

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Marius Kazlauskas

Duomenų integravimas panaudojant XML

Magistro darbas

Darbo vadovas

doc. V. Kiauleikis

Kaunas

2005

TURINYS

1.	ĮVADAS	4
2.	INFORMACINĖS SISTEMOS KŪRIMO ANALIZĖ	6
2.1.	Tyrimo sritis.....	6
2.2.	Analizės atlikimo metodų ir priemonių parinkimas	7
2.3.	Organizacijos (IT departamento) veiklos analizė	9
2.3.1.	<i>Analizės tikslas</i>	9
2.3.2.	<i>Veiklos sąveikų modelis (kontekstinė diagrama)</i>	9
2.3.3.	<i>Veiklos tikslų modelis</i>	10
2.3.4.	<i>Veiklos panaudojimo atvejų modelis</i>	11
2.3.5.	<i>Veiklos objektų modelis</i>	11
2.3.6.	<i>Organizacinės struktūros modelis</i>	12
2.3.7.	<i>Veiklos procesų modelis</i>	12
2.4.	Projekto tikslas ir kokybės kriterijų apibrėžimas.....	13
2.5.	Esamų sprendimų duomenų integravimo problemai spręsti analizė	15
2.5.1.	<i>Standartiniai duomenų bazių įrankiai duomenų importavimui/eksportavimui XML formatu</i>	15
2.5.2.	<i>IBM „XML Data Mediator“ – duomenų transformavimui tarp XML ir duomenų saugyklų</i>	16
2.5.3.	<i>Duomenų sandėliai</i>	17
2.5.4.	<i>Analizės išvados</i>	20
2.6.	XML technologijos duomenų integravimui.....	21
2.6.1.	<i>XML dokumentų nagrinėjimas</i>	21
2.6.2.	<i>Korektiškumo tikrinimo priemonės</i>	23
2.6.3.	<i>XML dokumentų transformavimas</i>	24
2.6.4.	<i>Duomenų išgavimas ir manipuliavimas</i>	25
3.	SISTEMOS STRUKTŪRA IR REALIZACIJA	27
3.1.	Reikalavimai informacinei sistemai.....	27
3.1.1.	<i>Vartotojų panaudojimo atvejų diagrama</i>	27
3.1.2.	<i>Specifikacijos panaudojimo atvejams</i>	28
3.1.3.	<i>Dalykinės srities klasių diagrama</i>	31
3.1.4.	<i>Vartotojų sąsajos modelis</i>	32
3.1.5.	<i>Papildomi reikalavimai sistemai</i>	32
3.2.	Technologinės informacijos centro projektas	34
3.2.1.	<i>Projekto tikslas</i>	34
3.2.2.	<i>Sistemos panaudojimo atvejų diagramos</i>	34
3.2.3.	<i>Panaudojimo atvejų scenarijų diagramos</i>	35
3.2.4.	<i>Sistemos architektūra</i>	38
3.2.5.	<i>Sistemos elgsenos modelis</i>	39
3.2.6.	<i>Duomenų bazės modelis</i>	40
3.2.7.	<i>Realizacijos modelis</i>	44

3.2.8.	<i>Testavimo modelis</i>	45
3.2.9.	<i>Reikalavimai sistemos funkcionavimo palaikymui</i>	45
4.	EKSPERIMENTINIS TYRIMAS	46
4.1.	Duomenų integracija.....	46
4.2.	Sistemos valdymas administratoriui	51
4.3.	Sistemos valdymas vartotojui	53
4.4.	Sistemos kokybės tyrimas.....	55
4.5.	Tolimesnės sistemos tobulinimo ir plėtojimo galimybės	56
5.	IŠVADOS	57
6.	LITERATŪRA	58
7.	TERMINŲ IR SANTRUMPŲ ŽODYNAS	59
8.	SANTRAUKA ANGLŲ KALBA	60
9.	PRIEDAI.....	61
	<i>1 priedas. Sistemos naudojimo instrukcija</i>	
	<i>2 priedas. Devintojoje magistrantų ir doktorantų konferencijoje pristatytas straipsnis „Organizacijos kompiuterinių sistemų integracija duomenų lygyje“</i>	
	<i>3 priedas. Sistemos sugeneruotos ataskaitos pavyzdys</i>	

1. ĮVADAS

Įmonės daug investuoja į informacijos technologijas, kad galėtų efektyviau organizuoti ir plėtoti veiklą, išlaikyti ir stiprinti savo pozicijas rinkoje. Per pastaruosius dešimtmečius vis didesni duomenų kiekiai saugomi elektroninėse laikmenose. Ši tendencija vis daugiau reikšis ateityje. Tačiau kuriant naujas informacines sistemas, dažniausiai jos būna nesuderintos su savo pirmtakėmis ir duomenų perkėlimo poreikis tampa gan svarbus. Taipgi iškyla viena aktualiausių informacinių sistemų problema – duomenų naudojimas ir pasidalinimas tarp skirtingų vartotojų, skirtingų įmonių ir skirtingų duomenų saugyklų.

Skirtingų informacinių sistemų naudojimas ir galimas korektiškas bendravimas tarp jų tampa vis aktualesnis. Organizacijų duomenų resursai gali būti daugybėje skirtingų formų: reliacinių duomenų bazių lentelėse, XML dokumentuose, objektinėse duomenų bazėse, elektroniniuose duomenų apsikeitimo pranešimuose ir daugelyje kitų [9]. Kadangi pasaulyje naudojamos įvairios aparatūrinės ir programinės platformos, skirtingų sistemų bendravimas įmanomas tik tuo atveju, kai nustatytas ryšys tarp jų leidžia informacinius mainus. Organizacijų kompiuterinių sistemų integracija (EAI) turi išspręsti bendrą skirtingų sistemų naudojimąsi verslo procesais ir duomenimis.

Informacinės sistemos integravimą galima vykdyti keliais būdais:

- duomenų lygio integracija
- uždavinio sąsajos lygio integracija
- metodo lygio integracija
- vartotojo sąsajos lygio integracija

Darbe nagrinėtas integravimas duomenų lygyje. Šio būdo privalumas yra pigumas, nes nėra daromi taikomųjų programų pakeitimai, testavimai ir įdiegimai. Duomenų lygio integracija apeina programinę logiką, paima ir perkelia duomenis tiesiogiai iš vienos sistemos į kitą. Tai yra paprastas ir greitai realizuojamas organizacijoje sprendimas. Šiam būdui įgyvendinti užtenka žinoti kur yra duomenys, surinkti informaciją apie duomenis, ir taikant veiklos principus nustatyti kurie duomenų srautai reikalingi realizacijai. Tai gali būti informacijos išgavimas iš vienos duomenų bazės, informacijos apdorojimas (jei reikalingas) ir atitinkamas kitos bazės pakeitimas. Tai naudinga tuo aspektu, kad esamų programų dažniausiai keisti nereikia. Duomenų lygio integracija yra bene lanksčiausiai ir lengviausiai įgyvendinama iš visų integracinių metodų. Čia dalijamasi duomenimis kaip galutiniu, nedalomu elementu, kas leidžia keisti, papildyti ar panaikinti dalį informacijos pagal reikalingą jos svarbą [1,2].

Darbo tikslas yra išnagrinėti XML galimybes duomenų lygio informacinių sistemų integracijai. Tai vienas iš pažangiausių integravimo būdų. Taikant XML galimybes, bet kada galima pakeisti esamus formatus, pridėdant ar panaikinant žymes, patikrinti ar duomenys yra struktūriškai teisingi, realizuota galimybė atlikti transformacijas į įvairius formatus, visiškai nepriklausomumas nuo sistemos platformos ar gamintojo. Šios technologijos darbe ir aptariamos.

Šiame darbe duomenų integravimo, panaudojant XML technologijas galimybės realizuotos konkrečioje aplinkoje – komunalinių paslaugų apskaitos įmonėje. Esant poreikiui formuoti ataskaitų pateikimą vartotojams per Interneto sąsają bei jų spausdinimui, realizuotas tų pačių XML srautų panaudojimas keliems taikomiesiems uždaviniams.

Darbo pabaigoje pateikiamos išvados, nurodoma naudota literatūra, terminų ir santrumpų žodynas bei priedai. Priedų skyriuje yra pateikiamas devintojoje magistrantų ir doktorantų konferencijoje pristatytas straipsnis „Organizacijos kompiuterinių sistemų integracija duomenų lygyje“, sistemos naudojimo instrukcija bei sugeneruotos ataskaitos pavyzdys.

2. INFORMACINĖS SISTEMOS KŪRIMO ANALIZĖ

2.1. Tyrimo sritis

Darbo tyrimo sritis – organizacijų kompiuterinių sistemų integracija duomenų lygyje.

Duomenų integravimas taikant XML technologijas. Duomenų surinkimas iš skirtingų duomenų bazių, apdorojimas ir transformavimas XML formatu. XML dokumentų panaudojimas duomenų atvaizdavimui per Internetinę sąsają ir spausdinimo ataskaitų generavimui. XML formato dokumentas naudojamas:

- Atitinkamos kitos duomenų bazės pakeitimui;
- Transformavimui į HTML dokumentą atvaizdavimui klientui per Internetinę sąsają;
- Formavimui PDF dokumento, standartizuotos spausdinimo formos kūrimui.

Procesai, metodai ir technologijos, skirtos duomenų perkėlimui iš vienu duomenų saugyklų į kitas (integracija duomenų lygyje). Analizuojamos XML technologijos – XML dokumentų nagrinėjimo, korektiškumo tikrinimo, transformavimo, duomenų išgavimo ir manipuliavimo taikymams. Nagrinėjama konkreti įmonė, jos kompiuterizavimo galimybės.

2.2. Analizės atlikimo metodų ir priemonių parinkimas

Informacinių sistemų inžinerija gali būti atlikta įvairiais metodais. Toliau pateikiamas F³, MERODE ir RUP metodų palyginimas.

F³ (angl. *From fuzzy to formal*, nuo migloto iki formalaus) – tai metodika, kuri skirta padėti informacinės sistemos vartotojams, reikalavimų savininkams ir reikalavimų inžinieriams gauti suderintą ir patvirtintą kuriamos sistemos specifikaciją sudarant integruotą reikalavimų inžinerijos specifikaciją. Ši metodika nėra objektinė.

MERODE (angl. *Model-based Existence-dependency Relationship Object-oriented Development*) – tai modeliu pagrįsta sistemos kūrimo metodika. Pasižymi abstrakčia objektų sąveikos sąvoka ir formaliu visų priemonių aprašymu. Čia išspręstos tokios problemos, kaip savybių, garantuojančių specifikacijos kokybę, trūkumas; objektiškai orientuotos analizės priemonių polinkis į realizaciją; aukšto lygio agregavimo priemonių trūkumas taikant objektiškai orientuotą metodiką.

RUP (angl. *Rational Unified Process*) - yra panaudojimo atvejų valdomas, architektūra grindžiamas, iteracinis, vykdomas palaipsniui, riziką mažinantis projektavimo procesas. Tai pilnai objektinė metodika.

Pagrindiniai RUP elementai:

1. Sukurti viziją
2. Valdyti projekto planą
3. Identifikuoti ir mažinti riziką
4. Paskirstyti darbus ir sekti rezultatus
5. Atlikti ekonominę analizę
6. Projektuoti komponentinę architektūrą
7. Palaipsniui konstruoti ir testuoti produktą
8. Testuoti ir įvertinti rezultatus
9. Kontroliuoti ir valdyti pasikeitimus
10. Teikti vartotojui paramą

1 lentelė. Informacinių sistemų inžinerijos metodų palyginimas

Etapas	F ³	MERODE	RUP
Organizacijos veiklos analizė	Apima	Pritaikoma	Apima
Sistemos analizė	Apima	Pritaikoma	Apima
Sistemos projektavimas	Apima	Pateikti tik principai	Apima
Sistemos realizavimas	Tik prototipų sukūrimas	Pateikti tik principai	Apima

Sistemos testavimas	Galima atlikti trasavimą nuo reikalavimų iki realizuojamų komponentų	-	Apima
---------------------	--	---	-------

Atlikus metodų palyginimą (žr. 1 lentelę) galima daryti išvadą, kad RUP metodas plačiausiai apima informacinių sistemų inžineriją. Šiame darbe ir pasirinktas naudoti būtent šis metodas.

2 lentelė. UML CASE įrankių pasirinkimo kriterijai

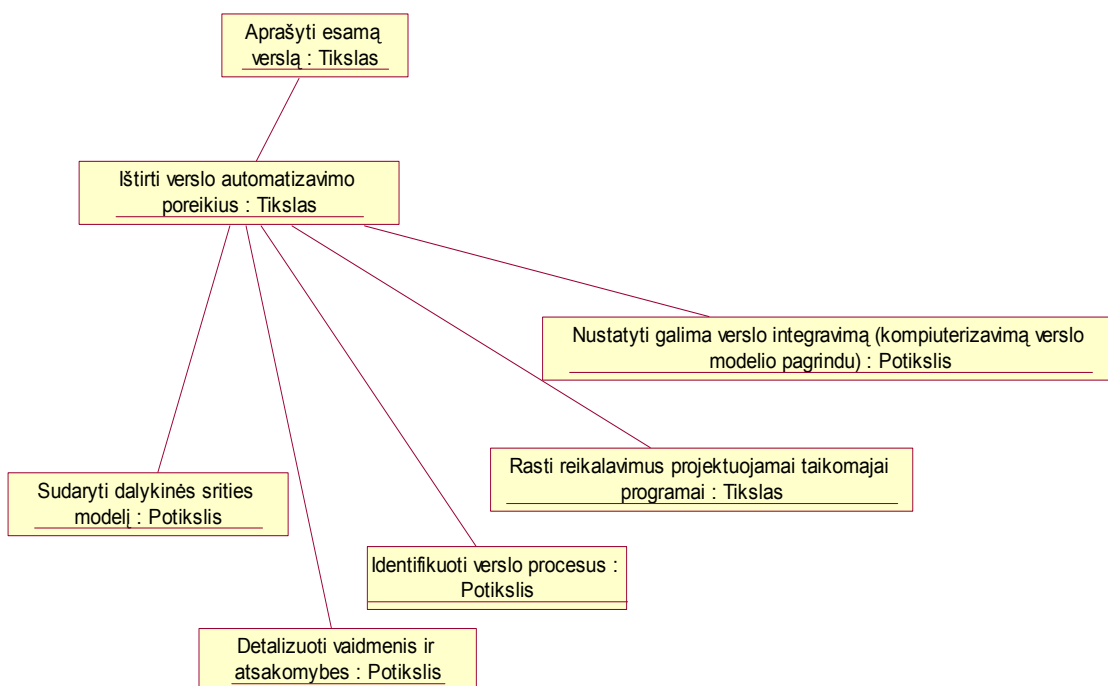
	Rational Rose 2000	MagicDraw	Visio2000 Enterprise
Pilnas UML (1.4 versijos) palaikymas	+	+	+
Diagramų suderinamumo kontrolė	+	+	+
Diagramų pasirinkimo sąrašai	+	+	+
Modelio navigavimas	+	+	+
Spausdinimas	+	+	+
Dokumentavimas HTML	+	+	+
Įrankio gyvybingumas (robustness)	+	+	+
Diagramų eksportas	+	+	+
Versijų palaikymas	+	+	+
Platformų palaikymas	C++, VisualC, Java, VBasic	VisualC, Java	VisualC, Java
Kodo generavimas	+	+	+
Duomenų modeliavimas	+	-	-
Tiesioginis ir atvirkštinis projektavimas	+	+	+
Repozitoriaus palaikymas	+	+	+

Aptarti įrankiai (žr. 2 lentelę) daugmaž panašūs, Rational Rose išsiskiria turintis duomenų modeliavimo įrankiu. Ši priemonė, atliekant modeliavimą, bus naudojama darbe.

2.3. Organizacijos (IT departamento) veiklos analizė

2.3.1. Analizės tikslas

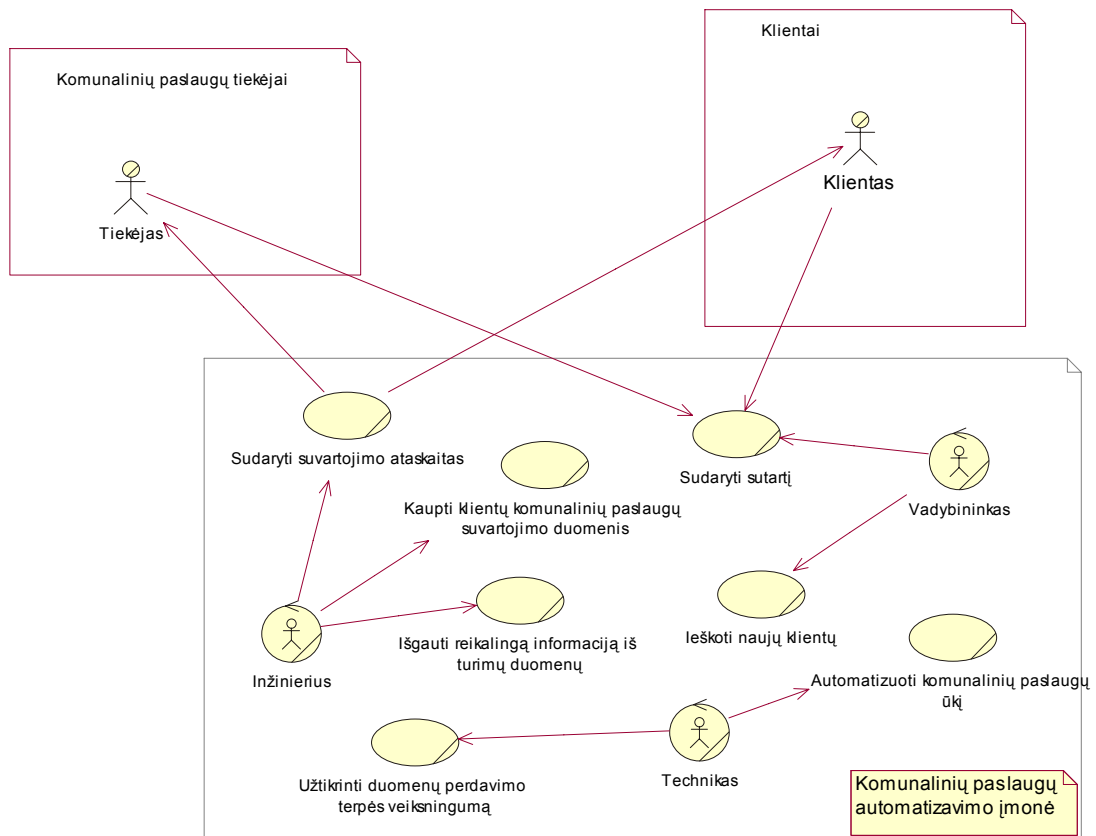
Pagrindinis organizacijos veiklos analizės tikslas – aprašyti esamą verslą ir ištirti verslo kompiuterizavimo poreikius bei galimybes – nustatyti tas sritis kurias įmanoma ir tikslinga kompiuterizuoti. Iš to seka potiksliai – sudaryti dalykinės srities modelį, identifikuoti ir nustatyti verslo procesus, detalizuoti vaidmenis ir atsakomybes organizacijos viduje, rasti reikalavimus projektuojamai taikomajai programai ir galiausiai nustatyti galimą verslo integralumą, kompiuterizavimo laipsnį. Analizės tikslas pavaizduotas 1 paveiksle.



1 pav. Analizės tikslų modelis

2.3.2. Veiklos sąveikų modelis (kontekstinė diagrama)

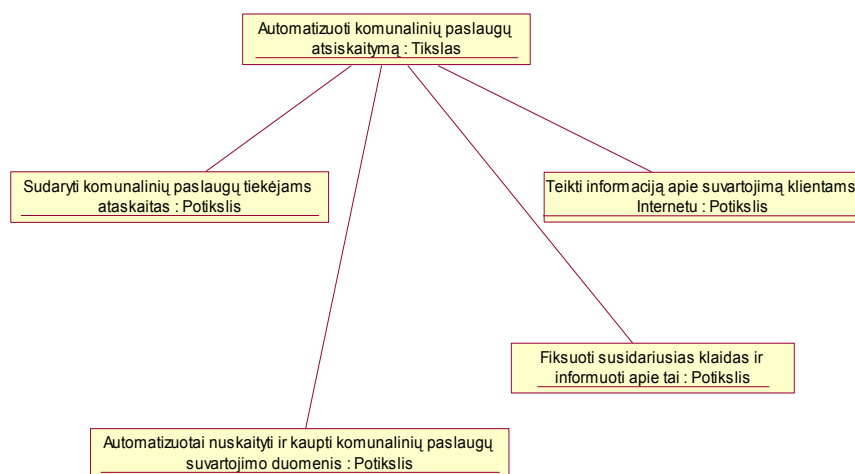
Organizacija sudaro trišales sutartis su klientais ir komunalinių paslaugų tiekėjais. Automatizuojamas klientų komunalinių paslaugų ūkis, tam tikslui, kad galima būtų automatizuotai kaupti ir analizuoti paslaugų suvartojimą. Klientai turi galimybę peržiūrėti informaciją apie savo prietaisų parodymus Internetu, tiekėjai automatizuotai gauna ataskaitas apie kiekvieno kliento gaunamų paslaugų suvartojimą.



2 pav. Kontekstinė diagrama

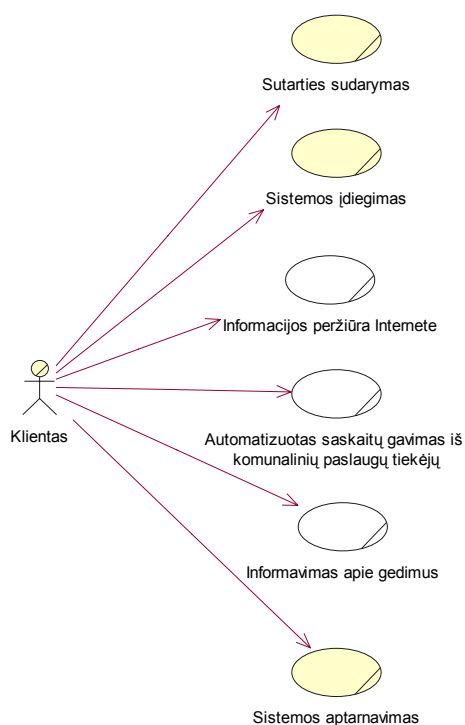
2.3.3. Veiklos tikslų modelis

Pagrindinis organizacijos IT skyriaus veiklos tikslas yra automatizuoti komunalinių paslaugų atsiskaitymą. Tai sudaro keli potiksliai: automatizuotai nuskaityti ir kaupti paslaugų suvartojimo duomenis, sudaryti ataskaitas komunalinių paslaugų tiekėjams, teikti informaciją klientams Internetu, automatiškai fiksuoti susidariusias klaidas ir daryti atitinkamus veiksmus.



3 pav. Veiklos tikslų modelis

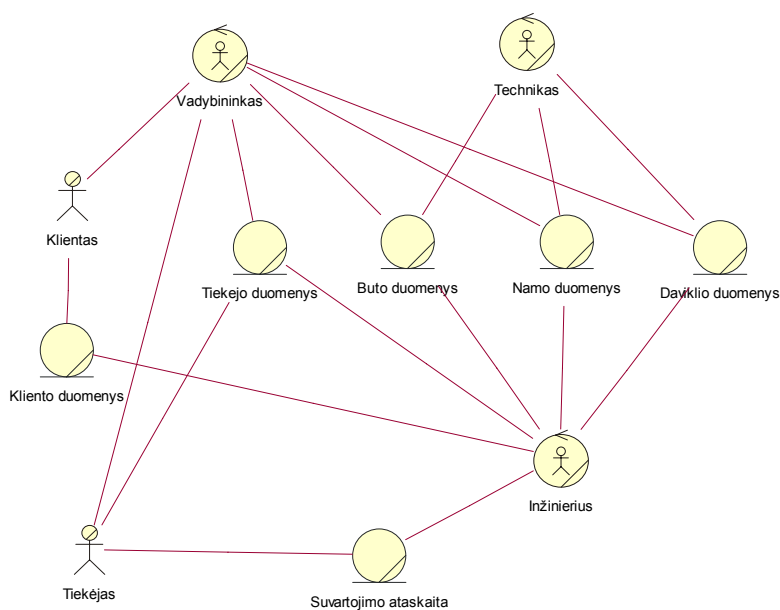
2.3.4. Veiklos panaudojimo atvejų modelis



4 pav. Veiklos panaudojimo atvejų modelis

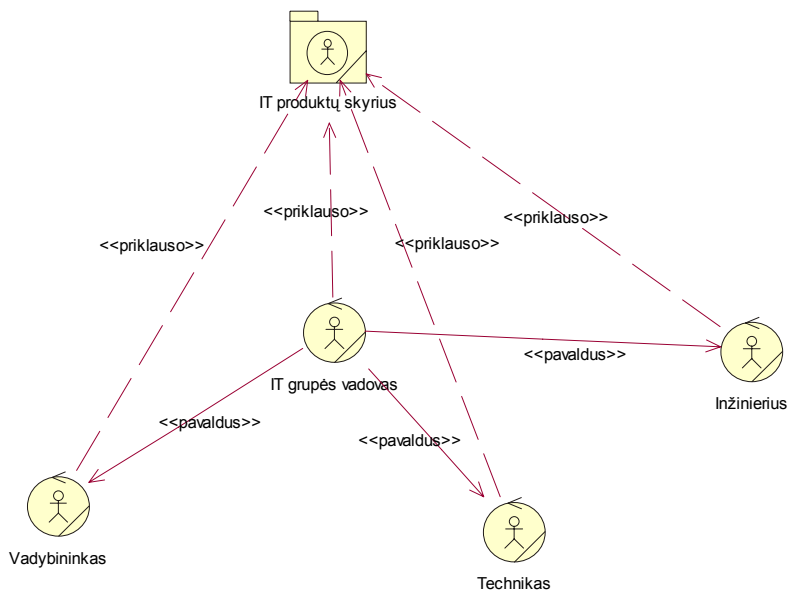
Organizacijos veiklą galima išskirstyti į 4 paveiksle pavaizduotus kliento panaudojimo atvejus. Tie panaudojimo atvejai, kuriuos numatoma kompiuterizuoti, pavaizduoti šviesesniame fone.

2.3.5. Veiklos objektų modelis



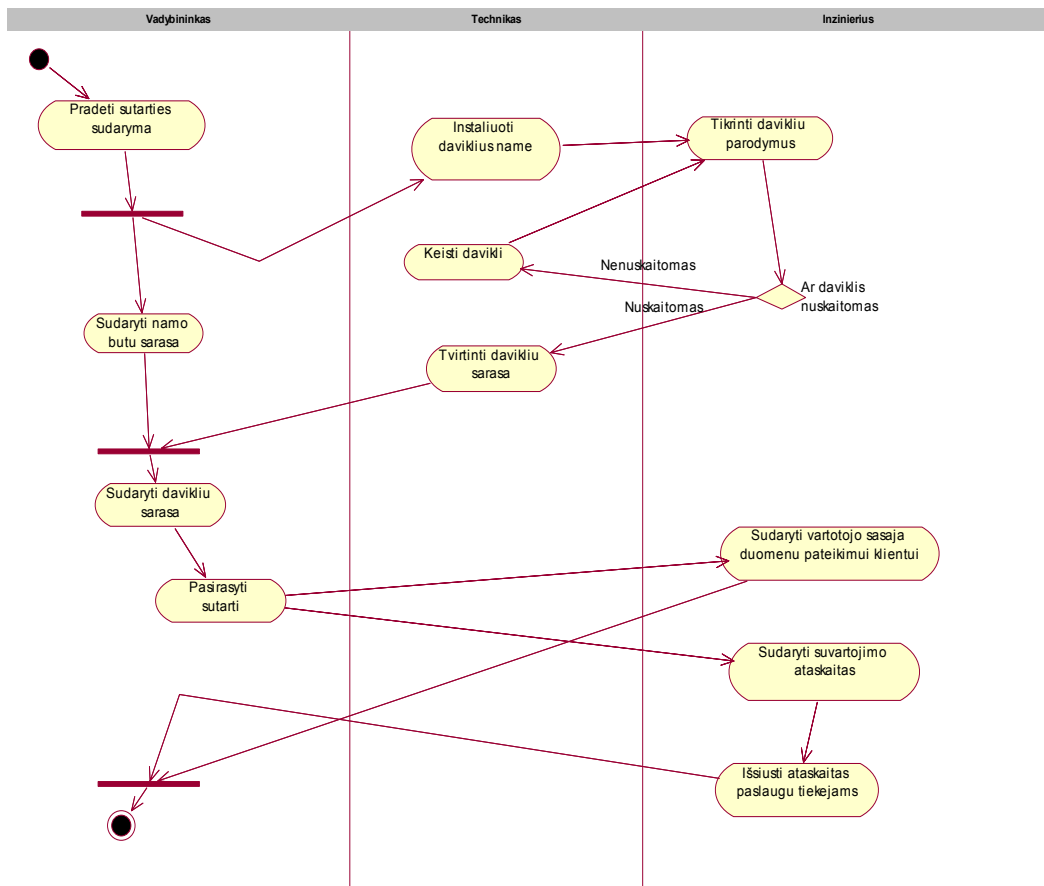
5 pav. Veiklos objektų modelis

2.3.6. Organizacinės struktūros modelis



6 pav. Organizacinės struktūros modelis

2.3.7. Veiklos procesų modelis

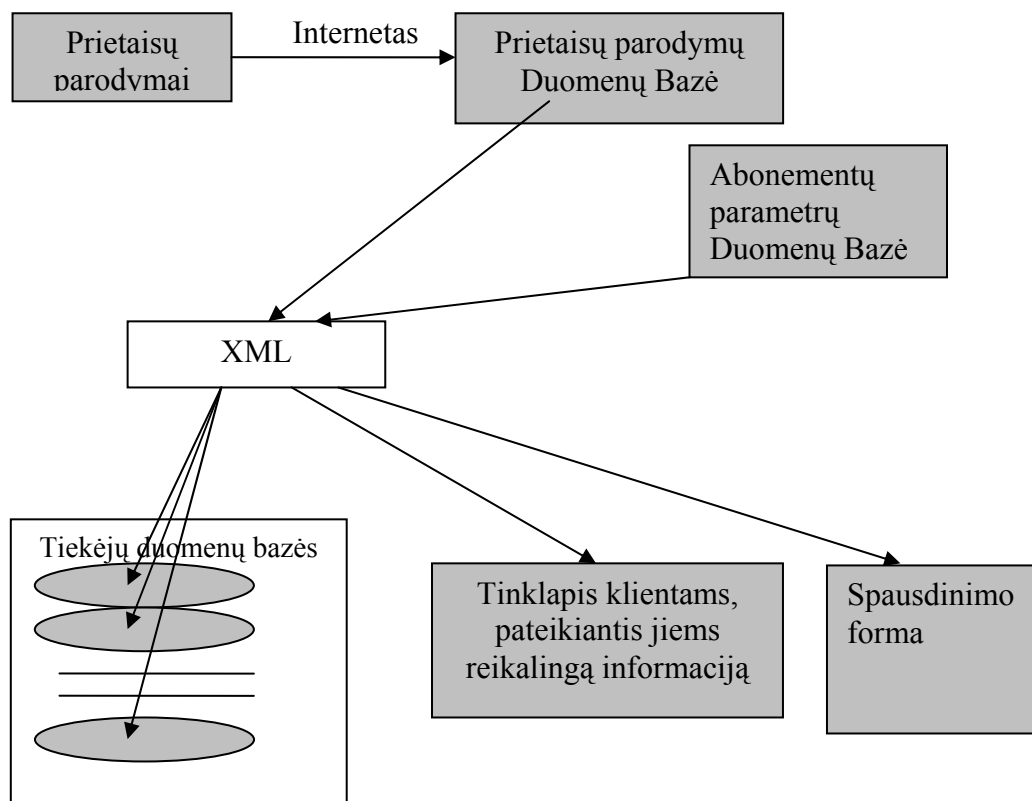


7 pav. Veiklos procesų modelis

2.4. Projekto tikslas ir kokybės kriterijų apibrėžimas

Darbas projektuojamas konkrečioje aplinkoje – komunalinių paslaugų automatizavime. Šis procesas apima įvadinių ir kontrolinių skaitiklių duomenų surinkimą ir pateikimą, supaprastina sąskaitų sudarymą. Žmonėms nereikia nurašinėti skaitiklių parodymų, pildyti lapelių ir patiems atlikinėti skaičiavimų. Vykdamas tokią sistemą, iškyla duomenų integracijos poreikis. Sistema surenka duomenis apie vartotojų paslaugų naudojimą (vandens, šilumos, elektros) ir saugo jų archyvą. Problema kyla tuomet, kai reikia sudaryti ir teikti reikalingas ataskaitas paslaugų tiekėjams (šilumos, vandens, elektros suvartojimą), taip pat realizuojant Internetinę sąsają klientams, pavaizduojant kiekvieno kliento informaciją jam patogią formą. Kiekviena šių sistemų atskirai naudoja skirtingus duomenų modelius, skirtingas duomenų bazines, joms reikalinga skirtinga informacija. Pvz., iš klientų šilumos daviklių gaunama informacija apie kiekvieno prietaiso parodymus, to prietaiso parametrus ir kt., o šilumos tiekėjui reikalinga tik bendras kliento šilumos suvartojimas per nurodytą laikotarpį.

Sistemos tikslas yra perduoti duomenis iš pradinio taško – namo duomenų bazės į galinį – tiekėjams ir klientams, pakeitus juos pagal poreikį. Šiuo atveju projekto tikslas būtų naudojantis standartizuotu, XML pagrįstu principu, keisti duomenų formatą į pageidaujamą tiesiog perduodant duomenis iš vienos duomenų bazės į kitą.



8 pav. Sistemos, taikant XML technologijas, struktūros schema

Pagrindinis projekto kokybės kriterijus – korektiškai veikianti programinė įranga. Pagrindinė sistemos veikimo funkcija – formuoti reikiamus XML srautus ir sudaryti galimybę, pakartotinai juos naudojant, keisti duomenų bazes, kurti ataskaitų atvaizdavimą Interneto puslapiuose ir sudaryti spausdinimo ataskaitas. Visa programinė įranga turi atitikti reikalavimuose apibrėžtus kriterijus. Taip pat svarbi sąlyga yra sistemos veikimo laikas. XML srautų sudarymas, transformavimas į Interneto puslapio struktūrą ir spausdinimo ataskaitų formavimas neturi užtrukti ilgą laiką.

2.5. Esamų sprendimų duomenų integravimo problemai spręsti analizė

2.5.1. Standartiniai duomenų bazių įrankiai duomenų importavimui/eksportavimui XML formatu

Yra daug būdų išspręsti efektyvaus, automatizuoto XML duomenų konvertavimo į ir iš reliacinių duomenų bazių problemą. Duomenų bazių kūrėjai, tokie kaip IBM, Microsoft, Oracle ir Sybase yra išvystę programinius įrankius padedančius konvertuojant XML dokumentus į reliacines lenteles. Štai keli siūlomi sprendimai [8]:

- Oracle XML SQL įrankis modeliuoja XML dokumentų elementus kaip lizdinę lentelių kolekciją. Pridėtas elementas yra modeliuojamas kaip Oracle objekto duomenų tipas. „SQL į XML“ konvertavimas sudaro XML dokumentą naudojant „vienas su vienu“ asociaciją tarp lentelių, nurodant objekto duomenų tipą ir lizdinį elementą. „XML į SQL“ reikalauja arba duomenų modelio redagavimo (konvertuojant jį iš reliacinio į objektiškai orientuotą) arba pertvarkymo originalaus XML dokumento.
- IBM DB2 XML plėtinys leidžia talpinti XML dokumentus kaip BLOB objektus arba kaip suskaidytus į lenteles. Naujausia transformacija, žinoma kaip XML kolekcija, apibrėžta XML 1.0 sintaksės.
- Microsoft sprendžia problemą praplėsdamas SQL ir pateikdamas OPENXML eilučių seką.
- Sybase Adaptive Server įveda „ResultSetXml“ Java klasę kaip abiejų kryptių XML konvertavimo bazę.

XML dokumentų konvertavimas natūraliai yra simetrinis. XML talpinimas neleidžia modifikuoti daugiau kaip vieną lentelę. XML išgavimas transformuoja SQL užklausos rezultatus į plokščios struktūros dokumentą (angl. - flat file).

Kūrėjų palyginimas pateikiamas 3 lentelėje.

3 lentelė. Kūrėjų palyginimai[8]

Kūrėjas	Konvertavimo taisyklės	Viena lentelė / daug lentelių	Konvertavimo priemonės	Simetrinis konvertavimas
Oracle	Besąlygiškai; sudarant objektų – reliacinių duomenų modelį	Daug	Nustatytos Java klasės	Simetrinis, jei XML dokumentas ir objektų – reliacinis modelis derinasi

IBM	Duomenų priėjimo apibrėžimo failas	Daug	Suprojektuotos talpinimo procedūros	Simetrinis
Microsoft	SQL išplėtimas; eilučių kolekcijos funkcijos	Daug išgaunant XML Viena talpinant XML	Naudojant SQL konstruktorių FOR XML ir eilučių kolekciją OPENXML	Asimetrinis
Sybase	DTD	Viena; užklausa gali apimti kelias lenteles	Naudojant Java klases	Simetrinis

Bendros savybės:

1. Visos priemonės importuoja ar eksportuoja duomenis į ar iš duomenų bazės, tačiau neturi galimybių tai atlikti pagal tam tikrą logiką, esant poreikiui transformavimo metu atlikti tam tikrus veiksmus ir pagal tai spręsti kokius duomenis naudoti;
2. Šios priemonės netinka esant poreikiui sujungti kelių skirtingų technologijų duomenų bazių duomenis;
3. Nėra bendrų taisyklių konvertuojant duomenų bazes į XML; specialus žymėjimas reikalingas, jei XML dokumentas naudoja naują sintaksę;
4. XML transformavimas į duomenų bazę reikalauja duomenų paruošimo, pvz., skaičių/datos performavimo atitinkamiems reikalavimams; XSLT gali būti naudojamas šiam tikslui.

2.5.2. IBM „XML Data Mediator“ – duomenų transformavimui tarp XML ir duomenų saugyklų

XML duomenų tarpininkas (anksčiau vadintas XML integratoriumi (angl. „XML Integrator“) – tai įrankis abipusiam duomenų konvertavimui tarp XML ir struktūrinių duomenų formatų, tokių kaip reliacinės duomenų bazės. Šis įrankis įkūnija taisykles duomenų atvaizdavimo tarp XML ir reliacinių duomenų bazių ir pakeičia programinį būdą, kai paprasčiausiai programiniu kodu aprašomas XML ir reliacinių duomenų konstrukcijų ryšys. „XML Data Mediator“ gali būti naudojamas kaip atskiras produktas arba gali būti integruojamas kaip biblioteka į kitus programinius paketus[6].

„XML Data Mediator“ palengvina užduotį atvaizduoti XML ir kitų duomenų struktūrų kūrimą, leisdamas deklaratyvias specifikacijas ryšių tarp XML struktūrų (tokių kaip elementai ar atributai) nurodymui ir reliacinių schemų elementų (tokių kaip eilutės ar stulpeliai) aprašymui. Kai toks atvaizdavimas yra specifikuotas schemas lygyje, „XML Data Mediator“ gali automatiškai konvertuoti duomenis iš ir į XML per „DUOMENYSĮXML“ ir „XMLĮDUOMENIS“ komponentus. „DUOMENYSĮXML“ komponentai gali surinkti duomenis iš vienos ar kelių duomenų saugyklų ir sutvarkyti juos į nuoseklų XML dokumentą, aprašytą specifikuotoje schemoje. „XMLĮDUOMENIS“ komponentai išgauna reikalingas apdorojamo XML dokumento dalis ir patalpina juos į vieną ar kelias duomenų saugyklas pagal duomenų keitimo komandas (tokias kaip įterpti, pakeisti, ištrinti, ar kitas, aprašytas vartotojo).

„XML Data Mediator“ suprojektuotas priimti įeinantį duomenų srautą kaip šabloną transformacijų atlikimui. Priklausomai nuo įeinamo srauto ar transformavimo krypties (iš duomenų saugyklos į XML ar iš XML į duomenų saugyklą), „XML Data Mediator“ sukuria atitinkamą medžio struktūrą transformacijų vykdymui. Medžio struktūra charakterizuoja tiek duomenų saugyklos vykdomas procedūras, tiek XML struktūros informaciją, kuri yra arba XML hierarchija, kuri bus sukurta vykdant duomenų ištraukimą iš duomenų saugyklų, arba bus atitinkamos duomenų saugyklos hierarchijos atitikmuo. Vykdamas duomenų paieškos procesą duomenų saugykloje, „XML Data Mediator“ sukuria atvaizdavimą, vadinamą „žymių medžiu“ – struktūrą, kuri aprašo hierarchiją XML elementų, kurie bus sukurti vykdant duomenų išgavimą. Atvirkštiniame procese sukuriamas medis, kuris aprašo atitinkamas XPath išraiškas, naudojamas reikalingų elementų išrinkimui iš skaitomo XML dokumento ir duomenų talpinimo procedūras korektiškam duomenų surašymui į duomenų saugyklą.

2.5.3. Duomenų sandėliai

Kas yra duomenų sandėliavimas

Duomenų sandėliavimo principai ir funkcijos aptariamos remiantis IBM DB2 duomenų sandėlio pavyzdžiu. Sistemos, kurios naudoja operatyvinius duomenis (duomenys, su kuriais vykdomos kasdieninės transakcijos) saugo informaciją, naudingą verslo analizei. Pavyzdžiui, analitikams gali būti naudinga informacija apie konkrečius produktus, kurie buvo parduoti konkrečiame regione konkrečiu periodu ruošiant ateities prognozes. Keletas problemų gali iškilti, kai analitikas dirbs tiesiogiai su operatyviniais duomenimis [7,11]:

- Analitikas gali neturėti kompetencijos atlikti užklausas operatyvinėje duomenų bazėje. Duomenų bazių užklausų kūrimui reikalingos taikomosios programos, kurios

naudoja specializuotą duomenų valdymo kalbą. Paprastai programuotojai, kurių kompetencija yra operatyvinių duomenų bazių valdymas, dirba ištisas dienas prižiūrėdami duomenų bazes ir jų taikomas programas. Sparta yra kritinė savybė daugumai operatyvinių duomenų bazių, kaip kad pvz., banke. Sistema negalės reikiamai aptarnauti vartotojų tuo pat metu vykdydama papildomas, ne visada korektiškas užklausas.

- Operatyviniai duomenys paprastai nėra optimaliame formate, reikalingame verslo analitikui. Pavyzdžiui, pardavimų duomenys, kurie yra aprašyti pagal produktus, regioną, sezonus yra daug naudingesni analitikui, nei neapdoroti duomenys. Duomenų sandėliavimas išsprendžia šią problemą. Duomenų sandėlyje kuriamos informacinių duomenų saugyklos. Informaciniai duomenys – tai duomenys, kurie gaunami iš operatyvinių duomenų ir gauti atliekant sprendimų priėmimo transformacijas. Pavyzdžiui, duomenų sandėliavimo įrankis gali nukopijuoti visus pardavimo duomenis iš operatyvinių duomenų bazių, išvalyti duomenis, atlikti skaičiavimus apibendrinant duomenis ir surašyti apibendrintus duomenis į atskirą duomenų saugyklą, atskirtą nuo operatyvinių duomenų. Vartotojai gali daryti užklausas šioje duomenų saugykloje, nedarydami įtakos operatyvinėms duomenų bazėms.

Duomenų sandėlio užduotys

- Duomenų sandėlio kūrimas, į kurį įeina:
- Identifikavimas šaltinių duomenų (arba operatyvinių duomenų) ir aprašymas jų sandėlio duomenų šaltinių naudojimui.
- Duomenų bazės sukūrimas, kuri naudojama kaip duomenų sandėlis ir aprašymas kaip sandėlio duomenų gavėjų.
- Procesų grupių aprašymai, kurie bus aprašyti duomenų sandėlyje.
- Specifikuoti duomenų perkėlimą ir transformavimą iš šaltinių į duomenų sandėlių saugyklų formatus, aprašant po žingsnį šį procesą.
- Aprašytų žingsnių testavimas ir grafiko, kada juos vykdyti automatiškai, nustatymas.
- Duomenų sandėlio valdymas nustatant apsaugas ir kontrolę.
- Duomenų informacijos katalogo kūrimas sandėlyje. Informacijos katalogas yra duomenų bazė, kuri susideda iš verslo metaduomenų. Verslo metaduomenys padeda vartotojams identifikuoti organizacijos duomenis ir jiems reikalingą informaciją. Duomenų sandėlio metaduomenys gali būti skelbiami informacijos kataloge. Jame atsispindi kokie duomenys saugomi sandėlyje.

- Sandėlio (ar dažniau duomenų mugės) duomenų modelio žvaigždinės schemos aprašymas. Žvaigždinė schema – tai specializuotas projektas, kuris susideda iš kelių dimensijų lentelių, aprašančių verslo aspektus ir vienos faktų lentelės, susidedančios iš faktų ar matmenų apie verslą. Pavyzdžiui, gamybinei įmonei, dimensijų lentelės gali būti produktas, rinka ir laikas. Faktų lentelė susideda iš transakcijų informacijų apie produktus, kurie užsakyti tam tikrų regionų tam tikru sezonu.

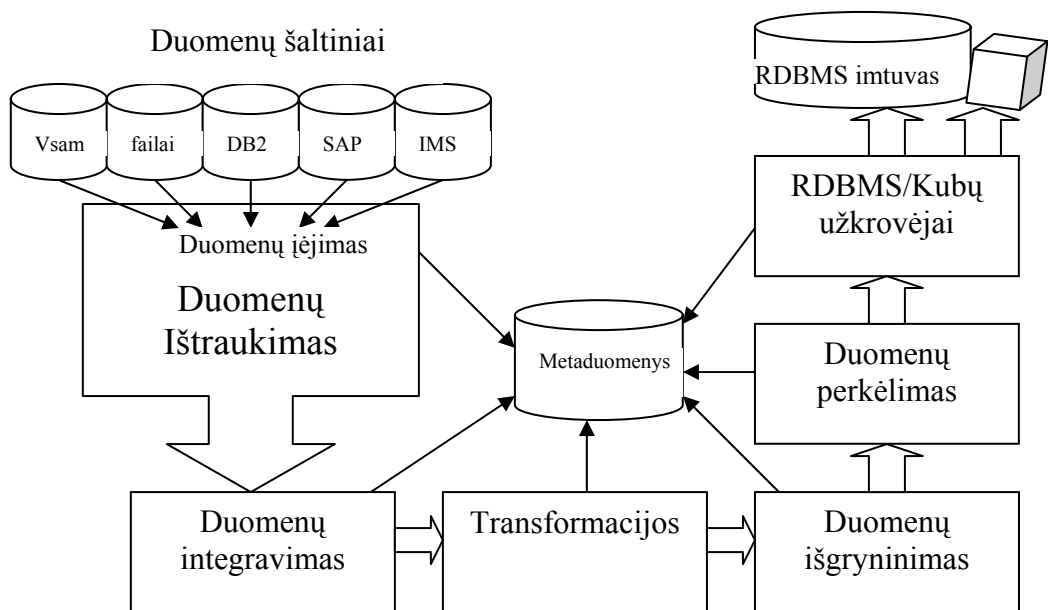
Savybės:

- Automatiškai sukuria ir užpildo duomenų sandėlius ir duomenų mugės esama informacija.
- Paprastina duomenų sandėlių kūrimo, vykdymo ir palaikymo procesus. Integruoti įrankiai duomenų išgavimui, transformavimui, užkrovimui, darbų planavimui, sandėlio valdymui ir kontrolei, vartotojų užklausoms ir ataskaitoms.

Privalumai:

- Suteikia galimybę verslo klientams turėti pačią tiksliausią informaciją, todėl jie gali priiminti teisingus sprendimus turėdami reikalingus duomenis reikalingu laiku reikalingoje vietoje.
- Suteikia administratoriui vieną kontrolės tašką valdyti, palaikyti ir suprasti centralizuotą ir paskirstytą duomenų sandėlio procesą.

Duomenų išgavimo, pakeitimo ir apskritai palaikymo valdymas pavaizduotas 9 paveiksle:



9 pav. Duomenų sandėlio veikimo schema [7]

2.5.4. Analizės išvados

Apžvelgus esamus produktus, duomenų perdavimui ir integravimui XML formatu, galima daryti šias išvadas:

- Duomenų bazių įrankiai duomenų apsikėitimui XML formatu dažniausiai neturi galimybių tai atlikti pagal tam tikrą logiką, nėra numatyta transformavimo metu atlikti tam tikrus veiksmus ir pagal tai spręsti kokius duomenis naudoti;
- Standartiniai duomenų bazių įrankiai netinka kelių skirtingų technologijų duomenų bazių duomenų sujungimui, eksportuojant duomenis į XML formatą;
- Duomenų sandėliai – ideali priemonė duomenų sujungimui, transformavimui ir perdavimui tarp bet kokių technologijų duomenų bazių, tačiau dėl šių produktų kainos, tai nėra kiekvienam prieinamas įrankis;
- Yra sukurta priemonių, išsprendžiančių XML srautų formavimui ir nuskaitymui, taip pat jų panaudojimui formuojant Internetines (HTML) ar spausdinimo formas (PDF), tačiau vieningo, viską apjungiančio produkto rasti nepavyko.

2.6.XML technologijos duomenų integravimui

2.6.1. XML dokumentų nagrinėjimas

Norint patikrinti XML dokumento korektišką sandarą ar jį panaudoti, dokumentas prieš tai turi būti apdorotas. Tai atlieka programos, vadinamos XML nagrinėtojai (angl. XML parser). Jos sugeba nuskaityti ir (dažniausiai) patikrinti XML dokumento duomenų korektiškumą. Nagrinėtojas leidžia programai naudotis XML dokumente esančiais duomenimis. Žinomiausi ir patys pagrindiniai yra du nagrinėtojų tipai [5]:

- Medžiais pagrįsti nagrinėtojai, kurie XML dokumentą programoje paverčia į vidinę atminties struktūrą (DOM).
- Įvykiais pagrįsti nagrinėtojai, kurie skaitydami XML dokumentą generuoja įvykius ir praneša apie tai programai (SAX).

Pasirinkimą nulemia programos reikalavimai. Dažniausiai yra pasirenkamas DOM, kadangi jo programavimo modelis žymiai aiškesnis, nei SAX įvykių apdorojimas. Tačiau, kad ir koks neįprastas būtų SAX programavimo modelis, jis gali būti truputį efektyvesnis. SAX programos gali būti žymiai spartesnės už savo DOM variantus, ir beveik visada naudoja žymiai mažiau atminties. Ypač patogu naudoti SAX dirbant su dokumentais srautiniu režimu. Jeigu įmanoma suskaldyti sudėtingus apdorojimo procesus į kelis filtrus, SAX yra nepamainomas. Naudojančios DOM programos linkusios būti daugiau monolitiškos, kur viena programa atlieka visą darbą.

Šie programos bruožai nurodo, kad programoje tikriausiai geriau būtų naudoti SAX (arba traukiantįjį) nagrinėtoją:

- Dokumentai netelpa į atmintį. Tai pati svarbiausia priežastis pasirinkti vieną ar kitą programavimo sąsają. Jeigu dokumentai netelpa į atmintį, nėra kitos galimybės tik srautinės darbo su XML sąsajos.
- Dokumentą galima apdoroti trumpais gabaliukais. Nereikia užsikrauti viso dokumento, kad galėtume pradėti su juo dirbti.
- Apdorojant dokumentą, užtenka tik žinių apie prieš tai buvusius elementus, nereikia žiūrėti į priekį.
- Apdorojimas gali būti išskaidytas į kelias nuoseklias operacijas.

Kiti bruožai nurodo, kad programa tikriausiai turėtų naudoti vieną iš DOM palaikančių nagrinėtojų:

- Programa vienu metu turi kreiptis į atskiras dokumento dalis ar net kelis dokumentus vienu metu.

- Vidinės duomenų struktūros yra panašaus sudėtingumo kaip ir pats dokumentas.
- Programa turi modifikuoti dokumentą pakartotinai.
- Jei dokumentas saugomas ilgą laiką ir su juo vis atliekamos operacijos, o ne vieną kartą apdorojamas ir pamiršamas.

Būna atvejų, kai reikia naudoti ir SAX, ir DOM. Pavyzdžiui, galima naudoti bibliotekas SAX XMLReader su prie jo prikabinutų XMLFilter serija, kurių galutinis suformuoja DOM dokumentą. Darant atvirkščiai galima pereiti per visas DOM medžio atšakas, formuojant SAX įvykius ir perduodant juos į biblioteką ContentHandler.

Atsižvelgiant į aptartus bruožus, darbe naudojamas DOM metodas. Toliau pateikiama skirtingų DOM modelių palyginimo lentelė [5,12].

4 lentelė. Skirtingų DOM modelių palyginimas

Savybė	W3C DOM	dom4j	JDOM
Pagrįstas Java interfeisais	Taip	Taip	Ne
Palaiko Java kolekcijas	Ne	Taip	Taip
Gali naudoti bet kokį SAX nagrinėtoją ar XMLFilter	Dažniausiai	Taip	Taip
Konvertuoja iš ir į DOM medžius	Taip	Taip	Taip
Pats įgyvendina DOM interfeisus	Taip	Taip	Ne
Integruotas XPath palaikymas	Ne	Taip	Taip
Įvykiais pagrįsto nagrinėjimo palaikymas	Galimas	Taip	Ne
Gali dirbti su XML srautais	Ne	Taip	Ne
XML Schema duomenų tipų palaikymas	Ne	Taip	Ne
Didelių XML dokumentų palaikymas	Ne	Taip	Ne

JDOM tikriausiai populiariausias dokumento modelis, nes jo programavimo sąsaja labai patogi. Nors DOM suteikia daug galimybių dirbti orientuojantis į dokumento struktūrą, tačiau jis turi trūkumų pateikdamas objektiškai orientuotą supaprastinimą. Programuotojams, kurie dirba su daugiau į duomenis orientuotomis struktūromis, kitaip nei knygomis, straipsniais ar kitais pilnaverčiais dokumentais, žymiai patogiau naudotis JDOM ar dom4j, kurie labiau atitinka jų poreikius. Todėl pasirenkamas JDOM modelis.

2.6.2. *Korektiškumo tikrinimo priemonės*

XML dokumentas yra tekstas, kuris aprašo hierarchinę duomenų struktūrą. XML standartas aprašo sintaksę, kaip turi būti sudarytas dokumentas. Atitinkantis XML sintaksę dokumentas vadinamas taisyklingu. Susidūrę su netaisyklingais dokumentais, įrankiai turėtų pranešti apie klaidas, nes tai jau nebėra XML dokumentas. Tačiau norint žinoti, ar dokumentas yra prasmingas, neužtenka vien taisyklingumo tikrinimo. Teisingi duomenys turi atitikti tam tikrą loginę struktūrą, kurią aprašo schemų kalbos. Būtent schemų kalbomis aprašomos naujos XML kalbos. Bet kuris dokumento korektiškumą tikrinantis nagrinėtojas gali užtikrinti, kad nagrinėjamas dokumentas atitinka nustatytas kalbos taisykles. Taip nukeliama didelė našta nuo programuotojo pečių, kuris turėdavo savo programoje pats tikrinti ar pateikiami duomenys yra korektiški. Kadangi, kartu su XML standartu pateikta schemų kalba DTD turėjo mažai galimybių, įvairios organizacijos sukūrė daug įvairių schemų kalbų [10].

Seniausia schemų kalba yra dokumento tipo apibrėžimas (angl. DTD – Document Type Definition), kuris susideda iš taisyklių apie dokumento struktūrą ir turinį. DTD išvardina visus galimus elementus, jų pasirodymo tvarką ir atributus.

Nors, palyginus su kitomis schemų kalbomis, DTD turi mažai galimybių, bet jis yra pirmasis priimtas standartas, populiarus ir dėl savo paprastumo. DTD yra sudedamoji XML standarto dalis, todėl jame yra keletas dalykų, kurie nepriklauso schemų kalbai, pvz., esybės. Net jei ilginiui schemų kalbose įsitvirtins XML Schema, panašu, kad DTD ir toliau bus plačiai naudojamas, ypač ten, kur pakanka jo galimybių.

Nors dokumento tipo apibrėžimo metodas puikiai tiko pirmiems bandymams, bet kuriant sudėtingas sistemas iš karto paaiškėjo jo trūkumai. Viena pagrindinių DTD problemų – tai nėra korektiškas XML dokumentas. Todėl darbui su juo nebetinka standartiniai XML nagrinėtojai.

Dar daugiau problemų kelia tai, kad DTD turi tik ribotas galimybes dokumento struktūros ir turinio aprašymui. Juo neįmanoma aprašyti sudėtingų dokumento elementų ryšių, be to, vienam elementui aprašytos taisyklės negali būti panaudotos kito elemento aprašymui.

Šias problemas išsprendžia kitas standartas – XML Schema. Šie dokumentai yra taisyklingi XML dokumentai, jie palaiko vardų sritis bei turi priemones elementams priskirti tipus, taip leisdama tikrinti dokumentus ne tik pagal elementų struktūrą, bet ir pagal jų turinį. Viena, labiausiai išsiskirianti iš kitų schemų kalbų XML Schema savybė, yra tipų paveldimumas.

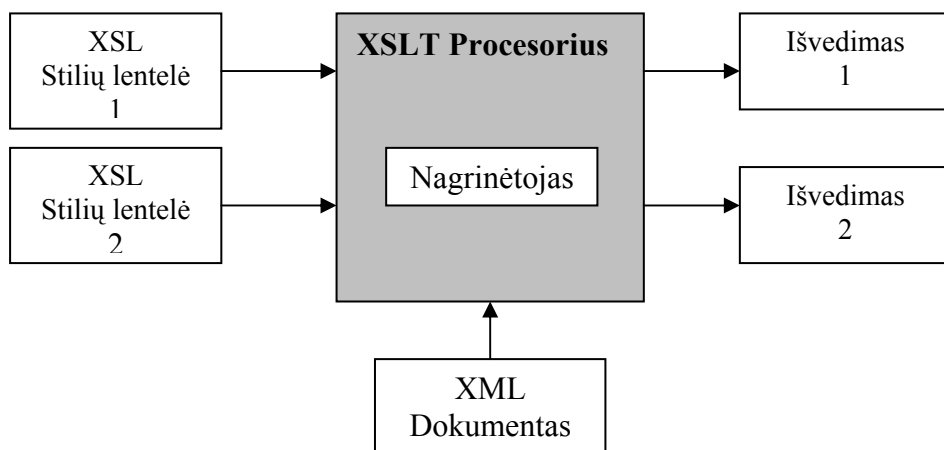
Savybė	DTD	XML schema
Naudojama XML sintaksė	Ne	Taip
Turi duomenų tipų savyje	10	37
Duomenų tipai gali būti naujai aprašyti pagal vartotojo poreikius	Ne	Taip
Atributo reikšmė pagal nutylėjimą	Yra	Yra
Elemento reikšmė pagal nutylėjimą	Nėra	Dalinai
Paveldimumas	Nėra	Yra
Unikalumas atributams	Galimas	Galimas
Unikalumas ne atributams	Nėra	Galimas
Atributų, elementų raktai	Nėra	Yra
Paties schemos dokumento tikrinimas	Negalimas	Galimas

Po truputį XML Schema tampa pagrindine schemų kalba, kadangi ji yra oficialus standartas ir leidžia aprašyti žymiai daugiau informacijos apie XML dokumentą. Ja remiasi ir kiti, naujesni XML šeimos standartai: XQuery, XPath, XSLT. Ypač svarbus šis standartas XML duomenų surišime, kadangi geras schemas aprašas leidžia apsieiti be surišimo schemos dokumento. Atlikus šių dviejų XML dokumentų schemų savybių ir galimybių palyginimą, darbe pasirinkta naudoti XML schemas.

2.6.3. XML dokumentų transformavimas

Vienas pagrindinių XML technologijos privalumų – tai galimybė konvertuoti XML duomenis iš vienos formos į kitą, naudojantis bendrinėmis priemonėmis. Technologija, kuri padeda mums tai atlikti, vadinasi XSLT (angl. eXtensible Stylesheet Language for Transformations) – išplečiama stilių kalba transformacijoms.

XSL stilių lentelė – tai transformavimo instrukcijų rinkinys, kuriomis išeitinis XML dokumentas verčiamas kitu dokumentu (nebūtinai XML). Norint atlikti XSLT transformacijas reikalingas XSLT procesorius, kuris transformuoja išeities XML dokumento duomenis, susiedamas juos su XSL stilių lentelės šablonais. Deja, XSLT procesoriai yra ganėtinai lėti, todėl kartais reikia griebtis įvairių gudrybių norint generuoti dokumentus realiu laiku [3,12].



10 pav. XSLT procesoriaus darbo schema [5]

XSLT kalba yra aukšto lygio programavimo kalba, kurios pagrindai ateina iš funkcinio programavimo teorijos. Ji žymiai galingesnė nei panašios paskirties standartai CSS ar CSS2, todėl vidutiniam programuotojui ją įvaldyti yra gana sunku. Lygiai tokia pati situacija buvo ir su SQL jos atsiradimo metu, tai tik dar kartą įrodo, kad XSLT nepaprastai galinga technologija.

XSL Formatavimo objektai – tai instrukcijos, kurios apibrėžia informacijos išsidėstymą ir išvaizdą. XSL-FO leidžia naudoti žymiai sudėtingesnius puslapio elementų išsidėstymo modelius nei HTML+CSS, todėl formatavimo objektais galima aprašyti visą spausdintą knygą. Dažniausiai XSL-FO naudojami norint gaminti PDF failus, kurie skirti verslo duomenų atvaizdavimui. Konvertavimui į galutinį formatą naudojami formatavimo objektų procesoriai.

Nors XML-FO dokumentus galima kurti ir rankomis, bet dažniausiai tam yra rašoma XSL transformacija ir formatavimo objektai generuojami iš duomenų failų.

2.6.4. Duomenų išgavimas ir manipuliavimas

Atskirų XML dokumento dalių išskyrimas ir manipuliavimas jomis yra svarbios veiklos, todėl sukurta daug susijusių specifikacijų, kurios apibrėžia, kaip nurodyti tam tikras XML dokumento vietas ar atlikti informacijos paiešką su parametrais. Toliau aptariamos dvi svarbiausias šios srities kalbos.

XPath yra kalba, skirta XML dokumento dalių adresavimui, kurios sintaksė primena kelius failų sistemoje ar URL. XPath palaiko funkcijas darbui su atrinktais duomenimis. Funkcijos skirtos gauti informaciją apie dokumento atšakas bei darbui su eilutėmis, skaičiais ir loginiais kintamaisiais. Programuotojai gali plėsti prieinamų funkcijų biblioteką. XPath

naudoja kompaktišką, ne XML sintaksę, kad galėtų būti naudojama XML atributuose arba nuorodose. XPath rekomendacija remiasi XSLT, XML Schema, XPointer specifikacijomis.

XPath dirba su abstrakčia, logine XML dokumento struktūra, o ne tekstine sintakse. Ji skirta darbui su vienu XML dokumentu, į kurį žvelgia kaip atšakų medį. XPath užklauso rezultato reikšmės konceptualiai irgi yra laikomos atšakomis. XPath duomenų modelyje egzistuoja šių tipų atšakos: tekstinės, elementų, atributų, šakninės, vardų sričių, apdorojimo instrukcijų, komentarų atšakos.

XQuery – tai bandymas pateikti užklauso kalbą, kuri būtų panašaus funkcionalumo ir turėtų stiprų formalizmą, kaip SQL reliacinėms duomenų bazėms. XQuery yra funkcinė kalba, kurioje kiekviena užklausa yra išraiška. XQuery išraiškos skirstomos į septynias plačias kategorijas: kelio išraiškos (XPath), elementų konstruktoriai, FLWR išraiškos (FROM, LET, WHERE, RETURN), išraiškos su operatoriais ir funkcijomis, sąlyginės išraiškos, įvertinimo išraiškos ir išraiškos, kurios tikrina ar keičia duomenų tipus. Skirtingų išraiškų sintaksė ir semantika labai skiriasi, todėl, kad XQuery kūrimui turėjo įtakos daug skirtingų sistemų.

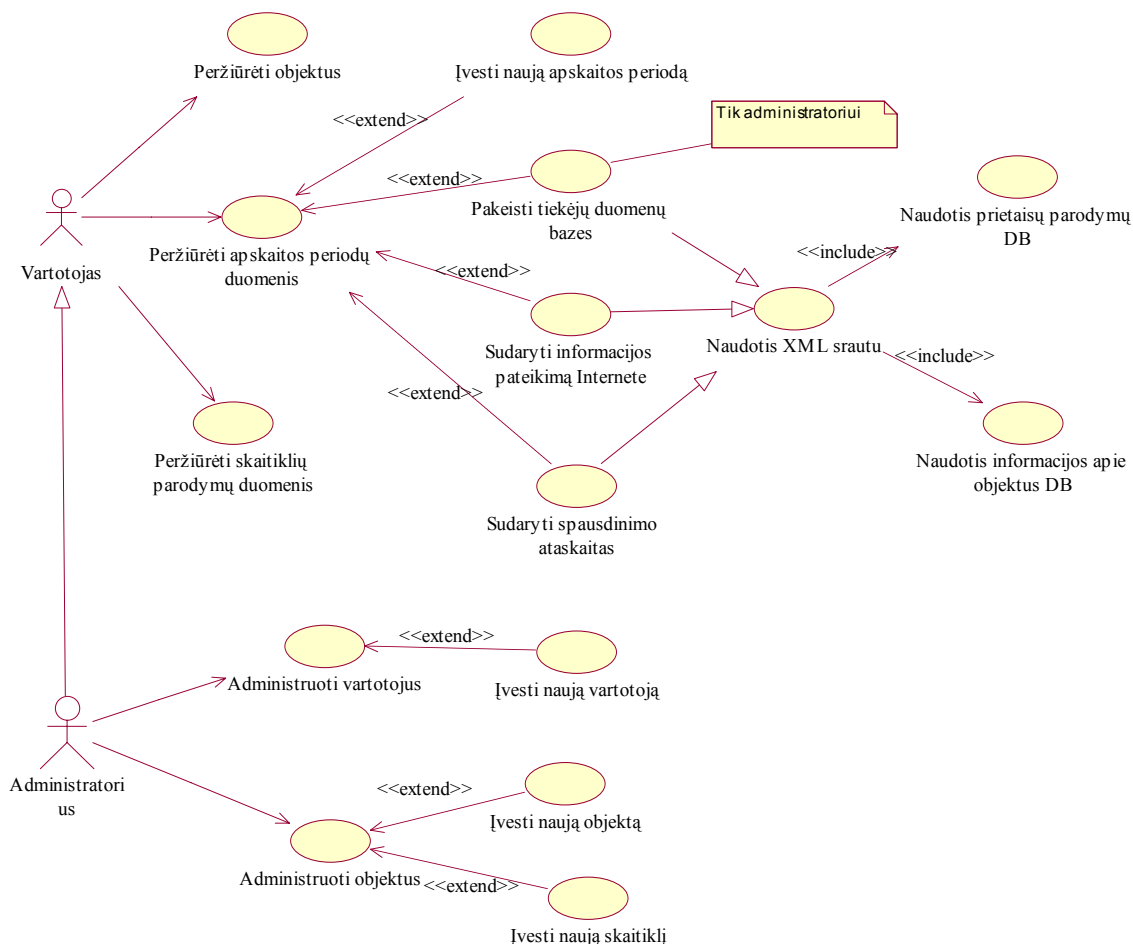
XQuery turi sudėtingą tipų sistemą, kuri paremta XML Schema duomenų tipais ir, kitaip nei XPath, palaiko manipuliavimą dokumento atšakomis. Taip pat XQuery duomenų modelis sukurtas darbui ne tik su pavieniu XML dokumentu, bet ir taisyklingu jo fragmentu, dokumentų seka ar dokumentų fragmentų seka [5].

Atsižvelgiant į darbo srities poreikius, naudoti pasirenkama XPath užklauso kalba, kadangi sudėtingų išraiškų skaičiavimų nėra, dirbama su vienu XML dokumentu, kurio loginė struktūra bus aiškiai aprašyta.

3. SISTEMOS STRUKTŪRA IR REALIZACIJA

3.1. Reikalavimai informacinei sistemai

3.1.1. Vartotojų panaudojimo atvejų diagrama



11 pav. Vartotojų panaudojimo atvejų diagrama

Vartotojai turi būti dviejų lygių – paprasti sistemos naudotojai ir administratoriai.

Sistemos vartotojui turi būti suteikta galimybė peržiūrėti esamą informaciją apie objektus, apskaitos periodus, skaitiklių parodymus, kurti naujus apskaitos periodus, sudaryti reikiamas ataskaitas, jas peržiūrėti ir parsisiųsti ataskaitą spausdinimui.

Administratorius turi turėti galimybę vykdyti visas paprasto vartotojo funkcijas, taip pat administruoti vartotojus ir objektus (sukurti naujus, redaguoti ar išmesti esamus, priskirti vartotojams teises matyti duomenis apie konkrečius objektus ar uždrausti priėjimą prie duomenų), bei siųsti pakeitimus tiekėjų duomenų bazėms.

3.1.2. Specifikacijos panaudojimo atvejams

6 lentelė

Panaudojimo atvejis	<i>Peržiūrėti objektus</i>
Aktoriai	Vartotojas, administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Vartotojas prisijungęs prie sistemos ir jam suteiktos teisės
Pagrindinis įvykių srautas	Sistemos reakcija
1. Pasirenkamas objektų peržiūros mygtukas 2. Pasirinkus norimą objektą, spaudžiama ant jo pavadinimo pelės mygtuku	1.1 Sistema išveda peržiūrai objektų sąrašą, prie kurių prieiti vartotojas turi teises 2.1 Sistema išsima, kad vartotojas prisijungė prie šio objekto 2.2 Sistema išveda peržiūrai objekto informaciją
Po sąlyga	Išvedami esami objekto apskaitos periodai, atsiranda mygtukas skaitiklių parodymų peržiūrėjimui
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, grąžinama į pradinį prisijungimo puslapį
Vykdyto variantai	
Veiklos taisyklės	Vartotojas turi turėti teisę matyti objekto duomenis
Specialūs (nefunkciniai) reikalavimai	Patogi sąsaja su vartotoju
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

7 lentelė

Panaudojimo atvejis	<i>Peržiūrėti apskaitos periodų duomenis</i>
Aktoriai	Vartotojas, administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Vartotojas prisijungęs prie sistemos, pasirinkęs peržiūros objektą
Pagrindinis įvykių srautas	Sistemos reakcija
1. Pasirenkamas apskaitos periodas 2. Pasirenkamas skaitiklių tipas 3. Norėdamas pakeisti konkretų įrašą, vartotojas įrašo norimą reikšmę ir spaudžia mygtuką „Įrašyti“	1.1 Sistema išveda peržiūrai objekto skaitiklių tipų sąrašą 2.1 Sistema išveda peržiūrai objekto apskaitos periodo pasirinkto tipo skaitiklių duomenis 3.1 Pakeisti duomenys įrašomi į duomenų bazę ir puslapis atidaromas naujai
Po sąlyga	Išvedami objekto apskaitos periodo duomenys, atsiranda mygtukas periodo ataskaitų peržiūrėjimui
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, grąžinama į pradinį prisijungimo puslapį
Vykdyto variantai	Vartotojas pasirenka peržiūrimų skaitiklių tipą
Veiklos taisyklės	Vartotojas turi turėti teisę matyti objekto duomenis, įvedant naujus duomenis, jie turi būti korektiški.
Specialūs (nefunkciniai) reikalavimai	Patogi sąsaja su vartotoju, koreguojami duomenys turi saugiai ir korektiškai pasiekti duomenų bazę
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

Panaudojimo atvejis	<i>Pakeisti tiekėjų duomenų bazes</i>
Aktoriai	Administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Prisijungęs vartotojas su administratoriaus teisėmis, pasirinkęs objektą, apskaitos periodą ir sudaromos ataskaitos tipą; suformuoti apskaitos periodo duomenys
Pagrindinis įvykių srautas	Sistemos reakcija
1. Vartotojas pasirenka norimos ataskaitos tipą ir nurodo, kad norima siųsti duomenis į tiekėjų duomenų bazes	1.1 Tikrinama ar yra suformuotas reikiamo tipo ataskaitos XML srautas, jei taip, pereinama prie 1.2 punkto 1.1.1 Jungiamasi prie prietaisų parodymų duomenų bazės ir nuskaityti reikiami duomenys 1.1.2 Jungiamasi prie informacijos apie objektus duomenų bazės ir nuskaityti reikiami duomenys 1.1.3 Nuskaityti duomenys susiejami į vieną bendrą reikiamos formos struktūrą 1.1.4 Suformuoti duomenys saugomi XML formate 1.2 Nuskaitytas XML srautas 1.3 Iš XML srauto formuojami įrašai į tiekėjų duomenų bazes
Po sąlyga	Pakeista tiekėjų duomenų bazė
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, grąžinama į pradinį prisijungimo puslapį; Nepavyko prisijungti prie tiekėjų duomenų bazių, sistema siūlo jungimąsi pakartoti arba nutraukti
Vykdyto variantai	Jei XML srautas suformuotas, jis nėra formuojamas
Veiklos taisyklės	Turi būti sudarytas priėjimas prie tiekėjų duomenų bazių; Turi būti paprašyta įvesti administratoriaus slaptažodį, jei prisijungęs paprastas vartotojas
Specialūs (nefunkciniai) reikalavimai	Patogi sąsaja su vartotoju, saugus ir savalaikis duomenų bazių pakeitimas.
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

Panaudojimo atvejis	<i>Sudaryti informacijos pateikimą Internetu</i>
Aktoriai	Vartotojas, administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Vartotojas prisijungęs prie sistemos, pasirinkęs objektą, apskaitos periodą ir sudaromos ataskaitos tipą; suformuoti apskaitos periodo duomenys
Pagrindinis įvykių srautas	Sistemos reakcija
1. Vartotojas pasirenka norimos ataskaitos tipą ir nurodo kad norima peržiūrėti ataskaitą naršyklės lange	1.1 Tikrinama ar yra suformuotas reikiamo tipo ataskaitos XML srautas, jei taip, pereinama prie 1.2 punkto 1.1.1 Jungiamasi prie prietaisų parodymų duomenų bazės ir nuskaityti reikiami duomenys 1.1.2 Jungiamasi prie informacijos apie objektus duomenų bazės ir nuskaityti reikiami duomenys 1.1.3 Nuskaityti duomenys susiejami į vieną bendrą reikiamos formos struktūrą 1.1.4 Suformuoti duomenys saugomi XML formate

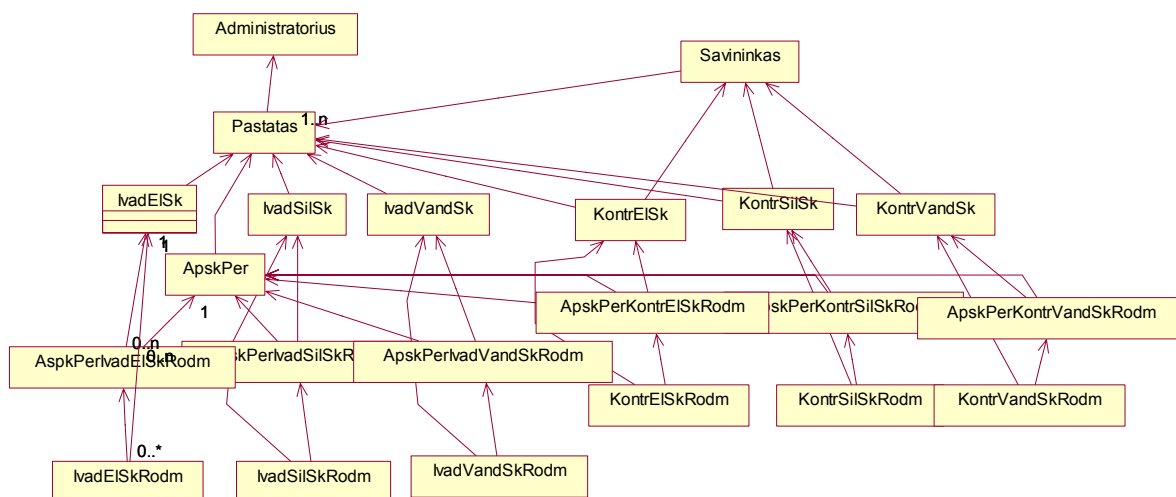
	1.2 Nuskaitomas XML srautas 1.3 XML srautas transformuojamas į HTML formą 1.4 Sudarytas HTML tekstas išvedamas naujame lange
Po sąlyga	Pavaizduota ataskaita vartotojo naršyklės lange
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, gražinama į pradinį prisijungimo puslapį;
Vykdyto variantai	Jei XML srautas suformuotas, jis nėra formuojamas
Veiklos taisyklės	
Specialūs (nefunkciniai) reikalavimai	Ataskaitos formavimas neturi užtrukti ilgą laiko tarpą (ne ilgiau kaip 1 – 2 min.)
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

10 lentelė

Panaudojimo atvejis	<i>Sudaryti spausdinimo ataskaitas</i>
Aktoriai	Vartotojas, administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Vartotojas prisijungęs prie sistemos, pasirinkęs objektą, apskaitos periodą ir sudaromos ataskaitos tipą; suformuoti apskaitos periodo duomenys
Pagrindinis įvykių srautas	Sistemos reakcija
1. Vartotojas pasirenka norimos ataskaitos tipą ir nurodo, kad norima sudaryti ataskaitą spausdinimui	1.1 Tikrinama ar yra suformuotas reikiamo tipo ataskaitos XML srautas, jei taip, pereinama prie 1.2 punkto 1.1.1 Jungiamasi prie prietaisų parodymų duomenų bazės ir nuskaityti reikiami duomenys 1.1.2 Jungiamasi prie informacijos apie objektus duomenų bazės ir nuskaityti reikiami duomenys 1.1.3 Nuskaityti duomenys susiejami į vieną bendrą reikiamos formos struktūrą 1.1.4 Suformuoti duomenys saugomi XML formate 1.2 Nuskaitomas XML srautas 1.3 Iš XML srauto formuojama ataskaita PDF formoje 1.4 Sudarytas PDF ataskaita atidaroma arba išsaugoma
Po sąlyga	Sudaryta spausdinimo ataskaita
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, gražinama į pradinį prisijungimo puslapį;
Vykdyto variantai	Jei XML srautas suformuotas, jis nėra formuojamas
Veiklos taisyklės	
Specialūs (nefunkciniai) reikalavimai	Ataskaitos formavimas neturi užtrukti ilgą laiko tarpą (ne ilgiau kaip 1 – 2 min.)
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

Panaudojimo atvejis	<i>Peržiūrėti skaitiklių parodymų duomenis</i>
Aktoriai	Vartotojas, administratorius
Sistema	Technologinės informacijos centras
Prieš sąlyga	Vartotojas prisijungęs prie sistemos, pasirinkęs peržiūrimą objektą
Pagrindinis įvykių srautas	Sistemos reakcija
1. Vartotojas pasirenka norimą skaitiklių tipą 2. Vartotojas pasirenka peržiūros pradžios ir pabaigos datas	1.1 Sistema išveda peržiūros pradžios ir pabaigos pasirinkimo formą 2.1 Sistema išveda pasirinktų skaitiklių parodymus pasirinktoms datoms 2.2 Pasirinkus konkretų skaitiklį ar jo vieną skalę, sistema išveda parodymus vaizdinėje grafiko formoje
Po sąlyga	Pavaizduoti skaitiklių parodymai
Alternatyvos (nesėkmės atvejai)	Vartotojo prisijungimo laikas baigėsi, grąžinama į pradinį prisijungimo puslapį; Nurodytai datai parodymai neužfiksuoti, sistema vaizduoja artimiausius parodymus
Vykdyto variantai	Vartotojas pasirenka peržiūrimų skaitiklių tipą
Veiklos taisyklės	Vartotojas turi įvesti pradžios ir pabaigos datas, kurių parodymus norės pamatyti
Specialūs (nefunkciniai) reikalavimai	Vaizduojant grafikus turi būti sudaryta galimybė vartotojui pasirinkti reikšmių ėmimo iš parodymų duomenų bazės norimus periodus
Kitos sistemos, su kuriomis sąveikauja sistema	Nėra

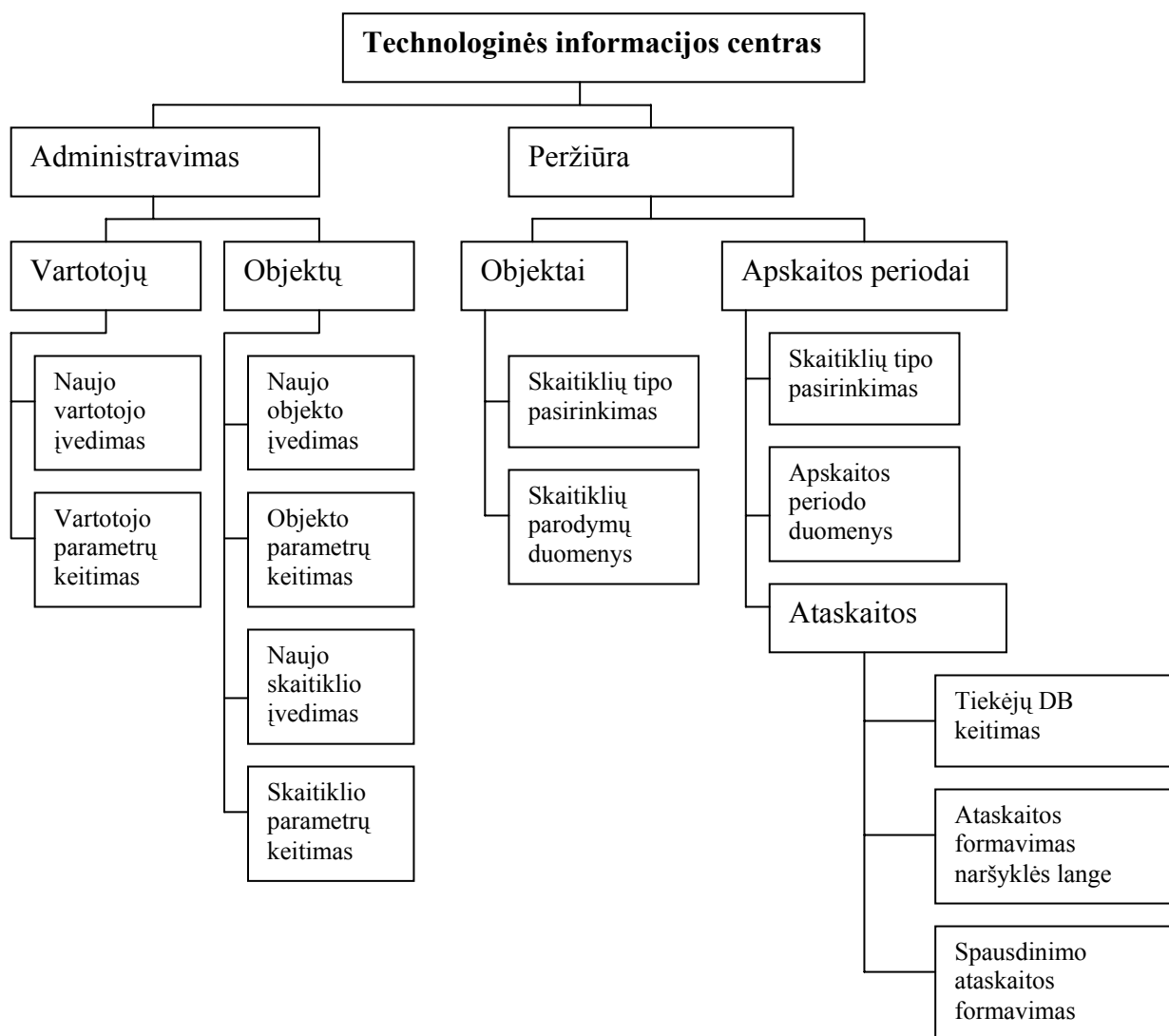
3.1.3. Dalykinės srities klasių diagrama



12 pav. Dalykinės srities klasių diagrama

Pagrindinės sistemos klasės skirstomos į dvi grupes: klasės priklausančios objekto (pastato) informacijai – įvadiniai skaitikliai ir kt., ir klasės, priklausančios pastato savininkų (nuomininkų, butų savininkų) informacijai – kontroliniai skaitikliai ir kt.

3.1.4. Vartotojų sąsajos modelis



13 pav. Vartotojų sąsajos modelis

3.1.5. Papildomi reikalavimai sistemai

Funkcionalumui

Sistema, naudodamasi prietaisų parodymų duomenimis, turi formuoti apskaitos periodų duomenis ir pagal nustatytą algoritmą sudaryti reikiamas ataskaitas. Numatytas naujų duomenų įvedimas, taip pat esamų redagavimas ir šalinimas. Pradiniai duomenys – sistemos administratoriaus rolė ir kiti reikalingi parametrai sistemos veikimui turi būti suvedami diegimo metu.

Sistema turi dirbti ypač tiksliai. Formuojami ataskaitiniai duomenys privalo visiškai atitikti esamus parodymų duomenis. Kiekvienos ataskaitos konkrečių duomenų tikslumas (skaičių eilė) turi atitikti konkrečius nurodymus.

Sistema turi integruoti duomenis į tiekėjų duomenų bazes pagal jų nurodytą duomenų struktūrą. Visos kuriamos ataskaitos saugomos XML standartu.

Turi būti užtikrintas priėjimo prie duomenų saugumas. Kiekvienam sistemos vartotojui turi būti nurodyta prie kurių duomenų jis turi teisę prieiti. Neužregistruoti sistemoje vartotojai matyti duomenų negali.

Patikimumui

Sistema turi pilnai funkcionuoti. Apsisaugojimui nuo nenumatytų gedimų ir duomenų praradimų, duomenys turi būti archyvuojami ir saugomi kitoje, fiziškai atskirtoje archyvineje duomenų bazėje. Duomenų atstatymo galimybė. Klaidingų duomenų įvedimo atvejais sistema privalo pateikti atitinkamą pranešimą ir duomenų į duomenų bazes neįrašyti.

Patogumui

Sistemos išvaizdai specialių reikalavimų nėra, vartotojo sąsaja turi būti lengvai suprantama, pritaikyta įvairioms aplinkoms. Vartotojo programinei įrangai turi būti keliami minimalūs reikalavimai.

Efektyvumui

Ataskaitų formavimas, atvaizdavimas ir persiuntimas turi užimti kuo mažiau laiko. Formuojant ataskaitą negali būti sugaišta daugiau kaip 2 – 3 minutės. Visi taikomųjų programų uždaviniai turi būti vykdomi sistemos serveryje, ne kliento pusėje.

Perkeliamumui

Sistema turi turėti įdiegimo programą. Numatytas įdiegimas skirtingose operacinėse platformose.

Priežiūros savybės

Sistema turi dirbti stabiliai, įvykus gedimams turi būti numatyta duomenų atstatymo galimybė.

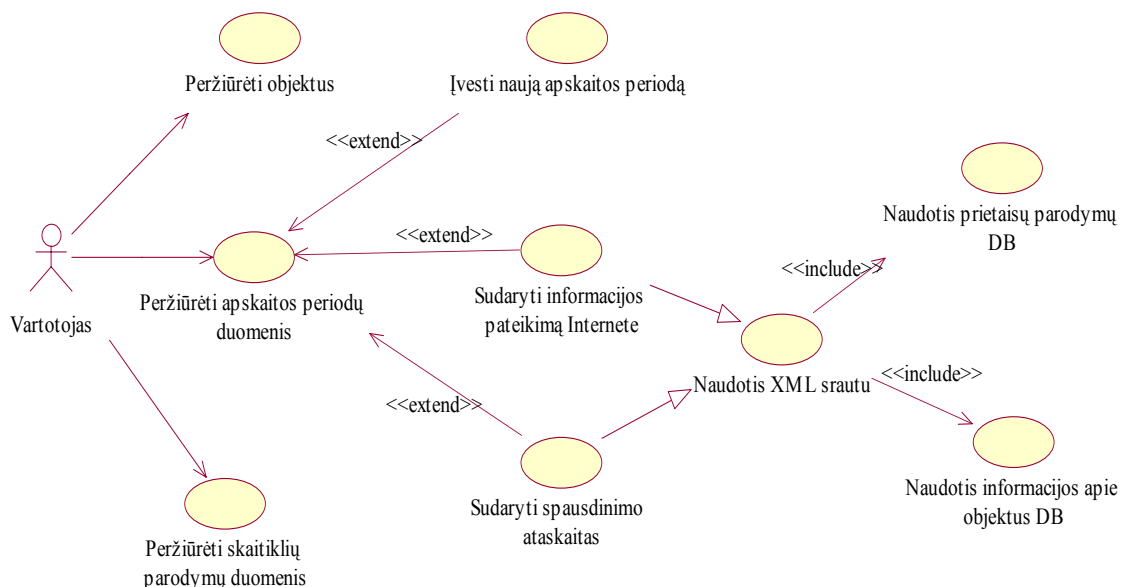
3.2. Techninės informacijos centro projektas

3.2.1. Projekto tikslas

Suprojektuoti ir realizuoti analizės dalyje išnagrinėtą techninės informacijos centro kompiuterinę sistemą.

3.2.2. Sistemos panaudojimo atvejų diagramos

Vartotojo panaudojimo atvejų diagrama



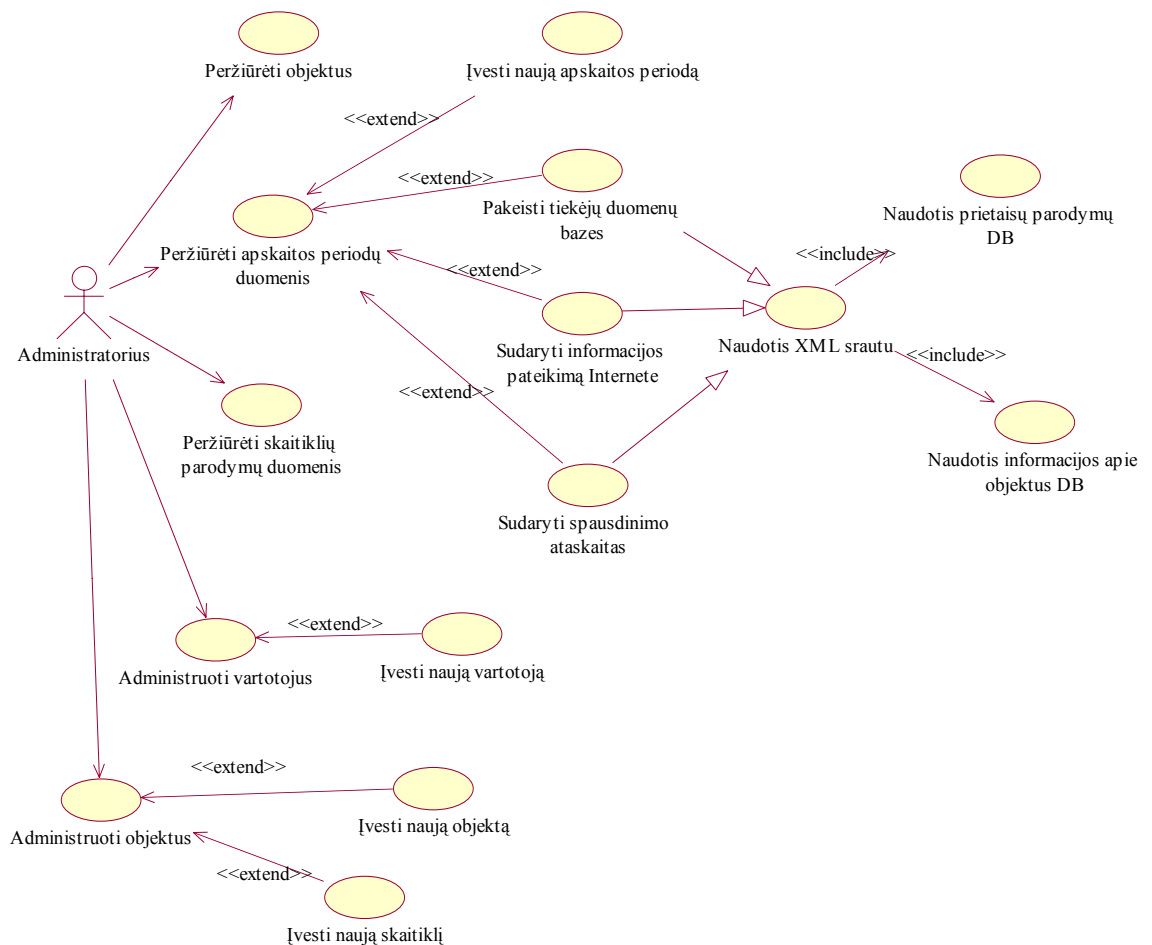
14 pav. Vartotojo panaudojimo atvejų diagrama

„Vartotojas“ – tai bet kuris sistemoje užregistruotas vartotojas. Jis gali peržiūrėti apskaitos periodų, skaitiklių parodymų duomenis, kurti naujus apskaitos periodus. Gali sudaryti norimas ataskaitas (tam sistema prisijungia ir naudojami prietaisų parodymų duomenų baze ir informacijos apie objektus duomenų baze) ir peržiūrėti jas naršyklės lange arba parsisiųsti ataskaitą spausdinimo formoje.

Administratoriaus panaudojimo atvejų diagrama

„Administratorius“ – tai sistemoje užregistruotas vartotojas su administratoriaus teisėmis. Administratorius gali vykdyti visas paprasto vartotojo funkcijas. Taip pat gali kurti naują vartotoją, naują objektą, skaitiklius, juos redaguoti, ištrinti.

Administratorius turi teisę pasirinkti norimą ataskaitą ir jos duomenis nusiųsti tiekėjams (tam sistema prisijungia ir naudojami tiekėjo duomenų baze), nusiunčiant pakeitimą į tiekėjų duomenų bazes.



15 pav. Administratoriaus panaudojimo atvejų diagrama

3.2.3. Panaudojimo atvejų scenarijų diagramos

Pateikiamos tik pagrindinių panaudojimo atvejų diagramos, kurios atspindi darbo idėją ir rezultata.

Tiekėjų duomenų bazių pakeitimas

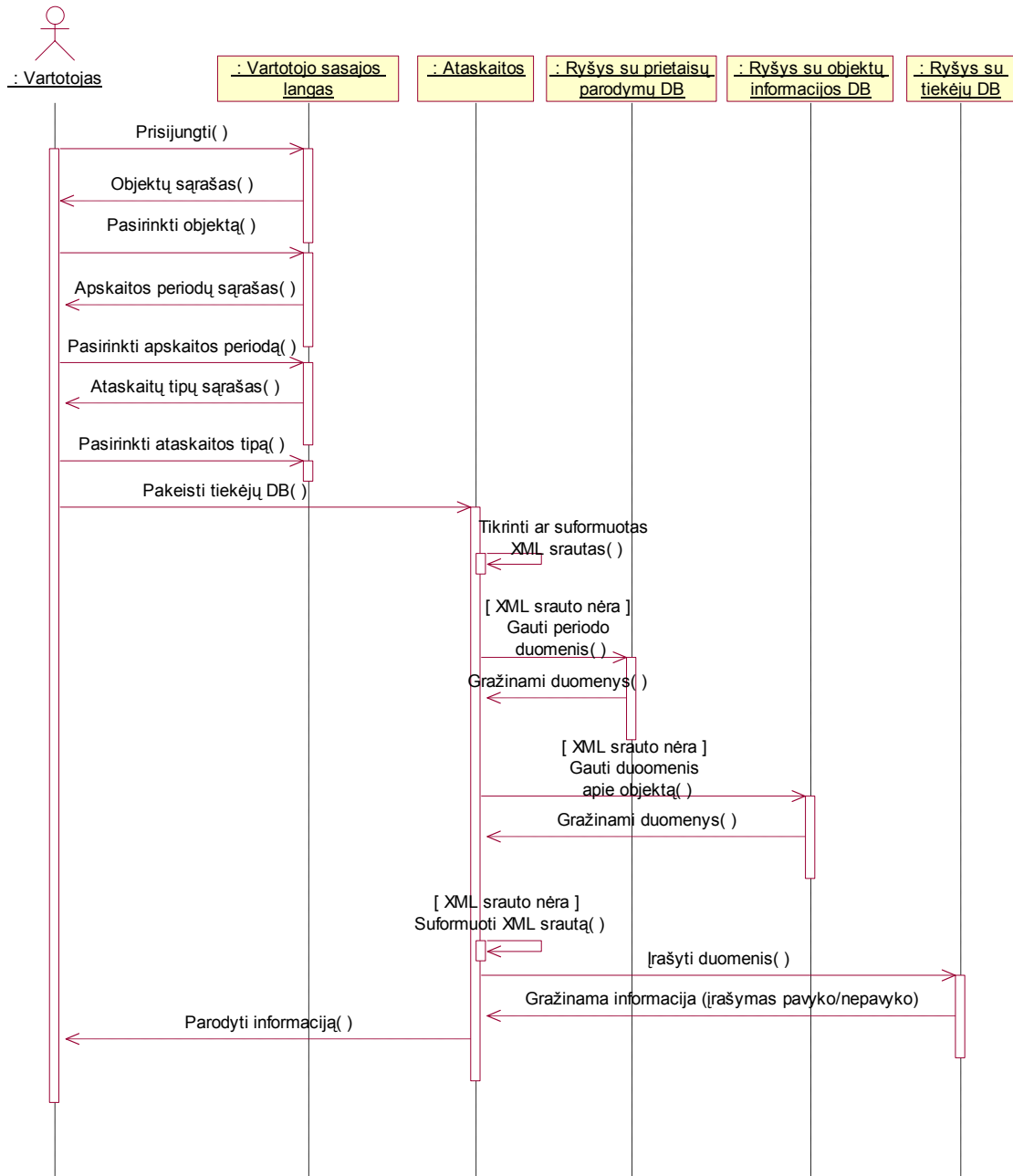
Norėdamas išsiųsti duomenis tiekėjams, atnaujindamas jų duomenų bazes, sistemos vartotojas, turintis administratoriaus teises, turi prisijungti prie sistemos, pasirinkti norimą objektą (pastatą), apskaitos periodą, reikiamą formuoti ataskaitos tipą ir pasirinkti, kad norima siųsti duomenis paslaugų tiekėjui.

Sistema tuomet tikrina, ar yra suformuotas XML srautas. Jei ne, tuomet sistema jungiasi prie prietaisų parodymų duomenų bazės ir užklausią reikiamo apskaitos periodo duomenų. Tada jungiamasi prie informacijos apie objektus duomenų bazės ir užklausiama reikiamo objekto duomenų. Iš gautų duomenų formuojamas XML duomenų struktūra.

Jei sistemoje XML srautas jau buvo suformuotas, tuomet jo formuoti nereikia. Šiuo atžvilgiu sistema tampa gan universali, nes tą pačią suformuotą XML struktūrą gali naudoti

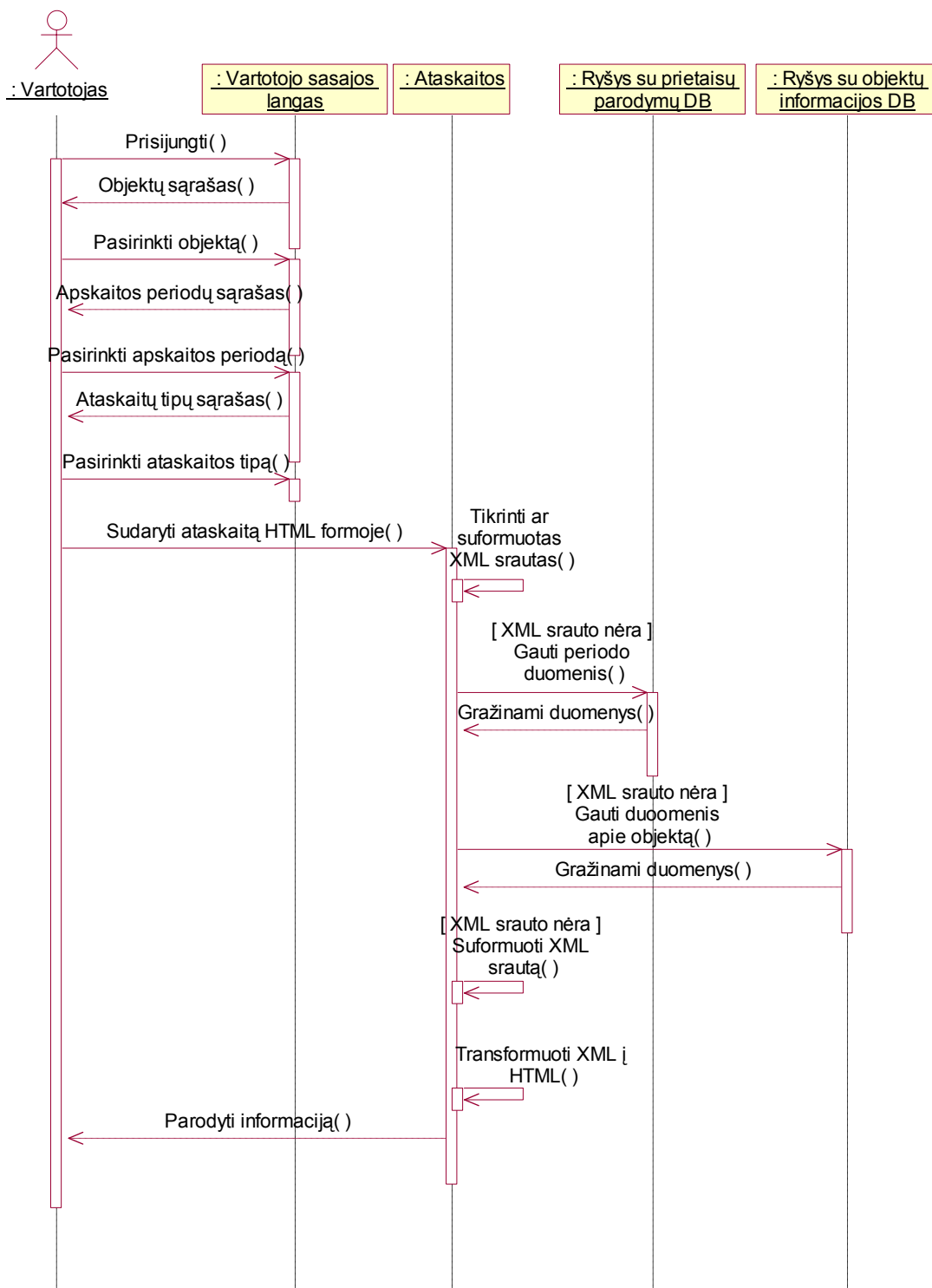
keliuose panaudojimo atvejuose: pakeičiant tiekėjų duomenų bazes, pavaizduojant informaciją vartotojams Interneto naršyklės lange, bei formuojant spausdinimo ataskaitas.

Suformuota XML struktūra keičiama į reikiamą duomenų bazės modelį ir siunčiami pakeitimo įrašai tiekėjų duomenų bazėms. Vartotojui ekrane išvedamas pranešimas apie pavykusį arba nepavykusį duomenų persiuntimą.



16 pav. Panaudojimo atvejo „Tiekėjų duomenų bazės pakeitimas“ scenarijus

Sudaryti informacijos pateikimą Internetu



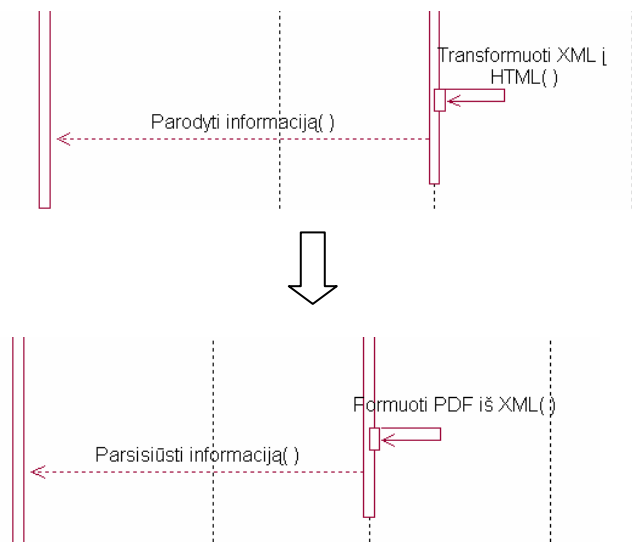
17 pav. Panaudojimo atvejo „Sudaryti informacijos pateikimą Internetu“ scenarijus

Sistemos veikimo scenarijaus seka panaši į aprašytą „Tiekėjų duomenų bazės pakeitimo“ scenarijaus seką. Suformuotas XML srautas čia nesiunčiamas tiekėjui, bet

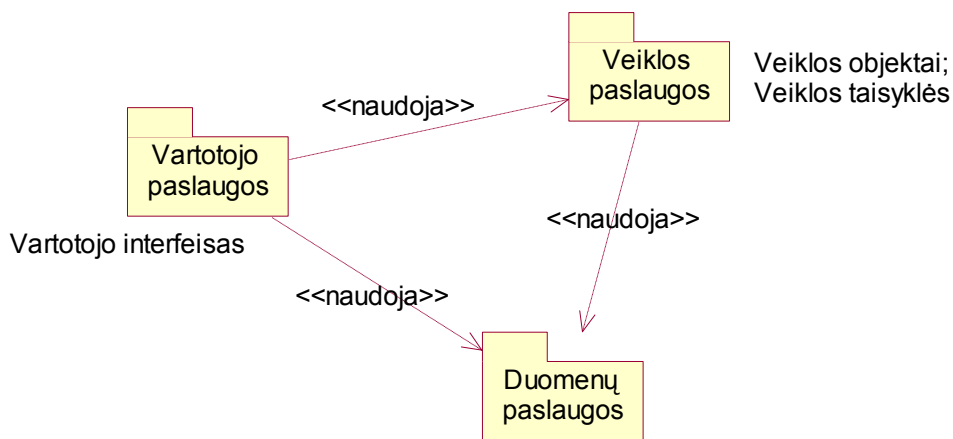
transformuojamas į HTML formą ir siunčiamas vartotojui, kuris gali peržiūrėti ataskaitą savo Interneto naršyklės lange.

Sudaryti ataskaitą spausdinimui

Scenarijus analogiškas „Informacijos pateikimo Internete“ scenarijui. Skiriasi tik paskutinės sistemos vykdymo sekos. Vietoj transformavimo XML srauto į HTML formą, čia formuojamas PDF failas ir siunčiamas vartotojui:



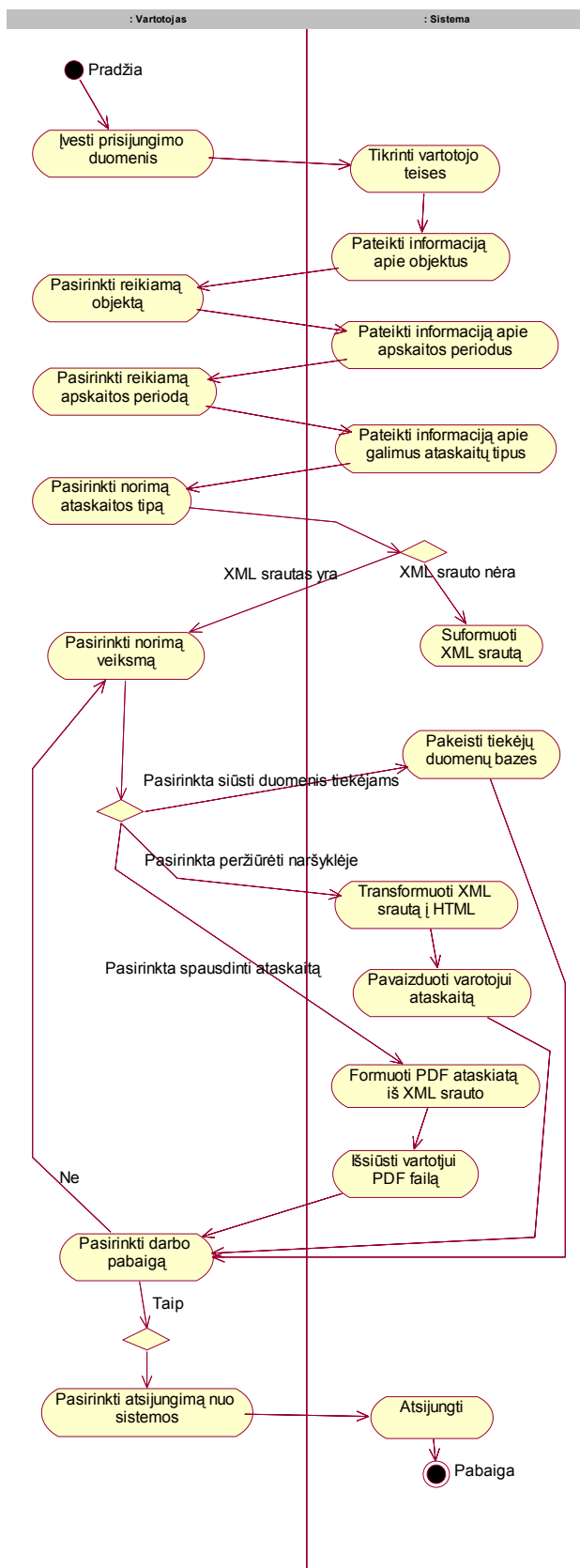
3.2.4. Sistemos architektūra



18 pav. Sistemos architektūra

Taikomosios programos architektūra atrodo taip. Veiklos paslaugos – tai visi objektai ir jų veiklos taisyklės (sudaryti informacijos pateikimą Internete, sudaryti ataskaitą spausdinimui, siūsti duomenis tiekėjams ir kt.). Vartotojo paslaugos – tai vartotojo sąsaja su sistema. Šioje sistemoje visos paslaugos realizuotos Interneto naršyklės pagalba. Duomenų paslaugos – tai visi ryšiai su duomenų bazėmis (prisijungimas, duomenų gavimas, pakeitimas, išsiuntimas).

3.2.5. Sistemos elgsenos modelis



19 pav. Sistemos elgsenos modelis

3.2.6. Duomenų bazės modelis

Prietaisų parodymų duomenų bazės modelis (MySQL)

Building – duomenys apie objektus (pastatus, daugiabučius namus, verslo centrus ir t.t.)

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
Code	Bigint(20)	Pastato unikalus kodas
Adresas	Varchar(100)	Pastato adresas
Picture	Varchar(100)	Pastato nuotrauka

Accper – apskaitos periodų duomenys (visuose objektuose sąskaitos sudaromas pagal tam tikrus apskaitos periodus, pvz., daugiabučiuose namuose apskaitos periodas yra mėnuo)

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDBuilding	Bigint(20)	Nuoroda į pastatą, kuriam priklauso apskaitos periodas
Begin	Date	Apskaitos periodo pradžia
End	Date	Apskaitos periodo pabaiga
Remarc	Varchar(100)	Pastaba

User – aprašo sistemos vartotojus

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
ID2	Bigint(20)	Antras identifikacinis numeris, naudojamas vartotojų surišimui su pastatais
Name	Varchar(20)	Vartotojo vardas (pavadinimas)
Login	Varchar(20)	Prisijungimo vardas
Passwd	Varchar(100)	Vartotojo slaptažodis (saugomas užkoduotoje formoje)
Level	Smallint(6)	Vartotojo teisių lygis

Passwd – lentelė vartotojų ir objektų surišimui, nurodo kurie vartotojai turi teises prieiti prie konkrečių objektų

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDUser	Bigint(20)	Nuoroda į vartotoją
IDBuilding	Bigint(20)	Nuoroda į objektą (pastatą)

Ownerlodger – lentelė skirta aprašyti objekto patalpų savininkams (butų savininkams, nuomuotojams ir pan.)

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
Name	Varchar(20)	Vardas (pavadinimas)
IDBuilding	Bigint(20)	Nuoroda į pastatą, kuriam priklauso savininkas

Metertype – aprašo skaitiklių tipus (elektros/vandens/šilumos, įvadiniai/kontroliniai)

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
Code	Smallint(6)	Tipo unikalus kodas
TableName	Varchar(20)	Duomenų bazės lentelės pavadinimas, kuriame saugomi šio tipo skaitiklių duomenys
Description	Varchar(50)	Skaitiklių tipo apibūdinimas

Toliau pateikiamos lentelės, skirtos įvadinių elektros skaitiklių parodymų duomenims, kitų skaitiklių (įvadinių/kontrolinių, vandens/šilumos) – analogiškos.

Accpermainelmeterrd – įvadinių elektros skaitiklių apskaitos periodų duomenys

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDBuilding	Bigint(20)	Nuoroda į pastatą, kuriam priklauso šie parodymai
IDMainElMeter	Bigint(20)	Nuoroda į skaitiklį, kuriam priklauso šie parodymai
IDAccPer	Bigint(20)	Nuoroda į apskaitos periodą, kuriam priklauso šie parodymai
IDElScaleIdentCod	Bigint(20)	Nuoroda į skaitiklio skalę, kuriam priklauso šie parodymai
Pmonth	Double	Daugiklio koeficientas, iš kurio reikia dauginti parodymų duomenis
From	Double	Parodymas nuo
To	Double	Parodymas iki
Remarc	Varchar(100)	Pastaba

Elmetertarif – aprašo elektros skaitiklių tarifus

Pavadinimas	Tipas	Reikšmė
Type	Bigint(20)	Identifikacinis numeris
Description	Varchar(100)	Tarifo apibūdinimas
Remarc	Varchar(100)	Pastaba

Elscaleidentcod – aprašo elektros skaitiklio skales

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDElMeterTarif	Bigint(20)	Nuoroda į skaitiklio tarifą, kuriam ši skalė priklauso
Description	Varchar(100)	Skalės apibūdinimas
Remarc	Varchar(100)	Pastaba

Mainelmeter – aprašo įvadinius elektros skaitiklius

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDBuilding	Bigint(20)	Nuoroda į pastatą, kuriam skaitiklis priklauso
IDMeterType	Bigint(20)	Nuoroda į skaitiklių tipą, kuriam skaitiklis priklauso
IndNb	Varchar(20)	Skaitiklio indikatoriaus numeris
IndProdCod	Int(11)	Indikatoriaus gamintojo kodas
IndInstallDate	Date	Indikatoriaus instaliavimo data
IndReviseDate	Date	Indikatoriaus revizijos data
IsActive	Tyniint(1)	Ar skaitiklis aktyvus (veikiantis)
TarifCode	Bigint(20)	Nuoroda į tarifą (tarifo kodas)
MeterNb	Varchar(100)	Skaitiklio numeris
MeterInstallDate	Date	Skaitiklio instaliavimo data
IndDevKoeff	Double	Indikatoriaus poslinkio koeficientas (skaitiklio ir indikatoriaus parodymų skirtumas)
IndMultiplKoeff	Double	Indikatoriaus daugiklio koeficientas
MeterMultiplKoeff	Double	Skaitiklio daugiklio koeficientas
MeterProdCod	Bigint(20)	Skaitiklio gamintojo kodas
MeterAccPtCod	Varchar(100)	Apskaitos taško numeris
Remarc	Varchar(100)	Pastaba

MeterProducer – aprašo skaitiklių gamintojus

Pavadinimas	Tipas	Reikšmė
Code	Bigint(20)	Identifikacinis numeris
Description	Varchar(100)	Gamintojo apibūdinimas
Remarc	Varchar(100)	Pastaba

Prdmainelmeterrd – įvadinųjų elektros skaitiklių parodymai

Pavadinimas	Tipas	Reikšmė
ID	Bigint(20)	Identifikacinis numeris
IDMainMeter	Bigint(20)	Nuoroda į skaitiklį, kuriam šie parodymai priklauso
Time	Datetime	Laikas
WT1P	Double	Pirmo tarifo aktyviosios energijos suvartojimas
WT1M	Double	Pirmo tarifo aktyviosios energijos sugeneravimas
QT1P	Double	Pirmo tarifo reaktyviosios energijos suvartojimas
QT1M	Double	Pirmo tarifo reaktyviosios energijos sugeneravimas
WT2P	Double	Antro tarifo aktyviosios energijos suvartojimas
WT2M	Double	Antro tarifo aktyviosios energijos sugeneravimas
QT2P	Double	Antro tarifo reaktyviosios energijos suvartojimas
QT2M	Double	Antro tarifo reaktyviosios energijos sugeneravimas
WT3P	Double	Trečio tarifo aktyviosios energijos suvartojimas
WT3M	Double	Trečio tarifo aktyviosios energijos sugeneravimas
QT3P	Double	Trečio tarifo reaktyviosios energijos suvartojimas
QT3M	Double	Trečio tarifo reaktyviosios energijos sugeneravimas
WT4P	Double	Ketvirto tarifo aktyviosios energijos suvartojimas
WT4M	Double	Ketvirto tarifo aktyviosios energijos sugeneravimas
QT4P	Double	Ketvirto tarifo reaktyviosios energijos suvartojimas
QT4M	Double	Ketvirto tarifo reaktyviosios energijos sugeneravimas
Remarc	Varchar(100)	Pastaba

Informacijos apie objektus duomenų bazės modelis (MS Access) – pateikiamos tik sistemos naudojamos lentelės vienam skaitiklių tipų informacijai identifikuoti (kitos analogiškos)

T Pastatas – duomenys apie pastatus

Pavadinimas	Tipas	Reikšmė
IDPastatas	Autonumber	Identifikacinis numeris
NamoId	Text	Pastato unikalus kodas
Miestas	Text	Miesto pavadinimas
Gatve	Text	Gatvės pavadinimas
Namas	Text	Namo numeris
NmApsildPl	Number	Namo bendras apšildomas plotas
NmApsildTur	Number	Namo apšildomas tūris
NmGyvSk	Number	Gyventojų skaičius
EIVartKod	Text	Elektros vartotojo kodas
SilVartKod	Text	Šilumos vartotojo kodas
SilBySilSk	Yes/No	Skaičiavimai atliekami pagal šilumos skaitiklius (ne daliklius)
KodNmSilSuv	Number	Namo šilumos suvartojimo paslaugos kodas
KodNmSilSuvPaslDet	Number	Namo šilumos suvartojimo paslaugos detalizavimo kodas
IsKvVandSuv	Yes/No	Ar skaičiuojamas karšto vandens suvartojimas
KodNmKvVandSuv	Number	Namo karšto vandens suvartojimo paslaugos kodas
IsSvVandSuv	Yes/No	Ar skaičiuojamas šalto vandens suvartojimas
KodNmSvVandSuv	Number	Namo šalto vandens suvartojimo paslaugos kodas

T Nuomininkas – aprašo pastato patalpos savininko duomenis

Pavadinimas	Tipas	Reikšmė
IdNuomininkas	Autonumber	Identifikacinis numeris
ApskTipas	Number	Įtrauktas į automatizuotą kaštų paskirstymą ar ne
Kodas	Text	Unikalus vartotojo kodas
PilnasPav	Text	Pilnas vartotojo pavadinimas (vardas, pavardė)
SutrPav	Text	Sutrumpintas pavadinimas
ImKodas	Text	Nuomininko įmonės kodas
NuomRegAdresas	Text	Nuomininko įmonės adresas (nesutampa su ofiso adresu)
Tel	Text	Telefono numeris
IsActive	Yes/No	Ar aktyvus (gyvena)
ApmokestPagrPl	Number	Pagrindinis patalpų plotas
ApmokestPapPl	Number	Papildomas patalpų plotas (rūsys, pan.)

T PastatIvSk_El – aprašo pastato įvadinčius elektros skaitiklius

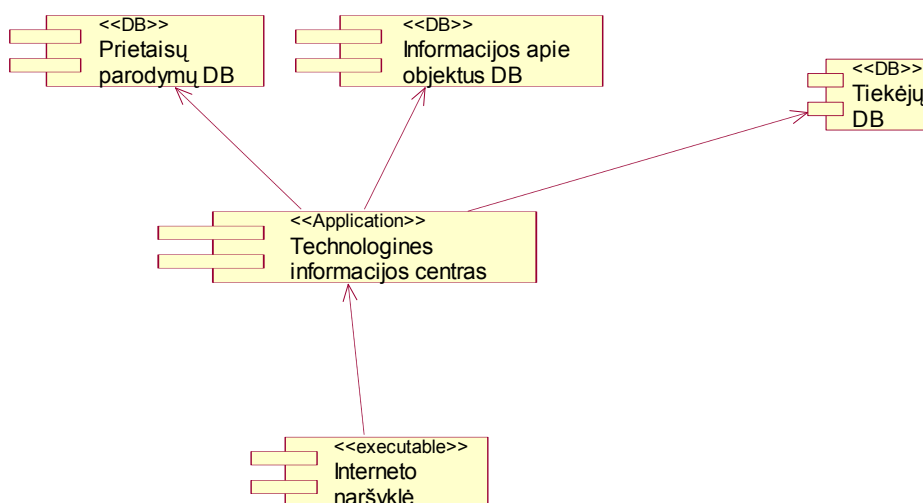
Pavadinimas	Tipas	Reikšmė
IdPastatIvSk_El	AutoNumber	Identifikacinis numeris
Kodas	Number	Unikalus skaitiklio kodas
IdPastatIvSk_El_Tarif	Number	Nuoroda į skaitiklio tarifų lentelę, kuriam tarifui skaitiklis priklauso
IndNrBt	Number	Indikatoriaus numeris bute (kiekviename bute pradedamas nuo 1)
IndTipas	Text	Indikatoriaus tipas
IndNumeris	Text	Indikatoriaus numeris
SkApskTskKodas	Text	Apskaitos taško kodas
SkId	Text	Skaitiklio identifikatorius, kuri nurodo el. tinklai
SkPastVieta	Text	Skaitiklio pastatymo vieta
SkNumeris	Text	Skaitiklio numeris
IsActive	Yes/No	Ar aktyvus (veikiantis)

Sudarytos ataskaitos XML schemas aprašymas (pateiktas vienas ataskaitos tipas – elektros suvartojimo ataskaitą elektros tinklams):

Pavadinimas	Tipas	Reikšmė
Adress	Tekstas	Objekto adresas
Code	Tekstas	Objekto kodas
Id	Tekstas	Skalės identifikacinis numeris
Iki	Tekstas	Parodymas iki
Multiplkoeff	Tekstas	Daugiklis
Note	(title, subtitle, periodBegin, periodEnd, owner, ownerContact, code, building+)	Visa ataskaita
Nuo	Tekstas	Parodymas nuo
Building	(adress, pointNr, skNr, scale+)	Objektas
Owner	Tekstas	Objekto savininkas
ownerContact	Tekstas	Objekto savininko kontaktiniai duomenys
periodBegin	Tekstas	Periodo pradžia
periodEnd	Tekstas	Periodo pabaiga
pointNr	Tekstas	Apskaitos taško numeris
Scale	(id, Pmen, iki, nuo, multfkoeff)	Elektros skaitiklio skalė
skNr	Tekstas	Skaitiklio numeris
Subtitle	Tekstas	Ataskaitos paantraštė
Title	Tekstas	Ataskaitos pavadinimas

3.2.7. Realizacijos modelis

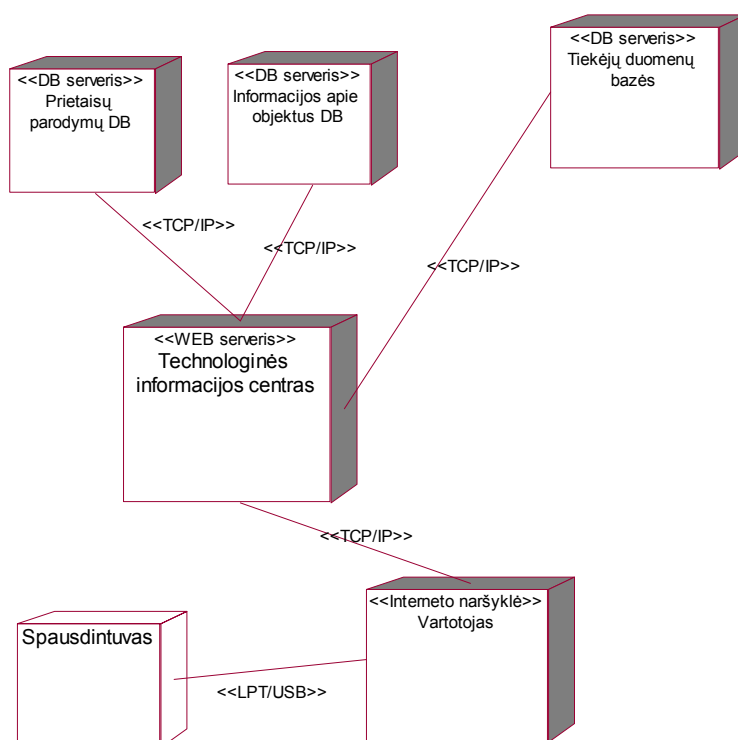
Komponentų diagrama



20 pav. Komponentų diagrama

Sistema sudaro šie pagrindiniai komponentai: duomenų bazės, iš kurių imami duomenys (prietaisų parodymų DB ir informacijos apie objektus DB); tiekėjų duomenų bazės, į kurias išsiunčiami duomenys; sistemos serveris, kuriame veikia taikomoji programa ir vartotojo pusėje esanti Interneto naršyklė.

Paskirstymo diagrama



21 pav. Paskirstymo diagrama

Įdiegus informacinę sistemą, visas valdymas ir naudojimas vyksta per paprastą Interneto naršyklės langą.

3.2.8. Testavimo modelis

Testavimo metu tikrinama kaip funkcionuoja sukurta informacinė sistema, ar teisingai atliekamos reikalavimuose apibrėžtos funkcijos:

- Prisijungimo prie sistemos teisingumas.
- Korektiškas vartotojo teisių suteikimas.
- Vartotojų administravimas: naujų vartotojų sukūrimas, esamų koregavimas, teisių keitimas.
- Objektų administravimas: naujų objektų sukūrimas, esamų koregavimas, naujų ir esamų skaitiklių koregavimas.
- Informacijos apie objektus, apskaitos periodus, skaitiklių parodymų korektiška peržiūra.
- Pasirinkto periodo duomenų grafiko atvaizdavimas.
- Ataskaitos pavaizdavimas naršyklės lange.
- Ataskaitos spausdinimo formos sukūrimas (PDF).
- Korektiškas tiekėjų duomenų bazių keitimas, duomenų persiuntimas.
- Ataskaitos formavimo laikas (turi atitikti reikalavimus).

3.2.9. Reikalavimai sistemos funkcionavimo palaikymui

Serveriui

Programinė įranga: Tomcat 5.0 (ar naujesnis) WEB serveris.

MySQL 4.1 (ar naujesnė) duomenų bazė

Vartotojui (klientui)

Kompiuteris su Interneto naršykle (rekomenduojama IE 5 arba naujesne)

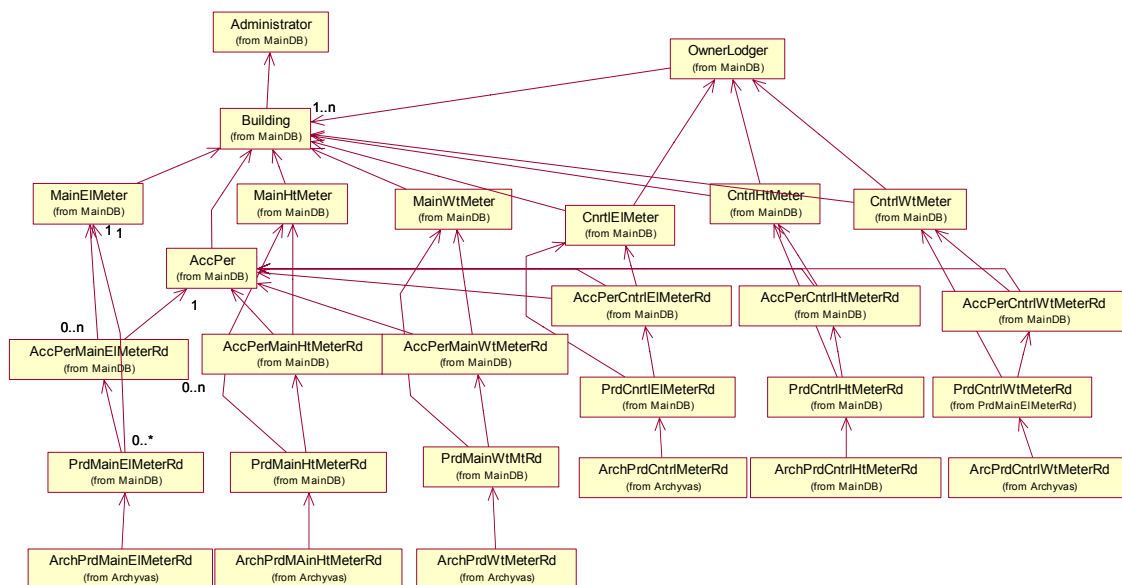
Ataskaitų spausdinimo formų peržiūrai ir spausdinimui: Adobe Acrobat Reader.

4. EKSPERIMENTINIS TYRIMAS

4.1. Duomenų integracija

Suprojektuota sistema yra realizuota ir šiuo metu testuojama UAB „Katreks“ aplinkoje su realiais duomenimis. Sistema apjungia duomenis iš dviejų skirtingų duomenų bazių – abonentų parametrų (MS Access) ir prietaisų parodymų (MySQL). Ataskaitų generavimui ir tiekėjų duomenų bazių pakeitimui reikalinga informacija tiek apie abonentus (pavadinimai, kodai), tiek apie jų suvartojimo duomenis (šilumos, elektros, vandens). Tam tikslui apjungiami duomenis ir formuojamas bendras XML srautas.

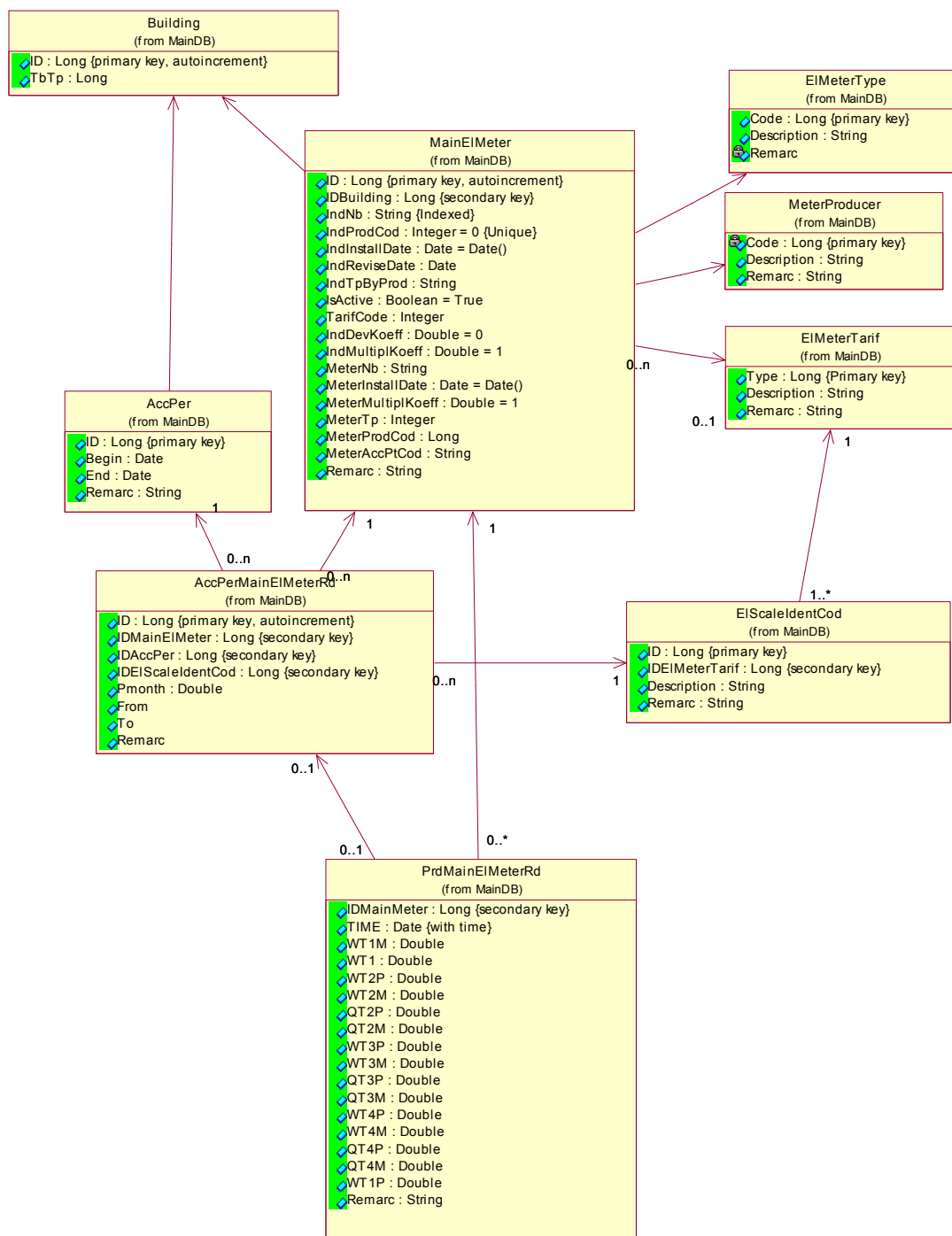
Prietaisų parodymų duomenų bazės schema pavaizduota 22 paveiksle.



22 pav. Prietaisų parodymų duomenų bazės schema

Duomenų bazės lentelės skirstomos į dvi pagrindines grupes: įvadinių (pastato) skaitiklių parodymai ir kontrolinių (nuomininkų, butų) skaitiklių parodymai.

Sistema sudaro ir generuoja ataskaitas visiems skaitiklių tipams ir grupėms (įvadiniams ir kontroliniams; šilumos, vandens ir elektros). Toliau pateikiamas detali schema vienam tipui – įvadinių elektros skaitiklių grupės lentelės (kiti tipai analogiški).



23 pav. Prietaisų parodymų, skirtų įvadiniams elektros skaitikliams, duomenų bazės schema

Duomenų bazė schema atitinka 3.2.6 skyrelyje „Duomenų bazės modelis“ aprašytą modelį ir laukų reikšmes.

Abonentų parametrų duomenų bazės pagrindinių lentelių schema pavaizduota 24 paveiksle.

Duomenis integruojami iš prietaisų parodymų duomenų bazės („katreks2“) ir abonentų parametrų duomenų bazės („Namų_administravimas“) į XML failą („pazyma“).

Toliau pateikiamas sugeneruoto XML srauto XML Schema. Parametrų reikšmės atitinka 3.2.6 skyrelyje pateiktas „Sudarytos ataskaitos XML schemas aprašymas“ laukų reikšmės.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Pmen" type="xs:string"/>
  <xs:element name="adres" type="xs:string"/>
  <xs:element name="building">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="adres"/>
        <xs:element ref="pointNr"/>
        <xs:element ref="skNr"/>
        <xs:element ref="scale" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="code" type="xs:string"/>
  <xs:element name="id" type="xs:string"/>
  <xs:element name="iki" type="xs:double"/>
  <xs:element name="multifkoeff" type="xs:double"/>
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="title"/>
        <xs:element ref="subtitle"/>
        <xs:element ref="periodBegin"/>
        <xs:element ref="periodEnd"/>
        <xs:element ref="owner"/>
        <xs:element ref="ownerContact"/>
        <xs:element ref="code"/>
        <xs:element ref="building" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="nuo" type="xs:double"/>
  <xs:element name="owner" type="xs:string"/>
  <xs:element name="ownerContact" type="xs:string"/>
  <xs:element name="periodBegin" type="xs:string"/>
  <xs:element name="periodEnd" type="xs:string"/>
  <xs:element name="pointNr" type="xs:string"/>
  <xs:element name="scale">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="id"/>
        <xs:element ref="Pmen"/>
        <xs:element ref="iki"/>
        <xs:element ref="nuo"/>
        <xs:element ref="skirtumas"/>
        <xs:element ref="multifkoeff"/>
        <xs:element ref="suvargota"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="skNr" type="xs:string"/>
  <xs:element name="skirtumas" type="xs:double"/>
  <xs:element name="subtitle" type="xs:string"/>
  <xs:element name="suvargota" type="xs:double"/>
  <xs:element name="title" type="xs:string"/>
</xs:schema>

```

Sistema sugeneruoja XML failą, kuris vėliau naudojamas perduodant duomenis paslaugų teikėjų duomenų bazių pakeitimams, generuojant ataskaitas ir transformuojant XML srautą į HTML puslapio struktūrą. Pateikiamas sistemos sugeneruotas XML failas:

```

<?xml version="1.0" encoding="UTF-8"?>
<note>
  <title>PAŽYMA</title>
  <subtitle>Apie suvartotą elektros energiją</subtitle>
  <periodBegin>2004 m. sausio mėn. 1 d.</periodBegin>
  <periodEnd>2004 m. liepos mėn. 26 d.</periodEnd>
  <owner>UAB „TURTO VYSTYMO ĮMONĖ“</owner>
  <ownerContact>Savanorių per. 247 Vilnius, tel. 686 630, fax. 686636</ownerContact>
  <code>05811</code>
  <building>
    <adres>Kaunas, Krėvės 14b</adres>
    <pointNr>1</pointNr>
    <skNr>50908</skNr>
    <scale>
      <id>Q+</id>
      <Pmen>123</Pmen>
      <iki>150098</iki>
      <nuo>150091</nuo>
      <skirtumas>7</skirtumas>
      <multifkoeff>1</multifkoeff>
      <suvartota>7</suvartota>
    </scale>
    <scale>
      <id>Q-</id>
      <Pmen/>
      <iki>133367</iki>
      <nuo>133365</nuo>
      <skirtumas>2</skirtumas>
      <multifkoeff>1</multifkoeff>
      <suvartota>2</suvartota>
    </scale>
    <scale>
      <id>T1</id>
      <Pmen/>
      <iki>227315</iki>
      <nuo>227311</nuo>
      <skirtumas>4</skirtumas>
      <multifkoeff>1</multifkoeff>
      <suvartota>4</suvartota>
    </scale>
  </building>
  <building>
    <adres>Kaunas, Krėvės 14b</adres>
    <pointNr>2</pointNr>
    <skNr>50918</skNr>
    <scale>
      <id>Q+</id>
      <Pmen>123</Pmen>
      <iki>150097</iki>
      <nuo>150091</nuo>
      <skirtumas>6</skirtumas>
      <multifkoeff>1</multifkoeff>
      <suvartota>6</suvartota>
    </scale>
    <scale>
      <id>Q-</id>
      <Pmen/>
      <iki>133367</iki>
      <nuo>133365</nuo>
      <skirtumas>2</skirtumas>
      <multifkoeff>1</multifkoeff>
      <suvartota>2</suvartota>
    </scale>
    <scale>
      <id>T1</id>
      <Pmen/>
      <iki>227319</iki>
      <nuo>227311</nuo>
      <skirtumas>8</skirtumas>
      <multifkoeff>2</multifkoeff>
      <suvartota>16</suvartota>
    </scale>
  </building>
</note>

```

4.2. Sistemos valdymas administratoriui

Realizuota prietaisų parodymų duomenų bazės valdymo sistema. Sukurtas Internetinė vartotojo sąsaja. Prisijungęs administratorius gali įvesti naujus, koreguoti ar ištrinti esamus vartotojus, objektus (pastatus) ar objektų prietaisus.

Pasirinkęs objektų administravimą ir norėdamas įvesti naują ar pakoreguoti jau užregistruotą sistemoje objektą, administratorius turi užpildyti sekančią formą:

Objekto įvedimas

Grįžti

Adresas:
Kaunas, Krėvės 14b

Objekto administratorius:
UAB "TURTO VYSTYMO ĮMONĖ"

TbTp (skaitmenys):
300

Vartotojų priėjimas prie objekto:
nmarius2
Marius Kazlauskas



Keisti objekto paveiksluką:

26 pav. Objekto įvedimo langas

Pasirinkus vartotojų administravimą ir norint įvesti naują ar pakoreguoti esamą sistemos vartotoją atsiveria toks sistemos langas:

Vartotojų administravimas

[Atgal į vartotojų sąrašą](#)

Prisijungimo vardas(privalomas, unikalus, be lietuviškų simbolių):

Slaptažodis(privalomas):

Vardas(pavardė):

Identifikacinis numeris(skaitmenys, privalomas, unikalus):

Administratorius:

Priėjimas prie objektų:

Kaunas, Trakų 20

Kaunas, Krėvės 14b

27 pav. Vartotojo įvedimo langas

Norėdamas administruoti objekto prietaisus, administratorius turi pasirinkti prietaisų (skaitiklių) tipą:

Kaunas, Krėvės 14b

Objekto savininkas: UAB "TURTO VYSTYMO ĮMONĖ"

Skaitikliai

Pasirinkite skaitiklių tipą

Pasirinkite skaitiklių tipą

Įvadiniai elektros skaitikliai

Kontroliniai elektros skaitikliai

Įvadiniai šilumos skaitikliai

28 pav. Prietaisų tipo pasirinkimo langas

Pasirinkus norimą tipą atsiveria prietaiso parametrų užpildymo langas (29 pav.). Administratorius gali užpildyti parametrus pasirinkdamas iš juos iš jau esamų arba įvesdamas naujus.

Kaunas, Krėvės 14b

Įvadinio elektros skaitiklio įvedimas

Atgal į įvadinį elektros skaitiklių sąrašą

Skaitiklio numeris: 43329

Skaitiklio instaliavimo data: 2004-10-20

Skaitiklio gamintojas: Elgama

Skaitiklio transf. koef.: 1.0

Indikatoriaus numeris: 43329

Indikatoriaus gamintojas: Elgama

Indikatoriaus instaliavimo data: 2004-10-20

Indikatoriaus revizijos data: 2004-10-20

Indikatoriaus poslinkio koef.: 0.0

Indikatoriaus transf. koef.: 1.0

Aktyvus:

Tarifas: Keturi tarifai, šešios skalės

Apskaitos taško numeris: 3

Pastaba:

Ištrinti skaitiklį (visi duomenys bus ištrinti!)

29 pav. Naujo prietaiso įvedimo langas

4.3. Sistemos valdymas vartotojui

Prisijungęs prie sistemos vartotojas gali peržiūrėti esamus duomenis, kurti naujus apskaitos parodymus, sudaryti periodų parodymų duomenų srautus ir juos peržiūrėti reikiamoje formoje (Internetiniame lange ar PDF formoje).

Pasirinkus apskaitos periodą ir norimą ataskaitos tipą atsiveria sekantis langas:

Kaunas, Krėvės 14b

Apskaitos periodo 1999-10-01 - 2004-10-20 duomenys

Baigti ataskaitų peržiūrą

Elektros skaitiklių ataskaita už periodą:

30 pav. Formuojamos ataskaitos langas

Suformuotus apskaitos periodo duomenis galima peržiūrėti, juos koreguoti, realizuota grafinė parodymo peržiūrėjimo forma (galima rinktis parodymų fiksavimo dažnį, vaizduojamus parametrus). Detalesnis vartotojo vadovas pateikiamas 1 priede „Sistemos naudojimo instrukcija“.

Kaunas, Krėvės 14b

Apskaitos periodo 1999-10-01 - 2004-10-20 duomenys

Baigti įvadinių elektros skaitiklių parodymų peržiūrą

Periodo įvadinių elektros skaitiklių parodymai

Performuoti periodo parodymus

[Skaitiklis nr. 43329. Apskaitos taškas 3](#)
[Skaitiklis nr. 44192. Apskaitos taškas 4](#)
[Skaitiklis nr. 50908. Apskaitos taškas 1](#)
[Skaitiklis nr. 51779. Apskaitos taškas 2](#)

Skalė	Parodymas nuo	Parodymas iki	Skirtumas	Koregavimas
Skaitliukas Nr. 43329				
Q+	310375	311946	1571	<input type="button" value="Įrašyti"/>
Q-	1264	1264	0	<input type="button" value="Įrašyti"/>
T1	115184	115561	377	<input type="button" value="Įrašyti"/>
T2	293220	293979	759	<input type="button" value="Įrašyti"/>
T3	160676	161469	793	<input type="button" value="Įrašyti"/>
T4	147925	148523	598	<input type="button" value="Įrašyti"/>
Skaitliukas Nr. 44192				
Q+	69240	69529	289	<input type="button" value="Įrašyti"/>

31 pav. Apskaitos periodo duomenys

32 pav. pateikiamas Interneto naršyklės lange pavaizduotos ataskaitos vaizdas (XSLT pagalba transformuotas XML srautas) . Sugeneruota ataskaita PDF formoje pateikta 3 priede.

XSL (PDF formavimui) ir XSLT (XML transformavimui į HTML) išeities kodai pateikti kompaktiniame diske.

Atgal

PAŽYMA

Apie suvartotą elektros energiją nuo 1999 m. Spalio mėn. 01 d. iki 2004 m. Spalio mėn. 20 d.

UAB "TURTO VYSTYMO ĮMONĖ"
Savanorių pr. 247 Vilnius, tel. 686 630, fax. 686636

Kodas 05811

№	Adresas	Aplauko arbo Nr.	Skaitiklio Nr.	Skaitiklio ID	P. mėn. kWh	Esdmavo "IKI" arba (W/m ² .C/m ² .kWh)	Esdmavo "NUO" arba (W/m ² .C/m ² .kWh)	Šilumos kWh	Isanf. coef.	Šilumos arba (W/m ² .C/m ² .kWh)
1	Kaunas, Krėvės 14b	1	50908	Q+	0.0	150091.0	149540.0	551.0	1.0	551.0
				Q-	0.0	133365.0	132922.0	443.0	1.0	443.0
				T1	0.0	227311.0	226282.0	1029.0	1.0	1029.0
				T2	0.0	585538.0	583400.0	2138.0	1.0	2138.0
				T3	0.0	351407.0	349157.0	2250.0	1.0	2250.0
				T4	0.0	338131.0	336557.0	1574.0	1.0	1574.0
2	Kaunas, Krėvės 14b	2	51779	Q+	0.0	233945.0	233940.0	5.0	1.0	5.0
				Q-	0.0	141630.0	140376.0	1254.0	1.0	1254.0
				T1	0.0	196271.0	195657.0	614.0	1.0	614.0
				T2	0.0	524454.0	523198.0	1256.0	1.0	1256.0
				T3	0.0	309666.0	308239.0	1427.0	1.0	1427.0
				T4	0.0	247236.0	246726.0	510.0	1.0	510.0
3	Kaunas, Krėvės 14b	3	43320	Q+	0.0	311946.0	310375.0	1571.0	1.0	1571.0
				Q-	0.0	1264.0	1264.0	0.0	1.0	0.0
				T1	0.0	115561.0	115184.0	377.0	1.0	377.0

pav. 32 Sugeneruotos ataskaitos langas

4.4. Sistemos kokybės tyrimas

Pagrindinis projekto kokybės kriterijus – korektiškai veikianti programinė įranga. Pagrindinės sukurtosios programinės įrangos funkcijos – XML srautų formavimas, jų panaudojimas duomenų bazių pakeitimui, Interneto puslapių ir spausdinimo formų generavimui. Sistema šias funkcijas atlieka.

Sudaryta Internetinė vartotojo sąsaja. Numatytas naujų vartotojų, naujų objektų įvedimas, jų koregavimas ir ištrynimasis veikia korektiškai.

Duomenų įvedimo, redagavimo metu vykdoma duomenų kontrolė.

Ataskaitų (XML srautų) formavimo, atvaizdavimo ir persiuntimo laiko analizė:

12 lentelė. Sistemos atliekamų veiksmų laiko palyginimas

Duomenų kiekis	XML suformavimo laikas, s	XML transformavimo į Internetinį puslapį laikas, s	PDF failo formavimo iš XML srauto laikas, s
4 objektai	1312ms	1022ms	8052ms
40 objektų	1643ms	1512ms	13680ms
100 objektų	2764ms	1753ms	23033ms

Apžvelgus sukurtoje informacinėje sistemoje realizuotą funkcionalumą, efektyvumą, galima teigti, kad pagrindiniai projekto kokybės kriterijai yra įvykdyti.

4.5. Tolimesnės sistemos tobulinimo ir plėtojimo galimybės

Sistema yra išbaigta pagal reikalavimus ir paruošta naudojimui. Esant reikalui, sistemą galima atnaujinti pagal pageidaujamus poreikius. Yra galimybė įvesti naujus parametrus – naujo tipo ataskaitas, naujus tiekėjus, naujas atvaizdavimo formas. Duomenų integravimas gali būti vykdomas iš ir į kitokių tipų ir gamintojų duomenų bazes. Tobulinant sistemą, galima realizuoti pilną galimų klaidų ar nestandartinių situacijų aptikimo ir nustatymo mechanizmą. Fiksuojant skaitiklių parodymus, galima daryti atitinkamus sprendimus, jei parodymai kurį laiką nekinta, arba jei yra netipiškai pakitę. Taipgi žinant įvadinių ir kontrolinių prietaisų parodymus galima spręsti apie nesutapimus ir galimas to priežastis.

5. IŠVADOS

Atlikta komunalinių paslaugų automatizavimo įmonės veiklos analizė ir nustatytos kompiuterizavimo galimybės.

Apžvelgtos kelios esamos priemonės, skirtos duomenų lygio kompiuterinių sistemų integracijai ir nustatyta, kad įgyvendinant duomenų integravimą nagrinėjamoje srityje tikslinga apjungti dvi technologijas – XML srautų panaudojimą duomenų integravimui ir XML galimybes PDF bei HTML formatų kūrimui.

Išnagrinėtos XML technologijų galimybės duomenų integracijai ir duomenų atvaizdavimui bei pritaikytos konkrečioje aplinkoje.

Analizuojamoje įmonėje duomenys saugomi keliuose skirtingose duomenų bazėse, todėl reikalingas jų apjungimas. Suprojektuota informacinė sistema, integruojanti šių duomenų bazių duomenis, formuojant XML srautus. Numatytas tų pačių srautų pakartotinas panaudojimas kuriant Internetinį puslapį ir spausdinimo ataskaitas.

Realizuoti programiniai moduliai ir praktiškai pritaikyti komunalinių paslaugų automatizavimo įmonės veikloje. Sistemos pagalba sutaupomas saugomų duomenų kiekis ir paspartinamas generuojamų ataskaitų laikas. Šiuo metu sistema yra testuojama realioje aplinkoje.

6. LITERATŪRA

1. **Boris Lublinsky.** Achieving the Ultimate EAI Implementation. Part 3: Data-Level Integration. EAI Journal, February 20, 2001 [žiūrėta 2003-10-12]. Prieiga per Internetą: <http://www.bijonline.com/Article.asp?ArticleID=301&DepartmentId=6>
2. **David S. Linthicum.** Data Level vs. Method Level. The Design Patterns for EAI. EAI Journal [žiūrėta 2004-02-10]. Prieiga per Internetą: <http://www.bijonline.com/Article.asp?ArticleID=224&DepartmentId=6>
3. **David S. Linthicum.** What XSLT Brings to the Application Integration Party. EAI Journal, August 2001, pages 12 – 15.
4. **Dongwon Lee, Wesley W. Chu.** Comparative Analysis of Six XML Schema Languages [žiūrėta 2004-09-10]. Prieiga per Internetą: <http://www.cobase.cs.ucla.edu/tech-docs/dongwon/ucla-200008.html>
5. **Elliotte Rusty Harold.** Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP and TrAX. 2003 ISBN: 0-201-77186-1.
6. **IBM alpha Works.** XML Data Mediator [žiūrėta 2004-09-10]. Prieiga per Internetą: <https://secure.alphaworks.ibm.com/tech/XI>
7. **IBM.** DB2 Warehouse Manager. Data Warehousing Overview [žiūrėta 2004-09-09]. Prieiga per Internetą: <http://www-306.ibm.com/software/data/db2/datawarehouse/overview.html>
8. **Igor Dayen.** Storing XML in Relational Databases [žiūrėta 2003-12-01]. Prieiga per Internetą: <http://www.xml.com/pub/a/2001/06/20/databases.html>
9. **Joshua Fox, Ph. D.** Active Information Models for Data Transformation. EAI Journal, May 2003, pages 26 – 30.
10. **JP Morgenthal.** XML for Data Integration. EAI Journal, October 2001, pages 13 – 16.
11. **Katherine Hammer.** Data Integrity in e-Business. EAI Journal [žiūrėta 2004-09-10]. Prieiga per Internetą: <http://www.bijonline.com/Article.asp?ArticleID=197&DepartmentId=6>
12. **W3C®.** W3C Technical Reports and Publications [žiūrėta 2004-01-10]. Prieiga per Internetą: <http://www.w3.org/TR/>

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

- CSS – (angl. Cascading Style Sheets) – stilių lentelės
- DB – duomenų bazė
- DOM – (angl. Document Object Model) – dokumento objektinis modelis
- DTD – dokumento tipų aprašymas
- EAI (Enterprise Application Integration) – organizacijų kompiuterinių sistemų integracija
- Flat failas – failas, kuris neturi struktūrinio ryšio tarp jo duomenų elementų
- HTML – hiperteksto žymių kalba
- OPENXML – transakcinė SQL užklausa, palaikanti eilučių rinkinį, suformuotą iš XML dokumento
- PDF – Portable Document Format
- SAX – Simple API for XML
- SQL – struktūrizuota užklausų kalba
- W3C - World Wide Web Consortium
- XML – išplėstinė žymių kalba
- XML schema – XML kalba parašytas dokumentas, skirtas XML dokumentui apibrėžti
- XPath – tai kalba skirta identifikuoti XML dokumento dalis
- Xpointer – yra kuriama sintaksė, nurodanti kreipinius į XML bylos dalis
- XQuery – duomenų paieškos XML dokumentuose užklausų kalba
- XSL-FO – XML pagrįsta žymių kalba aprašanti ir formuojanti XML duomenų atvaizdavimą
- XSLT - eXtensible Stylesheet Language for Transformations

8. SANTRAUKA ANGLŲ KALBA

XML – based data integration

Summary

Using different information systems and correct communication between them are very important problem at this time. Modern companies have a large number of major applications that take care of running the business. At different times, different people used different technologies to write these applications for different purposes.

Data resources can be at many different forms in organization: relational databases, object databases, XML documents and etc. Databases can operate with different systems and use distinct software. This problem must be solved by enterprise application integration (EAI) systems. The project analyzes data level EAI. This type of integration is relatively inexpensive and it doesn't need to incur the expense of changing, testing, and deploying the application.

XML can be a powerful ally for data integration. The main purpose of this project was to analyze usage of XML technologies possibilities for data integration. The problem solved is how to use the same XML flow for a several purposes: the change of another database(s), transforming to a HTML document and forming a PDF document for printing.

These problems are analyzed and solved in the particular range – organization of management of public utility. Here is designed and realized an information system.

9. PRIEDAI

Toliau prieduose pateikiama:

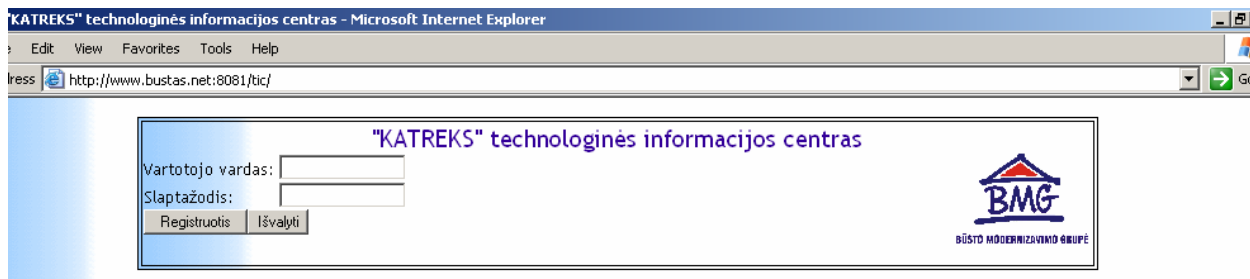
1 priedas. Sistemos naudojimo instrukcija

2 priedas. Devintojoje magistrantų ir doktorantų konferencijoje pristatytas straipsnis „Organizacijos kompiuterinių sistemų integracija duomenų lygyje“

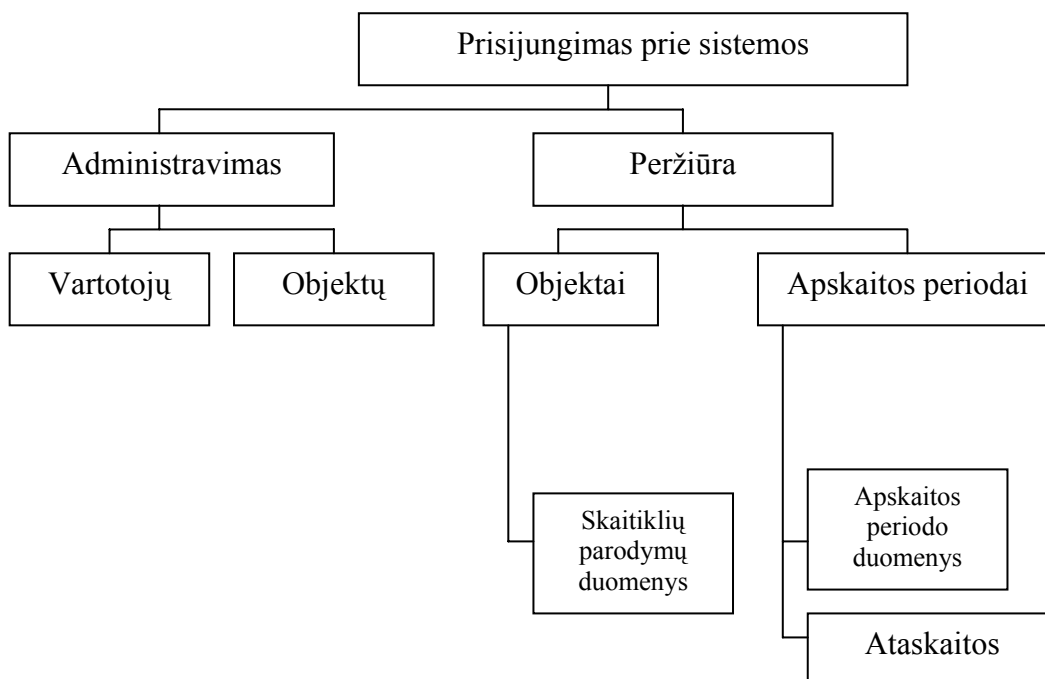
3 priedas. Sistemos sugeneruotos ataskaitos pavyzdys

Sistemos naudojimo instrukcija

Atidarius sistemos puslapį, atsiveria prisijungimo forma:



Pagrindinis sistemos realizacijos vartotojo meniu atrodo taip:



Prisijungus prie sistemos administratoriui, jis gali rinktis meniu punktus – administravimą arba peržiūrą. Prisijungus paprastam vartotojui, atsidaro peržiūros langas.

Administratorius, pasirinkęs administravimą, gali rinktis vartotojų administravimą arba objektų administravimą.

Atsidarius peržiūros langui, vartotojas turi pasirinkti norimą objektą:

"KATREKS" technologinės informacijos centras - Microsoft Internet Explorer

File Edit View Favorites Tools Help


Address <http://www.bustas.net:8081/tic/>

"KATREKS" technologinės informacijos centras

Prisijungęs: Marius Kazlauskas


Atsijungti

Baigti peržiūrą




Objektai

[Kaunas, Krėvės 14b](#)



[Kaunas, Trakų 20](#)



Prisijungus prie reikiamo objekto, aktyvuojasi meniu punktas „Skaitiklių parodymai“. Ekrane vartotojas mato sukurtus apskaitos periodus, jis gali kurti naują periodą.

Pasirinkus meniu punktą „Periodo ataskaitos“ ekrane pasirodo visų galimų suformuoti ataskaitų sąrašas. Pasirinkus reikiamą ataskaitos tipą pasirodo langas, kuriame galima pasirinkti vieną iš galimų veiksmų:

- Suformuoti (arba jei jau yra suformuotas – performuoti ataskaitos XML srautą)
- Pavaizduoti ataskaitą HTML formoje
- Suformuoti ataskaitą PDF formate
- Siųsti XML duomenų srautą tiekėjų duomenų bazių atnaujinimui
- Peržiūrėti suformuotą XML srautą

Puslapio vaizdas atrodo taip:

Kaunas, Krėvės 14b

Apskaitos periodo 1999-10-01 - 2004-10-20 duomenys

Baigti ataskaitų peržiūrą

Elektros skaitiklių ataskaita už periodą:

Suformuoti HTML formoje (Atidaryti kitame puslapyje) Suformuoti PDF formoj

Parsisiūsti XML formoje

Performuoti ataskaitą XML formoje **Ištrinti ataskaitą**

Pasirinkus skaitiklių parodymų peržiūrą atsidaro visų objekte registruotų skaitiklių tipų sąrašas. Reikia pasirinkti norimą tipą. Tuomet atsiveria langas su norimo peržiūrėti periodo datos įvedimu:

Kaunas, Krėvės 14b

Įvadinių elektros skaitiklių parodymai

Baigti įvadinių elektros skaitiklių parodymų peržiūrą

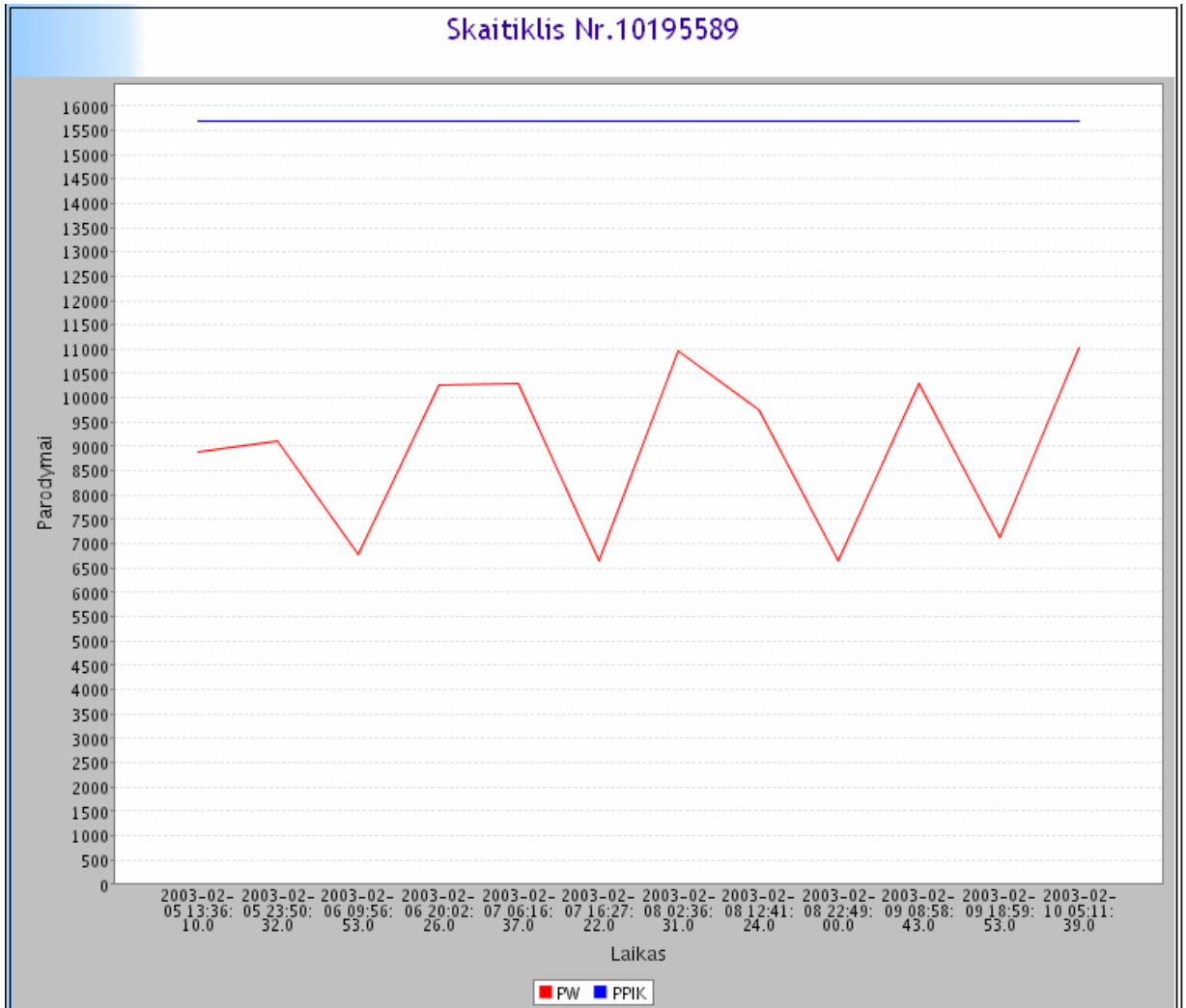
Nuo: 1999-11-01 iki: 2004-11-25 Įvesti Atstatyti

Lapkritis 1999

Sek	Pir	Ant	Tre	Ket	Pen	Šeš
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Pasirinkus reikiamas datas, ekrane pavaizduojami esami duomenys. Galima peržiūrėti parodymus grafinėje formoje:

Skaitiklis Nr.10195589



Analogiškai galima peržiūrėti apskaitos periodų parodymų duomenis.

**VYTAUTO DIDŽIOJO UNIVERSITETAS
KAUNO TECHNOLOGIJOS UNIVERSITETAS
VILNIAUS UNIVERSITETAS**

**INFORMACINĖ VISUOMENĖ IR
UNIVERSITETINĖS STUDIJOS**

9-OJI TARPUNIVERSITETINĖ MAGISTRANTŲ IR DOKTORANTŲ KONFERENCIJA

PRANEŠIMŲ MEDŽIAGA

KAUNAS
2004

ORGANIZACIJOS KOMPIUTERINIŲ SISTEMŲ INTEGRACIJA DUOMENŲ LYGyje

Marius Kazlauskas

Kauno technologijos universitetas, Informacijos sistemų katedra

Studentu g. 50, LT-3031 Kaunas, Lietuva

Apžvelgiama svarbi šiai dienai informacinių sistemų suderinamumo organizacijose problema. Duomenų integravimo uždaviniai nagrinėjami konkrečioje srityje – komunalinių paslaugų valdyme. Integracija duomenų lygyje ir galimi būdai išspręsti skirtingų sistemų suderinamumą. XML galimybių panaudojimas duomenų integravimui ir pilnai automatizuojant namo valdymą, duomenų surinkimą, perdavimą ir patogų pateikimą vartotojui.

Įvadas

Skirtingų informacinių sistemų naudojimas ir galimas korektiškas bendravimas tarp jų tampa vis aktualesnis. Organizacijų duomenų resursai gali būti daugybėje skirtingų formų: reliacinių duomenų bazių lentelėse, XML (eXtensible Markup Language) dokumentuose, objektinėse duomenų bazėse, elektroniniuose duomenų apsikeitimo pranešimuose ir daugelyje kitų. Kadangi pasaulyje naudojamos įvairios aparatūrinės ir programinės platformos, skirtingų sistemų bendravimas įmanomas tik tuo atveju, kai nustatytas ryšys tarp jų leidžia informacinius mainus. Organizacijų kompiuterinių sistemų integracija (EAI - Enterprise Application Integration) turi išspręsti bendrą skirtingų sistemų naudojamą verslo procesais ir duomenimis.

Duomenų lygio EAI

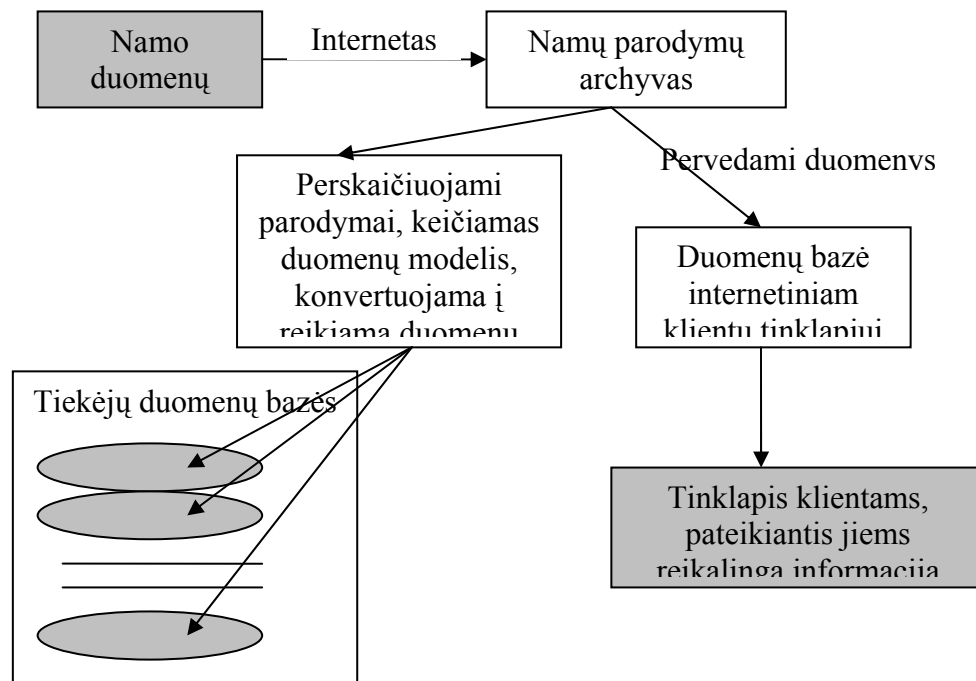
Informacinės sistemos integravimą galima vykdyti keliais būdais:

- duomenų lygio integracija
- uždavinio sąsajos lygio integracija
- metodo lygio integracija
- vartotojo sąsajos lygio integracija

Nagrinėsime integravimą duomenų lygyje. Šio būdo privalumas yra pigumas, nes nėra daromi taikomųjų programų pakeitimai, testavimai ir įdiegimai. Duomenų lygio integracija apeina programinę logiką, paima ir perkelia duomenis tiesiogiai iš vienos sistemos į kitą. Tai yra paprastas ir greitai realizuojamas organizacijoje sprendimas. Šiam būdui įgyvendinti užtenka žinoti kur yra duomenys, surinkti informaciją apie duomenis, ir taikant veiklos principus nustatyti kurie duomenų šrautai reikalingi realizacijai. Tai gali būti informacijos išgavimas iš vienos duomenų bazės, informacijos apdorojimas (jei reikalingas) ir atitinkamas kitos bazės pakeitimas. Tai naudinga turint galvoje, kad esamų programų dažniausiai keisti nereikia. Duomenų lygio integracija yra bene lanksčiausiai ir lengviausiai įgyvendinama iš visų integracinių metodų. Tai yra todėl, kad čia dalijamasi duomenimis kaip galutiniu, nedalomu elementu, kas leidžia keisti, papildyti ar panaikinti dalį informacijos pagal reikalingą jos svarbą [2].

Komunalinių paslaugų automatizavimas

Komunalinių paslaugų atsiskaitymo automatizavimas apima įvadinį ir butų skaitiklių duomenų surinkimą ir pateikimą, supaprastina sąskaitų sudarymą. Žmonėms nereikia nurašinėti skaitiklių parodymus, pildyti lapelius ir patiems atlikinėti skaičiavimus. Vykdydamas tokią sistemą, išskyla duomenų integracijos poreikis. Sistema surenka duomenis apie gyventojų paslaugų naudojimą (vandens, šilumos, elektros) ir saugo jų archyvą. Problema susidaro kai reikia sudaryti ir teikti reikalingas ataskaitas paslaugų tiekėjams (šilumos, vandens, elektros suvartojimą), taip pat realizuojant Internetinę sąsają klientams, pavaizduojant kiekvieno kliento informaciją jam patogią formą. Kiekviena šių sistemų atskirai naudoja skirtingus duomenų modelius, skirtingas duomenų bazines, joms reikalinga skirtinga informacija. Pvz., iš klientų šilumos daviklių gaunama informacija apie kiekvieno prietaiso parodymus, to prietaiso parametrus ir kt., o šilumos tiekėjui reikalinga tik bendras kliento šilumos suvartojimas per nurodytą laikotarpį. Duomenų modelio pakeitimui naudojama sistemos struktūra pavaizduota 1 pav. Namo prietaisų parodymai kaupiami vietinėje duomenų bazėje ir kartą paroje sinchronizuojami su namų duomenų archyvu. Reikiamos struktūros duomenims gauti (pvz. šilumos tiekėjams, Internetiniam puslapiui), taikomos atskiros programos arba duomenys perdujami tiesiog rankiniu būdu.



1 pav. Esamos duomenų keitimo struktūros schema.

Duomenų integravimo sistema, panaudojant XML technologijas

Minėtas sistemos principas tikrai nėra patogus. Sistemos tikslas yra perduoti duomenis iš pradinio taško – namo duomenų bazės į galinį – tiekėjams ir klientams (paveikslėliuose pavaizduota patamsintu fonu), pakeitus juos pagal poreikį. Galima koreguoti naudojamus duomenų tipus, perprogramuoti visas esamas sistemas. Tačiau šiuo atveju priimtinesnis ir pigesnis sprendimas būtų palikti esamą duomenų modelį ir, naudojantis standartizuotu, XML pagrįstu principu, keisti duomenų formatą į pageidaujamą tiesiog perduodant duomenis iš vienos duomenų bazės į kitą.

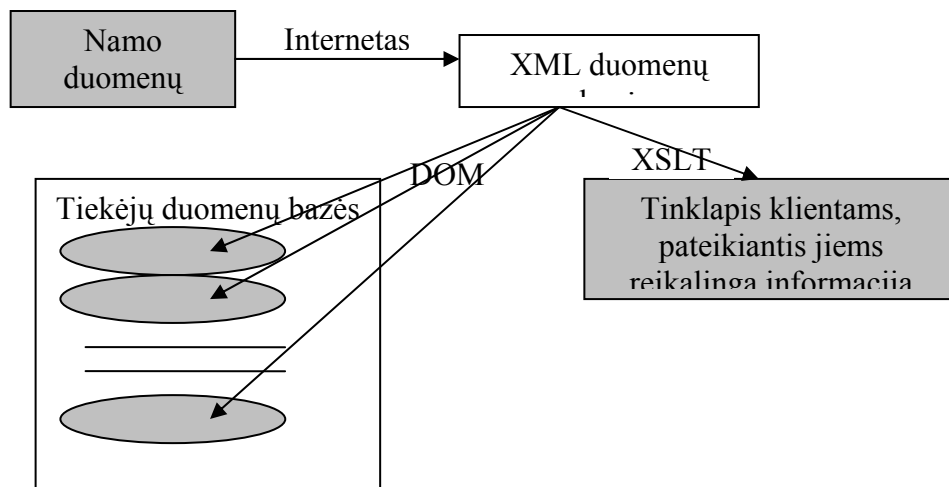
XML yra metakalba, skirta formuoti kitas kalbas. Iš esmės tai yra duomenų formatas. Jis aprašo žymes ir jų pateikimą, kurias visos programos, suprantančios XML, gali suprasti. Tai taisyklių rinkinys, apibrėžiantis, kaip tekstiniu formatu aprašyti struktūrinę informaciją. Žymės naudojamos tik informacijos atskyrimui, interpretavimą paliekant dokumentą apdorojančiai programai. Šiuo būdu galima aprašyti bet kokio sudėtingumo informaciją. Taikant XML galimybes, bet kada galima pakeisti esamus formatus, pridėdant ar panaikinant žymes, patikrinti ar duomenys yra struktūriškai teisingi, realizuota galimybė atlikti transformaciją į įvairius formatus, visiškai nepriklausomumas nuo sistemos platformos ar gamintojo [3].

Saugant duomenis standartizuota forma (šiuo atveju XML), vienas iš būdų keisti duomenų modelį juos perduodant yra XSLT (eXtensible Stylesheet Language for Transformations) naudojimas. Nagrinėjamai sistemai tai būtų vienas iš geriausių sprendimų informacijos išvedimui į tinklapį, skirtą klientams. XSLT yra kalba vieno XML dokumento transformavimui į kitą (pvz. HTML), duodanti galimybę keisti tiek duomenų modelį tiek pačius duomenis proceso metu. Transformacijų atlikimui naudojamas XSLT procesorius, kurio realizacija yra ir daugumoje programavimo kalbų [1].

Apdorojant duomenis, teikiant informaciją tiekėjams, neišvengiamai reikalinga atlikti skaičiavimus. Tam reikia turėti duomenis patogioje objekto tipo formoje, todėl reikia nuskaityti XML dokumentą. Tai atlieka programos, vadinamos XML nagrinėtojai (XML Parser). Vienas iš XML dokumentų nagrinėjimo metodų – dokumento objektinis modelis (DOM - Document Object Model). Naudojant DOM, nuskaitomas visas XML dokumentas ir sukuriama jo medžio tipo struktūra. Programiškai galima manipuluoti dokumentu, pertvarkant, pridėdant ar ištrinant reikiamas atšakas, kurti naujas ir užrašyti objekcinį modelį atgal į XML failą ar tiesiogiai į reikiamą reliacinę duomenų bazę [4].

Nagrinėjamu atveju, panaudojant DOM technologiją, duomenų reikšmės konvertuojamos į nurodyto objekto tipo medį, kad galima būtų juos patogiai apdoroti. Informacinės sistemos kūrėjui sudaroma galimybė dirbti su objektu, žinant tik jo struktūrą, nepriklausomai nuo pasirinktos programavimo kalbos. Taigi, turėdami reikalingo tipo objektą, galime sukurti norimos struktūros išvesties dokumentą, taip pat galime tiesiogiai išvesti informaciją į duomenų bazę. XML atvaizdavimas į reliacinę duomenų bazę vykdomas dviem žingsniais. Pirma, iš XML duomenų modelio schemas yra generuojama objekcinė schema, kuri atvaizduojama

į duomenų bazės schemą. Šios schemų transformacijos yra palaikomos daugelio programavimo kalbų. Siūloma duomenų transformavimo schema pavaizduota 2 pav. Namų parodymų duomenys kaupiami XML duomenų bazėje. Pritaikant XML technologijas (pvz. DOM, XSLT), duomenys transformuojami į reikiamą formą ir struktūrą.



2 pav. Siūlomos, taikant XML technologijas, struktūros schema.

Išvados

Vykdamas duomenų apdorojimą komunalinių paslaugų automatizuotame procese reikalingas duomenų modelio keitimas. Vienas iš geriausių sprendimų duomenų integravimui iš vienos sistemos į kitą yra XML technologijų panaudojimas. Taikant standartizuotą duomenų aprašymą ir technologijas, sudarančias galimybes patogiai pateikti ir dirbti su reikiama duomenimis, šį procesą galima visiškai automatizuoti.

Literatūros sąrašas

- [1] **D. Linthicum**. What XSLT Brings to the Application Integration Party. Prieiga per Internetą: <<http://www.bijonline.com/PDF/XSLTLinthicum202bb.pdf>>.
- [2] **D. Linthicum**. Data Level vs. Method Level The Design Patterns for EAI. Prieiga per Internetą: <<http://www.bijonline.com/Article.asp?ArticleID=224&DepartmentId=6>>
- [3] **JP Morgenthal**. XML for Data Integration. Prieiga per Internetą: <<http://www.bijonline.com/PDF/XMLDIMorgenhalbb.pdf>>
- [4] XML DOM [interaktyvus]. Prieiga per Internetą: <<http://www.w3schools.com/dom/default.asp>>

Data level EAI

Here is overviewed an important problem of IT systems compatibility in the nowadays organization. Data integration problems are analyzed in the particular range - management of public utility. Data level integration and possible ways to resolve compatibility of different systems. Usage of XML possibilities for data integration and full automation of house management, data collection, transferring and convenient presentation to user.

PAŽYMA

Apie suvartotą elektros energiją nuo 1999 m. Spalio mėn. 01d. iki 2004 m. Spalio mėn. 20d.

UAB "TURTO VYSTYMO ĮMONĖ"

Savanorių pr. 247 Vilnius, tel. 686 630, fax. 686636

Kodas 05811

Nr	Adresas	Apskaitos taško Nr.	Skaitiklio Nr.	Skalės ID	P mėn. kW	Rodmuo "IKI" arba (Wsum.;Qsum.),kWh	Rodmuo "NUO" arba (Wsum.;Qsum.),kWh	Skirtumas, kWh	Transf. koef.	Suvertota arba (Wmėn.;Qmėn.),kWh
1	Kaunas, Krėvės 14b	1	50908	Q+	0.0	150091.0	149540.0	551.0	1.0	551.0
				Q-	0.0	133365.0	132922.0	443.0	1.0	443.0
				T1	0.0	227311.0	226282.0	1029.0	1.0	1029.0
				T2	0.0	585538.0	583400.0	2138.0	1.0	2138.0
				T3	0.0	351407.0	349157.0	2250.0	1.0	2250.0
				T4	0.0	338131.0	336557.0	1574.0	1.0	1574.0
2	Kaunas, Krėvės 14b	2	51779	Q+	0.0	233945.0	233940.0	5.0	1.0	5.0
				Q-	0.0	141630.0	140376.0	1254.0	1.0	1254.0
				T1	0.0	196271.0	195657.0	614.0	1.0	614.0
				T2	0.0	524454.0	523198.0	1256.0	1.0	1256.0
				T3	0.0	309666.0	308239.0	1427.0	1.0	1427.0
				T4	0.0	247236.0	246726.0	510.0	1.0	510.0

3	Kaunas, Krėvės 14b	3	43329	Q+	0.0	311946.0	310375.0	1571.0	1.0	1571.0
				Q-	0.0	1264.0	1264.0	0.0	1.0	0.0
				T1	0.0	115561.0	115184.0	377.0	1.0	377.0
				T2	0.0	293979.0	293220.0	759.0	1.0	759.0
				T3	0.0	161469.0	160676.0	793.0	1.0	793.0
				T4	0.0	148523.0	147925.0	598.0	1.0	598.0
4	Kaunas, Krėvės 14b	4	44192	Q+	0.0	69529.0	69240.0	289.0	1.0	289.0
				Q-	0.0	3957.0	3957.0	0.0	1.0	0.0
				T1	0.0	39816.0	39765.0	51.0	1.0	51.0
				T2	0.0	138545.0	138294.0	251.0	1.0	251.0
				T3	0.0	80475.0	80081.0	394.0	1.0	394.0
				T4	0.0	131842.0	131196.0	646.0	1.0	646.0

Iš viso suvartota aktyviosios el. energijos: 14667 kWh

Iš viso suvartota reaktyviosios el. energijos: 2416 kWh

Iš viso generuota reaktyviosios el. energijos: 1697 kWh

Vadovas _____
 (vardas, pavardė) (parašas)

A.V.

Vyr.energetikas _____
 (vardas, pavardė) (parašas)