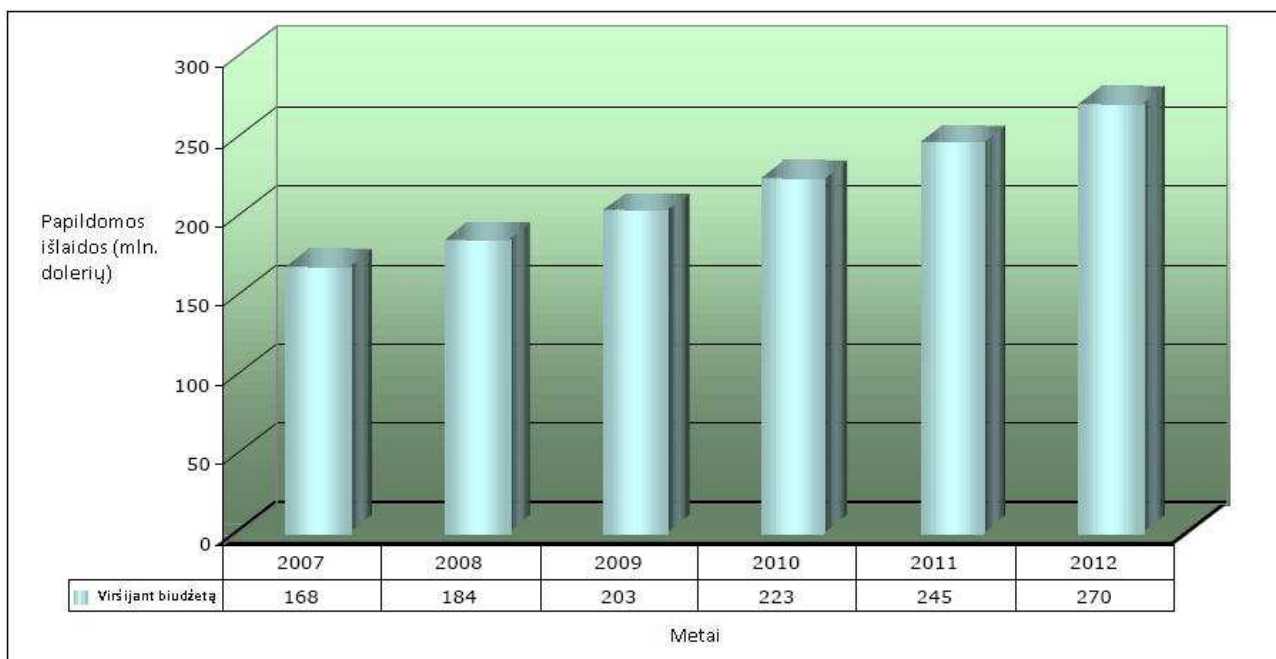
Žemiau pateikti grafikai, kiekybės atžvilgiu parodantys biudžeto pereikvojimo pasekmes globaliai ir pagal didžiuosius geografinius regionus [9].



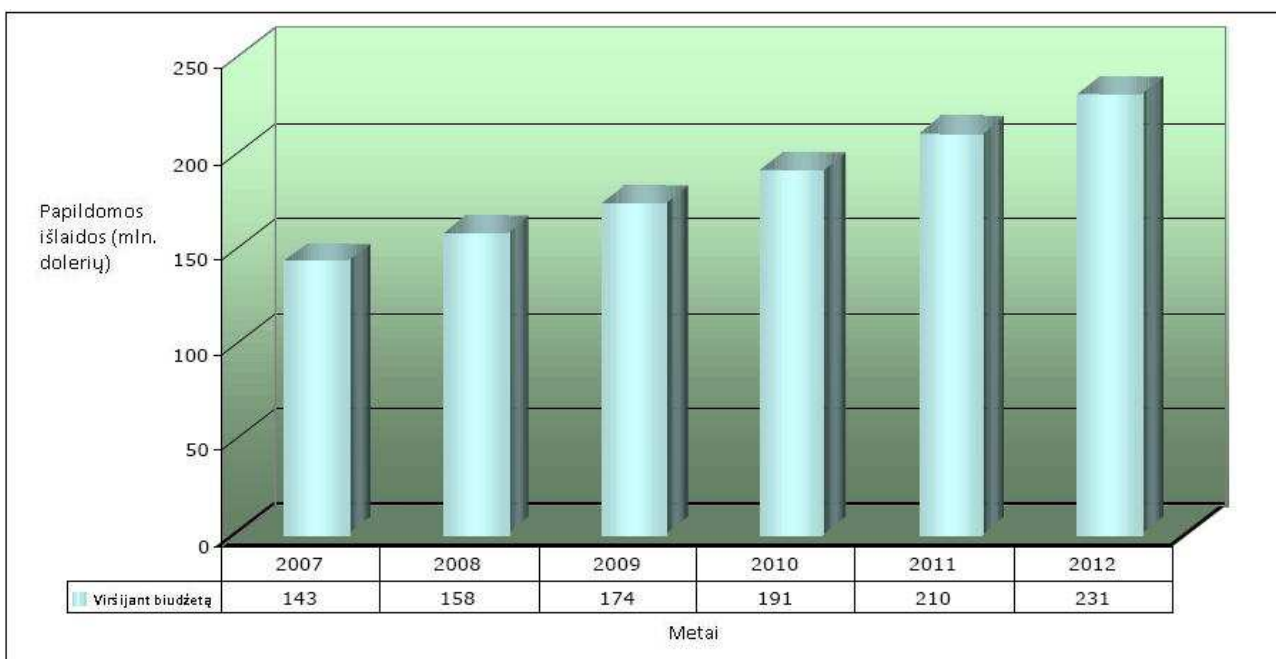
7 pav. Prognozuojamos globalios išlaidos, patiriamos pereinant duomenų perkėlimo projektų biudžetus



8 pav. Prognozuojamos išlaidos Šiaurės Amerikos regione, patiriamos pereinant duomenų perkėlimo projektų biudžetus



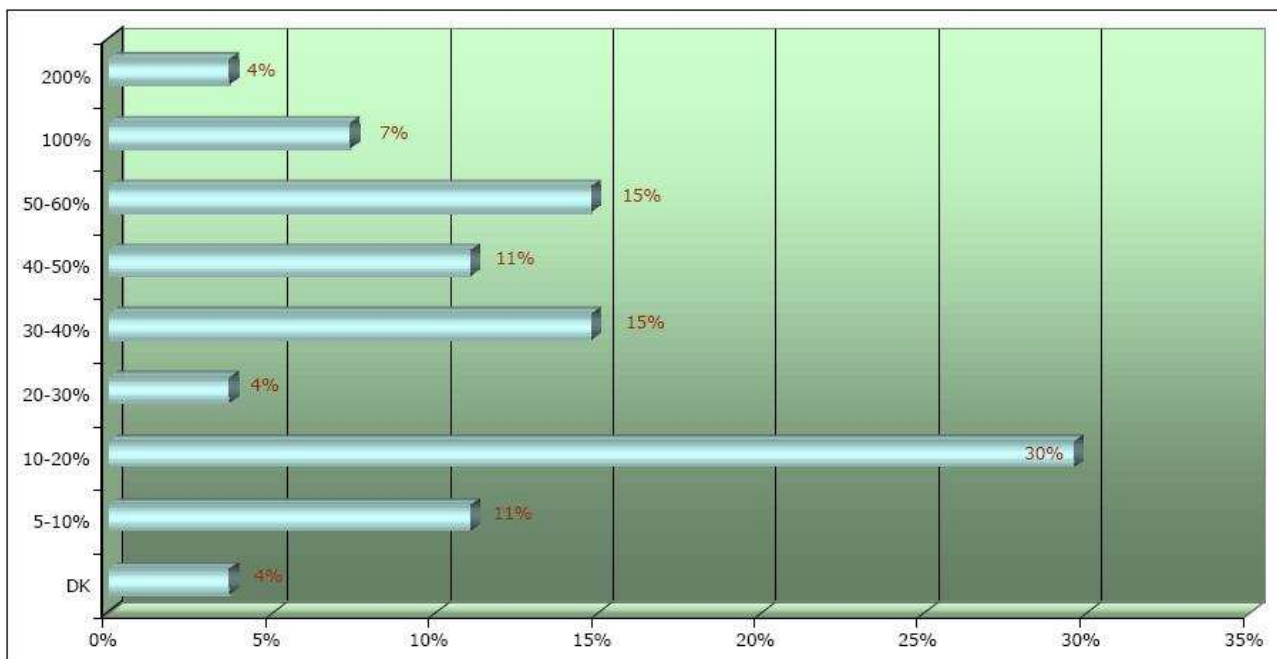
9 pav. Prognozuojamos išlaidos Azijos ir Ramiojo vandenyno regione, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus



10 pav. Prognozuojamos išlaidos Vakarų Europoje, patiriamos pereikvojant duomenų perkėlimo projektų biudžetus

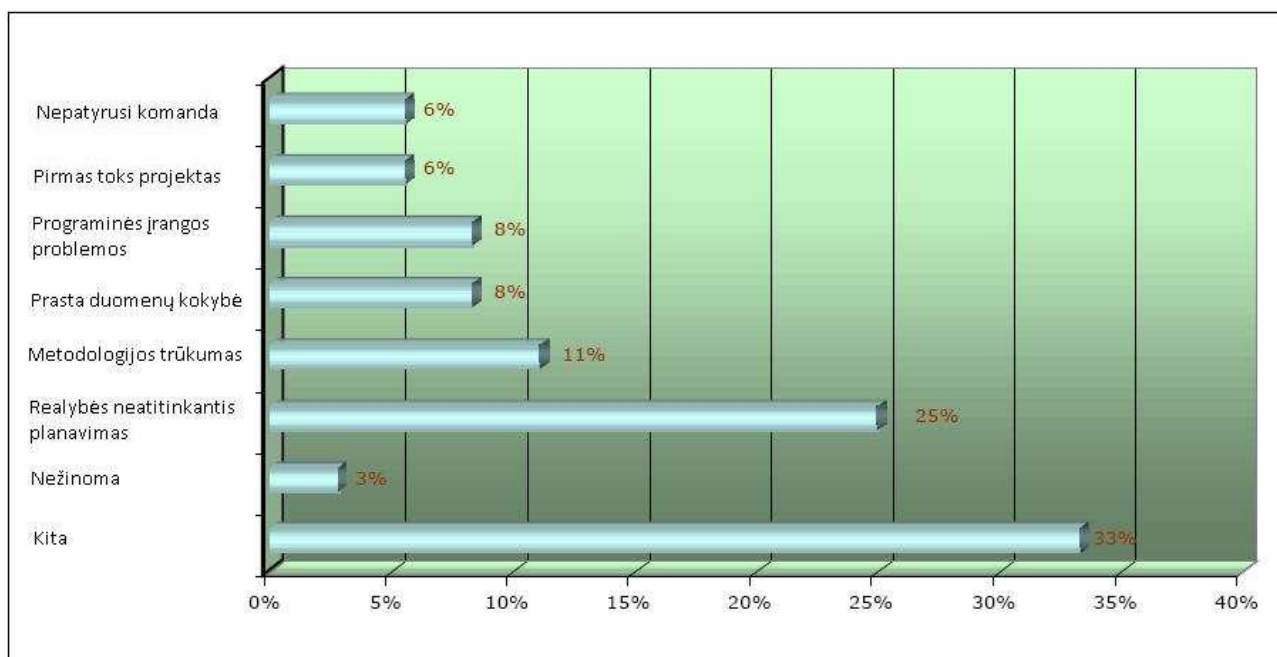
Projekto laiko panaudojimo planavimas

Tik 33% procentai respondentų pranešė, kad jų projektai buvo įvykdyti laiku. Kas nebuvo faktas, kurį buvo planuojama nagrinėti tyrimo metu. Tyrimo tikslas buvo nagrinėti projekto laiko viršijimo mastą, viršijimo priežastis ir to sukeliamas problemas, veiklos rezultatai yra pateikti 11 ir 12 paveiksluose.



11 pav. Perviršio mastai

Po normalizavimo, vidutinis procentinis pereikvojimas buvo 40.87 %.



12 pav. Perviršio priežastys

Tų 42% respondentų, kurie pasirinko atsakymo variantą „Kita“, pateiktos problemos priežastys buvo daugiau mažiau susiję su projekto sudėtingumo vertinimu, sudarant bendrą 67% respondentų dalį. Vienas respondentas kaltino įrankių trūkumą [9].

4.5. Tolesnės pasekmės

Prieš pradedant svarstyti, kas buvo negerai, būtų naudinga įvertinti anksčiau aptartų problemų gilesnį poveikį. Kaip jau anksčiau buvo minėta, duomenų perkėlimas visada vykdomas didesnio projekto kontekste, kuris savo ruožtu užima vietą dar platesniame verslo srities kontekste. Todėl kyla sekantis klausimas: kokį poveikį turi duomenų perkėlimo projekto laiko poreikvojimas taikomajam projektui, ar netgi visam verslo objekto kontekstui.

Yra aišku, kad „jeigu duomenų perkėlimo projektas sužlugs, ar techniškai pavyks, tačiau bus netinkamas naudoti vartotojams, tada taikomasis projektas, kurio dalimi yra duomenų perkėlimas, žlugs finansiškai pakirsdamas verslo objektą, dėl veltui panaudotų pastangų ir kuriamai sistemai puoselėtų vilčių. Gali atsirasti išlaidų ir dėl neišpildytų kliento reikalavimų.

Respondentai nebuvo apklausti (greičiausiai jie ir nebūtų galėję atsakyti), kaip problemos (laiko poreikvojimas ir biudžeto finansinių galimybių viršijimas) su duomenų perkėlimu paveikė „tėvinį“ projektą. Visgi, duomenų perkėlimo problemos būtų tik „simptomai“ visam projektui, neminint poveikio verslo objekto kontekstui. Šiuo atžvilgiu yra svarbu atspindėti tai, kad „giminiškai“ smulkūs projektai gali turėti didelę įtaką verslui. Pavyzdžiui, viena kompanija, kuri išleido daug mėnesių tik duomenų perkėlimo projektui, pačiam duomenų perkėlimui turėjo tik vieną savaitgalį. Tačiau viskas susiklostė nesėkmingai, ir nauja programos versija po savaitgalio neveikė. Buvo nuspręsta skubiai ir efektyviai išspręsti kilusias problemas ir bandyti duomenų perkėlimą dar kartą, kai tik tai bus įmanoma. Tam prireikė dviejų savaitžių, ne toks ir didelis projekto laiko ir kaštų poreikvojimas, ypač žiūrint viso projekto atžvilgiu. Tačiau sistema, kurios duomenys buvo perkeliama, buvo finansų valdymo sistema, todėl dvi savaites buvo neįmanoma išrašyti bet kokią sąskaitą! Kitais žodžiais tariant, projekto atžvilgiu nedidelis perviršis gali būti „naikinant“ kitų atžvilgiu [9].

4.6. Problemų priežastys

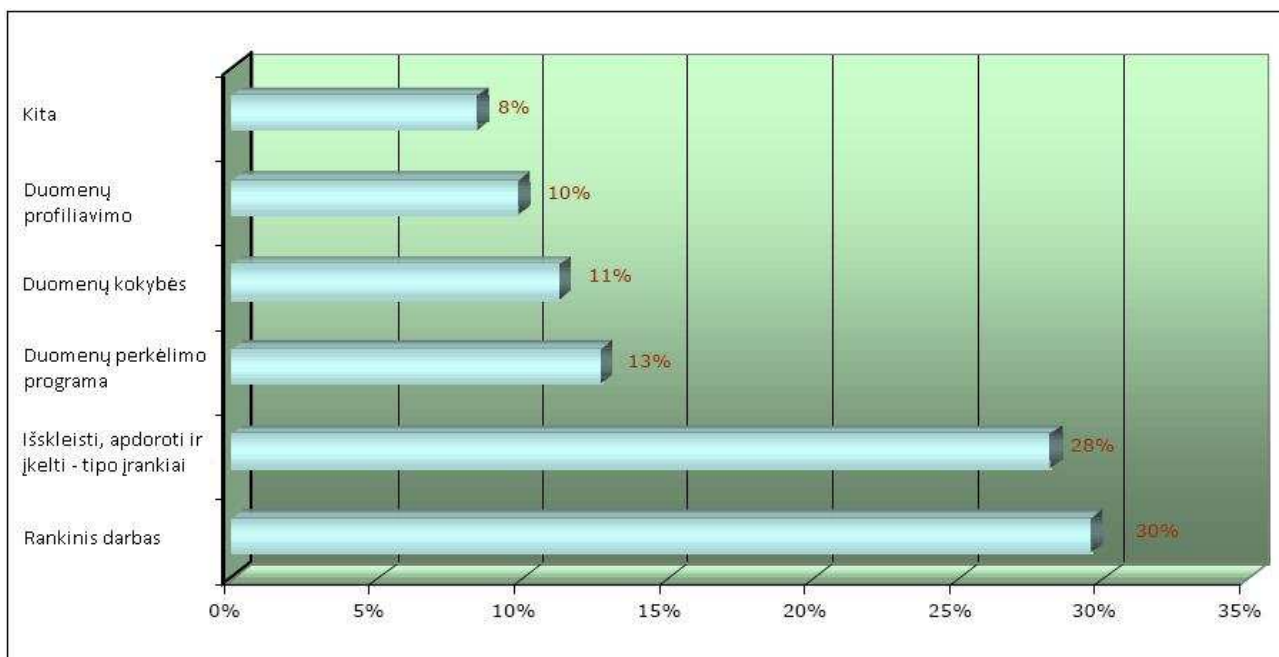
Buvo išnagrinėti visi atsakymai į klausimus, susijusius su biudžetu ir laiko planavimu, tiksliau nagrinėjant klausimus, susijusius su sudėtingumo vertinimu, metodologija, projekto valdymu, įrankiu pritaikymu ir naudojimu, bei ką reiktų pakeisti, jei tas pats projektas būtų vykdomas dar kartą. Kitaip tariant, buvo bandoma išgauti visą informaciją, kuri galėtų padėti surasti proceso dalis, kuriose kyla problemos. Kai kurie atsakymai buvo ganėtinai informatyvūs, kai kurie - mažiau. Visgi iš gautų rezultatų atrinkti įdomūs faktai:

- Beveik trys ketvirtadaliai respondentų turėjo formalią dokumentuotą metodologiją, kurią jie taikė duomenų perkėlimo projekto užduočių vykdyme. Daugiau nei trys ketvirtadaliai jų buvo sukurti organizacijos viduje. Pagal biudžeto poreikvojimo ir laiko perviršio santykį,

tarp tyrime dalyvavusių respondentų, galima manyti, kad šios metodologijos nebuvo tokios geros, arba nebuvo tvirtai jų laikomasi. Respondentų paklausus, ar juos sudomintų bendrų tikslų geros praktikos metodologijos taikymas jų organizacijoje, vėlgi daugiau nei trys ketvirtadaliai respondentų pareiškė norą pabandyti, juos domintų galimybė pabandyti arba labai susidomėjo tokiu dalyku. Grubiai tariant, apklausos dalyviai po lygiai pasiskirstė pasirenkant galimybę tokią praktiką priimti kaip konsultacinę paslaugą, dalinantis informacija bendravimo paskirties tinklalapyje arba dalinantis dokumentacija.

- Nepaisant atsakymų rezultatų, leidžiančių manyti, kad sudėtingumo vertinimas buvo problema, daugiau nei 85% respondentų manė, kad jų projekto sudėtingumo vertinimo vykdymas buvo sėkmingas, labai sėkmingas ar net ypatingai geras. Tai gali būti dėl to, kad respondentai siauriau žiūri, vertindami projekto sudėtingumą, arba tai yra tiesiog dėl vertinimo galvojant, kad viskas pasiseks. Šį klausimą papildė klausimas apie tai, kaip galima būtų panaudoti sudėtingumo vertinimo ir planavimo išteklius, ir kaip respondentai juos taikytų ateities projektuose. Daugiau nei trečdalis norėtų sukurti tokius išteklius organizacijos viduje, o maždaug pusė visų respondentų svarstė galimybę naudoti esamus vidinius išteklius. Ir 22% susidomėjo galimybe tam panaudoti išorinius resursus.
- Kai tyrimo dalyviai buvo paklausti, ką darytų kitaip kitąkart, buvo pateikta nemaža aibė atsakymų, tarp kurių dažniausiai pasitaikė specialių programinių įrankių naudojimas, metodologijų taikymas, išorinių resursų panaudojimas ir duomenų analizavimas ir jų korektiškumo tikrinimas. Beveik 25% anksčiau paminėtus dalykus, o kita panašaus dydžio dalis pripažino duomenų analizavimo ir korektiškumo tikrinimo, profiliavimo ir sudėtingumo vertinimo poreikį.
- Detalės apie projekto valdymui naudojamus įrankius pateiktos tolesniame tekste. Tačiau respondentams uždavus klausimą apie tai, kurias projekto puses būtų galima patobulinti, gerinant projekto valdymą, pagrindiniai atsakymai (apie 20% kiekvienas) buvo patobulinti bendradarbiavimą duomenų perkėlimo projekto viduje ir pagerinti bendradarbiavimą tarp duomenų perkėlimo ir bendrai viso projekto. Toliau apklausama, ar sudėtingas buvo toks bendradarbiavimas praktikoje. Bendradarbiavimas duomenų perkėlimo projekte, pagal 43 % respondentų nebuvo komplikotas ar nebuvo labai komplikotas.

Naudojamų programinių įrankių tipai pateikti 13 paveiksle.



13 pav. Naudojamų įrankių tipai

Kaip galima pastebėti, rankinis programų kūrimas vis dar yra plačiai naudojamas. Ypač galima pastebėti mažą duomenų profiliavimo procentą. Tai leidžia manyti, kad 90 % procentų projektų nenaudoja duomenų profiliavimo. Kaip galima tinkamai įvertinti duomenų projekto sudėtingumą, neprofilijuojant duomenų? Duomenis profilijuojant rankomis viskas visada bus paprasta ir kadangi duomenų profiliavimo įrankiai dar nėra laikoma „panacėja“, manoma, kad tokių produktų naudojimas yra vienas iš esminių dalykų, tam, kad realistiškai įvertinti sudėtingumą, pageidautina, prieš įvertinant projekto apimtį laiko atžvilgiu ir planuojant biudžetą.

Iš visų tyrime surinktų duomenų galima padaryti keletą esminių išvadų:

1. Duomenų perkėlimo rinka yra labai reikšminga ir auganti.
2. Žymi duomenų perkėlimo projektų dalis užsakovui yra pateikiama pavėluotai ar/ir viršijant planuojamą biudžetą.
3. Metodologijų naudojamų duomenų perkėlime dažniausiai nėra pilnai laikomasi.
4. Projekto laiko ir reikalavimų sudėtingumo vertinimas yra didelė problema.
5. Tinkamas programinių įrankių naudojimas ir suvokimas yra dalykas, kuriuo reiktų iš esmės susirūpinti.
6. Bendradarbiavimas tiek duomenų perkėlimo projekto viduje, tiek bendrai kuriamos informacinės sistemos ribose turėtų būti gerinamas.

Pagal egzistuojančias teorijas yra dvi priežastys, kodėl duomenų perkėlimo projektai pereikvoja laiką ar skirtą biudžetą: arba projektas nėra tinkamai įvertinamas užduoties sudėtingumo atžvilgiu, todėl prastai įvertinami projektui būtini laiko ir finansų resursai, arba projekto eigoje iškyla nenumatytų problemų, kurios ir sudaro situacijas, sąlygojančias kylančias problemas. Antroji problemų priežastis gali būti sąlygojama elementarių dalykų, kaip kad netikėtas visos duomenų

perkėlimo komandos susirgimas gripu arba užsakovo reikalavimų esminis pasikeitimas. Visgi dauguma neplanuotų problemų neturėtų būti netikėtomis ir dažniausiai kyla dėl netinkamo ar nepakankamo problemos ir užduoties suvokimo, kitais žodžiais tariant, sudėtingumo įvertinimo.

Tam, kad visiškai suvokti, ko reikia vykdant bet kokių duomenų perkėlimo projekto procesą, pirmiausia reikia žinoti visus duomenų šaltinius, kurie turėtų būti perkeliami, kas dažniausiai nėra paprasta užduotis, kadangi dažnas atvejis kai duomenys yra atskirai talpinami „Excel“ ar „Access“ programų tipų rinkmenose, apie kurias nieko nežino įmonės informacinių technologijų skyrius. Taip pat reikia suprasti tų duomenų prigimtį ir informaciją esančią juose. Dėl šios priežasties atsiranda duomenų profiliavimo ir analizavimo poreikis, analizę ir profiliavimą atliekant dar prieš derinant projekto trukmę, sudėtingumą ir finansinius klausimus. Tačiau visgi tik keletas kompanijų naudoja programinius įrankius, skirtus duomenų profiliavimui ir analizei. Todėl faktas, kad tiek daug projektų pereikvoja suplanuotus laiko ir finansinius resursus, nestebina, tačiau akivaizdu, kad tokios programinės įrangos naudojimas yra būtinas ne tik šioje stadijoje, bet ir viso projekto vykdymo metu, tam kad įsitikinti duomenų korektiškumu, jų transformavimo kokybe bei pačiu perkėlimu.

Galiausiai reiktų paminėti, kad visame „Forbes Global 2000“ sąrašė vidutiniškai yra 4.48 informacinės sistemos sukuriamos per metus. Kitais žodžiais tariant, šis tyrimas nėra tik kažkurio projekto aptarimas. Viename projekte įdiegtos technologijos ir naudojami įrankiai gali būti pritaikomi ir naudojami bet kuriam kitam projektui vykdyti.

Kaip ir bet kurioje srityje, akivaizdu, kad prevencinės priemonės ir jų taikymas yra geriausias būdas pasiekti, kad duomenų perkėlimas vyktų užtikrintai ir sėkmingai, negu reaguoti į kylančias problemas, nesigilinant į jų priežastis. Jeigu duomenų valdymo principai projekte yra įgyvendinti, tada galima manyti, kad duomenų vieta ir prigimtis yra tinkamai išanalizuota ir suvokta ir apdorota informacija yra aukštos kokybės. Kitais žodžiais tariant, grubi duomenų integravimo infrastruktūra yra tinkamoje vietoje. Jeigu to yra laikomasi, problemos, aptartos šiame tyrime, neturėtų atsirasti.

Toliau, sekant vykusias diskusijas ir remiantis tyrimą atlikusių žmonių patirtimi, galima pateikti keletą esminių rekomendacijų, ką reiktų padaryti prieš pradedant vykdyti projektą:

Prevenciniai veiksmai:

- Įgyvendinti duomenų valdymą.
- Turėti duomenų korektiškumo tikrinimo programą, kurią galima būtų panaudoti bet kurioje projekto fazėje.

Jeigu duomenų valdymo nėra kliento sistemoje, tada reiktų atitinkamai įvertinti finansinius ir projekto trukmės poreikius. Reiktų pabrėžti, kad duomenų perkėlimo projektas turi atskirą biudžetą ir laiko tvarkaraštį. Tokioje situacijoje tolesni veiksmai turėtų būti:

- Parinkti ar apibrėžti duomenų perkėlimo metodologiją, kurios bus laikomasi viso projekto metu.
- Įsitikinti, kad bendraujate su visais kliento pusės sričių atstovais, kad žinoti, iš kur visi duomenys ateina.
- Tuo atveju, kai duomenys yra keliuose šaltiniuose, nustatyti, kurį šaltinį reikėtų naudoti.
- Imtis „tuščių vietų“ analizės, kad sužinoti, ar tose vietose turėtų būti duomenys, kurių esamu momentu nėra ir nuspręsti, iš kur juos reikės paimti.
- Profiliuoti ir analizuoti visus duomenų šaltinius (greičiausiai tam tikrų įrankių pagalba) tam, kad suvokti ir surasti visas galimas problemas dėl pačių duomenų, kad tinkamai prognozuoti projektui reikiamo laiko kiekį ir tam reikalingą biudžetą.
- Naudoti įrankius, kurie palaiko projekte taikomą metodologiją.
- Kad ir kokius įrankius pasirinksite naudoti, dar prieš pradedant projektą, įsitinkite, kad jūs esate pilnai kompetentingi jais naudotis ir mokėsite juos naudoti [9].

5. Duomenų perkėlimo procesas

Duomenų perkėlimo procesas yra panašus į kitus informacinių technologijų projektus, kuriuose detalus planavimas yra vykdomas pagal struktūrizuotą metodologiją. Ši sekcija apibūdina duomenų perkėlimo projekto fazes.

- Fazė 0: Keliamų užduočių ir jų įvykdymo galimybių įvertinimas
- Fazė 1: Planavimas ir projektavimas
- Fazė 2: Pasikeitimų valdymas
- Fazė 3: Duomenų perkėlimas
- Fazė 4: Rezultatų analizė ir apžvalga po migravimo

5.1.Fazė 0: Keliamų užduočių ir jų įvykdymo galimybių įvertinimas

Pirmojoje duomenų perkėlimo fazėje yra apibrėžiamas projekto funkcionalumas ir reikalavimai, bei atliekama preliminarinė esamos aplinkos analizė, tam kad nustatyti taikomų standartų atitikimo lygį. Šios fazės metu pagal apibrėžtus reikalavimus yra parenkama duomenų perkėlimo metodologija. Tinkamai praėjus šią fazę, pasiekiami atitinkami rezultatai:

- Saugus, struktūrizuotas duomenų perkėlimo sprendimas, mažinantis riziką.
- Įmanomi įvykdyti ir prognozuojami rezultatai.
- Pagal užsakovo specifinius pageidavimus sumodeliuotas procesas.

Duomenų migravimo technologiją ir metodiką galima pasirinkti iš anksčiau aptartų variantų. Dažniausiai ši pasirinkimą diktuoja užsakovo pateikiami reikalavimai, parenkami pagal kiekvieno projekto specifiką. Tam, kad pasirinkti tinkamą metodą, labai svarbu atlikti preliminarą analizę, surenkant sekančią informaciją:

- Koks perkeliama duomenų kiekis. Reikia surinkti specifinę informaciją, susijusią su duomenų kiekiu, pobūdžiu ir struktūra.
- Informacinės sistemos prieinamumas ir apribojimo galimybės. Dažnai priklausomai pagal tai, kiek laiko gali būti išjungta informacinė sistema, yra parenkamas tinkamas duomenų perkėlimo metodas.
- Naudojamų techninių kompiuterinių sistemų pajėgumas. Kadangi duomenų perkėlimas dirba su didžiuliais duomenų kiekiais, todėl jų apdorojimas, perkėlimas ir analizė gali įtakoti tinklo, serverio ar/ir sistemos darbą. Ši įtaka turi būti tinkamai įvertinta ir į ją reiktų atsižvelgti.
- Darbų tvarkaraščio reikalavimai. Priklausomai nuo esamos, ne tik duomenų perkėlimo, bet viso projekto situacijos, duomenų perkėlimui gali būti skirtas labai trumpas laiko tarpas. Tam tikri įrankiai leidžia duomenų perkėlimą vykdyti per ilgesnį laikotarpį, duomenis perkeliama porcijomis. Tas gali būti naudinga, žinant, jog duomenų perkėlimą bus galima vykdyti daugiau nei kartą per tam tikrą laiko periodą, tokiu būdu atitinkamai paskirstant perkėlimo vykdymą.
- „Atsitraukimo“ planas. Tam, kad sumažinti duomenų perkėlimo žalos riziką užsakovo informacinių sistemų struktūroms, būtinas detalus „atsitraukimo“ darbų planas, sudaromas dar prieš pradėdant vykdyti duomenų perkėlimą [2].

5.2.Fazė 1: Planavimas ir projektavimas

Šios fazės metu yra sudaromas darbų vykdymo planas ir suprojektuojamas sistemos modelis, atitinkantis nulinės fazės metu surinktus reikalavimus. Šios fazės įvykdymas būtinas bet kuriam tinkamai įgyvendintam pakeitimų valdymo procesui.

Šią fazę sudaro užduočių įgyvendinimo plano sudarymas, kuriame yra ne tik specifinės projekto detalės, bet ir ateities reikalavimų analizė, remiantis galimu taikymo sričių augimu. Pasirinkta duomenų perkėlimo metodika turėtų būti tinkamai dokumentuota, tam kad įsitikinti, jog pasirinktas duomenų perkėlimo įgyvendinimo metodas atitinka užsakovo reikalavimus.

Dalis informacijos, kuri turėtų būti užfiksuota ir aprašyta, šios fazės metu:

- Specifiniai duomenų šaltinio ir tikslinės aplinkos duomenys.
- Duomenų saugumo reikalavimai.

- Duomenų perkėlimo būdo specifi­ka (vietinis tinklas, nutolę tinklo klientai ir t.t.).
- Planuojamų perkelti duomenų kiekis ir dydis.

Tam, kad apibrėžti ateities duomenų perkėlimo reikalavimus, labai svarbu apsvarstyti duomenų užimamos vietos augimą, tam kad galima būtų prognozuoti duomenų talpyklų poreikius. Saugumo ir našumo reikalavimai įtakoja tikslinės aplinkos fizinę konfigūraciją ir našumo galimybes.

Žvilgsnis iš vidaus į sudėtingą duomenų perkėlimo prigimtį ir įtaką apima bendrai naudojamus resursus, įskaitant serverio apkrovimą, tinklo pralaidumą ir talpumą, šaltinio bei tikslo duomenų talpyklų našumą ir įtaką informacinei sistemai. Visa tai turi įtakos duomenų kiekiui, kuris gali būti perkeltas iš vienos platformos į kitą per tam tikrą laiko tarpą. Tinkamas sistemos projekto sprendimas leidžia deramai įvertinti projekto išteklių poreikius ir prognozuoti projekto trukmę. Nuo to priklauso projekto sėkmė.

Sėkmės kriterijus yra nusakomas planavimo fazės metu ir suderinamas su užsakovo tarpininkais. Šie kriterijai skirstomi į dvi kategorijas: duomenų integrumo ir našumo. Savaime suprantama, duomenų integrumas yra pirmaeilis bet kuriame duomenų perkėlimo projekte. Projekto vykdytojai ir užsakovo atstovai turi suderinti testavimo planą. Toks planas kartais gali būti tiesiog eilinis kontrolinių sumų patikrinimas ar sėkmingai paleista ir tinkama naudojimui sistema. Visgi sudėtingesnis testavimas yra reikalingas tam, kad užtikrinti, jog visi duomenų rinkinių variantai yra perkelti teisingai. Taip pat duomenų perkėlimo našumo reikalavimai turi būti apibrėžti planavimo fazės metu ir dar kartą patikrinti po duomenų perkėlimo. Duomenų įvedimo/išvedimo parametrų statistika turėtų būti sudaroma bent jau dieną prieš duomenų perkėlimą tam, kad rezultatus galima būtų palyginti su situacija po duomenų perkėlimo.

Galimi įvairūs kiti sėkmingo duomenų perkėlimo patikrinimo būdai. Kuriant sudėtingas informacines sistemas dažniausiai dirba testuotojų komanda, kurie taip pat gali užsiimti ir duomenų perkėlimo funkcionalumo testavimu. Šis funkcionalumas gali būti tikrinamas tiek apkrovimo, našaus duomenų apdorojimo, perkeliama duomenų korektiškumo bei funkcionalumo testais [2]. Kai duomenų perkėlimo projekto priežastis yra technologinis informacinės sistemos procesų atnaujinimas, papildomas dėmesys turėtų būti skiriamas šioms sritims:

- Techniniai informacinės sistemos reikalavimai, apdorojamo duomenų srauto atžvilgiu.
- Techniniai informacinės sistemos reikalavimai, duomenų apdorojimo greičio atžvilgiu.
- Informacinės sistemos reikalavimai serverio operacinės sistemos lygmenyje.

5.3.Fazė 2: Pasikeitimų valdymas

Duomenų perkėlimo projektai turi atitikti ir netgi taikytis prie valdymo proceso pasikeitimų. Toliau pateikiamas trumpas pasikeitimų valdymo komponentų sąrašas:

- **Rizikos valdymas**

Esminis pasikeitimų valdymo proceso komponentas yra pilnavertiškas rizikos įvertinimas. Rizikos įvertinimas yra analizė to, kas gali nutikti blogo, kaip to išvengti ir kaip sumažinti nesėkmingo pasikeitimo poveikį. Tokia analizė turėtų būti dokumentuota, įtraukiant išbandytą testavimo planą.

- **Perkėlimo planas**

Čia apžvelgiamas duomenų perkėlimo įgyvendinimo planas, kuris buvo sukurtas planavimo ir projektavimo fazės metu. Čia apibrėžiama šaltinio ir tikslinė aplinka, taip pat įtraukiant visą procesą, suskirstytą žingsniais.

- **Būtinios sąlygos ir ryšiai**

Bet kokia būtina sąlyga, tokia kaip sistemos aparatinės dalies palaikymo susitarimas, papildomų resursų pateikimas užsakovo aplinkoje, turėtų būti suderinta, bei iš anksto dokumentuota.

- **Laiko apribojimai ir darbų grafikas**

Jeigu vykdomi darbai yra kritiškai svarbūs, laiko apribojimas ir darbų grafikas turėtų būti labai tiksliai apibrėžti, su tikslu, kad atitikti ribotus sistemos išjungimo, ar profilaktinių darbų vykdymo laiko tarpus. Būtina įsitikinti, ar atsarginis planas yra paruoštas, jei įgyvendinimas būtų atšauktas.

- **Išteklų planas**

Plane turėtų būti įtrauktas žmogiškųjų resursų naudojimas, jų įgūdžiai ir užimtumas [2].

5.4.Fazė 3: Duomenų perkėlimas

Šioje fazėje projektavimo fazės metu sukurtas planas yra įgyvendinamas. Bendros tinkamo duomenų perkėlimo įgyvendinimo praktikos specifiką sudaro:

- Pirmiausiai, prieš atliekant bet kokiais operacijas, sukuriama visavertė visų duomenų perkėlimo metu apdorojamų duomenų kopija.
- Peržiūros detalios procedūros. „Scenarijaus“ tipo metodologija yra tinkamiausias būdas. Tai reiškia, kad tiksliai dokumentuojami kiekvieno proceso dalyvio veiksmai ir atsakomybė. Taip pat jam skiriamos užduoties ir informacijos apie situaciją pristatymas kitiems komandos nariams.

- Įsitikinti, kad tinkamai apibūdinti bei prieinami proceso vykdymui reikalingi bet kokie ištekliai. Taip pat, kad atsitraukimo bei paties duomenų perkėlimo planai yra tinkamai suprasti ir įsisavinti.
- Surengti bandomąjį duomenų perkėlimą ar atlikti išankstinį testavimą kiekviename žingsnyje. Testas, vykdomas prieš tvarkaraštyje apibrėžtą įvykį, gali būti labai vertingas.
- Stebėti duomenų pralaidumo galimybes nuo pat ankstyvos projekto egzistavimo dalies. Jei duomenų perkėlimas yra paremtas tam tikru duomenų pralaidumo lygiu ir jei projekto vykdymo metu pastebimas žymus skirtumas tarp to, ko buvo tikimasi ir koks dabar yra realus duomenų pralaidumas, galbūt dar būtų galima atrasti laiko pakoreguoti tvarkaraščius, našumą ar metodologiją prieš projektui pasiekiant ribą, po kurios yra viršijamas projektui skirtas laikas.
- Labai svarbu laikytis testavimo plano ir viską dokumentuoti. Didžiausia duomenų perkėlimo proceso rizika yra pastebima duomenų integravimo problemos radime, neturint jokių žinių ir priemonių, kaip ji galėjo atsirasti. Jei planas yra dokumentuotas ir jo yra tinkamai laikomasi, grėsmės galima išvengti, atrandant problemą vienoje iš ankstesnių testavimo iteracijų, žinant atliktus pakeitimus. Labai svarbu gauti užsakovo patvirtinimą apie tinkamai atliktą duomenų perkėlimo etapą, prieš judant prie kito duomenų šaltinio perkėlimo.
- Nedelsiant kelkite problemas į „viršų“, kad jos būtų sprendžiamos iš esmės. Gali atrodyti lengviau kurį laiką palaukti, tikintis, kad problema pati išsispręs. Tačiau, jei kyla problemų, kurių sprendimas yra nenumatytas plane, labai svarbu įtraukti projekto vykdytojo įmonės projektams vadovaujančius ir resursus skirstančius asmenis, kad įsitikinti, jog bus įmanoma gauti pagalbą, kai jos reiks, ypač jei projekto komanda turi mažai patirties duomenų perkėlime [2].

5.5.Fazė 4: Rezultatų analizė ir apžvalga po migravimo

Tam, kad įsitikinti, kad nėra jokio šalutinio duomenų perkėlimo poveikio, labai svarbu atlikti sekančius veiksmus:

- Sistemos našumo statistikos palyginimas prieš ir po duomenų perkėlimo yra būtinas.
- Pašalinti ar išjungti duomenų perkėlimo įrankius ar/ir funkcionalumą iš sistemos, jei jis pastoviai veikia su sistema, kad nenaudoti veltui sistemos resursų ir turėti tinkamą sistemos našumo statistiką.
- Peržiūrėti duomenų perkėlimo metu kilusias problemas, metodus, kuriais jos buvo išspręstos, ir rezultatus [2].

5.6. Išteklių priskyrimas

Yra dar viena problema, su kuria susiduriama perkeliant duomenis. Kadangi duomenų perkėlimas yra iteracinis procesas, todėl tokio projekto metu reikia daugelio sričių specialistų trumpam laikui. Vadinasi, savų išteklių naudojimo procentas yra labai svarbus veiksnys norint tinkamai paskirstyti projekto išlaidas. Pavyzdžiui, jei projektui užbaigti reikia 1000 savo išteklių dienų, bet jei galima bus naudoti tik 75% savo išteklių, į tai reiks atsižvelgti formuojant biudžetą. Tokiu atveju reikia naudotis lanksčių organizacijų paslaugomis. Atliktų tyrimų duomenimis, 200 pasaulinių įmonių atlieka apie 4,48 duomenų perkėlimus per metus, taigi logiška būtų naudoti tuos pačius išteklius lygiagrečiai ir paskirstyti duomenų perkėlimo laiką taip, kad visi ištekliai turėtų pakankamai darbo [3].

Yra daugelis priežasčių, kodėl organizacijos pradeda duomenų perkėlimo projektus. Tai gali būti esamos programinės įrangos atnaujinimas, naujos sistemos kūrimas ar pirkimas. Tačiau nepaisant įvairovės, visi duomenų perkėlimo projektai turi bendrą savybę: jie yra sudėtiniai, reikalaujantys daug laiko ir pinigų. Šie projektai dažniausiai užtrunka kur kas ilgiau nei buvo planuota iš anksto dėl įvairių nenumatytų priežasčių. Kartais dėl šių priežasčių jie netgi gali būti nutraukti dėl didelių sąnaudų. Taigi kyla klausimas, kaip galima išvengti tokių atvejų ir projekto žlugimo? Pirmiausia yra projekto metodika. Naudojant teisingą projekto metodiką, duomenų perkėlimą galima padaryti žymiai paprastesnį ir lengvesnį [8].

5.7. Tiesinis duomenų perkėlimas

Logiška duomenų perkėlimą laikyti tiesiniu procesu, turinčiu keturias skirtingas fazes:

1. Duomenų šaltinio analizė.
2. Duomenų išgavimas ir transformavimas.
3. Duomenų patikrinimas ir klaidų ištaisymas.
4. Duomenų įkėlimas į gaviklį.

Tačiau duomenų perkėlimas yra labiau iteracinis procesas, todėl bandant pritaikyti tiesinę duomenų perkėlimo metodiką gali būti didelė klaida. Štai kas paprastai nutinka atliekant duomenų perkėlimą. Iškart pradėjus perkelti duomenis gali paaiškėti, jog trūksta informacijos apie duomenis ir duomenų šaltinio sistemą. Informaciją galima analizuoti ir analizuoti, bet pradėjus kelti duomenis gali atsirasti naujų nenumatytų apribojimų, dėl kurių reikia atlikti išsamesnę informacijos analizę. Kai pagaliau analizė atlikta, galima pereiti prie antros duomenų perkėlimo fazės, tai duomenų išgavimo ir transformavimo fazė. Joje gali atsirasti problemų dėl besikeičiančių gaviklio sistemos reikalavimų (tiek aparatinės įrangos, tiek programinės įrangos). Perėjus į duomenų tikrinimo fazę, vėl gali kilti įvairių problemų, kurios privers iš naujo išanalizuoti duomenų šaltinį ir sukurti naujus

metodus duomenims išgauti ir transformuoti. Dėl tokių klaidų gali tekti vėl ir vėl grįžti į ankstesnes fazes. Tiesinė duomenų perkėlimo metodika, kuri kiekvieną projekto fazę traktuoja atskirai, švaisto laiką ir pinigus. Tiesinis metodas paprasčiausiai nėra lankstus, kad galėtų efektyviai valdyti duomenų perkėlimą, kuris yra iteracinio pobūdžio [8].

5.8. Iteracinis duomenų perkėlimas

Tinkamesnė metodika, kuri lengvai pritaikoma prie duomenų perkėlimo, yra iteracinio pobūdžio. Naudojant iteracinę metodiką, galima nuolatos kartoti duomenų analizę, išgavimą, tikrinimą ir t.t., nešvaistant laiko ir pinigų iš biudžeto, kadangi visos keturios fazės yra sujungtos, o ne pavienės. Naudojant iteracinį procesą, kliūtys, trukdančios sėkmingai atlikti duomenų perkėlimą, yra sumažinamos arba visiškai pašalinamos. Visada galima grįžti į ankstesnes fazes ir kartoti duomenų analizę tol, kol apie duomenis bus gauta visa reikiama informacija. Šis lankstus būdas leidžia kur kas lengviau valdyti duomenų perkėlimo procesą, kad, atsiradus tam tikriems pakeitimams, netektų iš naujo suplanuoti viso projekto.

Ir paskutinis, bet labai reikšmingas dalykas yra tas, kad sukurama struktūra, kurią sudaro instrukcijos ir būdai, kaip efektyviai atlikti duomenų perkėlimą organizacijoje. Ši struktūra gali būti nuolat atnaujinama, atliekant vis kitokius projektus [8].

6. Duomenų perkėlimo įgyvendinimas

Duomenų perkėlimo metodika ir šios srities specifika buvo apžvelgta pirmame skyriuje. Toliau bus pateiktas realus duomenų perkėlimo projektas, jo įgyvendinimas bei funkcionalumas. Duomenų perkėlimo projektas yra vis dar vykdomas įmonėje UAB „Exigen Services Vilnius“. Šioje įmonėje aš dirbu programuotoju, ir pats prisidėjau kuriant duomenų perkėlimo įrankį. Kadangi įmonė kurioje aš dirbu, pasirašė kontraktą su klientu, dėl informacinės sistemos, sutrumpintai vadinamos IPB kūrimo bei pritaikymo jo reikmėms. Viena iš suderintų užduočių buvo duomenų perkėlimas iš senos kliento informacinės sistemos. Buvo suderinta, jog klientas pats vykdytų duomenų eksportą iš savo senos informacinės sistemos. Dėl šios priežasties duomenų perkėlimo įrankio kūrimas turėjo savitą specifiką, kadangi reikėjo kažkaip suderinti bendrą eksportuojamų duomenų struktūrą, kurią suprastų ir naudotų tiek kliento duomenų eksporto įrankis, tiek kuriamas duomenų perkėlimo įrankis.

6.1. Duomenų šaltinis – sena informacinė sistema

Duomenų perkėlimo šaltinis yra sena kliento informacinė sistema. Kadangi klientas pats nusprendė atrinkti duomenis, kuriuos jis norės perkelti į naują informacinę sistemą, detalūs duomenys apie šaltinį – seną informacinę sistemą, nebuvo labai detalūs. Daugiausiai ką pavyko sužinoti, kad kliento informacinių technologijų skyrius pats kūrė senąją sistemą. Ji yra nelanksti pasikeitimams ir buvo nuspręsta įsigyti naują informacinę sistemą. Sistemos funkcionalumą pilnai atkartoja ir papildo naujoji informacinė sistema, todėl tikslesnis šių sistemų funkcionalumo aprašymas bus pateiktas IPB aprašyme.

The screenshot displays the IPB (Individual Policy Builder) interface. At the top, the 'Policy Number' is 1049001. Below this, various policy details are listed in a form-like layout:

- Policy Status: In force
- Status Reason: Unhappy with Claims
- Suspended: ☐
- Scheme: Real Insurance
- Intermediary: Real
- Intermediary Reference Number: (empty)
- Intermediary Branch: (empty)
- Intermediary Agent: (empty)
- Original Inception Date: 06 Oct 2006
- Inception date: 06 Oct 2009
- Expiry Date: 05 Oct 2010
- Next Installment Date: 06 Oct 2010
- Payment Frequency: Annual
- Premium Balance: 0.00
- Confirmed Premium Balance: 0.00
- Premium: 398.16

Below the form, there is a 'Premium Summary' section with checkboxes for 'Current' (checked), 'Pending', and 'History'. A 'Benefit Movements' button is also present.

At the bottom, a table lists the policy details:

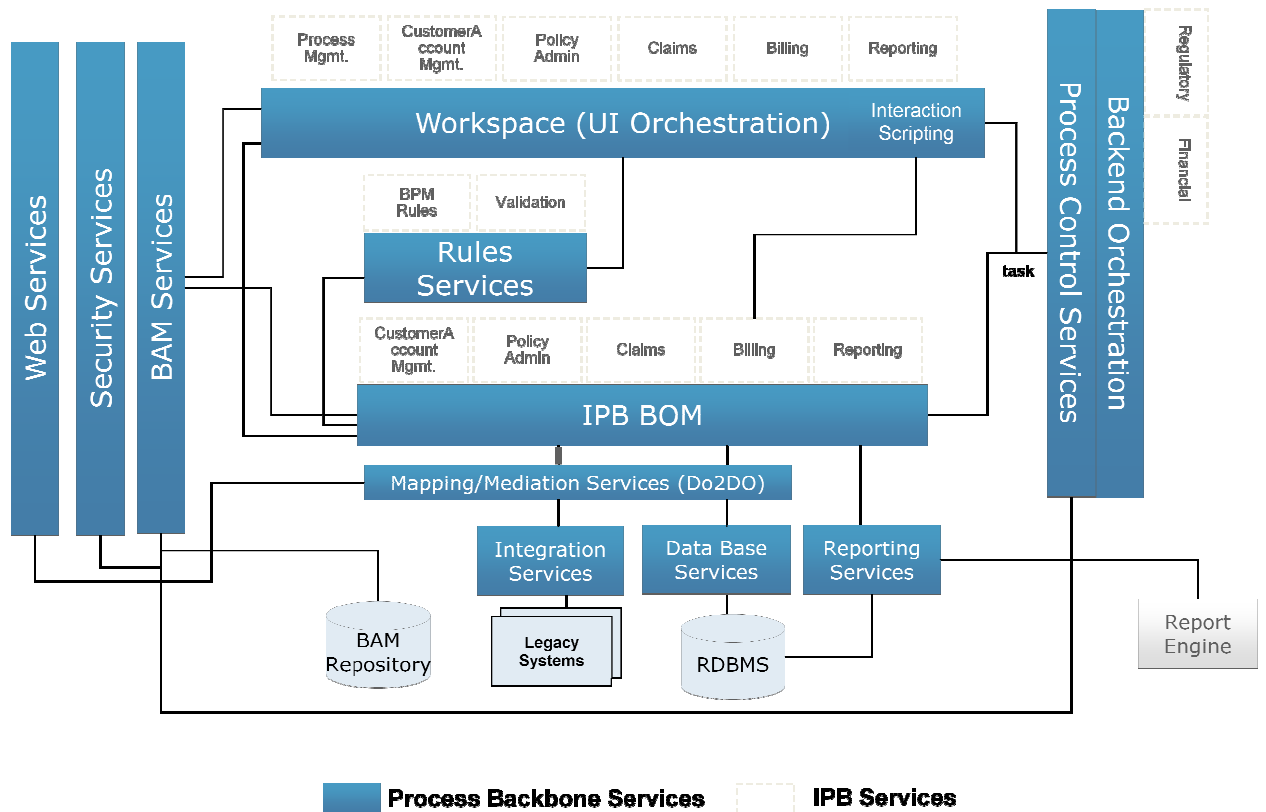
Category	Item	Product	Premium	SumInsured	Excess	Inception Date	Expiry Date
Motor	HONDA,CIVIC	Motor Comprehe...	398.16	20560.00	500.00	06 Oct 2009	05 Oct 2010

14 pav. Duomenų šaltinio taikomosios programos langas

6.2. Draudimo procesų sistema – IPB

6.2.1. Bendroji sistemos architektūra

IPB sistema suteikia draudimo sprendimus įgyvendinančios sistemos pagrindą, kuris yra plečiamas pagal individualius užsakovo poreikius. Bendroji IPB sistemos architektūrinė schema pateikiama 15 pav.



15 pav. Bendroji architektūrinė IPB sistemos schema

Sistema yra skaidoma į šias dalis:

- Pagrindinė posistemė – pateikia pagrindines sąsajas bei reikalingus duomenų tipus.
- Ryšių su klientais valdymo posistemė – pateikia ryšių su klientais valdymo servisus apimančius visą klientų ir jų paskyrų gyvavimo ciklą (CRUD), paskyrų tarpusavio sąryšių valdymą, paieškos ir išsaugojimo funkcionalumus.
- Draudimo produkto posistemė – pateikia visų užsakovo turimų draudimo produktų registrą, bei galimybę valdyti jau turimus, bei pridėti naujus draudimo produktus.
- Draudimo poliso posistemė – pateikia sąsajas bei servisus, reikalingus užtikrinti nuo verslo logikos priklausančius draudimo poliso išdavimo procesus bei įgyvendinančius visą polisų gyvavimo ciklą.
- Bankinių pavedimų valdymo posistemė – užtikrina tinkamą sistemos bendravimą su bankais bei kitomis bankinėmis sistemomis, pateikia tam reikalingas sąsajas ir servisus. Taip pat integruoja jau veikiančias bankines sistemas.
- Turinio valdymo posistemė – pateikia kliento, poliso bei bankinių pavedimų dokumentų šablonų, dokumentų generavimo taisyklių, dokumentų, skirtų spausdinimo, generavimo, paveikslėlių įkėlimo, indeksavimo ir saugojimo servisus.
- Verslo procesų modeliavimo posistemė – atsakinga už tinkamą verslo procesų modeliavimą ir modelių pateikimą kitoms posistemėms.

- Vartotojų valdymo/apsaugos posistemė – suteikia galimybę valdyti vartotojus bei yra atsakinga už bendrą sistemos saugumą, taip pat už duomenų užšifravimą ir iššifravimą.
- Verslo veiklos sekimo posistemė – suteikia galimybę stebėti, kiek laiko sistemos naudotojas, verslo procesai, verslo servisi ir apsaugos servisi praleido vykdydami vieną ar kitą užduotį.
- Ataskaitų posistemė - pateikia integruotą ataskaitų rengimo variklį ir verslo veiklos sekimo posistemės gautų rezultatų vaizdavimą ataskaitomis.
- Grupinių darbų atlikimo posistemė – pateikia darbų planuotojo funkcionalumą, kuris naudojamas bankinių pavedimų, ataskaitų ir dokumentų generavimo posistemėse.

6.2.2. Sistemai keliami reikalavimai

Užsakovai suformulavo tokius bendruosius kuriamos sistemos reikalavimus:

- turi būti suteikta galimybė apibrėžti žmogiškos darbų sekos žingsnius naudojant priimtinas modeliavimo priemones;
- organizacinis sistemos modelis turi palaikyti ataskaitų bei kitų funkcinių modulių tarpusavio sąryšius;
- rolėmis ir privilegijomis grįsti užduočių skirstymo bei atlikimo mechanizmai;
- patirtimi grįsti užduočių skirstymo ir atlikimo mechanizmai
- galimybė skirstyti darbų prioritetus pagal kompleksinius verslo duomenis;
- galimybė užsakovui pačiam nustatyti darbų atlikimo prioritetus ir skirstymo logiką;
- užtikrinti tokių koncepcijų kaip “paskutinis veiksmo atlikėjas”, “prioritetinis veiksmo atlikėjas” buvimą ir naudojimą;
- veiksmų sekimas ir pranešimų siuntimo galimybės, kurios padės užtikrinti veiksmų sekimą bei sutarčių su kitomis sistemomis laikymąsi;
- konfigūruojamo vartotojo užduočių sąrašo palaikymas;

Užsakovams pageidaujant užsakoma sistema turėjo tenkinti šiuos reikalavimus, bei integruotis į jau esamas sukurtas ir naudojamas sistemas:

- vartotojų autentifikavimas ir prieigos teisės;
- vartotojų paskyrų registravimas ir administravimas;
- klientų ir jų paskyrų valdymas;
- procesų ir darbų valdymas;
- draudiminių produktų įvedimas ir administravimas;
- poliso gyvavimo ciklo procesų valdymas;
- mokėjimų procesų valdymas;

- dokumentų generavimas;
- paketinių procesų valdymas;
- sistemos darbo ataskaitų generavimas;

Taip pat sistema turėjo tenkinti nefunkcinius, bet ne mažiau svarbius reikalavimus. Nefunkciniai reikalavimai buvo suformuluoti vieno kalendorinių metų laikotarpiui, laikant, jog sistema turėtų dirbti maždaug 300 dienų per metus. Reikalavimai buvo šie:

- iki 21000 automobilių draudimo polisų generavimas;
- iki 200 draudimo polisų pratęsi kiekvieną dieną;
- 327 nauji draudimo polisų įvedimai;
- 30 dokumentų kiekvienam polisui generavimas;
- gaunamų 30 dokumentų kiekvieną valandą apdorojimas;
- gaunamų 200 dokumentų faksu ir elektroninių paštu apdorojimas;
- gaunamų 200 dokumentų įprastu paštu apdorojimas;
- vidutinis kiekvieno dokumento dydis – 100kB;
- siunčiamų 70 dokumentų rūšių palaikymas;
- 625 išeinančių skambučių per dieną palaikymas;
- 1250 įeinančių skambučių per dieną palaikymas;

6.2.3. Pagrindiniai sistemos tikslai

Užsakovas suformulavo tokius kuriamos informacinės sistemos tikslus:

- Sistemos pristatymas laiku. Kadangi sistemos kūrimas yra padalintas fazėmis, todėl kiekviena sistemos vystymo fazė turi būti laiku pristatyta užsakovui. Sistemos vystymas yra padalintas fazėmis taip, jog kiekviena sistemos fazė suteiktų daugiausiai naudos ir kainuotų mažiausiai laiko bei įneštų mažiausiai trikdžių į dabar vykstančius procesus.
- Didinti efektyvumą. Pagrindinis šios sistemos tikslas yra padidinti produktyvumą automatizuojant, kryptingai vystant procesus bei naudojant pažangiausias technologijas.
- Integruoti veikiančias sistemas. Kur tik įmanoma, būtina naudoti jau esamas sistemas, užuot jas kūrus iš naujo ir pakeisti senąsias.

6.2.4. Ryšių su klientais valdymo posistemė

Ryšių su klientais valdymo posistemė yra sudėtinė informacinės sistemos dalis. Ši posistemė yra atsakinga už šias sistemos funkcijas:

- klientų registravimą;

- klientų susiejimą su paskyromis;
- redaguoti ir šalinti klientus iš sistemos;
- perkelti klientus iš vienos paskyros į kitą.

Sistemoje pagal užsakovo reikalavimus yra skiriami šie klientų tipai:

- anonimai;
- identifikuojami anonimai;
- fiziniai asmenys
- jungtiniai klientai;
- verslo klientai;
- šeimos patikėtiniai.

Anonimų ir identifikuotų anonimų klientų tipai yra naudojami užsienio pardavimams valdyti. Identifikuojamas anonimas, tai anonimas, pateikęs savo vardą ir pavardę. Tačiau joks polisas negali būti išduotas anonimo arba identifikuoto anonimo tipą turintiems klientams.

Įprastinis verslo procesas teigia, jog klientas, norintis gauti draudimo polisą, yra fizinis asmuo, jungtinis, verslo klientas arba šeimos patikėtinis. Egzistuoja skirtingos sąlygos ir yra apibrėžtos skirtingos taisyklės, kurie laukai (ir kiek vardų aibių) turėtų būti pildoma kiekvienam kliento tipui.

Fizinis asmuo yra vienas žmogus, jungtinis klientas yra bent du žmonės (sutuoktiniai arba partneriai), verslo draudimo sąskaita atidaroma verslininkams. Informacinės sistemos vartotojas dirbdamas bendrauja su klientu ir pagal jo pageidavimus ir poreikius pateikia draudimo poliso pasiūlymą. Kiekvienam vartotojui prieinamas IS funkcionalumas nustatomas pagal vartotojui suteikiamas roles ir privilegijas. Visų rolių ir privilegijų priskyrimas tam tikroms sistemos funkcijoms yra aprašytas saugumo posistemės panaudos scenarijaus dokumente. Jame aprašomos rolės ir privilegijos bei funkcijos, kurias galima atlikti turint viena ar kitą rolę ir privilegiją.

6.3. Duomenų perkėlimo metodika

Su kliento atstovais suderinus duomenų perkėlimo įrankio funkcionalumą, duomenų apimtį ir pobūdį, bei duomenų perkėlimo trukmę, buvo pasirinktas perkėlimo dalimis metodas. Pasirinkimą lėmė keletas priežasčių :

- Duomenų perkėlimą klientas pageidavo vykdyti savarankiškai
- Pradžioje buvo planuojama perkelti 100 000 draudimo polisų, kurių kiekvienas sukuria įrašus bent penkiolikai lentelių, priklausomai nuo poliso tipo ir draudžiamų objektų. Tokiu atveju, dėl didelio duomenų kiekio, perkėlimas vienu kartu truktų daugiau nei savaitę, kas nėra priimtina.

- Planuojama kurį laiką dar naudoti senąją sistemą, todėl gali atsirasti neplanuotas papildomas duomenų perkėlimo poreikis atlikus planuotą duomenų perkėlimą.

6.4. Perkeliamų duomenų pobūdis

Duomenų perkėlimo tikslas yra perkelti senus klientų duomenis. Kadangi klientas užsiima draudimo paslaugomis, tai reiks perkelti draudimo polisų duomenis. Norint atitikti naujos informacinės sistemos reikalavimus perkeliant polisą reikia turėti visą būtiną informaciją, todėl kartais panaudojama atitinkama visiems polisams vienoda reikšmė, jei šaltinyje reikalingų duomenų nėra.

Informacinėje sistemoje pirmiausia įvedami kliento duomenys, juos įvedus galime pasirinkti klientą ir sukurti jam polisą. Pasirinkus poliso rūšį įvedame visus reikalingus duomenis ir jei klientas atitinka keliamus reikalavimus ir nepriklauso jokiai rizikos grupei, jam sudaromas draudimo polisas.

Tokie pat duomenys turi būti užpildomi ir duomenų perkėlimo metu. Tam kad simuliuoti sistemos darbą, pirmiausiai įkeliami kliento duomenys. Dėl šios priežasties, eksporto metu bus sukurta atskira duomenų rinkmena, skirta tik kliento duomenims. Kadangi kiekvienas klientas gali turėti keletą draudimo polisų, kiekvienas draudimo polisas bus pateiktas atskiroje duomenų rinkmenoje.

XML duomenų rinkmenos tipas buvo pasirinktas, kad duomenis būtų galima patogiai eksportuoti iš informacinės sistemos ir juos perkelti į kitą informacinę sistemą. Šis duomenų rinkmenos tipas turi savų privalumų, kuriuos galima pritaikyti tokio tipo duomenų perkėlime, t.y. kai duomenų eksportas ir importas atliekamas skirtingų programinių įrankių.

6.5. XML dokumentas, XML standartas

Norint suprasti, kas yra XML, pirmiausia reikia suprasti, kas yra duomenų žymėjimas. Žmonės per amžius kūrė dokumentus ir tiek pat laiko juos žymėjo. Pavyzdžiui, mokyklose mokytojai visą laiką žymi studentų darbus. Jie nurodo studentams perkelti pastraipas, aiškiau išreikšti sakinį, ištaisyti rašybos klaidas ir t. t. Dokumentų žymėjimas yra dokumento informacijos struktūros, prasmės ir išvaizdos apibrėžimas. Jei esate naudoję programos „Word“ keitimų sekimo priemonę, naudojote kompiuterizuotą žymėjimo formą. Kompiuterijoje „žymėjimas“ tapo „žyme“. Žymėjimas yra kodų, vadinamų žymėmis (arba kartais ženklais), naudojimo procesas norint apibrėžti bet kokių duomenų struktūrą, išvaizdą ir (XML atveju) prasmę.

HTML ir XML dokumentuose yra duomenų, aplink kuriuos yra žymės, bet tai yra vienintelis dviejų kalbų panašumas. HTML žymės apibrėžia duomenų išvaizdą ir naudojimą – čia yra antraštės, čia prasideda pastraipa ir t. t. XML žymės apibrėžia duomenų struktūrą ir prasmę – kas yra duomenys, t.y. meta duomenimis aprašomi duomenys. Kai aprašoma duomenų struktūra ir prasmė, tuos duomenis galima įvairiais būdais pakartotinai panaudoti. Pavyzdžiui, jei turite pardavimo duomenų bloką, o kiekvienas bloko elementas aiškiai apibrėžtas, į pardavimo ataskaitą galite įkelti tik tuos elementus, kurių jums reikia, o kitus elementus įkelti į apskaitos duomenų bazę. Kitaip tariant, galite naudoti vieną sistemą savo duomenims generuoti ir žymėti XML žymėmis, tada apdoroti tuos duomenis bet kokioje kitoje sistemoje, neatsižvelgiant į aparatūros platformą ar operacinę sistemą. Šis mobilumas yra pagrindinė priežastis, kodėl XML tapo viena iš populiariausių duomenų mainų technologijų. Taip pat XML dokumentai pasižymi tuo, kad jie nepriklauso nuo kompiuterio platformos ir yra dažnai naudojami, kaip sąsaja tarp programų ar procesų [1].

Egzistuoja keletas esminių dalykų, susijusių su HTML ir XML naudojimu:

- Vietoj XML negalima naudoti HTML. Tačiau XML duomenis galima įkelti į HTML žymes ir rodyti tinklalapyje.
- HTML apribotas iki iš anksto apibrėžto žymių rinkinio, kurį bendrai naudoja visi vartotojai.
- XML leidžia kurti bet kokią žymę, kurios reikia norint apibrėžti jūsų duomenis ir jų struktūrą. Pavyzdžiui, jūs norite saugoti ir bendrai naudoti informaciją apie naminius gyvūnus. Galite sukurti toki XML kodą:

```
<?xml version="1.0"?>
<KATĖ>
  <VARDAS>Murkė</VARDAS>
  <VEISLĖ>Siamo</VEISLĖ>
  <AMŽIUS>6</AMŽIUS>
  <STERILIZUOTA>taip</STERILIZUOTA>
  <NUKIRPTI_NAGAI>ne</NUKIRPTI_NAGAI>
  <LICENCIJA>Izz138bod</LICENCIJA>
  <ŠEIMININKAS>Vardenis Pavardenis</ŠEIMININKAS>
</KATĖ>
```

16 pav. XML rinkmenos ištrauka

Matote, kad naudojant XML žymes, galima tiksliai žinoti, į kokius duomenis žiūrima. Pavyzdžiui, žinote, kad šie duomenys yra apie katę, ir galite lengvai rasti katės vardą, amžių ir kt. Dėl gebėjimo kurti žymes, kurios apibrėžia beveik bet kokio tipo duomenų struktūrą, XML yra „išplėstinis“.

Nesupainiokite šiame kodo pavyzdyje esančių žymių su HTML rinkmenos žymėmis. Pavyzdžiui, jei įklijuosite šią XML struktūrą į HTML rinkmeną ir peržiūrėsite rinkmeną savo naršyklėje, rezultatas atrodys maždaug taip:

17 pav. HTML kalba atvaizduotas kodas

Naršyklė nepaiso jūsų XML žymių ir rodo tik duomenis. Norint matyti gražiai pateiktus duomenis reikia pasirašyti XSLT transformacijas.

XML yra nepriklausomas nuo platformos, tai reiškia, kad bet kuri programa, skirta naudoti XML duomenis, gali skaityti ir apdoroti jūsų XML duomenis, neatsižvelgiant į kompiuterio aparatinę įrangą ar operacinę sistemą. XML yra labai mobili, todėl ji tapo viena iš populiariausių technologijų duomenų mainuose tarp duomenų bazių [10].

Schemos. Schema yra tiesiog XML failas, kuriame yra taisyklės, kas gali ir ko negali būti XML duomenų faile. Schemų rinkmenos paprastai naudoja XSD plėtinį, o XML duomenų rinkmenos naudoja XML plėtinį.

Schemas leidžia programoms tikrinti duomenis. Pagal schemoje pateikiamą duomenų struktūrą XML rinkmena yra pakraunama į atmintį, atitinkamai išdėstant XML struktūros elementus. Jei duomenys yra nekorektiški, pavyzdžiui: datos tipo lauke yra įvestas tekstas, XML struktūros pakrovimo į atmintį metu, bus išvesta vykdymo klaida, pranešanti apie duomenų tipų neatitikimą ir bus nutrauktas XML rinkmenos apdorojimas. Kai visa XML rinkmenoje saugoma informacija atitinka struktūrinius reikalavimus, saugomais duomenis galima bus patogiai naudotis, žinant jog buvo patikrintas jų atitikimas XSD schemai. Tokią XML rinkmeną, XML palaikanti programa gali naudoti duomenims skaityti, aiškinti ir apdoroti.

Kaip jau anksčiau buvo minėta, duomenų perkėlimui buvo pasirinkta naudoti ACORD standarto XSD schemą. ACORD yra draudimo industrijos tyrimų ir kūrimo ne pelno siekianti asociacija, užsiimanti standartų kūrimu. Tai informacijos šaltinis, naudojant XML ir kitas elektroninės komercijos priemones. Šis standartas plačiai naudojamas Jungtinėse Amerikos Valstijose ir visame pasaulyje. Geriausiai žinomas jų sukurtas XSD formato standartinių formų rinkinys, skirtas draudimo sričiai. Dauguma draudiminių įvykių Jungtinėse Amerikos Valstijose yra aprašomi ir persiunčiami tarp įvairių institucijų, naudojant ACORD formas.

```

<xsd:complexType name="Dwell_Type">
  <xsd:sequence>
    <xsd:element ref="PrincipalUnitAtRiskInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PolicyTypeCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PurchaseDt" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PurchasePriceAmt" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="OilStorageTankLocationCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="FuelLineLocCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="WiringTypeCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="WiringInspectedDt" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="HousekeepingConditionCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="StormShuttersCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="FloorUnitLocated" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="NumOtherStructures" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="OtherStructureAmt" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="AdditionalResidenceInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="NumEmployeesFullTimeResidence" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="NumEmployeesPartTimeResidence" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="LandArea" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="ThermostaticallyControlledCentralHeatInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="DogTypeCd" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="HillsidePct" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="HillsideIncline" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PublicOfficeInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="AreaTypeSurroundingsCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="Construction" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="DwellOccupancy" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="DwellRating" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="BldgExposures" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="BldgProtection" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="BldgImprovements" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="DwellInspectionValuation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="MobileHome" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="SwimmingPool" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="AnimalExposureInfo" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Coverage" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="QuestionAnswer" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="DwellStateSupplement" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="MinPremInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="HeatingUnitInfo" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="InspectionInfo" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="PlumbingConditionCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PlumbingLeaksInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="TotalInsurableReplCostAmt" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="FloodInfo" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="RoofConditionCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="HistoricHomeInd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="PayorCd" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="SafetyNetInd" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="ID"/>
  <xsd:attribute name="LocationRef" type="IDREF"/>
  <xsd:attribute name="SubLocationRef" type="IDREF"/>
  <xsd:attribute name="CondoManagerRef" type="IDREF"/>
</xsd:complexType>

```

18 pav. Standartinės ACORD XSD schemas ištrauka [5]

Duomenų perkėlimui konkrečioje situacijoje, ACORD standarto naudojimas yra labai naudingas, kadangi maždaug 60% reikalingos perkelti informacijos galima buvo patalpinti standartiniuose XSD schemas elementuose. Visgi likusius 40% teko pridėti. XSD schemas išplėtimo galimybės yra labai didelės, kadangi sintaksė nėra sudėtinga ir XSD schema yra labai lanksti. Pridedant naujai sukurtus XSD schemas elementus pradžioje buvo pridedamas:

„com.exigen“ tekstas, kad kuo paprasčiau būtų galima identifikuoti kurie schemos elementai yra ACORD standarte, o kurie yra naujai pridėti.

```
<xsd:complexType name="Dwell_Type">
  <xsd:sequence>
    <xsd:element minOccurs="0" maxOccurs="1" name="PrincipalUnitAtRiskInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PolicyTypeCd" type="PolicyType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PurchaseDt" type="Date"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PurchasePriceAmt" type="CURRENCY"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="OilStorageTankLocationCd" type="LocationType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="FuelLineLocCd" type="FuelLineLocation"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="WiringTypeCd" type="WiringType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="WiringInspectedDt" type="Date"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="HousekeepingConditionCd" type="HousekeepingCondition"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="StormShuttersCd" type="StormShutterType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="FloorUnitLocated" type="Long"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="NumOtherStructures" type="Long"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="OtherStructureAmt" type="CURRENCY"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="AdditionalResidenceInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="NumEmployeesFullTimeResidence" type="Long"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="NumEmployeesPartTimeResidence" type="Long"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="LandArea" type="MEASUREMENT"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="ThermostaticallyControlledCentralHeatInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="DogTypeCd" type="DogType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="HillsidePct" type="PercentLong"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="HillsideIncline" type="MEASUREMENT"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PublicOfficeInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="AreaTypeSurroundingsCd" type="LocationType"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="Construction" type="Construction_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="DwellOccupancy" type="DwellOccupancy_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="DwellRating" type="DwellRating_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="BldgExposures" type="PROXIMITY"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="BldgProtection" type="BldgProtection_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="BldgImprovements" type="BldgImprovements_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="DwellInspectionValuation" type="DwellInspectionValuation_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="MobileHome" type="MobileHome_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="SwimmingPool" type="SwimmingPool_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="AnimalExposureInfo" type="AnimalExposureInfo_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Coverage" type="Coverage_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="QuestionAnswer" type="QuestionAnswer_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="DwellStateSupplement" type="DwellStateSupplement_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="MinPremInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="HeatingUnitInfo" type="HeatingUnitInfo_Type"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="InspectionInfo" type="InspectionInfo_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PlumbingConditionCd" type="HousekeepingCondition"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PlumbingLeaksInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="TotalInsurableReplCostAmt" type="CURRENCY"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="FloodInfo" type="FloodInfo_Type"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="RoofConditionCd" type="HousekeepingCondition"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="HistoricHomeInd" type="Boolean"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="PayorCd" type="Payor"/>
    <xsd:element minOccurs="0" maxOccurs="1" name="SafetyNetInd" type="Boolean"/>
    <xsd:element name="com.exigen_CoverType" type="xsd:string" minOccurs="0"/>
    <xsd:element name="com.exigen_LossInfo" type="com.exigen_LossInfo_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="com.exigen_NotOccupiedDate" type="Date"/>
    <xsd:element name="com.exigen_ValueOfContentsInStorage" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="com.exigen_CisLocationRef" type="xsd:string" minOccurs="0"/>
    <xsd:element name="com.exigen_StoriesNumber" type="Long" minOccurs="0"/>
    <xsd:element name="com.exigen_RoomsNumber" type="Long" minOccurs="0"/>
    <xsd:element name="com.exigen_PropertyItem" type="com.exigen_PropertyItem_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="com.exigen_Form" type="Form_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="com.exigen_WeeklyRentalIncome" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="com.exigen_NumberOfTenants" type="Long" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="ID"/>
  <xsd:attribute name="LocationRef" type="IDREF"/>
  <xsd:attribute name="SubLocationRef" type="IDREF"/>
  <xsd:attribute name="CondoManagerRef" type="IDREF"/>
</xsd:complexType>
```

19 pav. Papildytos ACORD XSD schemos ištrauka [5]

ACORD schema pasižymi begale elementų kuriuos galima pritaikyti, patenkinant praktiškai visus draudimo srities poreikius, duomenų dalinimosi ir keitimosi tarpusavyje atžvilgiu. Nors tai gana didelis privalumas, tačiau jis labai apkraunantis atmintį, kas yra nepriimtina siekiant efektyvesnio duomenų perkėlimo. Dėl šios priežasties nereikalingi ACORD XSD schemos

elementai buvo pašalinti. Kadangi įmonė „Exigen Services“ ACORD standartą naudoja jau senokai ir ji yra pritaikę kitose srityse, duomenų perkėlimo įrankio kūrimo buvo panaudota seniau įgyta patirtis. Tam kad pašalinti nereikalingus elementus yra sukurta nesudėtinga programa, kuri pirmiausiai iš XSD schemos pašalina nenaudojamus elementus. Toliau yra tikrinami naujai pridėti schemos elementai, lyginant sumažintą standartinę ACORD schemą ir naujai pridėtų elementų schemą. Sujungus jas į vieną, gaunama schema, kurioje visi elementai yra naudojami.

6.6. Duomenų perkėlimo įrankio tikrinimas

Kadangi įmonėje įvesta griežta visų sistemos modulių tikrinimo kontrolė, taip pat buvo tikrinamas ir duomenų perkėlimo įrankis. Duomenų perkėlimo įrankio tikrinimui buvo paskirtas vienas žmogus. Tokiam darbui reikia specifinių žinių, kad suvokti duomenų perkėlimo įrankio veikimo specifiką ir tai ją išanalizavus pateikti idėjų kokius scenarijus reikėtų vykdyti, norint įsitikinti duomenų perkėlimo programos veikimo korektiškumu.

Pirmiausiai, paskirtas žmogus turėjo susipažinti su duomenų perkėlimo įrankio specifikacija ir pagal ją sudaryti tikrinimo planą. Tikrinimo planą sudaro atskiri, skirtingą funkcionalumą tikrinantys testai. Egzistuoja du tikrinimo lygiai, vykdomi skirtingais atvejais. Pagrindinio funkcionalumo tikrinimo lygmuo tikrina funkcionalumą kuris dažniausiai naudojamas. Šio lygio testai vykdomi kiekvieną kartą kai išleidžiama nauja visos sistemos versija, tame tarpe ir duomenų perkėlimo įrankio versija. Kitas testavimo lygis tikrina apskritai visą funkcionalumą. Tokie testai vykdomi pagal pareikalavimą. Pavyzdžiui: pirminėse projekto fazėse, kai programa dar kuriama, ir klaidų pasitaiko pakankamai daug, tikrinamas visas funkcionalumas, kol tikrinama programa veikia pagal specifikaciją.

Klientas taip pat turi savo testuotojų komanda, kurie vykdo tikrinimą savo nuožiūra, tačiau apie jų vykdomus testus daug informacijos nėra pateikta. Jie taip pat vadovaujasi bendrai suderinta specifikacija, dažniau atsižvelgdami į verslo procesus. Tačiau aišku viena, kad duomenų perkėlimo įrankis buvo kruopščiai tikrinamas viso kūrimo metu.

7. Duomenų perkėlimo specifikacija

7.1. Funkcionalumas

- Bus perkeliama tik viena, aktyvi poliso versija.
- Lentelių įrašai duomenų bazėje turėtų būti atitinkamai pažymėti.
- RPL sistemoje draudimo polisų dabartinis numeris yra septynženklis skaičius. Tam, kad užtikrinti jog nesusikirs RPL polisų numeriai su IPB sukurtų polisų numeriais įsitikinta, jog RPL polisų numeriai yra intervale, kurio paskutinis skaičius specifikacijos sudarymo metu buvo 11202001, kai tuo tarpu IPB sukurti polisai numeruojami nuo 50000001.
- Atidarius perkeltą polisą IPB sistemoje, turi būti žymė, jog jis buvo perkeltas iš RPL sistemos [6].

7.2. Duomenų kiekis ir nefunkciniai reikalavimai

- Viso planuojama perkelti mažiausiai po 100 000 kiekvieno tipo polisų. Šis skaičius gali augti, kadangi pavyzdžiui namų, butų ir pastatų draudimo polisų jau dabar yra apie 80 000 ir galimas 50% to skaičiaus augimas. Viso planuojama perkelti tris draudimo polisų tipus.
- Duomenų perkėlimui XML rinkmenomis bus naudojamas draudimo standartų rinkinys ACORD, XSD formato.
- Duomenų apdorojimas neturėtų sustoti, jei vienas ar keletas polisų turi klaidų ar neatitikimų. Toks polisas turėtų būti pažymėtas kaip neperkeltas, ir jo XML rinkmena turėtų būti perkelta į atitinkamą katalogą. Jei rinkmena apdorota sėkmingai, ji taip pat atitinkamai pažymima ir perkeliama į direktoriją, kur perkeliama sėkmingai importuoti failai.
- Jei polisas nebuvo perkeltas, turi būti galimybė pakeisti jo duomenis, ir perkelti dar kartą [6].

7.3. Techninis duomenų perkėlimo įrankio dizainas

- Duomenų perkėlimui bus naudojamos XML tipo rinkmenos
- Bus naudojama kuo labiau supaprastinta ACORD standarto XSD schema
- Duomenų perkėlimo programa neribos ryšių tarp kliento ir draudimo poliso duomenų.
- Perkeliams klientų ir polisų duomenims nėra jokių numeracijos apribojimų, t.y. jie gali būti numeruojami skirtingai nei paprasti polisai, pagal duomenų perkėlimo proceso reikalavimus.

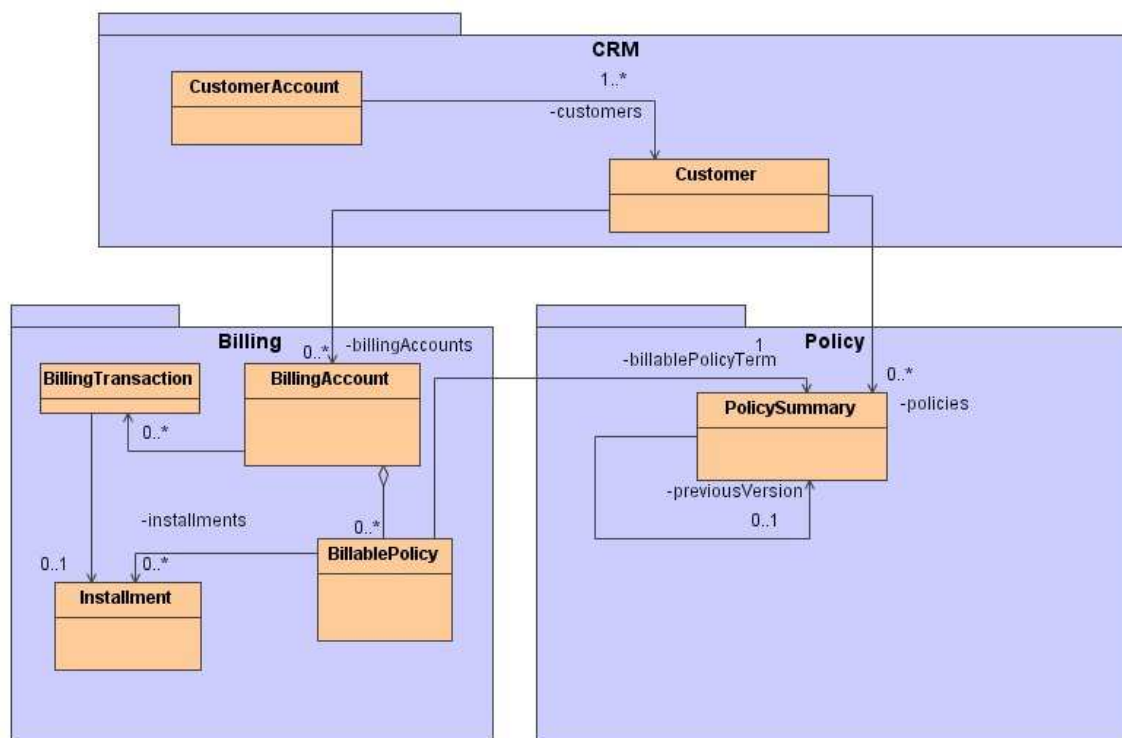
- Mokėjimų už draudimo polisą kredito kortele duomenys bus šifruojami naudojant AES duomenų šifravimo standartą.

Prielaidos

- Perkelti polisai turi atitikti visus IPB sistemos reikalavimus. Tai reiškia, kad negali būti jokių skirtumų, ar taisyklių (išskyrus suderintas išimtis) kur RPL polisas atrodys vienaip, o sukurtas IPB – kitaip.
- Duomenų perkėlimo metu, lentelėse kuriams naujiems įrašams, bus priskiriami visi reikalingi identifikatoriai [6].

7.4. Duomenų ryšiai lentelėse

Abstrakčiai žvelgiant, IPB egzistuoja pagrindinių lentelių medis, kurio kiekviena lentelė pateiktas kliento ir poliso duomenimis bus pildoma duomenų perkėlimo metu.



20 pav. IPB duomenų bazės lentelių medis

Lentelė CustomerAccount

Minėtoji lentelė susieja keletą klientų į vieną sąskaitą. Pavyzdžiui kai apdraudžiami skirtingi šeimos nariai, tačiau vienoje sąskaitoje. Tokiu atveju lentelėje atsiras tos sąskaitos įrašas. Joje taip pat laikomi tam tikri finansiniai duomenys, tokie kaip bendra sąskaitos polisų vertė ir informacija apie sąskaitos galiojimo terminą.

Duomenų perkėlimo metu, kliento duomenų XML rinkmenoje bus pateiktas sąskaitos numeris, tam kad pridėti klientą prie jau egzistuojančios sąskaitos, o jeigu tai naujas klientas – sukurti naują sąskaitą.

Lentelė Customer

Ši lentelė kaupia pagrindinę informaciją apie klientą, tokią kaip kliento vardas ir rinkodaros informacija, koks brokeris pardavė jam draudimo polisą. Gali egzistuoti keletas polisų, priskirtų vienam klientui. Šiai lentelei skirti duomenys bus perkeliama iš RPL sistemos, kadangi visi duomenys egzistuoja duomenų šaltinyje. Eksportuojant duomenis, kiekvienai kliento duomenų rinkmenai privalės būti priskirtas RPL kliento numeris, kuris nesidubliuos su IPB kuriamais šios lentelės įrašais [4].

Lentelė PolicySummary

Šioje lentelėje talpinama pagrindinė draudimo polisų informacija. Polisiai, kuriais IPB yra identifikuojami pagal unikalų poliso numerį, poliso galiojimo datą ir versijos numerį.

Polisas negali egzistuoti be kliento duomenų, esančių Customer lentelėje. PolicySummary lentelė su kliento duomenimis yra susieta kliento identifikaciniu numeriu. Identifikacinis poliso numeris turi būti perkeltas iš RPL duomenų šaltinio.

Lentelė BillingAccount

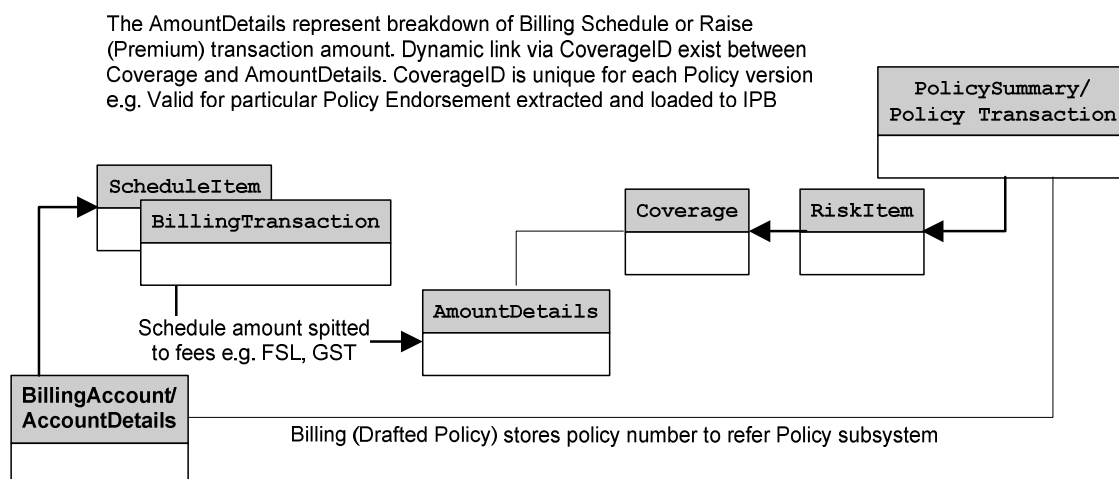
Šioje lentelėje yra saugoma finansinė poliso informacija. Pagal šios lentelės įrašus, galima surasti paskutinį aktualų mokėjimų grafiką (jei poliso apmokėjimo tipas yra mėnesinis, kitu atveju mokėjimas bus tik vienas), taip pat finansinių transakcijų – mokėjimų, grąžinimų istorinę informaciją.

Vienas BillingAccount lentelės įrašas yra sukuriamas vienam polisui. Skirtingos poliso versijos yra prijungiamos prie to paties įrašo.

Ši lentelė taip pat turi savo identifikacinį numerį – billingAccountNo. Duomenų perkėlimo reikmėms šis numeris bus toks pats kaip poliso numeris. Pagal duomenų perkėlimo proceso poreikius identifikacinis numeris gali būti keičiamas, kad atitiktų IPB kuriamų lentelės įrašų specifiką.

Minimalus duomenų perkėlimui reikalingų duomenų kiekis yra poliso finansinis balansas ir mokėjimų grafikas, su išskaidytais mokėjimais pagal mokesčius ir draudimo lygius.

Ryšis tarp poliso ir jo finansinių duomenų



21 pav. Ryšiai tarp poliso ir jo finansinius duomenis saugančių duomenų bazės lentelių

Finansinius duomenis saugančios lentelės prie poliso duomenų dinamiškai siejamos identifikaciniu poliso numeriu, bei draudimo lygį nusakančio įrašo numeriu. Finansinė informacija siejama su atitinkama poliso versija. Pavyzdžiui, jei turime aktyvią poliso versiją kuriai nebuvo sukurta jokių papildomų pakeitimo versijų, finansinę poliso informaciją galime nesudėtingai pasiekti, bet jei polisas buvo pakeistas, finansinę informaciją surasti tampa ganėtinai sudėtinga, dėl skirtingoms poliso versijoms kuriamų atskirų poliso vertės duomenų. Todėl tik su paskutine poliso versija susijusi finansinė informacija turėtų būti perkeliama [6].

Finansinius duomenis kaupiančių lentelių rinkinys turi būti pildomas atitinkama seka:

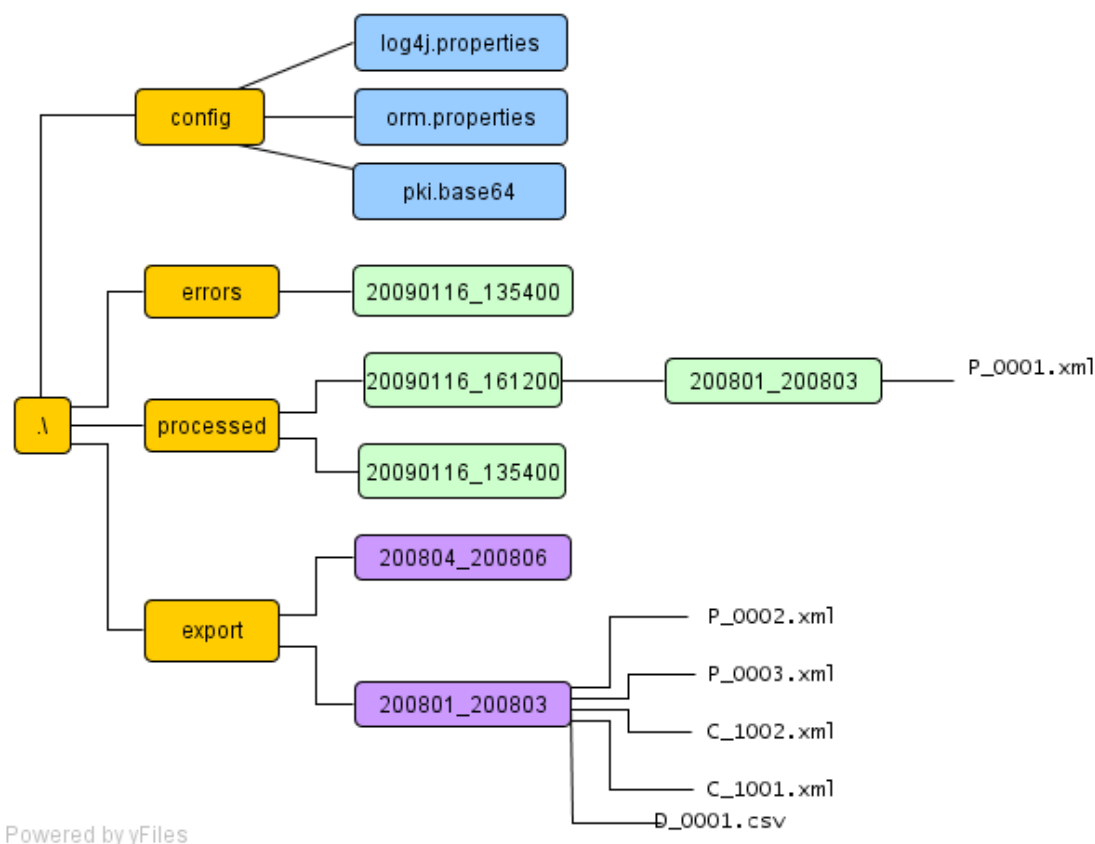
- Poliso posistemės lentelės
- Finansiniai poliso duomenys ir mokėjimų informacija
- Mokėjimų grafiko duomenys
- Mokėjimų transakcijų duomenys

7.5. Katalogų struktūra

Duomenų perkėlimo procesui turi būti sukuriami atitinkami katalogų struktūra. Katalogas, kuriame bus talpinamos perkėlimui skirtos XML tipo rinkmenos, turi turėti atitinkamus pakatalogius:

- Export – jame turėtų būti patalpintos perkėlimui skirtos XML rinkmenos.
- Processed – čia bus talpinamos sėkmingai apdorotos XML rinkmenos.

- Errors – šioje direktorijoje bus talpinamos XML rinkmenos, kurių apdorojimo metu buvo rasta klaidų ir rinkmenas būtina keisti.



22 pav. Duomenų perkėlimo įrankio katalogų ir rinkmenų struktūra

Yra sutarta, jog *Export* kataloge bus katalogai, kurie nurodys metų ketvirčius, pavyzdžiui draudimo polisai, kurių galiojimo data yra nuo 2007-01-01 iki 2007-03-31 turėtų būti kataloge, pavadintame 200701_200703. Tokių katalogų vardų šablonas yra MMMMmm (kur M – metams skirtų simbolių vieta, o m – mėnesiams), nurodant ketvirčio pradžią ir pabaigą, pvz. 200701_200703.

Tikimasi, kad dėl tokio rinkmenų skirstymo sumažės rinkmenų skaičius vienoje direktorijoje ir bus sumažinta tikimybė turėti rinkmenomis perpildytą katalogą.

Katalogai *Processed* ir *Errors* turės duomenų perkėlimo proceso kuriamus pakatalogius, kurių pavadinimas sudarytas iš perkėlimo vykdymo laiko ir proceso identifikacinio numerio. Tokio katalogo struktūra bus tokia pati kaip *Export* kataloge. Atitinkamai pakatalogiai, rasti kataloge *Export* turi būti kuriami tik vienoje iš *Processed* ar *Errors* direktorijų, pagal XML rinkmenos apdorojimo rezultatą [6].

Katalogų ir rinkmenų pavadinimų konvencija

- Katalogas, kuriame sukeltos draudimo polisų duomenų, kurių galiojimo data yra nuo 2007-01-01 iki 2007-03-31 turėtų bus kataloge, pavadintame 200701_200703. Toks pavadinimas yra sudarytas pagal katalogų pavadinimo šabloną - MMMMmm (kur M – metams skirtų simbolių vieta, o m – mėnesiams), nurodant ketvirčio pradžią ir pabaigą, pvz. 200701_200703.
- XML tipo rinkmena, kurioje saugomi duomenys apie klientą, bus vadinami, pavadinimo pradžioje pridedant „C_“, po kurio seka kliento identifikacinis numeris ir .xml galūnė. Pavyzdys: Kliento, kurio numeris – 10001, duomenys bus perkeliami rinkmenoje: C_10001.xml
- XML tipo rinkmena, kurioje saugomi draudimo poliso duomenys, bei poliso finansinė informacija bus pavadinta atitinkamai – pavadinimas prasidės P_, toliau seks poliso identifikacinis numeris ir .xml galūnė. Pavyzdys: Poliso, kurio numeris 100, duomenis bus patalpinti rinkmenoje: P_100.XML
- Rinkmenos, kuriose patalpintas sąrašas polisų, kurie turėtų būti ištrinti iš IPB sistemos, privalo pradžioje pavadinimo turėti – „D_“, po kurio seks bet koks, tam tikrai dienai unikalus skaičius ir .csv galūnė. Pavyzdys: D_0001.csv
- Katalogas, kuris identifikuoja perkėlimo proceso vykdymo identifikacinį numerį – MMMMmmDD_VVmmss. Pirma dalis yra pilna data, sudaryta iš metų, mėnesio ir dienos, o antra dalis sudaryta iš valandų, minučių ir sekundžių. Pavyzdys: 20070101_192012
- Duomenų perkėlimui *Export* direktorijoje turi būti sudėtos tik visiškai apdorojimui paruoštos rinkmenos. Duomenų perkėlimo metu rinkmenos negali būti perkelinėjamos į *Export* direktoriją.
- Duomenų perkėlimo proceso metu, apdorojama rinkmena bus pervadinta, pakeičiant galūnę į .tmp, o baigus apdorojimą ir prieš perkeliant rinkmeną į atitinkamą direktoriją, bus pervadintas grąžinant galūnę .xml.
- Šifravimo raktas, naudojamas iššifruoti mokėjimų duomenis, tiksliau kreditinės kortelės sąskaitos numerį, saugomas rinkmenoje config/pki.base64.
- Rinkmenos, esančios direktorijoje *config* ir turinčios galūnes .properties yra duomenų perkėlimo įrankio nustatymų rinkmenos.

7.6. Duomenų perkėlimo procesas

Duomenų perkėlimo proceso metu, periodiškai tikrinamas *Export* katalogas, ieškant XML rinkmenų. Kai randamos XML tipo rinkmenos, vykdomi sekantys veiksmai:

- Atrenkamos rinkmenos, kuriose patalpinti klientų duomenys, kadangi jos turi būti apdorojamos pirmiausiai
- Atrenkamos rinkmenos, kuriose yra draudimo polisų informacija, jos bus apdorojamos po to.
- Atrenkamos CSV tipo rinkmenos, kuriose surašyti numeriai polisų, kurie turės būti ištrinti. Pirmiausiai ištrinamas polisas, po to jo finansinė informacija, tačiau duomenys apie klientą ir sąskaitą išlieka nepakitę.

Abi pusės sutaria, kad:

- Duomenų perkėlimui naudojamas ACORD standartas, taikomas XML rinkmenoms
- Klientas eksportuos visus reikalingus duomenis į XML rinkmenas, kurios bus saugojamos rinkmenų sistemoje.
- Duomenų perkėlimo proceso metu bus tikrinama tam tikra direktorija, ieškant XML rinkmenų, kurias radus, bus vykdomas duomenų perkėlimas į IPB sistemą.

Duomenų perkėlimo metu yra kuriamos atskiros lentelės susieti RPL lentelių įrašus su IPB duomenų bazės lentelių įrašais. Tokios sąsajos sudaromos šiems įrašams:

- RPL polisų numerių susiejimui su IPB polisų numeriais (yra suderinta, kad jie bus vienodi)
- RPL draudimo lygio duomenims susieti su IPB draudimo lygio duomenimis
- RPL draudžiamo objekto duomenims susieti su IPB draudžiamo objekto duomenimis
- RPL poliso finansinės informacijos duomenims susieti su IPB poliso finansinės informacijos duomenimis

Tokių sąsajų sudarymo tikslas yra pažymėti perkėlimo metu sukurtus įrašus, tam kad pakartotinio perkėlimo metu nebūtų kuriami nauji, besidubliuojantys įrašai [6].

7.7. ACORD standarto naudojimas RPL duomenims

Visi perkeliami duomenys bus talpinami XML rinkmenoje. Tai bus daroma atitinkamai paskirstant:

- Duomenys apie klientą bus talpinami XML medyje - ACORD/CommonSvcRs/AccountInqRs/PartyInqInfo

- Duomenys apie polisą ir jo finansinę informaciją bus talpinami - ACORD/InsuranceSvcRq/PersAutoPolicyQuoteInqRq

Paminėtos XML struktūros dalys yra standartiniai ACORD schemas elementai, kurie ir bus naudojami duomenų perkėlimo procese. Detalesniam XML struktūros elementų reikšmių priskyrimui yra sukurtas atskiras dokumentas, nurodantis tikslų lauką IPB sistemoje ir jam priskirtą XML elementą. Šio dokumento ištrauka pateikta 23 pav.

7.8. Duomenų perkėlimo įrankio naudojimas

Duomenų perkėlimo įrankis turi du veikimo būdus:

- Duomenų perkėlimas, kai kuriami tik nauji polisai
- Duomenų perkėlimas, kai taisomi polisai, kurie anksčiau nebuvo perkelti dėl surastų klaidų ar neatitikimų

Duomenų perkėlimo įrankis yra programa, kurią galima paleisti bet kuriam kompiuteryje. Visi reikalingi nustatymai turėtų būti patalpinti kataloge *config*. Programa paleidžiama iš komandinės eilutės. Programos paleidimo metu pirmas komandinėje eilutėje įvestas parametras yra pagrindinio perkeliama duomenų katalogo adresas, o antras parametras naudojamas pasirinkti veikimo būdą.

7.9. Programos proceso registravimas

Duomenų perkėlimo procesas yra registruojamas į log tipo rinkmeną. Naudojama standartinė Java kalbos Log4j biblioteka, skirta programų aktyvumo registravimui į pultą ir į tekstinę rinkmeną.

7.10. Perkeliama duomenų tikrinimas

Perkeliami duomenys bus tikrinami dviem etapais. Kadangi IPB sistemoje naudojama objektų tikrinimo pagal iš anksto nustatytas taisykles sistema – BLS, tai ji bus naudojama ir duomenų perkėlimo metu. BLS posistemei turi būti pateiktas duomenimis užpildytas objektas, kas ne visada bus įmanoma duomenų perkėlimo metu, kadangi galima tokia situacija, kai pati XML rinkmena yra nekorektiška, nes trūksta būtinų duomenų. Tam tikslui buvo sukurta speciali XML struktūros tikrinimo funkcija, kuri ir bus naudojama tam, kad įsitikinti jog visi reikalingi duomenys yra pateikti XML rinkmenoje, bei yra korektiški.

IPB sistemoje, kuriant draudimo polisą, tam tikrų laukų reikšmės yra parenkamos iš galimų reikšmių sąrašo. Tam kad atitikti šį reikalavimą, duomenys XML rinkmenoje bus tikrinami lyginant jų reikšmes, su galimų reikšmių rinkiniu. Kai reikšmės nėra tarp galimų pasirinkti, apdorojama

XML rinkmena yra atmetama, kaip netinkama perkėlimui. Registruojama klaida, kurioje nurodoma XML struktūros vieta ir reikšmė, kuri turėtų būti pataisyta. Tokią XML rinkmeną bus galima dar kartą perkelti, tik tą reikšmę pataisius, kitu atveju rinkmena bus vėl atmesta.

Kadangi duomenų perkėlimą planuojama vykdyti etapais, galimi atvejai, kai poliso duomenis bus norima atnaujinti po pirmojo perkėlimo iteracijos praėjus mėnesiui ar dar daugiau. Jei polisas buvo perkeltas, antrą kartą bandant jį perkelti, programa patikrins ar jis nebuvo pakeistas IPB sistemoje. Atvejais, kai polisas pakeičiamas IPB sistemoje, labai sudėtinga nustatyti kokio duomenys buvo pakeisti, ir kiek kokių naujų poliso versijų galėjo ir buvo sukurta. Dėl šios priežasties, duomenų perkėlimas polisui, kuris buvo pakeistas IPB sistemoje, bus neįmanomas [6].

7.11. Perkeltų polisų papildomas žymėjimas IPB sistemoje

Tam kad aiškiai atskirti perkeltus polius, atidarius polisą IPB sistemoje turi būti matoma atitinkama žymė: „Perkeltas iš RPL“. Tokia žymė turi būti sukuriamą kiekvienam sėkmingai perkeltam polisui.

Perkelti polisai turi specifinės informacijos, susijusios su draudimo brokeriais, kadangi polisai galėjo būti pirkti dabar jau negaliojančių akcijų metu, taip pat iš brokerių įmonių kurios nutraukė veiklą ir t.t. Dėl šios priežasties IPB sistemoje bus pridėti nauji brokeriai, skirti perkeltiems polisams. Taip pat tokiais atvejais kai akcija jau yra pasibaigusi, tačiau jos kodas yra panaudotas polise, ir jis yra perkeliamas iš RPL, duomenų perkėlimo programa neturėtų tikrinti ar akcija dar galiojanti, bet duomenų bazėje patikrinti, jog ji kažkada egzistavo ir leisti perkelti tokį polisą. Tokie polisai bus atnaujinti vėliau, brokeriui susisiekus su klientu.

Perkeliamiems polisams kreditinės kortelės duomenys bus šifruojami, pagal AES algoritmą. Atvejais kai neįmanoma iššifruoti kreditinės kortelės numerio, bandant polisą perkelti į IPB, polisas bus atmestas, registruojant klaidą. AES algoritmo šifravimo ir dešifravimo parametrai yra šifravimo raktas ir inicializavimo vektorius. Algoritmo parametrai pateikiami base64 tipo rinkmenoje, patalpintoje *config* direktorijoje, *pki.base64* rinkmenoje.

7.12. Perkeltų polisų naudojimas IPB sistemoje

Kadangi tik paskutinė aktyvi poliso versija perkeliama į IPB sistemą, perkeltų polisų naudojimas ir keitimas turi tam tikros specifikos:

- Naujos perkeltos poliso versijos gali būti kuriamos tik į ateitį, po perkeltos poliso versijos galiojimo pradžios. Pavyzdys: Jei perkeltos poliso versijos galiojimo pradžia yra 2009-03-01, tai IPB sistemoje kuriamų transakcijų galiojimo data negali būti prieš šią datą. Poliso

versijos galiojimo data turi būti pateikiama XML struktūroje, //TransactionEffectiveDt elemente. [6]

7.13. Duomenų eksportavimas iš šaltinio

Pirmoji duomenų perkėlimo stadija yra duomenų eksportavimas iš šaltinio. Ši dalis vykdoma kliento, todėl ypatingai daug informacijos apie ją pateikti nėra tikslo. Eksportavimui naudojama suderinta XSD schema, taip pat XLS tipo rinkmena, pagal kurią lentelės įrašas išvedamas į XML elementą, pagal adresą nusakytą tame XLS dokumente.

Kaip jau buvo aptarta anksčiau, kliento ir polisų duomenys išvedami į skirtingas XML tipo rinkmenas, kurios taip pat apdorojamos atitinkama tvarka. Pirmiausiai apdorojamos rinkmenos, su kliento duomenimis, o po to – poliso duomenis saugančios XML rinkmenos [7].

Required(M) or Optional(O)	IPB UI			Accord reference	IPB Model mapping	Lookup and enums	Relation to CBS	Notes	notes
	Field Name	Type	Tab						
M	Cover Type			//HomeLineBusiness/Dwell/com.exigen.CoverType		CoverType			
M	(Location ref id)			//Dwell @LocationRef					
M	(country)			//Location/Addr[AddrTypeCd=PropertyAddress]/Country				Default to AU	
M	Post Code			//Location/Addr[AddrTypeCd=PropertyAddress]/PostalCode					
M	Suburb			//Location/Addr[AddrTypeCd=PropertyAddress]/City					
M	State			//Location/Addr[AddrTypeCd=PropertyAddress]/StateProvCd					
M	Street Name			//Location/Addr[AddrTypeCd=PropertyAddress]/DetailAddr/StreetName					
M	Street Type			//Location/Addr[AddrTypeCd=PropertyAddress]/DetailAddr/StreetType					
M	Street #			//Location/Addr[AddrTypeCd=PropertyAddress]/DetailAddr/StreetNumber					
O	Unit #			//Location/Addr[AddrTypeCd=PropertyAddress]/DetailAddr/UnitNumber					
O	Other Address Line			//Location/Addr[AddrTypeCd=PropertyAddress]/Addr4 - Other Line					
M	(Rate Effective Date)	always hidden		//PersPolicy/RateEffectiveDt					
M	Is it a home you own with just you and your family living there?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='OWN_HOME']/YesNoCd					
M	Occupancy Type			//HomeLineBusiness/Dwell/DwellOccupancy/DwellUseCd		OccupancyType			
C	Will the home be unoccupied for more than 60 consecutive days?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='UNOCCUPIED_OWN']/YesNoCd					
C	Are more than 3 unrelated people living in the home?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='UNRELATED_INF']/YesNoCd					
C	Is the property sublet?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='SUBLET']/YesNoCd					
M	Is the home well maintained and in good condition?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='GOOD_CONDITION']/YesNoCd					
M	Is the home on less than 10 acres?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='LT_10 Acres']/YesNoCd					
C	Is the property heritage listed?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='HERITAGE_LISTED']/YesNoCd					
M	What is the building type?			//Dwell/QuestionAnswer[CONSTRUCTION_CD]/YesNoCd		BuildingType			
C	Is the home under strata, body corporate, community or company title?			//HomeLineBusiness/Dwell/QuestionsAswer[QuestionCd='COMPANY_OWNED']/YesNoCd					
M	Is that a brick veneer home?			//HomeLineBusiness/Dwell/Construction/InsurerConstructionClassCd				If value = "Brick/Veneer" then	
C	What is the construction type?			//HomeLineBusiness/Dwell/Construction/InsurerConstructionClassCd		ConstructionType			
M	What is the roof type?			//HomeLineBusiness/Dwell/Construction/RoofMaterial/RoofMaterialCd		RoofMaterialType			
M	Year of construction			//HomeLineBusiness/Dwell/Construction/YearBuilt					

23 pav. XML elementų ir IPB sistemos laukų priskyrimo dokumento ištrauka

7.14. Duomenų perkėlimo įrankio nuostatos ir paruošimas darbui

Būtinios sąlygos:

- Kompiuteryje, kuriame bus naudojamas duomenų perkėlimo įrankis, turi būti įrašyta bent 1.5 versijos JAVA JRE aplinka ir nustatytas JAVA_HOME sisteminis kintamasis.

Visi programos nustatymai yra kataloge: c:\MigrationApp\classes" (katalogas c:\MigrationApp gali būti kitas, tai priklauso kuriame kataloge buvo išskleistas duomenų perkėlimo įrankio archyvas).

Viso yra trys nustatymų rinkmenos:

- db.properties – IPB duomenų bazės nustatymai
- migrationDataLoader.properties – duomenų perkėlimo įrankio bendri nustatymai
- pki.base64 – šifravimo ir iššifravimo parametrai

Prieš paleidžiant duomenų perkėlimo įrankį, turi būti sukurti trys katalogai: *Export*, *Errors* ir *Processed*. Katalogo adresas, kuriame sukurti šie trys katalogai yra nustatomas paleidimo metu.

Duomenų perkėlimo įrankis apdorojimą gali vykdyti keletu gijų, kurių skaičius yra nustatytas *migrationDataLoader.properties* rinkmenoje. Toks apdorojimas žymiai pagreitina programos darbą, kadangi darbas keletu gijų vyksta taip pat kaip paraleliai paleidus keletą duomenų perkėlimo programų [3].

7.15. Duomenų perkėlimo įrankio naudojimas

Komandinėje eilutėje nueiname į tą vietą, kur įrašyta duomenų perkėlimo programa ir egzistuoja rinkmena pavadinimu „import.bat“. Į komandinę eilutę įvedame tokią komandą:

```
import.bat <XML rinkmenų katalogo adresas> [<programos veikimo būdas>]
```

Kur:

<XML rinkmenų katalogo adresas> - katalogas, kuriame sukurti anksčiau minėti trys katalogai ir įdėtos perkelti paruoštos XML rinkmenos;

<programos veikimo būdas> - programos veikimo būdas, t.y. ar bus bandoma dar kartą perkelti, jau anksčiau perkeltas XML rinkmenas, ar bus perkeliama tik nauji duomenys. Jei bus leidžiamas perrašymas, rašoma Bulio reikšmė – „true“, jei neleidžiamas – „false“.

Komandos pavyzdys:

```
import.bat C:\dm\ true
```

Paleidus duomenų perkėlimo įrankį, pradedamas duomenų perkėlimo procesas. Pirmiausiai patikrinamas *Export* katalogas, jei jame randama XML rinkmenų, atskirai išskirstomos kliento ir poliso duomenų rinkmenos. Pirmiausiai apdorojamos kliento duomenis saugančios XML tipo rinkmenos [3].

Apdorojant kliento duomenų XML rinkmeną, pirmiausiai nuskaitoma rinkmena ir patikrinama jame saugoma informacija ir jos atitinkamumas IPB sistemai. Jeigu duomenys yra nekorektiški, registruojama klaida, kuri išvedama į ekraną. Toliau XML rinkmena yra perkeliama į *errors* katalogą ir pereinama prie kitos rinkmenos apdorojimo.

```
C:\WINDOWS\system32\cmd.exe - import.bat C:/dm6_HC true
C:\SUN\HRD-TRUNK\migration\iph-hrd-migration-app\target\classes>import.bat C:/dm6_HC true
[JRokit] Local management server started.
13:30:54.241 INFO [tools.LoadMigrationData]: Usage: <Directory structure root> [Override=false]
13:30:54.257 INFO [tools.LoadMigrationData]: Directory structure root: C:/dm6_HC
13:30:54.257 INFO [tools.LoadMigrationData]: Override indicator: true
13:30:54.272 DEBUG [tools.LoadMigrationData]: Initiate application context started
13:31:02.866 INFO [bam.DefaultActivityDefinitionsRegistry]: Registering 97 activity definitions
13:31:02.960 INFO [spring.BamDispatcherFactory]: Adding event processor activityPersisterIPB of type com.exigen.ipb.bam.ActivityPersister$$EnhancerBy
CGLIB$57d37525f
13:31:12.991 DEBUG [tools.LoadMigrationData]: Initiate application context finished
13:31:14.132 INFO [impl.DataMigrationServiceImpl]: 11 files found to process in 'export' directory of path provided: C:/dm6_HC
13:31:14.148 INFO [impl.MigrationThreadRunnableImpl]: [INFO] Thread-5 Starting import of file in: C:/dm6_HC/export/P_H_00000005.xml
13:31:15.601 ERROR [impl.PolicyAdditionalValidationServiceImpl]: Value should be present: ACORD/InsuranceSvcRq[1]/HomePolicyQuoteInqRq[1]/HomeLineBusi
ness[1]/Dwell[1]/QuestionsAnswer[QuestionCd="SPECIFIED ITEMS"/]/YesNoCd
13:31:15.616 INFO [impl.MigrationThreadRunnableImpl]: TaskExecutor-1 MigrationProcessingException: Provided XML file is not valid.
13:31:15.616 ERROR [impl.MigrationThreadRunnableImpl]: File: P_H_00000005 was not imported. Took: 1421 ms
13:31:15.616 DEBUG [impl.MigrationThreadRunnableImpl]: File move started.
13:31:15.632 DEBUG [impl.MigrationThreadRunnableImpl]: File: P_H_00000005.xml successfully moved to : C:/dm6_HC/errors/20100522_133114/export
13:31:15.632 INFO [impl.MigrationThreadRunnableImpl]: [INFO] Thread-5 Ending import of file: P_H_00000005, took: 1421 ms
```

24 pav. Klaidos XML rinkmenoje registravimas ir išvedimas į pultą

Jei patikrinimas praeina sėkmingai duomenų bazėje ieškoma kliento su tokiu pat numeriu. Jei įrašas apie tokį klientą jau egzistuoja, patikrinama ar pasirinktas programos veikimo būdas leidžia perrašyti anksčiau perkeltus kliento duomenis. Jei leidžia – duomenų perkėlimas vykdomas toliau, iš XML rinkmenoje esančių duomenų sudarant lentelių įrašus, jei neleidžia – duomenų perkėlimas nutraukiamas. Sukūrus atitinkamus įrašus lentelėms duomenų bazėje, įrašai yra išsaugomi, tuo pačiu užpildant atitinkamą lentelę duomenimis apie sėkmingą duomenų perkėlimo proceso baigimą. XML rinkmena perkeliama į *Processed* katalogą ir pereinama prie sekančios XML rinkmenos apdorojimo.

```
C:\WINDOWS\system32\cmd.exe - import.bat C:/dm6_HC true
C:\SUN\HRD-TRUNK\migration\iph-hrd-migration-app\target\classes>import.bat C:/dm6_HC true
[JRokit] Local management server started.
13:27:22.942 INFO [tools.LoadMigrationData]: Usage: <Directory structure root> [Override=false]
13:27:22.957 INFO [tools.LoadMigrationData]: Directory structure root: C:/dm6_HC
13:27:22.957 INFO [tools.LoadMigrationData]: Override indicator: true
13:27:22.973 DEBUG [tools.LoadMigrationData]: Initiate application context started
13:27:31.629 INFO [bam.DefaultActivityDefinitionsRegistry]: Registering 97 activity definitions
13:27:31.723 INFO [spring.BamDispatcherFactory]: Adding event processor activityPersisterIPB of type com.exigen.ipb.bam.ActivityPersister$$EnhancerBy
CGLIB$56ehfe77
13:27:41.739 DEBUG [tools.LoadMigrationData]: Initiate application context finished
13:27:42.895 INFO [impl.DataMigrationServiceImpl]: 11 files found to process in 'export' directory of path provided: C:/dm6_HC
13:27:42.911 INFO [impl.MigrationThreadRunnableImpl]: [INFO] Thread-5 Starting import of file in: C:/dm6_HC/export/P_H_00000005.xml
13:27:44.520 DEBUG [impl.PolicyDataMigrationServiceImpl]: [INFO] Mapping started.
13:28:09.395 DEBUG [impl.PolicyDataMigrationServiceImpl]: [INFO] Mapping completed.
13:28:30.317 DEBUG [impl.PolicyDataMigrationServiceImpl]: [INFO] Validation started.
13:28:30.317 DEBUG [impl.PolicyDataMigrationServiceImpl]: [INFO] Validation completed.
13:28:30.380 DEBUG [impl.PolicyDataMigrationServiceImpl]: [INFO] Mapped policy is correct, successfully mapped and will be imported.
13:28:30.942 DEBUG [impl.BillingDataMigrationServiceImpl]: Policy #H_00000005 found. Billing account will be created.
13:28:31.083 DEBUG [impl.BillingDataMigrationServiceImpl]: ****Billing: H_00000005 created.
13:29:27.428 INFO [impl.BillingDataMigrationServiceImpl]: Billing import took: 56532
13:29:29.662 INFO [impl.MigrationThreadRunnableImpl]: File: P_H_00000005 for Policy: #H_00000005 was successfully imported
13:29:29.678 DEBUG [impl.MigrationThreadRunnableImpl]: File move started.
13:29:29.678 DEBUG [impl.MigrationThreadRunnableImpl]: File: P_H_00000005.xml successfully moved to : C:/dm6_HC/processed/20100522_132742/export
13:29:29.678 INFO [impl.MigrationThreadRunnableImpl]: [INFO] Thread-5 Ending import of file: P_H_00000005, took: 106689 ms
```

24 pav. Sėkmingo XML rinkmenos apdorojimo vykdymo registro išvedimas į pultą

Išvados

- Apžvelgtos dažniausiai taikomos duomenų perkėlimo metodikos. Pasirinkta geriausiai problematiką atitinkanti duomenų perkėlimo metodika – perkėlimas dalimis. Situacija skiriasi nuo standartinių situacijų:
 1. Duomenų eksportas iš šaltinio informacinės sistemos atliekamas kliento programinės įrangos. Dėl šios priežasties pranyksta poreikis išjungti senąją sistemą.
 2. Duomenų perkėlimas gali būti vykdomas apdorojant tiek labai mažus duomenų kiekius, tiek didelius, kadangi tai leidžia perkeliama duomenų specifika. Duomenų perkėlimo įrankis atskirai apdoroja kiekvieną duomenų rinkmeną, todėl galima duomenų perkėlimą atlikti perkeliama duomenų kiekį sumažinant iki vieno poliso.
- Išnagrinėtos dažniausiai duomenų perkėlimo metu kylančios problemos. Pateiktos jų priežastys, bei bendros rekomendacijos, kaip galima jų išvengti. Daugiausiai problemų sukelia netinkamas duomenų perkėlimo projekto sudėtingumo įvertinimas.
- Sukurta duomenų perkėlimo specifikacija. Joje aprašyta, kaip veiks duomenų perkėlimo įrankis, specifiniai kliento pageidavimai ir kokie duomenys bus perkeliama.
- Sukurtas duomenų perkėlimo įrankis. Duomenims perduoti naudojama XML rinkmenos struktūra. Duomenų perkėlimo metu iš XML rinkmenos nuskaitomi duomenys, patikrinamas jų atitikimas IPB sistemai ir informacija įrašoma į duomenų bazės lenteles. Duomenų perkėlimo įrankis leidžia duomenis perkelti keletą kartų, nekuriant besidubliuojančių įrašų. Antrą kartą perkeliama tuos pačius duomenis programa suranda reikalingus įrašus duomenų bazėje ir jei duomenų patikrinimo rezultatas sėkmingas, atnaujinta informacija išsaugoma duomenų bazėje.
- Įgyvendintas duomenų perkėlimui pateikiamų duomenų tikrinimas. Duomenų korektiškumo tikrinimas bus vykdomas duomenų perkėlimo metu, kadangi tai geriausiai atitinka užduoties specifiką. Sukurta dviejų pakopų tikrinimo sistema:
 1. Tikrinama, ar XML rinkmenoje yra visi reikalingi duomenys ir ar jie atitinka IPB sistemos reikalavimus.
 2. Tikrinamas XML rinkmenoje esančių duomenų atitikimas verslo logikai, naudojant IPB posistemį – BLS.
- Duomenų perkėlimas būtų vykęs sklandžiau, jei bendravimas su klientais būtų kėlęs mažiau problemų. Daugiausiai problemų kilo dėl to, jog užsakovo atstovai eksportuodami duomenis, naudojosi savo intuicija, užuot paklausę, kokių duomenų yra tikimasi iš šaltinio. Todėl gana daug problemų iškilo klientams pateikus pirmuosius eksportuotų duomenų

pavyzdžius. Klaidos dažnai kartojosi, kadangi nebuvo tinkamai reaguojama į aprašytas problemas, duomenų perkėlimo procesas viršijo planuotą trukmę. Tai įvyko labiausiai dėl užsakovų kaltės, duomenų perkėlimo tvarkaraštis be didesnių konfliktų buvo pakoreguotas ir šiuo metu procesas beveik pabaigiamas.

Literatūros sąrašas

1. Böhme, T., Rahm E. Evaluation of XML Data Management Systems with XMach. Mokslinis straipsnis. Vokietija, 2008. [žiūrėta 2010-05-06]. Prieiga per internetą: <http://www.springerlink.com/content/1nvgeya68k9uqw6w/fulltext.pdf>
2. EMC Corportation Data Migration Considerations: A Customer Engineering Residency. Bedford, Usa, 2007, p. 4 – 8.
3. Exigen Services Administration guide. Vilnius, 2009, p. 45 – 48.
4. Exigen Services Customer Use Case 02.17. Vilnius, 2010, p. 20 – 21.
5. Exigen Services Data Loader XSD scheme. Vilnius, 2009.
6. Exigen Services Exigen Data Load Technical Document. Vilnius, 2009, p. 4 – 18.
7. Exigen Services HC product mappings 0.3. Vilnius, 2010, p. 2.
8. Howard, P. Data Migration. Towcester, United Kingdom, 2007. 2-9 p.
9. Howard, P., Potter, C. Data Migration in the Global 2000. Towcester, United Kingdom, 2007, p. 1 – 8.
10. O'Reilly XML Schema. JAV, 2005, 396 p.
11. Rasulovas, T. Duomenų perkeliavimo tarp verslo valdymo sistemų tyrimas. Vilnius, VGTU, 2009, p. 7 – 19.

Priedai

1 priedas. Kliento duomenis talpinanti XML rinkmena

```
<ACORD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns=
"http://www.ACORD.org/standards/PC_Surety/ACORD1.12.0/xml/">
  <CommonSvcRs xmlns="">
    <RqUID>12395</RqUID>
    <AccountInqRs>
      <PartyInqInfo>
        <InsuredOrPrincipal>
          <GeneralPartyInfo>
            <NameInfo>
              <CommlName>
                <CommercialName>Mrs Client Client</CommercialName>
              </CommlName>
              <PersonName>
                <Surname>Client</Surname>
                <GivenName>Client</GivenName>
                <TitlePrefix>Mrs</TitlePrefix>
              </PersonName>
            </NameInfo>
            <Addr>
              <AddrTypeCd>InsuredsAddress</AddrTypeCd>
              <DetailAddr>
                <StreetName>Street</StreetName>
                <StreetTypeCd>ST</StreetTypeCd>
                <StreetNumber>7</StreetNumber>
              </DetailAddr>
              <City>City</City>
              <StateProvCd>FSD</StateProvCd>
              <PostalCode>1111</PostalCode>
              <Country>LT</Country>
            </Addr>
            <Communications>
              <PhoneInfo>
                <PhoneTypeCd>homePhone</PhoneTypeCd>
                <CommunicationUseCd>Preferred</CommunicationUseCd>
                <PhoneNumber>0296740000</PhoneNumber>
              </PhoneInfo>
              <PhoneInfo>
                <PhoneTypeCd>mobilePhone</PhoneTypeCd>
                <PhoneNumber>0404868123</PhoneNumber>
              </PhoneInfo>
              <EmailInfo>
                <EmailAddr>Client@lt.lt</EmailAddr>
              </EmailInfo>
            </Communications>
          </GeneralPartyInfo>
        </InsuredOrPrincipalInfo>
        <PersonInfo>
          <GenderCd>F</GenderCd>
          <BirthDt>1963-02-05</BirthDt>
        </PersonInfo>
      </InsuredOrPrincipalInfo>
      <com.exigen_MarketingInfo>
        <com.exigen_BrandCd>Real Insurance</com.exigen_BrandCd>
        <com.exigen_CampaignCd>Radio</com.exigen_CampaignCd>
      </com.exigen_MarketingInfo>
    </PartyInqInfo>
  </AccountInqRs>
</CommonSvcRs>
</ACORD>
```

```

    <com.exigen_SubCampaignCd>TodayPM</com.exigen_SubCampaignCd>
    <com.exigen_ReferralCd>REAL</com.exigen_ReferralCd>
  </com.exigen_MarketingInfo>
  <com.exigen_CustomerInfo>
    <com.exigen_CustomerNo>1309591</com.exigen_CustomerNo>
    <com.exigen_AccessTrackInfo>
      <com.exigen_CreatedOn>2006-11-15T00:00:00.000000+11:00</com.exigen_CreatedOn>
      <com.exigen_UpdatedOn>2010-01-21T14:37:12.587000+11:00</com.exigen_UpdatedOn>
      <com.exigen_CreatedBy>caty</com.exigen_CreatedBy>
      <com.exigen_UpdatedBy>RPLDataConversion</com.exigen_UpdatedBy>
    </com.exigen_AccessTrackInfo>
    <CompanyProductCd>TRAVEL</CompanyProductCd>
    <com.exigen_CustomerType>INDV</com.exigen_CustomerType>
    <com.exigen_UserNote>
      <com.exigen_CreatedOn>2009-12-21T13:06:38.000000+11:00</com.exigen_CreatedOn>
      <com.exigen_PerformerId>RBonus</com.exigen_PerformerId>
      <com.exigen_UserText>Note: REAL Bonus Period: 16 Nov 2006 - 15 Nov 2009 REAL
Bonus Payment: $122.27</com.exigen_UserText>
    </com.exigen_UserNote>
    <com.exigen_CustomerAccountNo>9000000179378</com.exigen_CustomerAccountNo>
    <com.exigen_CustomerAccountName>Mrs Client Client</com.exigen_CustomerAccountName>
    <com.exigen_CustomerSince>2006-11-15T19:37:12.000000+11:00</com.exigen_CustomerSince>
  </com.exigen_CustomerInfo>
</InsuredOrPrincipal>
</PartyInqInfo>
</AccountInqRs>
</CommonSvcRs>
</ACORD>

```

2 priedas. 12-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“, įvykusios Vilniuje 2009 m. balandžio mėn. 13 d., mokslinis straipsnis.

„MOKSLAS – LIETUVOS ATEITIS“. 2009, Nr. 2 (2) Inžinerinė informatika

Mokslinių straipsnių rinkinys

ISSN 2029-2341 (print) / ISSN 2029-2252 (online)

WEB ORIENTUOTOS ARCHITEKTūros PRITAikomumas IR Palyginimas su SOA

Tomas Urbanavičius, Domas Pūras, Viktorija Briskmanaitė

*magistrantai,
Vilniaus Gedimino technikos universitetas,
tomciau@gmail.com, domaspuras@yahoo.com, kinzmaraul@yahoo.com*

Anotacija. Nagrinėjamas į paslaugas orientuotos architektūros (SOA) poaibis – Web orientuota architektūra (WOA). Trumpai apžvelgiamos WOA savybės ir naudojamos technologijos. Nagrinėjami esminiai skirtumai ir panašumai tarp WOA ir SOA.

Reikšminiai žodžiai: SOA, WOA, WEB, orientuota, architektūra.

Įvadas

Kadangi į paslaugas orientuota architektūra (angl. *Service Oriented Architecture* – SOA) tampa esmine paradigma lanksčiai taikomųjų programų kūrimo struktūrai, vystosi nauja žiniatinkliui orientuotų taikomųjų programų architektūra (angl. *WEB Oriented Architecture* – WOA), kuri SOA suteikia dar didesnę reikšmę. Tokiu būdu sukurtos aplikacijos suteikia servisams naujas galimybes kuriant sistemas, kurių sėkmė arba žlugimas priklauso nuo vykdymo, saugumo, patikimumo ir bendro taikomosios aplikacijos efektyvumo savybių.

Pastaruoju metu vis didėjant informacijos kiekiams, verslo sektorius vis dažniau pateikia turimą informaciją, viešam priėjimui internete. Keletą pastarųjų metų, dideli informacijos kiekiai vis aktyviau perduodami interneto ryšio kanalais, iš vieno verslo struktūrų, į kitas. Toks verslo struktūrų bendravimas yra sudėtingas, tačiau būtinas konkurencinėje kovoje. Nepaisant poreikio ir įdėtų pastangų (sukurtos SOA, EAI, ESB paradigmos), šios srities integravimo į bendras informacines sistemas, problemos yra vis dar sprendžiamos, ieškant esminių sprendimų. Trumpai tariant, integracija tarp sistemų neveikia tokia lygmenyje, kokiame turėtų veikti. Tačiau pastaruoju laikotarpiu, aktyviai besivystantis Žiniatinklio plėtojimas veda geresnio sistemos modelio link, Kuris turėtų padėti išspręsti adaptavimo ir efektyvumo problemas, pasitaikančias tradicinio SOA taikymo vykdytume {2}.

WOA

Daugelis iš mūsų galėtų pritarti, kad vis dar nėra sukurta lengvai prieinama ir panaudojama prieiga prie duomenų ir sistemų, kurios mums reikalingos ir naudojamos mūsų kasdieniame darbe. Darbuotojams daug laiko tenka praleisti kopijuojant ir perkeltant nukopijuotą medžiagą, dirbant su keletu taikomųjų aplikacijų. Duomenys perduodami suskaidant juos į paketus, o po to vėl importuojami, vykdant įvairias operacijas su IT sistemomis, milijonus kartų per dieną. Informacijos perkėlimas taip pat nevyksta visiškai efektyviai, kadangi tenka ją įvedinėti skirtingose vietose, įvairiais būdais. Visa tai vyksta nesavarankiškai, o žmogui atliekant įvairias užduotis. Nors ir SOA koncepcinės iniciatyvos juda teisinga linkme, tačiau įdėtos nemažos pastangos nedavė tikėtosi rezultato, tobulinant integracijos lygmenį ir gerinant sistemos veiklumą.

Visgi neviskas taip blogai kaip gali pasirodyti. Sistemų integravimo ir didelių informacijos kiekių perdavimo problemos iš esmės sprendžiamos pažangiausioje Žiniatinklio (angl. *World Wide Web*) erdvėje, dar kitaip vadinamoje Web 2.0. Iš esmės Web servisų atsiradimas sietinas su ankstyvąja Interneto istorija (apie 1999-2000 metus), nors pradžioje ir nebuvo sėkmingas sprendimas kaip ir SOAP (angl. *Simple Object Access Protocol*), WSDL (angl. *Web Services Description Language*), UDDI (angl. *Universal Description Discovery, and Integration*) ir WS-I

technologijos. To priežastys bus apžvelgtos tolesniame tekste.

Skirtingai nuo SOA, WOA vienareikšmiškai pateikia šiek tiek kitokį sprendimą, fundamentaliai labiau atsižvelgiant į kitus dalykus, tačiau visgi išlieka artimu SOA „giminaičiu“. Šis, daug labiau Tinklui orientuotas įgyvendinimas yra vadinamas „Tinklui Orientuota Architektūra“ iš esmės yra pačio Žiniatinklio didelio lankstumo ir po visu tuo besislepiančios architektūros tikslų ir idėjų esmė. WOA sudaro Žiniatinklio esminę koncepciją ir puikūs rezultatai, suteikę Žiniatinklui pačio didžiausio, laisvai naudojamo tinklo žemėje, kaip ir didžiausio egzistuojančio SOA „kūrinio“ vardą. {5}

Iš esmės WOA veikia SOA pagrindu, todėl SOA nuvertinti nevertėtų. Tolesniame tekste bus paaiškintos WOA idėjos, kurios kolkas nėra tiksliai standartizuotos. Kadangi WOA nėra tik standartas pritaikomas bet kokiose informacinių technologijų srityse, kuris buvo sukurtas, kuriant tiesioginio ryšio servisus, pritaikymui platesniam vartotojų ratui ir kurių sėkmė yra garantuota. Egzistuoja keletas pardavėjų, kurie „rikiuojasi“ už šio pragmatiško, efektyvaus ir gana populiarus pritaikymo, apribojant oficialų žinių kiekį ir palaikymą praktikoje, tiems kas nori naudoti šį naudingą ir paprastą sistemų sąveikavimo būdą.

Šiuo atveju reikėtų pabrėžti žodžio „paprastas“ vartojimą, kadangi tokia veikla negali būti paprasta, dėl joje vykstančių sudėtingų procesų ir veiklų. Iš esmės WOA nėra tiesiog paprastas Žiniatinklio servisų kūrimo būdas, tai dar ir ekosisteminis mąstymas, kuris kartais gali būti sunkiai suvokiamas nesistemiškai mastantiems, tiksliau sakant – ši alternatyva nėra tiksliai apibrėžta industriniu požiūriu. Nepaisant įvairių literatūrinių bandymų surasti WOA apibrėžimą, visiškai tikslus WOA kolkas dar nėra pateiktas. Toliau bus pateikti esminiai punktai tuo pačiu apibrėžiant skirtumus nuo SOA {3}.

Kas tai yra WOA?

- WOA informacija tinklu pateikiama resursų forma ir yra prieinama ir gali būti keičiama, laikantis universalaus resursų atpažinimo identifikatoriaus (angl. URI - *Universal Resource Identifier*) protokolo, dažniausiai HTTP.
- Kiekvienas resursų šaltinis gali būti pasiekiamas globaliu unikaliu adresu, žinomu kaip universalus resursų identifikatorius arba URI (angl. *Universal Resource Identifier*), veikiančiu pagal RFC 3986.{1}
- Resursai yra valdomi HTTP sintaksės pagalba (GET, PUT, POST, DELETE) naudojantis

technologija, vadinama, perdavimo būsenos nusakymu arba REST (angl. *Representational State Transfer*).

- Tinklo resursu naudojimas vykdomas išimtinai tik tinklo komponentais (iš esmės naudojantis, tik naršyklėmis ir Web serveriais).
- Prieiga prie resursų turi būti suskirstyta ir nereikalaujanti daugiau nei lokalsios tinklo informacijos.
- Tai komponentų atsakomybė suvokti pateikimą ir nusakyti būseną resursų kuriais manipuliuoja prieigos metu.
- WOA resursų servisų kontraktai yra numanomas, tai reprezentacinė dalis kuri gaunama.
- WOA resursams priklauso idėtiniai URI identifikatoriai, kurie sudaro didesnę tinklą iš tipinių būsenos elementų. (pavyzdžiui URL seka pagal kurią galima sužinoti resursų vietą).
- WOA įkūnyja Thomas Erl esminius SOA principus {3}, nors dažnai ir netikėtais būdais (toks kaip kontrakto turėjimas, nors ir tik numanomo).

Esminiai SOA ir WOA skirtumai

Vienas iš esminių būdų suprasti WOA yra pamatyti kaip tai skiriasi nuo SOA, nors ir egzistuoja tam tikras šių modelių persidengimas, integracijoje į žiniatinklį. Kadangi abu modeliai sąveikauja per HTTP, save apibūdinančiais duomenų formatais tokiais kaip XML, yra susiję su laisvų standartų naudojimu, ir gali būti naudojami kuriant sutartinio sudėtingumo sistemas, didžioji dalis jų paprastumo čia ir pasibaigia. Toliau pateikiami keletas kontrastų tarp šių modelių įgyvendinimų:

- SOA yra linkusi turėti nedidelį, tiksliai apibrėžtų galutinių adresų rinkinį, per kuriuos, daugelis duomenų tipų ir struktūrų gali būti perduodami. WOA yra linkusi turėti labai didelį ir neapibrėžtą skaičių galutinių adresų; kiekvienas kurių yra individualiai panaudojami resursai. Tam naudojami netgi ne galutiniai adresai, o URI identifikuojami galutiniai taškai kiekvienam resursų tipui ir struktūrai.
- Tradicinis SOA sukuria bendravimo lygmenį virš HTTP, naudojant SOAP ir pateikiant unikalius ir kartais pernelyg didelius apribojimus tinklo kūrėjui, kai WOA naudojasi HTTP ir susijusiais perdavimo mechanizmais, kaip idealiais aplikacijos lygmens abstrakcijos vienetais.

- SOA buvo sukurtas nuo viršaus iki apačios būti įrankiams patogiu, kai tuo tarpu WOA buvo sudaryta forma iš pačio Žiniatinklio natūraliai, ir turi geriausią palaikomumą paprasto, procedūrinio kodo ir XML perdavėjo.
- SOA naudoja Web servisų saugumo ir kitus modernizuotus saugumo standartus, kai tuo tarpu WOA teikiasi naudotis tik HTTPS.
- SOA turi kovoti su XML Schemas principais, servisų kontraktais, kai tuo tarpu WOA paprastai ignoruoja šią problemą, ir suteikia Web servisams galimybę savarankiškai pateikti pageidaujamų formatų tipus.
- Tradicinis SOA yra tiesiog grieždiškas naudoti naršyklėje kai tuo tarpu WOA yra ypatingai paprasta naudoti tiesiog bet kur {4}.

Išvados

1. Panagrinėtas WOA modelis, lyginant jį su SOA modeliu. Toks lyginimas yra ganėtinai informatyvus dėl išskiriamų esminių bruožų ir idėjų.

2. Dėl kolkas labai aktyvaus SOA naudojimo ir taikymo visose srityse WOA taikymas yra komplikotas. Tačiau alternatyvų ieškojimas, veda prie tinkamų sprendimų ir idėjų gerinant informacinių sistemų funkcionavimą.

3. WOA yra modelis tinkantis šiuolaikiniam informacinių technologijų pasauliui, dėl plataus panaudojimo galimybių, nesudėtingo resursų pasiekiamumo ir patogios sąveikos su juo.

Literatūra

- {1} Berners-Lee, T. Uniform Resource Identifier (URI): Generic Syntax. 2005, January. P. 61.
- {2} Casarez, V.; Cripe, B.; Weckerle, P.; Sini, J. Reshaping Your Business with Web 2.0, p.259.
- {3} Hinchcliffe, D. All About Service-Oriented. 2005.
- {4} Hinchcliffe, D. [What Is WOA? It's The Future of Service-Oriented Architecture](#) SOA. 2008.
- {5} O'Reilly Radar. Web 2.0 Principles and Best Practices, 2006. p. 101.

WEB ORIENTED ARCHTECTURE APPLICABILITY AND COMPARISION WITH SOA

T. Urbanavičius, D. Pūras, V. Briskmanaitė

Summary

Reviewing a subset of the System Oriented Architecture - Woa architecture. Woa Brief overview of features and technology used. Reviewing the fundamental differences and similarities between SOA and Woa.

3 priedas



UAB „Exigen Services Lietuva“
Ulonų g. 2, LT-08245 Vilnius
biuro tel.: +370 5 2754605, faks.: +370 5 2754604
biuro el. paštas: vilnius@exigenservices.com

Baigiamojo magistro darbo praktinis panaudojimas įmonėje

Baigiamasis magistro darbas „Duomenų perkėlimo tarp informacinių sistemų analizė ir įgyvendinimas“, kurį atliko Tomas Urbanavičius, buvo peržiūrėtas ir jame aprašyta duomenų perkėlimo programa yra naudojama įmonės klientų.

Direktorius

Tomas Vaičiūnas