

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

Donatas Veikutis

**Prekių profilio sinchronizacijos modelio sudarymas
ir tyrimas**

Magistro darbas

Vadovė dr. A. Slotkienė

Šiauliai, 2014

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

TVIRTINU

IT katedros vedėja dr. A. Slotkienė
2014-06-02

Prekių profilio sinchronizacijos modelio sudarymas ir tyrimas

Informatikos inžinerijos magistro darbas

Autorius

ITM-11 gr. magistrantas
2014 m. gegužės 26 d.

D. Veikutis

Vadovė

IT katedros docentė
2014 m. gegužės 26 d.

dr. A. Slotkienė

Recenzantai

IT katedros docentė
2014 m. birželio 2 d.
IT katedros docentė
2014 m. birželio 2 d.

dr. S. Ramanauskaitė

dr. A. Drukteinienė

Šiauliai, 2014

SANTRAUKA

Prekių profilio sinchronizacijos modelio sudarymas ir tyrimas

Plėtojantis elektroninei prekybai ir didinant prekių asortimentą, būtinas ir didesnis žmogiškųjų išteklių kiekis įmonėje, tam kad prekių profilių informacija būtų atnaujinama laiku. Norit sutaupyti laiko ir žmogiškųjų išteklių, reikalinga šiuos procesus automatizuoti.

Šiai problemai spręsti darbe pateikiamas prekių profilių sinchronizavimo modelis, kuris naudoja tarpinę duomenų bazę tiekėjo prekių profilių duomenims saugoti.

Atlikus tyrimą, įrodyta, jog naudojant šį duomenų sinchronizavimo modelį, pagrindinė el. parduotuvės sistema yra apkraunama mažiau, nes serverio resursų išnaudojimas pasiskirsto tarp el. parduotuvės ir tarpinės prekių profilių sistemų.

Prasminiai žodžiai: prekės profiliai, duomenų sinchronizavimas, elektroninė parduotuvė.

SUMMARY

Analysis and development synchronization model of product profile

With the development of electronic trading and increasing the range of products, is necessary for greater amount of human resources in the company, in order to product profile information is updated in a timely manner. To save time and human resources required to automate these processes.

To solve this problem, in this work is provided model of product profile synchronization that uses intermediate level, for data save of supplier product profile information.

The analysis has shown that the main e-shop system is loaded less when using of this data synchronization model, because use of server resources are distributed among the e-shop system and intermediate system of supplier products profiles.

Keywords: product profile, data synchronization, e-shop.

TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

- XML** - (angl. Extensible Markup Language) yra W3C konsorciumo rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.
- SOAP** - paskirstytųjų programų protokolas sukurtas Microsoft, IBM, DevelopMentor ir UserLand kompanijų.
- REST** - (angl. Representational State Transfer) yra alternatyvus architektūrinis stilius, skirtas interneto paslaugoms kurti gaunant ir manipuluojant ištekliais su nustatytais standartinėmis operacijomis, tokiomis kaip GET, POST, PUT ir DELETE.
- DOM** - (angl. Document Object Model) tai objektiškai orientuota standartinė sąsaja, naudotis ir manipuluoti struktūriniais duomenimis.
- SAX** - tai įvykiu pagrįstas analizatorius, kuris nuskaitytu duomenis iš XML dokumento.
- JSON** - tekstinis, žmogui suprantamas duomenų formatas atvaizduojantis duomenų struktūras ir asociatyvius masyvus.
- Google Analytics** - tai programinės įrangos įrankis, kuris suteikia galingą skaitmeninės analizės priemonę bet kam prieinamą prie žiniatinklio, nesvarbu, kokį įrenginį naudotų – didelį ar mažą.
- Elektroninė parduotuvė** - interneto tinklalapis, kuris suteikia galimybes peržiūrėti, pasirinkti, užsakyti prekes bei/arba atsiskaityti už jas.
- PHP** - plačiai paplitusi dinaminė interpretuojama programavimo kalba (angl. Hypertext Preprocessor), sukurta 1995 m. ir specialiai pritaikyta interneto svetainių kūrimui.
- JavaScript** - objektiškai orientuota skriptų programavimo kalba, besiremianti prototipų principu.
- Java** - objektiškai orientuota programavimo kalba, 1991 metais sukurta Džeimso Goslingo ir kitų Sun Microsystems inžinierių.
- RAM** - (angl. Random Access Memory) kompiuterio operatyvinė atmintis, kuri išsaugo įrašytą informaciją tol, kol tiekama maitinimo įtampa, todėl kol prijungtas maitinimo šaltinis.

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Elektroninės parduotuvės komponentai [1][17]	12
2 pav. Elektroninė parduotuvė ir vykstantys procesai.....	13
3 pav. Užsakymo administravimo schema	13
4 pav. JSON sintaksės pavyzdys	15
5 pav. XML sintaksės pavyzdys.....	16
6 pav. XML prekių profilio pavyzdys.....	17
7 pav. XML prekių profilio pavyzdys.....	17
8 pav. XML prekių profilio pavyzdys.....	18
9 pav. XML prekių gamintojų pavyzdys.....	18
10 pav. XML prekių grupių pavyzdys	18
11 pav. SOAP protokolo veikimo schema	19
12 pav. SOAP užklauso pavyzdys.....	19
13 pav. SOAP atsakymo pavyzdys	20
14 pav. Galimi resursų identifikatoriai naudojant REST	21
15 pav. Resursas visi užsakymai	21
16 pav. Resursas konkretus užsakymas	21
17 pav. SAX metodo architektūra.....	22
18 pav. DOM modelio objektas [12].....	24
19 pav. DOM panaudojimas PHP aplinkoje	24
20 pav. Nuskaitymo metodų vykdymo laiko diagrama [11].....	25
21 pav. Nuskaitymo metodų atminties sunaudojimo diagrama [11].....	25
22 pav. Tiekėjai ir elektroninė parduotuvė	28
23 pav. Prekės įrašymas į el. parduotuvę.	29
24 pav. Prekės aprašo pašalinimo/įrašymo schema	30
25 pav. Konceptinė modelio komponentų diagrama.....	32
26 pav. Esybių ryšių diagrama	33
27 pav. Tarpinės sistemos PĮ funkcijos.....	35
28 pav. Nuskaitymo modulio schema	36
29 pav. Prekių nuskaitymo ciklas.....	39
30 pav. Nerastų prekių XML faile deaktyvavimo algoritmas	40
31 pav. Atnaujinimo modulio algoritmo schema	42
32 pav. Prekės įrašymo į el. parduotuvę algoritmas.....	45
33 pav. Prekės atnaujinimas.....	46
34 pav. Prekės aprašo atnaujinimo algoritmas	48
35 pav. Prekės nuotraukų atnaujinimo algoritmas	50
36 pav. James A. McCal kokybės trikampis [19]	52
37 pav. Prekių profilių įrašymo diagrama.....	54
38 pav. Prekių profilių paklaidos koeficiento diagrama	55
39 pav. Bendras užklausių kiekis susumavus ir tarpinę sistemą	57
40 pav. Produkcijos sistemos užklausių kiekis	57
41 pav. El. parduotuvės lankomumo diagrama	58

42 pav. Įeinamų/išeinamų duomenų srautų diagrama.....	59
43 pav. Procesoriaus apkrovos diagrama	59
44 pav. RAM naudojimo serveryje diagrama	60
45 pav. Duomenų srauto tyrimo metu diagrama	60
46 pav. CPU apkrautumo tyrimo metu diagrama.....	61
47 pav. RAM naudojimo tyrimo metu diagrama	61
48 pav. El. parduotuvės apkrautumas nenaudojant tarpinės sistemos.....	65
49 pav. El. parduotuvės apkrautumas naudojant tarpinę sistemą.....	65
50 pav. Tarpinės sistemos duomenų bazės struktūra	70

LENTELIŲ SĄRAŠAS

1 lentelė. DOM ir SAX palyginimas [13]	26
2 lentelė. Duomenų bazės esybių sąryšiai	33
3 lentelė. Nuskaitymo modulio parametrai	36
4 lentelė. Metodų aprašymų lentelė	37
5 lentelė. Atnaujinimo duomenys	38
6 lentelė. Metodų aprašymų lentelė	39
7 lentelė. Nerastų prekių XML faile deaktivavimo algoritmo metodų aprašų lentelė	41
8 lentelė. Gražinami prekės duomenys iš duomenų bazės	41
9 lentelė. Prekės sukūrimo metodų lentelė.....	42
10 lentelė. Prekės atnaujinimo metodų lentelė.....	43
11 lentelė. Metodų lentelė	47
12 lentelė. Metodų lentelė	49
13 lentelė. Metodų aprašų lentelė.....	49
14 lentelė. Tyrimo situacijos	51
15 lentelė. Paklaidos koeficiento apskaičiavimo lentelė	54
16 lentelė. Užklausų skaičiaus tyrimo duomenys	56
17 lentelė. Tyrimo įvesties savybės	62
18 lentelė. Tyrimo išvesties duomenys	63
19 lentelė. Integralumo tyrimo rezultatai	63

TURINYS

Įvadas	11
1. Prekių aprašų automatizavimui naudojamos technologijos	12
1.1. Elektroninės prekyba	12
1.1.1. Elektroninės prekybos modelio schema	12
1.1.2. Automatizavimo ir sinchronizavimo svarba elektroninėje prekyboje	14
1.1.3. Prekių profilio aprašai.....	14
1.2. Prekių profilio aprašymo technika	15
1.2.1. Prekių profilių pateikimo būdai	15
1.2.2. XML nauda prekių aprašų formavime	16
1.2.3. Prekių profilių pateikimo skirtumai	17
1.3. Duomenų perdavimo architektūros.....	18
1.3.1. Simple Object Access Protocol (SOAP).....	18
1.3.2. REST architektūra.....	20
1.4. Galimi resursų nuskaitymo būdai	22
1.4.1. SAX nuskaitymo metodas XML dokumentui.....	22
1.4.2. DOM nuskaitymo metodas XML dokumentui	23
1.4.3. DOM ir SAX palyginimas	24
1.5. Skyriaus išvados.....	26
2. Prekių aprašų sinchronizacijos modelis	28
2.1. Prekių profilių sinchronizavimo problemos ir siūlomi sprendimo būdai.....	28
2.1.1. Duomenų stoka XML dokumente prekės įrašymui elektroninės parduotuvės duomenų bazėje	28
2.1.2. Skirtingas prekių profilių atnaujinimo dažnis.....	29
2.1.3. Neapibrėžtas prekės gyvavimo ciklas	29
2.1.4. Papildomi sistemos reikalavimai	30
2.1.5. Techniniai reikalavimai sistemai	30
2.2. Konceptinė prekių profilių sinchronizavimo modelio schema.....	31
2.3. Tarpinės sistemos duomenų bazės struktūra.....	33
2.4. Tarpinės sistemos programinė įranga	35
2.5. Nuskaitymo modulis	35
2.5.1. Prekių nuskaitymo algoritmas	38
2.5.2. Nerastų prekių deaktivavimo algoritmas.....	40
2.6. Atnaujinimo modulis	41
2.7. Duomenų sinchronizavimo modulis	44
2.7.1. Prekės įrašymas į el. parduotuvės informacinę sistemą	44
2.7.2. Prekės kainos ir likučio atnaujinimas	45
2.7.3. Prekės aprašymo atnaujinimas	47
2.7.4. Prekės nuotraukų atnaujinimas	49
2.8. Skyriaus išvados.....	50
Prekių aprašų sinchronizacijos modelio tyrimas	51
2.9. Tyrimo tikslai ir apimtis.....	51
2.10. Tyrimo techninė ir programinė aplinka	51
2.11. Testavimo kriterijai	52
2.12. Sistemos veikimo tyrimo rezultatai.....	53

2.13. Sistemos našumo tyrimo rezultatai	55
2.14. Duomenų vientisumo tyrimo rezultatai.....	62
2.15. Panaudojamumo tyrimas.....	63
2.16. Skyriaus išvados.....	65
Darbo išvados.....	67
Literatūra.....	68

IVADAS

Šiuo metu labai išpopuliarėjusi verslo niša yra elektroninės parduotuvės. Šiam verslui esant smulkiam, tai yra specializuojantis tik ties viena ar dvejomis prekių grupėmis, nereikia daug materialųjų išteklių. Taip yra todėl, kad duomenų kiekis elektroninėje parduotuvėje nėra didelis, iki 1000 prekių aprašų, ir vienas ar du darbuotojai gali sutvarkyti ir juos išlaikyti kokybiškus bei laiku kintamus. Bet jei prekių kiekis šokteli iki kelių, keliolikos ar net kelių dešimčių tūkstančių prekių, o pardavimų skaičius atitinkamai taip daug nekinta, neįmanoma pasamdyti tiek darbuotojų, kad visi prekių aprašų pokyčiai būtų sutvarkomi per 8 darbo valandas, nes informacija kartais keičiasi ne dienomis, bet ir valandomis. Taigi, šiai problemai išspręsti būtina pasitelkti automatinį prekių aprašų sinchronizavimą. Šis būdas leidžia, nepadidinus materialųjų kaštų elektroninės parduotuvės įmonėje, valdyti gan didelius kiekius informacijos nustatytu laiku.

Šio darbo tikslas – sudaryti e-parduotuvės prekių aprašų nuskaitymo, atnaujinimo ir sinchronizavimo modelį bei ištirti jo taikymo efektyvumą. Taigi, šiame darbe siūlomas modelis, kurio pagalba bus galima pasiekti tokį rezultatą.

Šiam tikslui pasiekti yra išskelti šie uždaviniai:

1. Išanalizuoti prekių profilių struktūrą ir sinchronizavimo aspektus.
2. Išnagrinėti duomenų nuskaitymo technologijas ir perdavimo architektūras.
3. Sudaryti e-parduotuvės prekių aprašų nuskaitymo, atnaujinimo ir sinchronizavimo modelį.
4. Ištirti sukurto modelio taikymo savybes.

1. PREKIŲ APRAŠŲ AUTOMATIZAVIMUI NAUDOJAMOS TECHNOLOGIJOS

1.1. Elektroninės prekyba

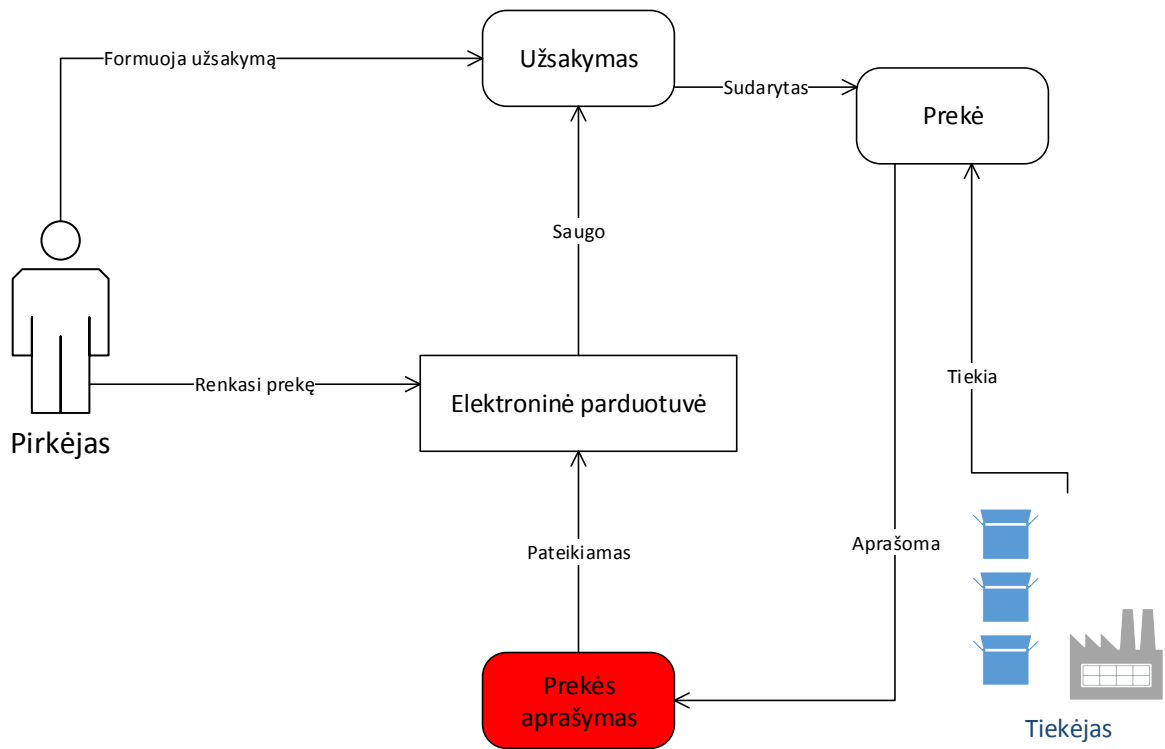
1.1.1. Elektroninės prekybos modelio schema

Diagramoje (žr. 1 pav.) pavaizduoti pagrindiniai elektroninės prekybos komponentai. Taigi, elektroninė parduotuvė susideda iš klientų duomenų bazės, darbuotojų (vadybininkų, buhalterių, IT specialistų, kurjerių ir pan.), prekių aprašų duomenų bazės, turinio valdymo sistemos, kuria naudojantis darbuotojai gali koreguoti prekių aprašus, klientų informaciją ar kitus svetainės duomenis, bei aišku svarbiausio veiksnio verslui, tai yra, pardavimų.



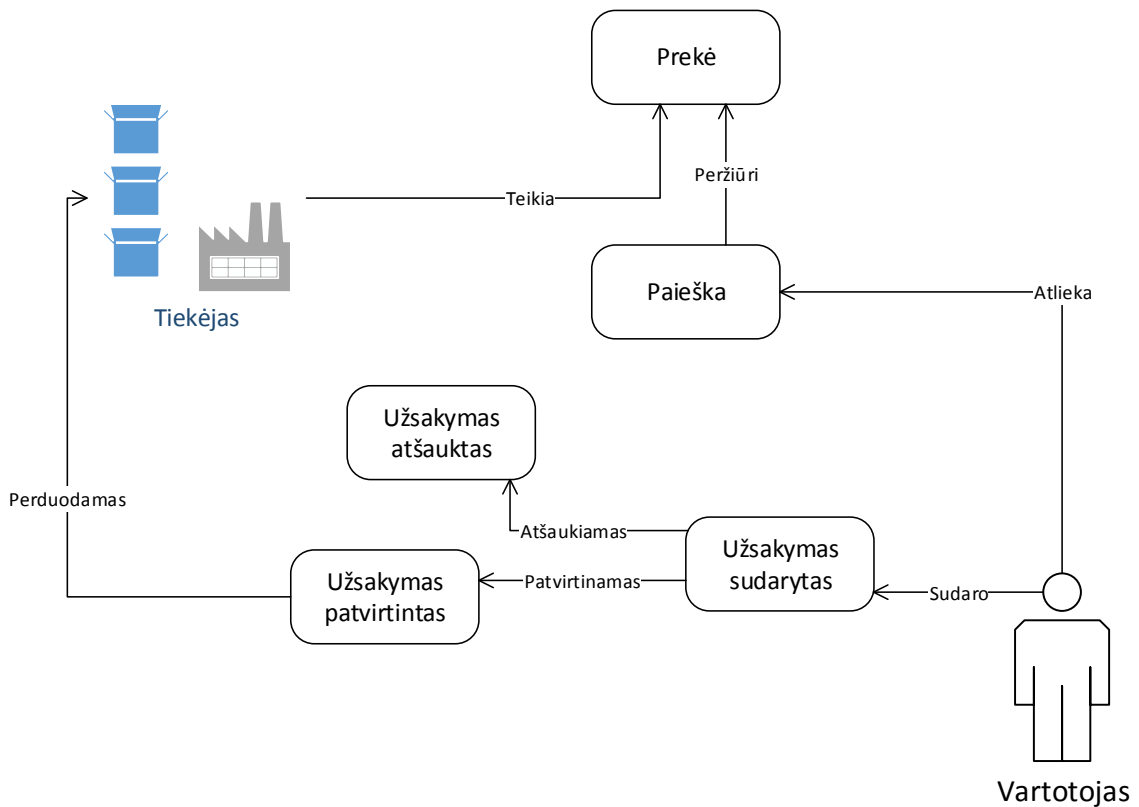
1 pav. Elektroninės parduotuvės komponentai [1][17]

E-parduotuvė susideda ne tik iš vidinių komponentų, jos veikimas priklauso nuo išorinių procesų. Pirkėjai renka prekes ir formuoja užsakymą, užsakymas yra sudarytas iš prekių, kurias tiekia tiekėjai. Kad pirkėjai rastų reikiamas prekes, tiekėjai turi jas aprašyti ir perduoti duomenis e-parduotuvei. Taip pilni prekių aprašai yra pateikiami e-parduotuvės svetainėje. Visi šie procesai pavaizduoti 2 pav. Šiame pavyzdyje išskiriamas elementas „Prekės aprašymas“, kadangi jis yra vienas svarbiausių objektų el. prekyboje, nes nuo prekės aprašymo informatyvumo priklauso, ar bus sudomintas klientas.



2 pav. Elektroninė parduotuvė ir vykstantys procesai

Detalesnė schema kaip pirkėjas įsigyja prekę pavaizduota 3 pav. Joje taip pat pavaizduota, kad pirkėjas atlieka paiešką, ir gali sudaryti užsakymą. Užsakymas gali būti atšauktas, taip jokie veiksmai daugiau neatliekami, bet kitu atveju, jei jis patvirtintas, užsakymas perduodamas tiekėjams, o tiekėjas pristato prekes pirkėjui.



3 pav. Užsakymo administravimo schema

1.1.2. Automatizavimo ir sinchronizavimo svarba elektroninėje prekyboje

Viena iš populiariausių dabartinių verslo nišų yra elektroninė prekyba, kadangi šis verslas nereikalauja papildomų patalpų, prekių sandėliavimo ar panašiai. Pardavėjas nurodo, kokias prekes turi, ar gali tūrėti, esant užsakymui ir jas parduoda klientui internetu, neturint jokio fizinio kontakto su žmogumi. Kiekviena elektroninė parduotuvė bando klientui pasiūlyti kuo platesnį ir įvairesnį prekių asortimentą ir dėl to pardavėjai susiduria su problema, jog fiziškai nebeišgali susitvarkyti su dideliu prekių kiekiu. Dėl šios priežasties jiems yra reikalinga investuoti į automatinę informacijos atnaujinimą.

Kuo informatyvesni prekių aprašai parduotuvėje, savaime suprantama, kad tuo didesnių pardavimų įmonė gali tikėtis. Taigi, automatizavus procesus elektroninės prekybos versle, galima padidinti pajamas, nepadidinus materialijų kaštų.

Šiam tikslui pasiekti gali padėti paskirstytos sistemos ir į paslaugas orientuotos sistemos (SOA). Paskirstytos sistemos susideda iš autonominių kompiuterių grupės, kurie sujungti tinklu ir paskirstyta tarpine įranga, kuri leidžia kompiuteriams koordinuoti veikimą kartu ir dalintis resursais. [14]

Paskirstytų sistemų privalumai [22]:

- Centralizuota architektūra, lengvesni komponentų pakeitimai, atnaujinimai.
- Palaiko daug klientų, kurie bendrauja per centrinis servisus.
- Užtikrina lygiagretų bendravimą tarp serverio dalies ir klientų.
- Didelė panaudojimo vertė – tam tikrus komponentus galima panaudoti ir kitose projektuose (pvz. servisus).
- Didesnis patikimumas.

Paskirstytų sistemų trūkumai[22]:

- Tokias sistemas sunku kurti, įdiegti ir palaikyti.
- Didelė priklausomybė nuo tinklo patikimumo ir ryšio kokybės.
- Daugiau komponentų, sudėtingesnė sistema, todėl didesnė tikimybė atsirasti defektams.

1.1.3. Prekių profilio aprašai

Prekių aprašai elektroninėje parduotuvėje yra vienas svarbiausių bei reikšmingiausių elementų prekyboje. Prekių aprašas turi būti informatyvus ir kokybiškas, tam, kad vartotojai gautų pilną informaciją apie prekę ir galėtų tinkamai apsispręsti dėl užsakymo.

Elektroninėje parduotuvėje prekės aprašai dažniausiai susideda iš šių elementų:

- Prekės pavadinimas.
- Prekės kategorija (grupė).
- Savikaina – kaina, tiekama elektroninei parduotuvei.
- Antkainis – procentas nuo savikainos, pagal kurį skaičiuojama mažmeninė kaina.
- Mažmeninė kaina – kaina, už kurią klientai gali įsigyti prekes.

- Detalus prekės aprašymas.
- Trumpas aprašymas – aprašymas, kuris gali būti rodomas prekių sąrašė, ne tik prekės puslapyje.
- Prekės nuotraukos.
- Prekės charakteristikos – prekės techninės charakteristikos galimai naudojamos kaip filtrai ar pan.
- Prekės kodas – unikalus prekės kodas.
- Prekės ID – unikalus prekės duomenų bazės identifikatorius.
- Gamintojas – prekės prekinis ženklas.
- Mato vienetas (t., vnt., kg., m. ir t.t.).

1.2. Prekių profilio aprašymo technika

1.2.1. Prekių profilių pateikimo būdai

Kiekviena el. verslo įmonė gali pateikti savo produktų profilių aprašymus skirtingais būdais, tokiais kaip - XML formatu, JSON duomenų masyvu ar tiesioginiu prisijungimu prie duomenų bazės.

Vienas nesaugiausių, bet įmanomų, būdų yra bendradarbiaujančioms įmonėms pateikti prisijungimo duomenis prie jų duomenų bazių. Taip el. parduotuvė, kuriai reikia tiekėjo prekių aprašų, gali savo nuožiūra eksportuoti duomenis ir juos importuoti į savo sistemą. Šiam būdai realizuoti įmonės turėtų labai pasitikėti vienas kita, kadangi el. parduotuvė gali laisvai matyti visus duomenis (ar jais manipuliuoti, jei nėra suteiktų ribotų teisių) turint prisijungimo parametrus prie tiekėjo sistemos.

Kitas informacijos eksportavimui naudojamas formatas yra JSON. Šis duomenų pateikimo formatas lengvai skaitomas žmonėms bei greitai apdorojamas ir naudojamas kompiuterinėse sistemose. Šio duomenų formato apdorojimo bibliotekos yra integruotos į JavaScript, PHP, Java ir daugelį kitų programavimo aplinkų, dėl to papildomos bibliotekos nereikalingos. JSON sintaksės pavyzdys pateiktas žemiau (žr. 4 pav.). JSON duomenys per duomenų perdavimo sąsają pateikiami tekstine išraiška kaip pateikta pavyzdyje, vėliau programavimo aplinkoje ji yra dekoduojama, paverčiama objektiniu masyvu, kuriuo naudojantis duomenys yra nuskaitomi. [23]

```
{
  "vardas" : "Vardenis",
  "pavarde" : "Pavardenis"
}
```

4 pav. JSON sintaksės pavyzdys

XML (ang. Extensible Markup Language) yra paprastas ir labai lankstus teksto formatas, kilęs iš SGML (ang. Standard Generalized Markup Language). Iš pradžių ši kalba buvo sukurta tam, kad padėtų susidoroti su didelio masto elektroninės leidybos problemomis, vėliau ji pradėta naudoti visur, kur reikia perduoti duomenis interneto tinklais ar kitomis technologijomis.[2] XML sintaksės pavyzdys pateiktas žemiau (žr. 5 pav.). XML formato dokumentas taip pat yra pateikiamas

naudojantis pasirinkta duomenų perdavimo sąsaja ir programavimo aplinkoje apdorojamas, ir nuskaitomas pasitelkiant tam skirtas bibliotekas.

```
<asmuo>
  <vardas>Donatas</vardas>
  <pavarde>Veikutis</pavarde>
</asmuo>
```

5 pav. XML sintaksės pavyzdys

Taigi, išanalizavus pagrindinius informacijos pateikimo būdus, galima teigti, jog kokių būdu bus pateikiami duomenys sprendžia tik pati įmonė, XML ir JSON pateikimo būdai yra lankstūs ir įskaitomi ne tik kompiuterių bet ir žmonių, o duomenų pateikimas atskleidžiant savo įmonės duomenų bazę galimas jei yra glaudus ir visiškas pasitikėjimas partneriais.

Išanalizavus keletą realių el. parduotuvės tiekėjų duomenų perdavimo formatų, pastebėta, kad visi jie naudoja XML formatą, todėl tolimesnėje darbo eigoje prekių profilių aprašai bus analizuojami naudojant šį duomenų pateikimo formatą.

1.2.2. XML nauda prekių aprašų formavime

XML kalba leidžia vartotojui organizuoti informaciją taip, kad ji būtų suprantama tiek kompiuteriui, tiek fiziniam asmeniui. Vartotojas gali naudoti XML, kad sukurtų savo struktūros formatą, tai yra informacijai, kuri privalo atsidurti šiame faile. Jis gali nurodyti savo taisyklėmis, nurodyti elementus, kurie turi atitikti tokią informaciją kaip: pavadinimas, aprašymas, nuotrauka, kaina ir pan. Naudojant tokius struktūrizuotus dokumentus įmonės gali keistis tokia informacija, kaip užsakymų istorija, prekių asortimentu, užsakymų statusais ir pan.

XML dokumento panaudojimo sritys [3]:

- **Paieška internete ir automatizuotos žiniatinklio užduotys.** XML apibrėžia, kokio tipo informacija yra įtraukta į dokumentą. Dėl to naudojant šį formatą duomenys yra grąžinami daug greičiau.
- **Bendrųjų programų naudojime.** XML naudoja standartinius metodus informacijai pasiekti, taigi bet kokia aplikacija ar įrenginys palaikantis šiuos standartus gali lengvai išvesti, išsaugoti ar dalintis duomenimis.
- **E-verslo reikmėms.** XML diegiamas elektroniniame versle lengvam B2B modelio įgyvendinimui ir duomenų apsikeitimui.
- **Metaduomenų programoms.** XML palengvina metaduomenų išreiškimą, daugkartinio naudojimo formatu.
- **Belaidžiuose prietaisuose ir pan.**

Pagrindiniai XML kalbos privalumai [3][15]:

- Naudojama ne tik kompiuteriui, bet ir žmogui suprantama kalba.
- Visiškai suderinama su JAVA, PHP bei kitomis programavimo kalbomis. Bet kuri programa, kuri gali naudoti XML, gali naudoti pateiktą informaciją.

- Išplečiama kalba. Tai yra, galima sukurti savo žymas arba naudoti jau sukurtas, taip pat sukurti atributus, kurių tik prireikia vartotojui.

Kadangi XML duomenų formatas yra lankstus ir lengvai suprantamas, tai lėmė jo populiarumą tarp elektroninių parduotuvių bei jų prekių tiekėjų. Naudojantis šio formato kalba yra teikiama informacija elektroniškai parduotuvei, o kai kuriais atvejais elektroniškai parduotuvė savo duomenis teikia ir partneriams.

1.2.3. Prekių profilių pateikimo skirtumai

Kaip jau minėta, XML dokumento formatas yra lankstus ir laisvai kuriamas, todėl kiekvienas XML šaltinis gali turėti savo skirtumą. Kai kurios įmonės, tokios kaip Yandex, Google, yra sukūrusios savo standartus prekių profilių aprašymui ir jų pateikimui atskiroms šių įmonių sistemoms. Bet tokiais standartais tikrai nebūtinai naudojasi kitos įmonės, kurios nori integruoti duomenų šaltinius į elektronines parduotuves. Dažniausiai yra kuriamas savas XML formatas, kuriame pateikiami prekių profiliai. Todėl sistemos turi prisitaikyti prie įvairių struktūrizuoto dokumento formatų.

Išanalizavus 5 tiekėjų XML failus, kurie bendradarbiauja su elektroniškai parduotuve, pastebėti prekių profilių pateikimo skirtumai.

```
<ProductCatalog>
  <Product>
    <ProductCode>CF/4GB</ProductCode>
    <Vendor>KINGSTON</Vendor>
    <ProductType>Atminties raktai (flash)</ProductType>
    <ProductCategory>Memory</ProductCategory>
    <ProductDescription>KINGSTON Memory ( flash cards ) 4GB Compact
Flash</ProductDescription>
    <AttrList>
      <element Name="Memory Technology" Value="NAND Flash"/>
      <element Name="Memory Form Factor" Value="Compact Flash"/>
    </AttrList>
    <MarketingInfo>
      <element>Since 1987, Kingston has been known for the highest quality memory
products available. The company recognizes the important role memory plays in
computing productivity and takes the necessary steps to guarantee a reliable memory
product.</element>
    </MarketingInfo>
    <Images>
      <Image>http://el-parduotuve.lt/1.jpg</Image>
      <Image>http://el-parduotuve.lt/2.jpg</Image>
    </Images>
  </Product>
  ...
</ProductCatalog>
```

6 pav. XML prekių profilio pavyzdys

```
<Products>
  <Item id="15" pavadinimas="Samsung N900" grupe="Planšetės" aprasymas="This
great tablet..." gamintojas="Samsung" image=http://el-parduotuve.lt/1.jpg" />
  ...
</Products>
```

7 pav. XML prekių profilio pavyzdys

Pavyzdžiuose (žr. 6 pav. bei 7 pav.) pateikti XML dokumentų struktūrų pavyzdžiai, kuriuose galima įžvelgti didelių skirtumų ne tik tarp elementų pavadinimų, bet ir pačios kodo sintaksės. 6 pav. pateikiamas toks prekių profilis, kai gausybė *Product* elementų, kurie savo viduje turi dar elementų, pagal kurių pavadinimus galima spręsti, kokia informacija juose yra teikiama. 7 pav. prekių profilis pateikiamas visai kitokia sintakse, tai yra, kai prekių informacija teikia tik vienas elementas *Item*, o savo viduje jis turi atributus su prekės duomenimis.

Galimas ir dar vienas labiau išskirstytos prekių profilio informacijos teikimo būdas. Tai yra, kai prekių informacija teikiama vienu XML dokumentu, bet, pavyzdžiui, prekės grupė ir gamintojas yra teikiami kaip duomenų bazės identifikatoriai (skaičius), o šias reikšmes galima gauti tik nuskaičius ir suradus identifikatorių atitinkamame XML faile. Tokios struktūros XML formatai pateikiami 8 pav. , 9 pav. ir 10 pav. pavyzdžiuose. Tokia prekių profilio pateikimo architektūra nėra optimali, gali būti ir tokių sistemų, kai kitaip šių duomenų pateikti negalima, ar tiesiog IT skyrius padarė tokį sprendimą, kad taip teikti prekių aprašus yra geriausia.

```

<ProductCatalog>
  <Product>
    <ProductCode>CF/4GB</ProductCode>
    <Vendor>10</Vendor>
    <ProductCategory>185</ProductCategory>
    <ProductDescription>KINGSTON      Memory      (flash      cards)      4GB      Compact
Flash</ProductDescription>
    ...
  </Product>
  ...
</ProductCatalog>

```

8 pav. XML prekių profilio pavyzdys

```

<Vendors>
  <Vendor>
    <id>10</id>
    <title>KINGSTON</title>
  </Vendor>
  ...
</Vendors>

```

9 pav. XML prekių gamintojų pavyzdys

```

<ProductCategorys>
  <ProductCategory>
    <id>185</id>
    <title>Memory</title>
  </ProductCategory>
  ...
</ProductCategorys>

```

10 pav. XML prekių grupių pavyzdys

1.3. Duomenų perdavimo architektūros

1.3.1. Simple Object Access Protocol (SOAP)

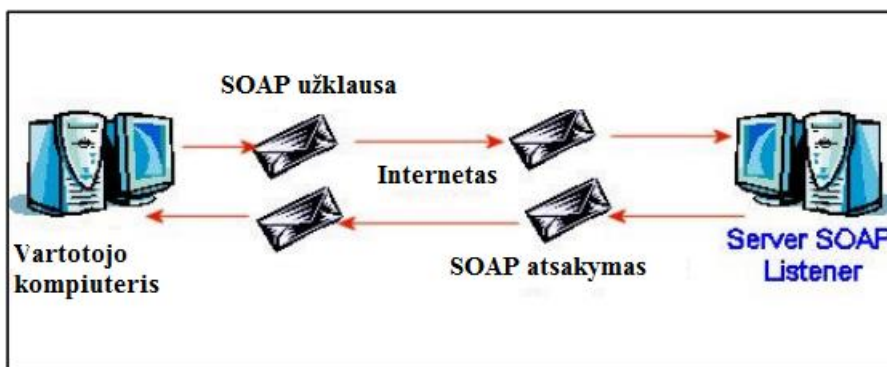
SOAP protokolas yra paskirstytųjų programų protokolas, sukurtas Microsoft, IBM, DevelopMentor ir UserLand kompanijų. SOAP protokolas yra labai lankstus ir gali būti suderinamas

su įvairiomis aplikacijomis, tačiau tam, kad būtų galima juo naudotis, svarbiausia įjungti nuotolinių procedūrų užklausas (RPC) per HTTP, naudojant XML.

Šis protokolas yra paremtas kreipimosi XML objektu ir iš pradžių buvo skirtas HTTP protokolu bendrauti tarp paskirstytų programų, naudojant prisijungimo duomenis tam, kad būtų galima gauti leidimą tolimesniems veiksams iš įmonėse esančių tinklo ugniasienių.

Pagrindiniai SOAP protokolo tikslai [4]:

- Suteikti standartinio objekto iškvietimo protokolą, paremtą tinklo standartais.
- Sukurti paprastą ir išplečiamą protokolą bei naudingą formatą, kuris gali vystytis.



11 pav. SOAP protokolo veikimo schema

11 pav. pateiktas standartinis SOAP užklausos pavyzdys:

- Vartotojas suformuoja SOAP užklausa ir siunčia ją internetu į SOAP WSDL serverį.
- SOAP serveris tikrina užklausos parametrus ir pateikia atsakymą.
- Vartotojas gauna atsakymą iš SOAP serverio.

Taigi, matoma, jog tai gana paprastas duomenų perdavimo ryšys, kadangi vartotojas bet kuriuo metu gali gauti naujausius duomenis, kuriuos sugeneruoja serveris, gaudamas užklausa su tinkamais parametrais. SOAP užklausos pavyzdys (žr. 12 pav.).

```

POST /StockPrice HTTP/1.1           // HTTP ANTRAŠTĖ
Host: www.stockpriceserver.com      //Tai antraštės dalis kuri
Content-Type: text/xml              //nustato turinio tipą bei ilgį,
Content-Length: 100                 //taip pat serverio URL

<SOAP-ENV:Envelope xmlns:SOAPENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/"> //Užklausos duomenys ir antraštė
<SOAP-ENV:Body> //SOAP užklausos duomenų pradžia
<ns1:getAddressFromName                xmlns:ns1="urn:AddressFetcher"
SOAPENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<nameToLookup xsi:type="xsd:string">John B. Good</nameToLookup>
</ns1:getAddressFromName>
</SOAP-ENV:Body> //SOAP duomenų pabaiga
</SOAP-ENV:Envelope> //SOAP užklausos pabaiga

```

12 pav. SOAP užklausos pavyzdys

13 pav. pateiktas SOAP protokolo atsakymo pavyzdys:

```
HTTP/1.1 200 OK //
Content-Type: text/xml // Antraštė
Content-Length: nnnn //

<SOAP-ENV:Envelope xmlns:SOAPENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema/">
<SOAP-ENV:Body> // SOAP atsakymo duomenų pradžia
<ns1:getAddressFromNameResponse xmlns:ns1="urn:AddressFetcher"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<return xmlns:ns2="urn:ibm-soap-address-demo"
xsi:type="ns2:address"> // Gražintas elementas
<city xsi:type="xsd:string">Anytown</city>
<state xsi:type="xsd:string">NY</state>
<phoneNumber xsi:type="ns2:phone">
<areaCode xsi:type="xsd:int">123</areaCode>
<number xsi:type="xsd:string">7890</number>
<exchange xsi:type="xsd:string">456</exchange>
</phoneNumber>
<streetName xsi:type="xsd:string">Main Street</streetName>
<zip xsi:type="xsd:int">12345</zip>
<streetNum xsi:type="xsd:int">123</streetNum>
</return>
</ns1:getAddressFromNameResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

13 pav. SOAP atsakymo pavyzdys

SOAP pranašumai [7]:

- SOAP žinutės – naudojant XML struktūrą SOAP žinutės gali būti lanksčiai formuojamos;
- SOAP žinučių perdavimo mechanizmas.

SOAP trūkumai [7]:

- sudėtingumas;
- besikeičiančios specifikacijos;
- pardavėjų poreikių skatinamos specifikacijos;
- reikia sudėtingų priemonių ir karkasų sudėtingumui palengvinti.

1.3.2. REST architektūra

REST (angl. Representational State Transfer) yra alternatyvus architektūrinis stilius, skirtas interneto paslaugoms kurti gaunant ir manipuliuojant ištekliais su nustatytais standartinėmis operacijomis, tokiais kaip GET, POST, PUT ir DELETE. [7]. Ši architektūra nuo SOAP architektūros skiriasi tuo, jog ji neturi kažkurio WSDL failo, kuriame nurodomos visos galimos operacijos kreipiantis į servisą. REST architektūra iš esmės yra kliento-serverio architektūra, paremta HTTP protokolu. Šioje architektūroje klientas ir serveris bendrauja URI nuorodų principu, ir perduodamomis HTTP protokolo antraštėmis.[8]

Grąžintas rezultatas gali būti įvairių tipų XML dokumentas, JSON objektas ir pan. Dokumente pateikiamas visų operacijų sąrašas, jų schema ir URI nuorodos, kurias naudodamas vartotojas gali toliau tęsti bendravimą su serveriu.

Svarbiausias REST architektūros reikalavimas, kad kiekvienas iš resursų turi turėti unikalius identifikatorius, kurie yra nurodomi URI nuoroda. Identifikatorius rekomenduojama pateikti HTTP GET metodo, o ne paslėptuoju POST, kadangi tai leidžia daug greičiau ir lanksčiau dalintis norima informacija. Taip pat rekomenduojama resursų identifikatorius vadinti ne veiksmažodžiu, kaip *getUzsakymus*, o daiktavardžiu *uzsakymai* [7].

Kelios galimos nuorodos resursų identifikavimui:

1. <http://www.svetaine.lt/uzsakymai>
2. <http://www.svetaine.lt/uzsakymai/99871>
3. <http://www.svetaine.lt/prekes/578>
4. <http://www.svetaine.lt/klientai/15478>

14 pav. Galimi resursų identifikatoriai naudojant REST

Tokio tipo nuorodos (žr. 14 pav.) gali pateikti galimus resursus, tokius kaip pavaizduotieji 15 pav. ir 16 pav.

```
<uzsakymai>
  <uzsakymas ref="http://www.svetaine.lt/uzsakymai/99871">
    <suma>200.00</suma>
    <pvm_kaina>42.00</pvm_kaina>
    <prekiu_kiekis>2</prekiu_kiekis>
  </uzsakymas>
  <uzsakymas ref="http://www.svetaine.lt/uzsakymai/99881">
    <suma>400.00</suma>
    <pvm_kaina>84.00</pvm_kaina>
    <prekiu_kiekis>4</prekiu_kiekis>
  </uzsakymas>
</uzsakymai>
```

15 pav. Resursas visi užsakymai

```
<uzsakymas ref="http://www.svetaine.lt/uzsakymai/99871">
  <klientas ref="http://www.svetaine.lt/klientas/99871">
    <vardas>Vardenis</vardas>
    <pavarde>Pavardenis</pavarde>
  </klientas>
  <preke ref="http://www.svetaine.lt/prekes/578">
    <kaina>100.00</kaina>
    <kiekis>2</kiekis>
  </preke>
  <pvm_kaina>42.00</pvm_kaina>
</uzsakymas>
```

16 pav. Resursas konkretus užsakymas

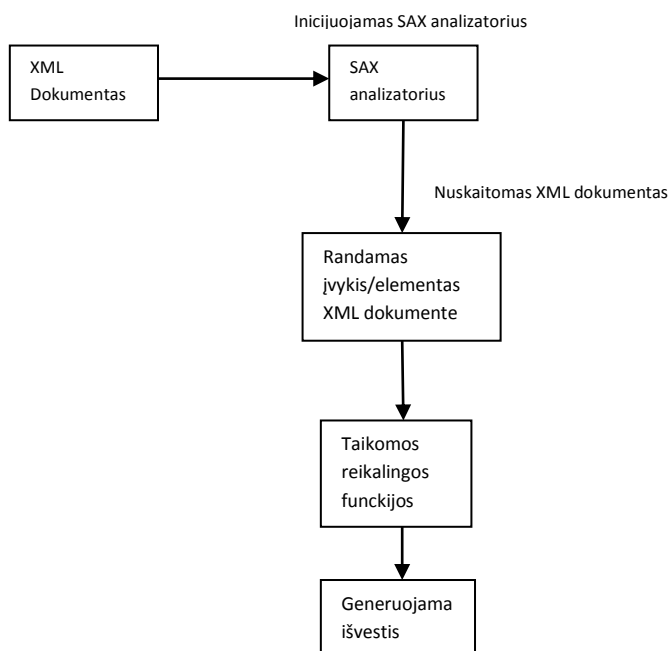
Pavyzdžiuose (žr. 15 pav. ir 16 pav.) pateikta kaip ne žmogaus valdoma programa gali komunikuoti su resursais. 15 pav. pateikti visi užsakymai, kurie yra elektroninėje parduotuvėje, kiekvienas užsakymo elementas turi savo atributą „*ref*“, jo reikšmė yra URI nuoroda (identifikatorius), kurio programa galima toliau komunikuoti su serveriu ir gauti detalesnę informaciją

apie elementą, šiuo atveju užsakymą. Taigi, kai programa kreipiasi į nuorodą, atidaromas kitas resursas, kuriame yra detalai pateikiamas užsakymas. Toliau, programai tikrinant užsakymą, ji ras prekės ir kliento elemente tą patį atributą „ref“, juo naudojant aplikaciją atitinkamai galės pasiimti prekės ir kliento detalią informaciją. Taigi, suformuota REST architektūra leidžia pakankamai paprastai susirinkti visą informaciją apie reikiamus resursus.

1.4. Galimi resursų nuskaitymo būdai

1.4.1. SAX nuskaitymo metodas XML dokumentui

SAX nuskaitymo metodas, tai įvykiu pagrįstas analizatorius, kuris nuskaitytu duomenis iš XML dokumento. Pirmoji SAX 1.0 versija buvo išleista 1998m liepos 11d. O dabartinė SAX 2.0.2 versija, buvo pristatyta 2004m. balandžio 4d. ir nuo to laiko dar nebuvo atnaujinta. [9]



17 pav. SAX metodo architektūra

17-tame pavyzdyje pateikta SAX metodo architektūra [9], joje vaizduojama kaip nuskaitytas XML failas.

- Visų pirma inicijuojamas SAX metodas su visais reikalingais jo parametrais, aprašomos funkcijos kada ir kas turėtų įvykti.
- Dalimis, kiek buvo nurodyta inicijuojant SAX metodą, nuskaitytas XML failas.
- Aptinkamas įvykis/elementas, po kurio turėtų įvykti funkcija.
- Funkcijos pritaikomos ir, esant reikalui, generuojama išvestis.

SAX analizatorius iškviečia tvarkykles (funkcijas) kiekvienam atidaromam ir uždaramam XML elementui. Jis taip pat iškviečia nurodytas funkcijas duomenų nuskaitymui iš XML elemento. Taigi, norit naudoti šį XML nuskaitymo metodą, PHP aplinkoje reikia [9]:

1. Inicijuoti SAX analizatorių PHP aplinkoje, kam reikia naudoti šią eilutę:

```
$xparser=xml_parser_create();
```

Pateiktu kodu sukuriamas xparser kintamasis, kuris priklauso SAX analizatoriui.

2. Identifikuoti įvykiams ir nurodyti kuriuo momentu kuri funkcija turi būtų iškviesta naudojamas šis kodas:

```
xml_set_element_handler($xparser, "startingHandler", "endingHandler");  
xml_set_character_data_handler($xparser, "cdataHandler");
```

- xml_set_element_handler() funkcija naudojama tam, kad nurodyti, kurie metodai bus iškviečiami atidarant ir uždarant XML elementus.
 - startingHandler funkcija bus iškviečiama, kai SAX analizatorius ras XML atidarymo elementą.
 - endingHandler funkcija bus iškviečiama, kai SAX analizatorius ras XML uždarymo elementą.
 - Xml_set_character_data_handler() funkcija nurodo, kurį metodą reikės iškviešti XML elemento duomenims nuskaityti.
3. Funkcijos startingHandler, endingHandler ir cdataHandler turi būti aprašytos programuotojo.
 4. Toliau, nurodyta funkcija, XML dokumentas yra atidaromas

```
$fp=fopen("student.xml", "r");
```

5. Ir XML dokumentas yra nuskaitytas bitų skaičių kiekiu nurodytu *while* cikle

```
while($data=fread($fp, 4096)) {  
    xml_parse($xparser, $data, feof($fp));  
}
```

6. Toliau nurodyta funkcija reikalinga tam, kad pašalinti XML duomenis iš PHP atminties.

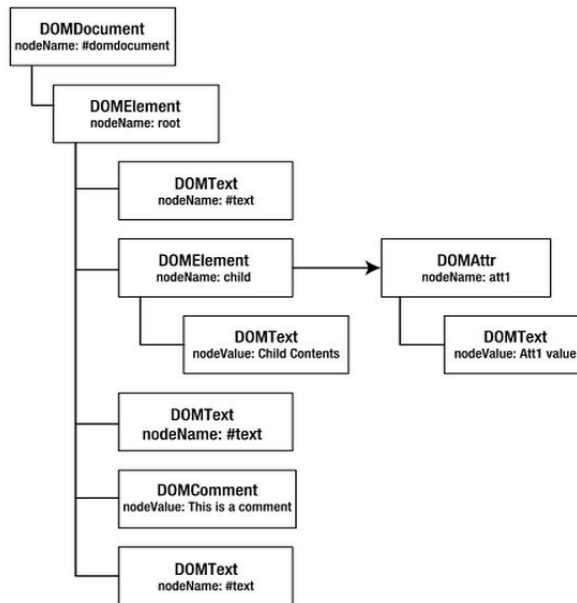
```
xml_parser_free($xml_parser);
```

1.4.2. DOM nuskaitymo metodas XML dokumentui

DOM (angl. Document Object Model) tai objektiškai orientuota standartinė sąsaja, skirta naudotis ir manipuluoti struktūriniais duomenimis.[11] Šis modelis yra standartizuotas WC3 konsorciumo, bet tuo pačiu ir išskaidytas į tris lygmenis DOM Level 1, DOM Level 2 ir DOM Level 3, visi šie lygmenys programuojamoms programoms duoda savitų funkcijų.

DOM modelio mintis yra visą struktūrizuotų duomenų dokumentą išskirstyti kaip medį, kuris sudarytas iš mazgų. Tai vaizdingiau pateikta 18 pav. , kuriame matoma visa DOM dokumento struktūra.

Visas DOMDocument objektas yra talpinamas į PHP atmintį, taigi, kuo dokumentas didesnis, tuo ilgiau jį apdoroja serveris ir daugiau atminties yra išnaudojama.[12] Kitu atveju, kadangi objektas yra sudarytas kaip medis, jį galima apdoroti įvairiai, tai suteikia didelį lankstumą. Kiekvienas iš mazgų gali būti prieinamas, o jo funkcionalumas priklausys nuo jo tipo.



18 pav. DOM modelio objektas [12]

Palyginus su SAX modeliu, DOM modelyje reikia parašyti ganėtinai mažiau kodo.

```

1. $doc = new DOMDocument();
2. $doc->loadXML(file_get_contents("procdcts.xml"));
3. $product = $doc->getElementsByTagName("preke");
4. foreach($product as $itm){
5. }
  
```

19 pav. DOM panaudojimas PHP aplinkoje

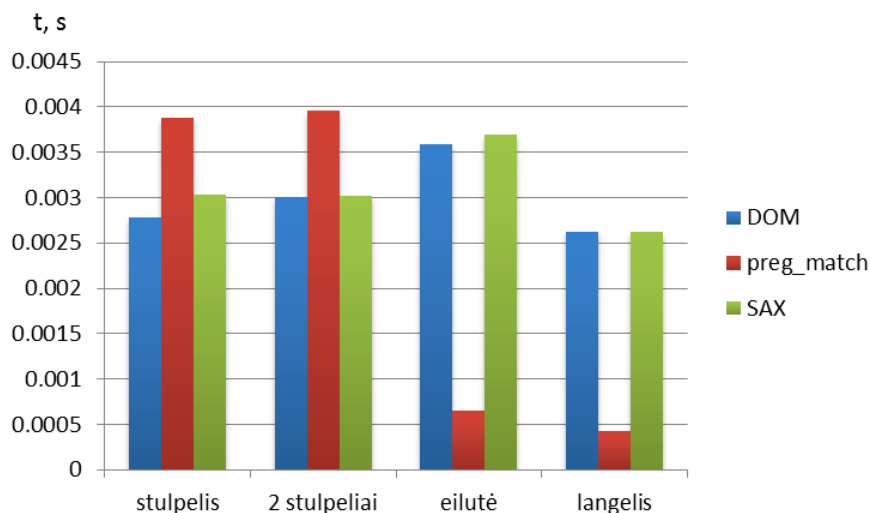
19 pav. pateikta kaip atrodytų paprastas *preke* elemento nuskaitymas naudojant DOM metodą.

- Pirmoje eilutėje sukuriamas DOMDocument objektas.
- Antroje į PHP atmintį įkraunamas visas XML domumentas.
- Trečioje eilutėje naudojant navigacijos funkcija *getElementsByTagName()* į kintamąjį *\$product* įrašomi visi galimi *preke* elemento mazgai.
- Ketvirta ir penkta eilutės – tai ciklas, kuris paima po vieną masyvo elementą ir vykdomi kol pasibaigs visi elementai.

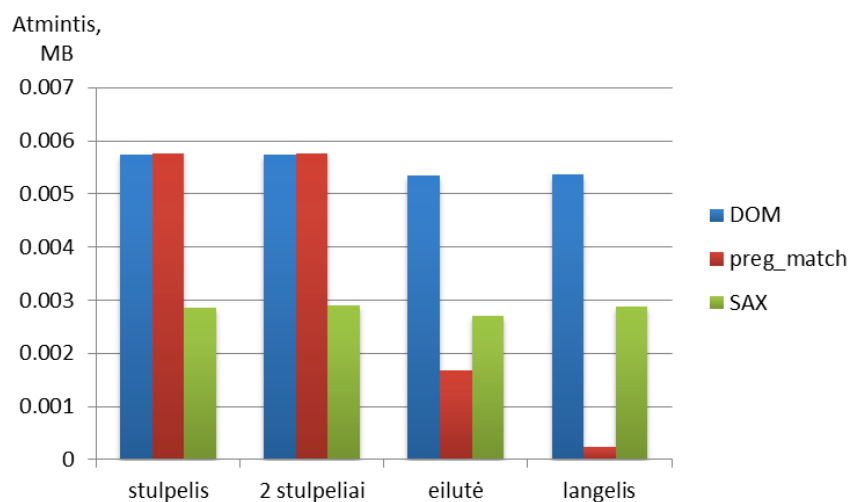
Taigi, naudojant DOM metodą, nebereikia aprašinėti atskirų funkcijų, ką reikia daryti elementą uždarant ir atidarant. Šio metodo esmė, ką daryti, kai randamas elementas ar atributas naudojant DOM metodo navigacijos funkcijas.

1.4.3. DOM ir SAX palyginimas

Kiril Griazev [11] magistro darbe palygino tris struktūrizuotų duomenų technologijas. Jo darbe buvo pateiktos dvi diagramų rūšys, tai – kai struktūrizuotų duomenų dokumentas atitinka standartus, ir kai jų neatitinka. Kadangi XML dokumentas yra griežtai standartizuotas, apžvelgiamos tik tos diagramos, kuriose buvo analizuojami tik standartus atitinkantys dokumentai.



20 pav. Nuskaitymo metodų vykdymo laiko diagrama [11]



21 pav. Nuskaitymo metodų atminties sunaudojimo diagrama [11]

Pateiktose diagramose (žr. 20 pav. ir 21 pav.) analizuojamas internetinis puslapis, kurio HTML kode sudaryta lentelė. Nuskaitant HTML dokumentą analizuojamos kelios situacijos, kai nuskaitomi [11]:

- visi konkrečios eilutės duomenys,
- visi konkretaus stulpelio duomenys,
- konkretaus lentelės langelio duomenys.

Taigi, laiko prasme (žr. 20 pav.), nuskaitant mažesnius duomenų elementus, pirmąją *regex* metodas, bet, nuskaitant daugiau informacijos. pastovesnį ir mažesnį laiką išlaiko SAX nuskaitymo metodas.

Atminties suvartojimo diagramoje (žr. 21 pav.) taip pat nuskaitant mažesnius duomenų kiekius vėl pirmąją *regex* metodas, bet iš šios diagramos matoma, kad SAX metodas išlieka lankstesni, nes nuo duomenų kiekio suvartojama atmintis nesikeičia.

Reziumuojant šias dvi diagramas (žr. 20 pav. ir 21 pav.), daroma išvada, kad mažiausiai parankus yra DOM metodas, kadangi jo vykdymo laikas yra gan didelis. Taip yra dėl visų pirma viso

dokumento įkėlimo į PHP atmintį. O kiti du - regex ir SAX - metodai vienu ar kitu atveju yra savaip patrauklūs, bet SAX metodo duomenys stabilesni.

IBM kompanija savo internetiniame puslapyje pateikia skirtumus tarp SAX ir DOM metodų, tai pavaizduota 1 lentelė.

1 lentelė. DOM ir SAX palyginimas [13]

	DOM	SAX
Interfeiso tipas	Objektiškai orientuota	Įvykiu orientuota
Objektinis modelis	Sukuriamas automatiškai	Turi būti sukuriamas atskirai programai
Elementų seka	Koncentruota	Ignoruojama pavieniais įvykiais
Atminties sunaudojimas	Aukštas	Žemas
Duomenų išgavimo greitis	Lėtas	Greitas
Duomenų saugojimas	Geresnis sudėtingoms struktūroms	Geresnis paprastoms struktūroms
XML patvirtinimas	Neprivalomas	Neprivalomas
Galimybė koreguoti XML dokumentą	Galima atmintyje	Negalima

1.5. Skyriaus išvados

Apžvelgus elektroninės parduotuvės elementus ir jų vaidmenį versle, galima teigti, jog pagrindinė šio verslo problema yra didelio kiekio prekių profilių informacijos automatizavimas ir apdorojimas. Be šios informacijos automatizavimo el. parduotuvė gali tiesiog neišgalėti samdyti didelio kiekio žmogiškųjų išteklių.

Išnagrinėjus XML struktūrą iš prekių profilių pusės, pastebėta, jog XML dokumentas teikiamas skirtingų asmenų gali kisti kardinaliai, kadangi jokios prekių profilių pateikimo specifikacijos nėra, kiekviena paslaugų teikėja ją nusistato pati, todėl kiekviena sistema turi galėti prisitaikyti prie įvairių XML dokumentų.

Išnagrinėjus struktūrizuotos informacijos XML duomenų šaltinio nuskaitymo metodus, pastebėta, kad prieš kuriant programinę įrangą reikia įvertinti, kokios apimties duomenų šaltiniai bus nuskaityti. DOM metodas yra skirtas sudėtingesniems XML formatams nuskaityti, bet jeigu dokumentas bus itin didelis (nuo 25000 prekių profilių) gali neužtekti operatyviosios (RAM) atminties sukurti DOM objektą programavimo aplinkoje, todėl, tokiu atveju, protingiau būtų naudoti SAX metodą.

Išanalizavus SOAP ir REST duomenų perdavimo architektūras, daroma išvada, jog REST architektūra gali būti naudojama reikiamam duomenų pasiekiamumui per naršyklę, kai nebūtinai prisijungimo duomenys, ir kai reikalingas duomenų kešavimas. SOAP architektūra dažniau yra

naudojama, kai reikalingas asinchroninis procesų vykdymas ar tada, kai žinutės tarpusavyje susijusios (prisijungimas vyksta tik vieną kartą).

Apžvelgus DOM ir SAX tyrimų duomenis, pastebėta, jog SAX yra efektyvesnė už DOM, kadangi ji gali būti pritaikyta ir didelių ir mažų apimčių dokumentams. SAX yra greitesnė už DOM ir sunaudoja mažiau atminties. Dėl šios priežasties dideli XML dokumentai (nuo 25000 prekių profilių) gali būti nuskaitomi naudojant SAX metodą.

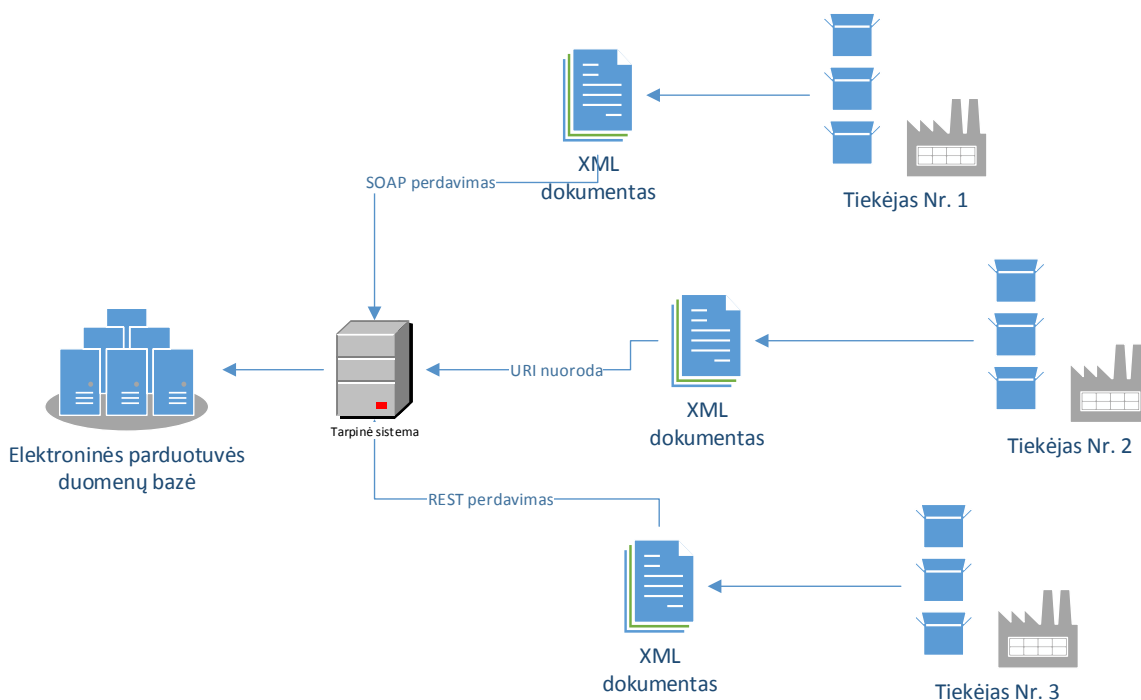
2. PREKIŲ APRAŠŲ SINCHRONIZACIJOS MODELIS

2.1. Prekių profilių sinchronizavimo problemos ir siūlomi sprendimo būdai

Lietuvos elektroninio verslo erdvėje dar ir šiuo metu yra naudojami prekių profilių atnaujinimo algoritmai. Bet daugelis jų turi galėti prisitaikyti prie esamos sistemos ir prekių profilių struktūros. Taigi, šiame darbe yra siūlomas modelis (22 pav.), kuris prekių profilių informaciją sinchronizuos per tarpinę sistemą, tam, kad užsakovo el. parduotuvės serveris būtų apkraunamas mažiau ir prekių profiliai korektiškai įsirašytų į el. parduotuvės duomenų bazę.

Prekių profilių sinchronizacijos modelis per tarpinę sistemą sprendžia kelias problemas elektroninėje parduotuvėje:

1. Netapačios struktūros ir skirtingos apimties prekių aprašai bei jų perdavimo būdai.
2. Duomenų stoka XML dokumente prekės įrašymui elektroninės parduotuvės duomenų bazėje.
3. Skirtingas prekių profilių atnaujinimo dažnis.
4. Neapibrėžtas prekės gyvavimo ciklas.

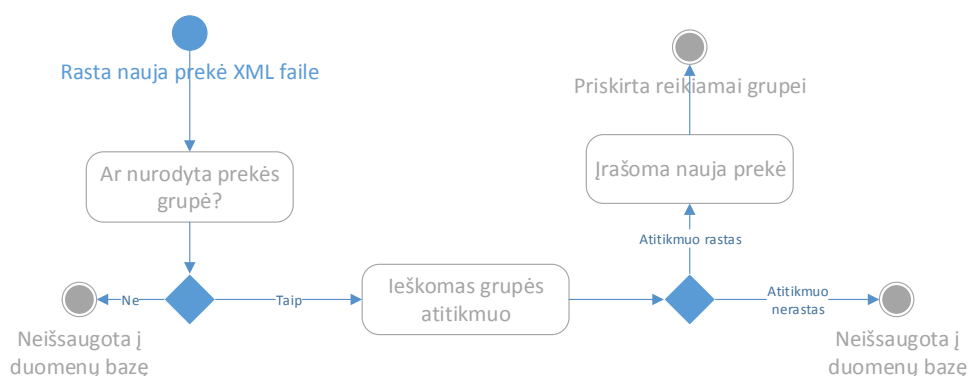


22 pav. Tiekėjai ir elektroninė parduotuvė

2.1.1. Duomenų stoka XML dokumente prekės įrašymui elektroninės parduotuvės duomenų bazėje

Visų pirma buvo susisiekti su keletu didžiausių prekių tiekėjų ir išanalizuoti jų teikiami XML failai, jų struktūra bei duomenys. 23 pav. vaizduojama, kaip įprastu atveju būtų galima įrašyti prekę į elektroninės parduotuvės duomenų bazę. Schemoje matomai trys atvejai, kai programa baigia darbą, du iš jų yra nepalankūs ir klaidingi. Tai yra, kai XML faile nerasta prekės grupė, ir kai nurodyta prekės grupė nerasta elektroninės parduotuvės grupių medyje. Taigi, integravus į parduotuvę tokio tipo sistemą, būtų prarasti kai kurie prekių aprašai ar jie netgi būtų patalpinti ne į reikiamą vietą, dėl to, kad

tiekėjo grupės pavadinimas savo duomenų bazėje turi kitokią reikšmę nei el. parduotuvės duomenų bazėje.



23 pav. Prekės įrašymas į el. parduotuvę.

Padaryta išvada, kad norint prisitaikyti prie esamos įmonės parduotuvės duomenų bazės ir buhalterijos programinės įrangos, būtina kurti tarpinę sistemą (*Middleware*) tiekėjų prekėms. Taip nuspėta todėl, kad parduotuvės ir tiekėjo prekių medis gali labai skirtis arba galimi atvejai, kai jų tiesiog nebūna ar jie būna anglų/lenkų/rusų kalbomis. Todėl tokia tarpinė sistema leis, įdėjus truputėlį vadybininko darbo, prekes įkelti į nurodytas grupes.

2.1.2. Skirtingas prekių profilių atnaujinimo dažnis

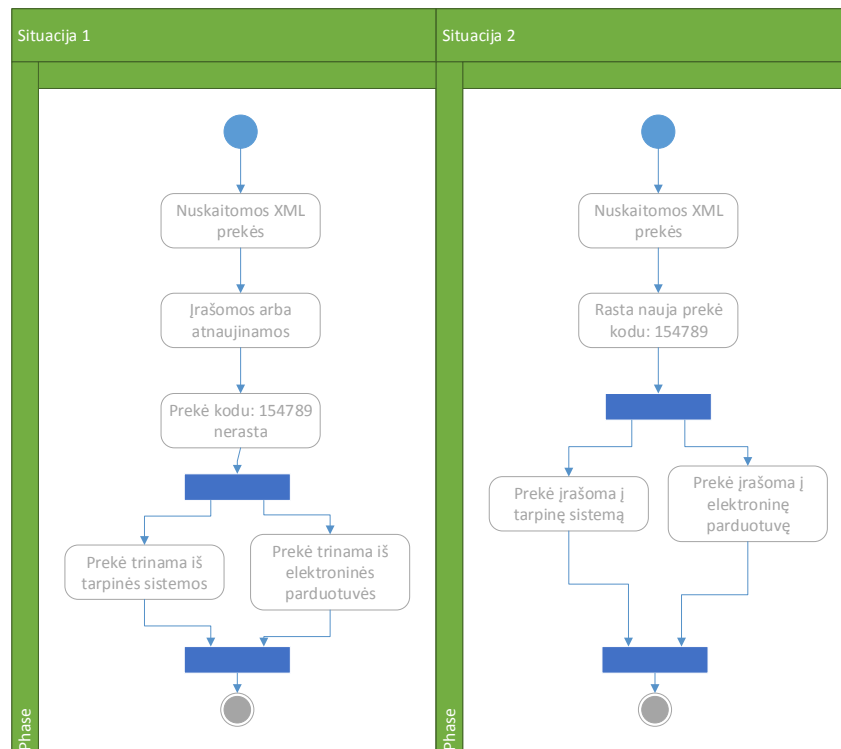
Skirtingi tiekėjai turi skirtingus prekių profilių kiekius. Todėl didesnę prekių nomenklatūrą turintis tiekėjas savo prekių profilių informaciją gali keisti daug dažniau, kai, pavyzdžiui, vienas svarbiausių pardavimo veiksnių – kiekis, sandėlyje gali kisti kas valandą. Dėl šios priežasties tarpinėje sistemoje turi būti numatyti parametrai, kurie leistų norimus tiekėjų XML dokumentus nuskaityti ir sinchronizuoti su el. parduotuvės sistema, ne kas 24 valandas, o taip pat ir kas 1 valandą. Taip bus sumažinta tikimybė, jog dėl greitai besikeičiančių sandėlių likučių prekių profiliuose bus parduota nebesama prekė.

2.1.3. Neapibrėžtas prekės gyvavimo ciklas

Dažniausiai tiekėjai neteikia duomenų apie prekes pašalintas iš jų sistemos, jie tiesiog nebeteikia duomenų apie jas XML faile. Bet būna atvejų, kai prekės vieną dieną teikiamos XML faile, o kitą jau nebe.

24 pav. vaizduojama kaip gali kisti duomenys sistemose jeigu prekę pašalintume jos neradus XML formato dokumente. Vienu laiko momentu ji dingtų iš sistemos, kitu laiko momentu ji vėl atsirastų joje kaip visiškai nauja prekė.

Dėl šios priežasties sistemai keliamas reikalavimas, kad, nerastos prekės XML faile, būtų žymimos kaip prekės, kurių tiekėjas neturi sandėlyje, o prekės, kurių redagavimo data yra senesnė nei 3 mėnesiai, būtų šalinamos iš sistemos. Taip bus sumažinta tikimybė dažniems duomenų šuoliams.



24 pav. Prekės aprašo pašalinimo/įrašymo schema

2.1.4. Papildomi sistemos reikalavimai

Papildomi sistemos reikalavimai buvo tokie:

- Jei tiekėjas nurodo, koks turi būti antkainis nuo prekės kainos, tokia kaina ir turi būti matoma klientui elektroninėje parduotuvėje. Kadangi kai kurie tiekėjai nori kontroliuoti savo prekių kainas rinkoje.
- Programinėje įrangoje, skirtoje vadybininkui, turi būti galimybė išjungti prekių atnaujinimą:
 - Aprašymams
 - Kainoms
 - Likučiams
 - Charakteristikoms
 - Nuotraukoms

Šis reikalavimas reikalingas tuo atveju, jei aprašymas ar kita informacija buvo netinkama vadybininkui, jis ją pakoregavo ir nenori, kad sistema tai pakeistų. Pasinaudojus programine įranga ir susiradus tą prekę, vadybininkas turėtų galėti išjungti atnaujinimą tai informacijos daliai. Tarpinė duomenų bazė čia pasitarnauja tuo, jog tiesiogiai atnaujinamų duomenų išjungti nebūtų galima, nes jų tiesiog nėra el. parduotuvės sistemoje.

2.1.5. Techniniai reikalavimai sistemai

Techniniai reikalavimai sistemai buvo tokie, kad sistema turi veikti nesutrikdant duomenų srauto kanalo, ir esant reikalui galėtų plėstis. Serverio techninės savybės:

- Procesorius: 19.20 GHz (8 x 2.40 GHz)
- Atmintis: 16384 MB + 8192 MB SWAP

- Disko vieta: 150 GB RAID
- RAID našumas: 12.5 MB/s skaitymas, 12.5 MB/s rašymas, 200 IOPS
- Pralaidumas: 100 Mbps
- Operacinė sistema: CentOS Linux 5.3
- Duomenų bazė: MySql 5.4
- PHP 5.3.14

2.2. Konceptinė prekių profilių sinchronizavimo modelio schema

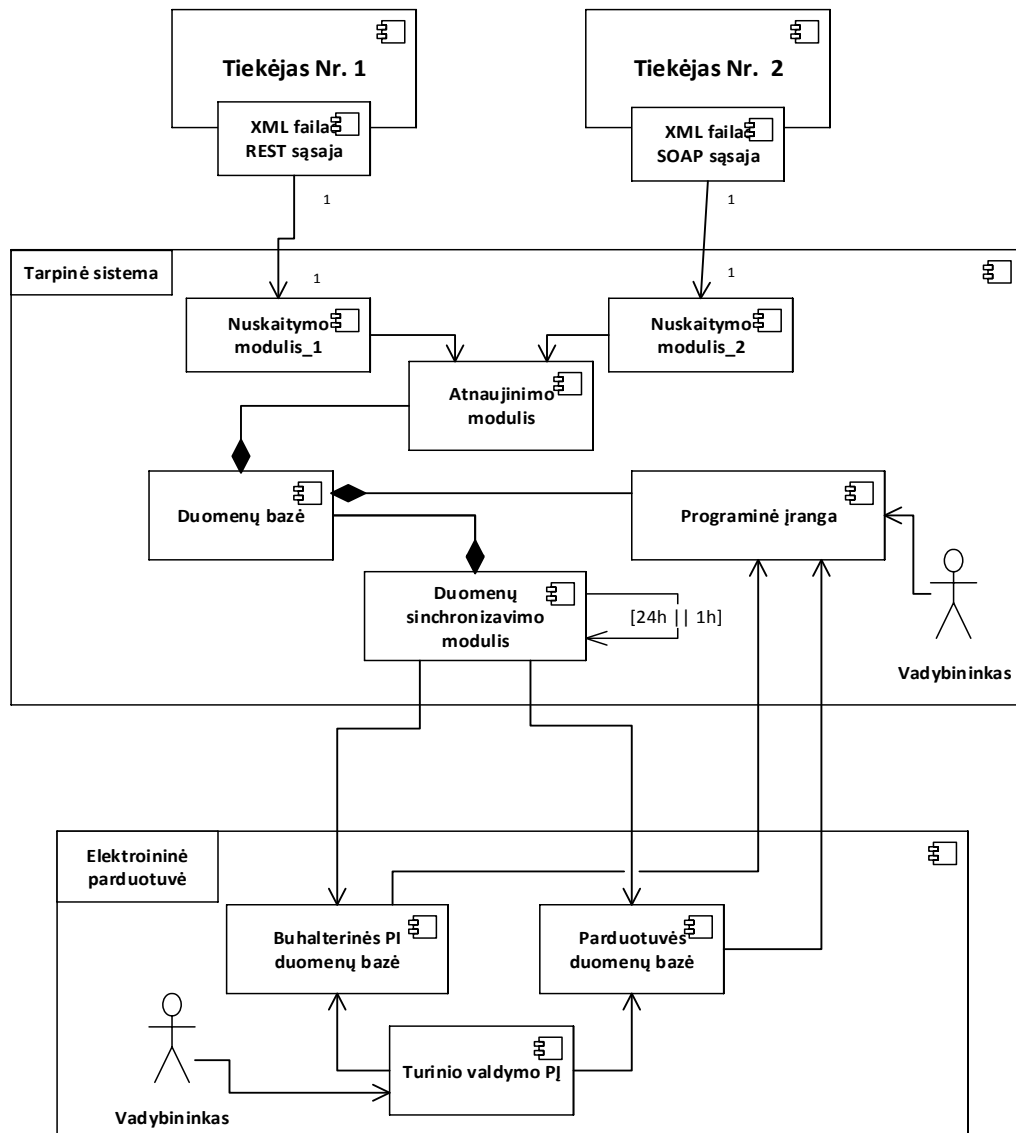
Atsižvelgiant į elektroninei prekybai kylančias problemas ir jų sprendimo būdus, sudarytas modelis, kurio konceptinė schema pateikta 25 pav. Visas modelis susideda iš trijų stambių dalių, tai yra: tiekėjai, elektroninė parduotuvė ir tarp jų egzistuojanti tarpinė prekių profilių sistema.

Kad sistema pradėtų savo darbą, tiekėjas turi pateikti dokumentaciją, koku būdu, SOAP, REST ar įrasta URI nuoroda galima patekti į jų XML failą. Struktūrizuoto dokumento aprašą, kokius duomenis jie pateikia, kokie elementai atitinka prekių profilio informaciją.

Programuotojas, išanalizavęs XML failą, savo ruožtu sistemoje turi sukurti modulį (klasės objektą), kuriuo būtų galima nuskaityti pateiktą XML failą. Kiekvienam tiekėjui šis modelis yra kuriamas atskirai, tai gali užimti nuo 20min iki 3h darbo, priklausomai kiek sudėtinga prekių perdavimo sistema yra. Taip yra daroma tam, kad visi prekių laukai būtų 100 proc. išsaugoti į duomenų bazę. Kitu atveju, naudojant kažkokius panašių žodžių algoritmus, išlieka tikimybė, kad informacija į duomenų bazę įsirašys nepilnai, tai būtų didelis trūkumas verslui, ir modelio įdėjai.

Kiekvienas nuskaitymo modulis kreipiasi į atnaujinimo modulio klasę, kuri savo ruožtu gali įrašyti ir atnaujinti prekes iš nuskaitytų XML prekės profilio duomenų.

Taigi visiems tiekėjo prekių profiliams atsidūrus tarpinės sistemos duomenų bazėje, vadybininkas gali jas matyti šios sistemos programinėje įrangoje. Jis tiesiog pasirenka tiekėją ir jam yra parodomos visos jo prekės, suskirstytos pagal tiekėjo prekių medį. Tada vadybininkas pats nusprendžia, kurias tiekėjo prekių grupes sukelti į el. parduotuvės prekių grupes. Prieš įkeliant vadybininkas gali pasirinkti, koku antkainiu pardavinėti įkeltą prekę (jei prekės antkainio nenustato tiekėjas), ar atnaujinti visas prekės aprašo dalis (aprašymus, kainas, likučius, charakteristikas, nuotraukas), ar įkelti esamas prekės nuotraukas (kartais nuotraukos gali būti su vandens ženklais ir jų vadybininkas neturėtų įkelti).



25 pav. Konceptinė modelio komponentų diagrama

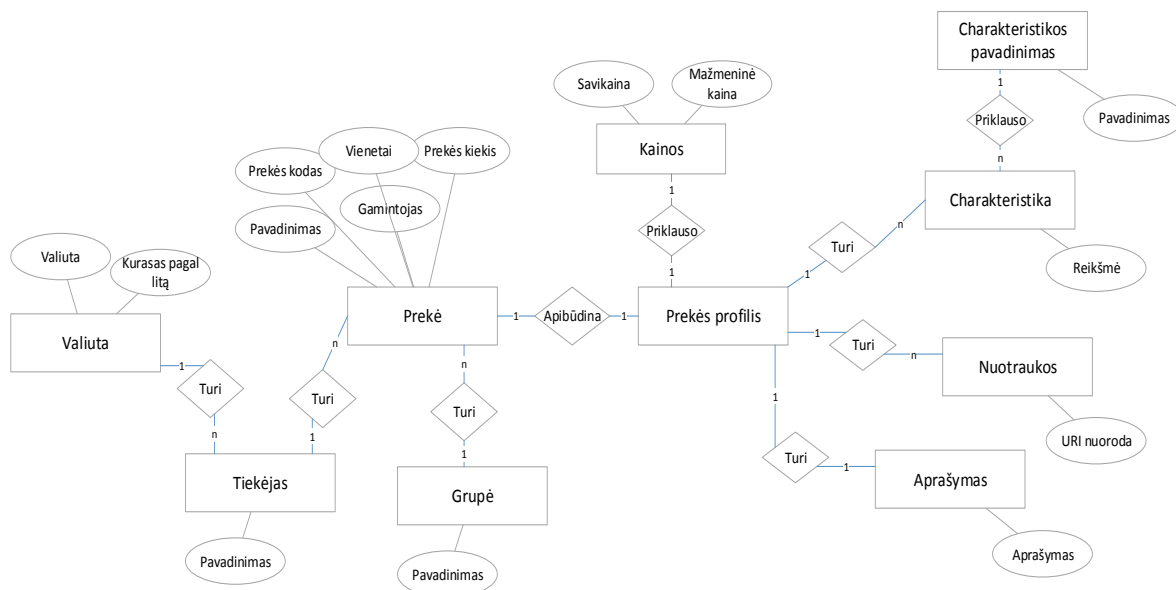
Vadybininkui, atlikus savo darbą nustatytu laiku, gali pradėti veikti duomenų sinchronizavimo algoritmas, kuris tikrina duomenis tarpinės sistemos duomenų bazėje. Patikrinama, kurie prekių aprašai joje buvo atnaujinti ir tuos pačius duomenis atnaujina elektroninės parduotuvės bei buhalterijos PĮ duomenų bazėse.

Toks duomenų ratas vyksta kiekvieną naktį, duomenys atnaujinami į tarpinės sistemos duomenų bazę, o vėliau, pagal datą, kurios prekės buvo atnaujintos, tą naktį duomenys perkeliama į elektroninės parduotuvės duomenų bazes.

Elektroninės parduotuvės dalyje yra dar vienas komponentas - turinio valdymo sistema, tai yra PĮ, kuri leidžia greitai koreguoti prekių informaciją. Ši PĮ šiek tiek susijusi su tarpinės sistemos duomenų baze, kadangi yra tikrinama ar prekės elektroninėje parduotuvėje yra susietos su tarpinės sistemos duomenų baze. Tokiu atveju vadybininkui nėra leidžiama koreguoti prekės savikainos, kadangi ji yra reguliuojama tarpinės sistemos. Arba, jei vadybininkas išjungia prekę iš parduotuvės, kad jos neįjungtų tarpinės sistemos duomenų sinchronizavimo modulis, automatiškai išjungiamas prekės likučių atnaujinimo dalis.

2.3. Tarpinės sistemos duomenų bazės struktūra

Siekiant sukurti lanksčią informacijos saugojimo struktūrą, kad įvairiomis formomis pateikiama tiekėjo informacija galėtų atsidurti tarpinėje sistemoje, sumodeliuota duomenų bazė, kurios esybių ryšiai pateikti 26 pav. , o detaliai aprašyti 2 lentelėje.



26 pav. Esių ryšių diagrama

2 lentelė. Duomenų bazės esybių sąryšiai

Pavadinimas	Prekė			
Aprašymas	Šie duomenys skirti pateikti pagrindinę prekės informaciją, ir ją identifikuoti.			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Pavadinimas	Tekstinis	1,..1000	1
	Prekės kodas	Tekstinis	1,..100	1
	Vienetai	Tekstinis	1,..50	0
	Gamintojas	Tekstinis	1,..100	0
	Kiekis	Slankaus kablelio	15,4	1
	Prekės profilis	Rinkinys	-	1
	Grupė	Rinkinys	-	1
	Tiekėjas	Rinkinys	-	1
Sąsaja	1:1 Prekės profilis n:1 Grupė n:1 Tiekėjas			

Pavadinimas	Grupė			
Aprašymas	Šie duomenys skirti apibūdinti prekės vietą kategorijų medyje tiekėjo inf. sist.			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Pavadinimas	Tekstinis	1,..1000	1
Sąsaja	1:n Prekė			

Pavadinimas	Tiekėjas			
Aprašymas	Šie duomenys skirti apibūdinti prekių tiekėją			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Pavadinimas	Tekstinis	1,..255	1

	Valiuta	Rinkinys	-	1
Sąsaja	1:n Prekė n:1 Valiuta			

Pavadinimas	Valiuta			
Aprašymas	Skirta nurodyti tiekėjo valiutos kursą lito atžvilgiu			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Valiuta	Tekstinis	1,..10	1
	Kursas pagal litą	Slankaus kabelio	10,4	1
Sąsaja	1:n Tiekėjas			

Pavadinimas	Prekės profilis			
Aprašymas	Duomenys skirti detaliam apibūdinti tiekėjo prekę.			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Kainos	Rinkinys	-	1
	Charakteristika	Rinkinys	-	0
	Nuotraukos	Rinkinys	-	0
	Aprašymas	Rinkinys	-	0
Sąsaja	1:1 Kainos 1:n Charakteristika 1:n Nuotraukos 1:1 Aprašymas			

Pavadinimas	Kainos			
Aprašymas	Duomenys skirti prekės kainų maržai.			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Savikaina	Slankaus kabelio	10,4	1
	Mažmeninė kaina	Slankaus kabelio	10,4	0
Sąsaja	1:1 Prekės profilis			

Pavadinimas	Charakteristika			
Aprašymas	Duomenys apibūdinantys prekės technines savybes			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Reikšmė	Tekstinis	0..., 65535	1
	Charakteristikos pavadinimas	Rinkinys	-	1
Sąsaja	n:1 Prekės profilis n:1 Charakteristika pavadinimas			

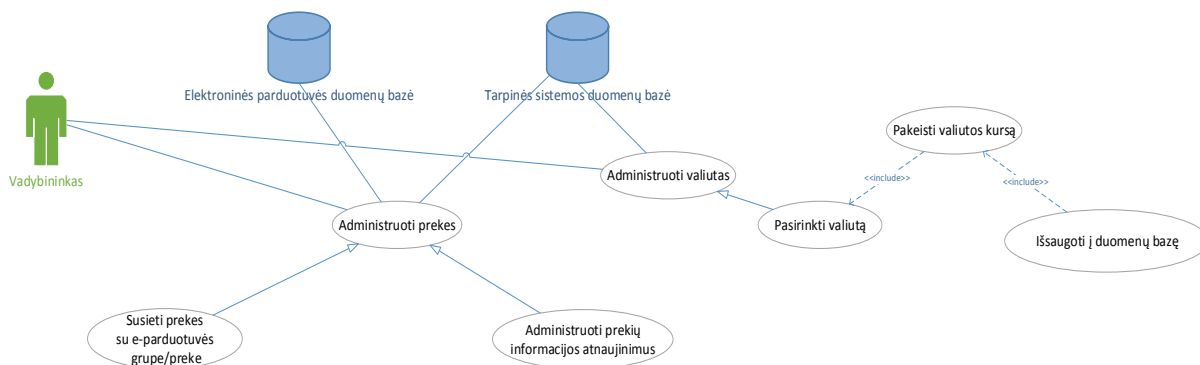
Pavadinimas	Charakteristika pavadinimas			
Aprašymas	Techninių savybių pavadinimai			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Pavadinimas	Tekstinis	0..., 255	1
Sąsaja	1:n Charakteristika			

Pavadinimas	Nuotraukos			
Aprašymas	Duomenys teikiantys vaizdinę prekės informaciją			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	URI nuoroda	Tekstinis	0...,2000	1
Sąsaja	n:1 Prekės profilis			

Pavadinimas	Aprašymas			
Aprašymas	Prekės detalaus tekstinio aprašymo duomenys			
Struktūra	Atributas	Tipas	Apribojimai	Būtinumas
	Aprašymas	Tekstinis	0..., 16777215	1
Sąsaja	n:1 Prekės profilis			

2.4. Tarpinės sistemos programinė įranga

Sprendžiant problemą kaip užtikrinti nuoseklų ir teisingą prekių įrašymą, į e-parduotuvės duomenų bazes buvo sumodeliuota programinė įranga. Pagrindinė šios programinės įrangos paskirtis - suteikti galimybes vadybininkui manipuluoti tiekėjo prekių informacija, nurodyti kurios tiekėjų grupės turi atitikti el. parduotuvės grupes, bei valdyti jau įkeltų prekių profilius.



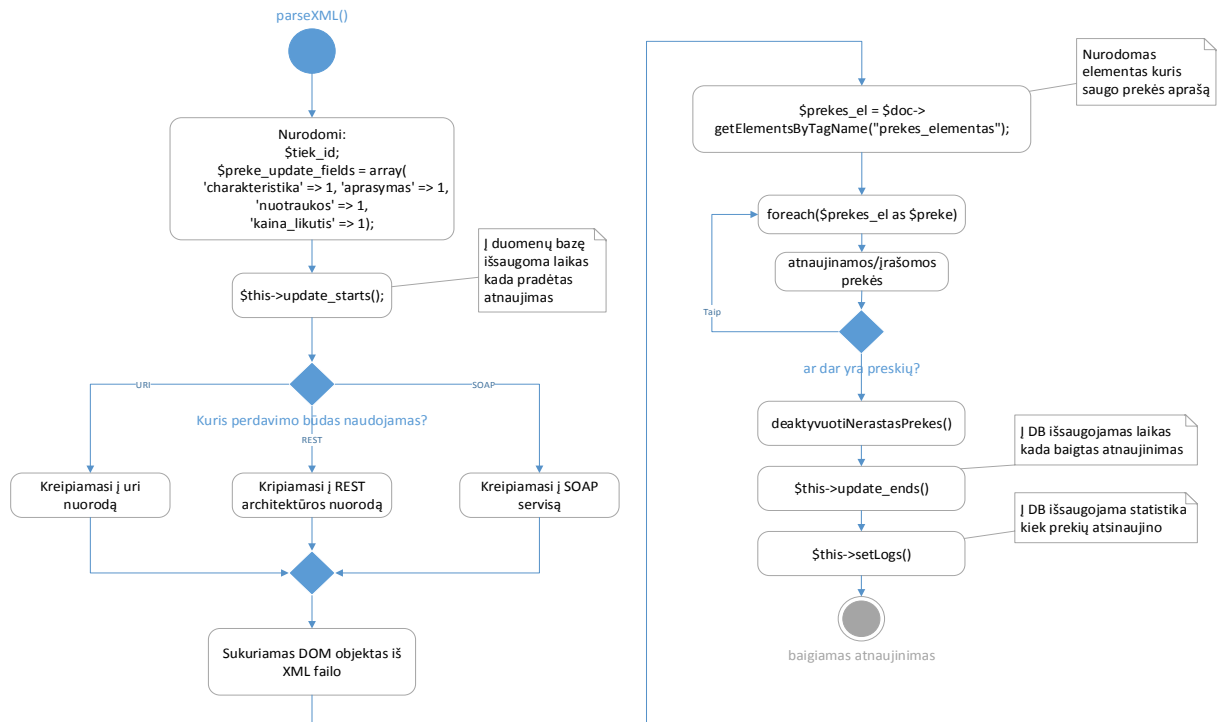
27 pav. Tarpinės sistemos PI funkcijos

Vadybininkui suteikiama laisvė atlikti tokius veiksmus (žr. 27 pav.):

1. Tiekėjo prekių ir grupių susiejimas su e-parduotuvės prekėmis ir grupėmis.
2. Tiekėjo prekių informacijos atnaujinimų administravimas.
3. Valiutų kursų administravimas.

2.5. Nuskaitymo modulis

Tarpinėje sistemoje norit nuskaityti tiekėjo XML failą, kiekvienam iš jų kuriamas nuskaitymo modulis (žr. 28 pav.).



28 pav. Nuskaitymo modulio schema

Algoritmas pradedamas nustatant pradines reikšmes, kurios reikalingos tolimesniam prekių atnaujinimui, šios reikšmės pavaizduotos lentelėje (žr. 3 lentelė.).

3 lentelė. Nuskaitymo modulio parametrai

Kintamasis	Tipas (reikšmė)	Aprašymas
tiek_id	Sveikas skaičius	Tiekėjo identifikacinis numeris iš parduotuvės duomenų bazės
valiuta	Slankaus kablelio skaičius	Jei tiekėjas prekes pateikia ne litais, iš duomenų bazės grąžinamas tos valiutos kursas.
preke_update_fields[` charakteristika `]	Sveikas skaičius (1/0)	Naujų prekių, charakteristikų atnaujinimo parametras
preke_update_fields [` aprasymas `]	Sveikas skaičius (1/0)	Naujų prekių, aprašymo atnaujinimo parametras
preke_update_fields [` nuotraukos `]	Sveikas skaičius (1/0)	Naujų prekių, nuotraukų atnaujinimo parametras
preke_update_fields[` kaina_likutis `]	Sveikas skaičius (1/0)	Naujų prekių, kainos ir likučio atnaujinimo parametras

Nustačius visus parametrus, atnaujinimas pradedamas išsaugant atnaujinimo pradžios laiką duomenų bazėje, naudojantis funkcija update_start();

Sekantis žingsnis yra XML failo gavimas iš tiekėjo sistemos. Jei tiekėjas naudoja paprastą URI ar REST architektūra paremtą nuorodą, XML failas gaunamas naudojantis `file_get_contents()` funkcija. Taip pat galimas ir SOAP architektūros duomenų perdavimo būdas. Naudojant SOAP yra sukuriamas specialus objektas, kuriame nurodoma WSDL nuoroda, taip pat sukuriamas SoapHeader objektas, kuriuo perduodami prisijungimo duomenys ir tiekėjo nurodytomis funkcijomis, kurios pateiktos WSDL faile, grąžinami duomenys.

Toliau sukuriamas DOM objektas, į kurį nurodomas XML failas. Ciklo pagalba, per nurodytą masyvą, nuskaitomos ir atnaujinamos/įrašomos prekės į tarpinę duomenų bazę. Nuskaitytus prekes yra vykdoma funkcija `deaktyvuotiNerastasPrekes()`, kurios pagalba visos, prieš tai buvusios prekės XML faile, o dabar nesamos, nustatomos kaip išparduotos prekės. Po šios funkcijos seka funkcija išsauganti nuskaitymo modulio darbo pabaigos laiką, paskutinė funkcija yra išsauganti statistinius duomenis, kiek atsinaujino prekių, kiek jų buvo pašalinta, kelioms iš jų buvo atnaujintos nuotraukos.

4 lentelė. Metodų aprašymų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
<code>update_starts()</code>	Metodas, kuris išsaugo, kada prasidėjo tiekėjo atnaujinimas	Dabartinė data	Įrašo duomenų bazėje ID
<code>getElementByTagName(name)</code>	DOM nuskaitymo algoritmo funkcija, reikalinga gauti reikiams elementams pagal jų pavadinimą	Elemento pavadinimas	Elementų rinkinys
<code>deaktyvuotiNerastasPrekes()</code>	Funkcija nustato prekių kiekius lygius nuliui, toms prekėms kurios nebebuvo pateiktos XML faile	Tiekėjo identifikacinis numeris, nuskaitytų prekių kodų rinkinys	-
<code>update_ends()</code>	Metodas fiksuojantis atnaujinimo pabaigą duomenų bazėje	Dabartinė data, įrašo identifikacinis numeris iš <code>update_starts()</code> funkcijos.	-
<code>setLogs();</code>	Metodas apskaičiuojantis ir išsaugantis naujų bei atnaujintų prekių statistiką, pasitelkiant SQL funkcijas.	-	-

2.5.1. Prekių nuskaitymo algoritmas

Norint labiau supažindinti su prekių nuskaitymo algoritmu, pateikiamas detalesnis pavyzdys (žr. 29 pav.). Kai grąžinamas XML failas, ciklo būdu po vieną nuskaitomi prekės elementai, kurie savyje saugo visą prekės informaciją. Informacijai saugoti PHP atmintyje naudojamas masyvas \$preke. Visos galimos šio masyvo reikšmės pateiktos lentelėje (žr. 5 lentelė.)

5 lentelė. Atnaujinimo duomenys

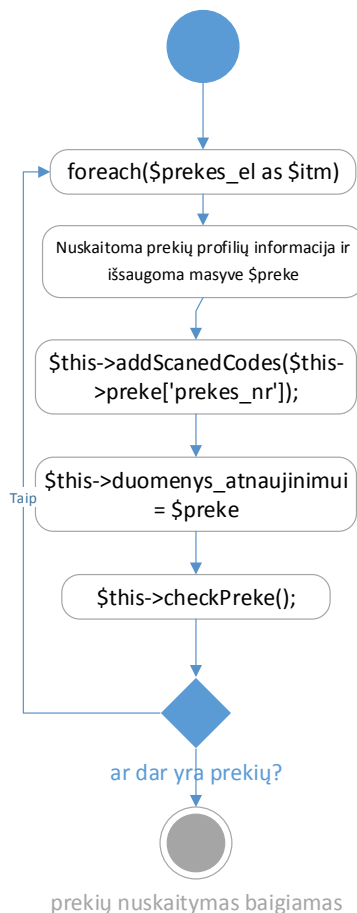
Kintamasis	Tipas	Aprašymas	Ar kintamasis privalomas?
\$preke["prekes_nr"]	Tekstinis	Prekės identifikacinis numeris(id) tiekėjo informacinėje sistemoje.	Taip
\$preke["pavadinimas"]	Tekstinis	Prekės pavadinimas tiekėjo informacinėje sistemoje.	Taip
\$preke["kiekis_sandelyje"]	Sveikas skaičius	Prekės kiekis sandėlyje tiekėjo informacinėje sistemoje. Šis skaičius gali ir nebūti tikslus, svarbiausia, jei kiekis >0, prekė bus parduodama, o jei kiekis <=0, prekė bus išparduota.	Taip
\$preke["grupe"]	Tekstinis	Grupės pavadinimas (ar visas kelias), priklausančiai prekei.	Taip
\$preke["kaina"]	Slankaus kablelio skaičius	Prekės savikaina elektroninei parduotuvei.	Taip
\$preke["mazmenine_kaina"]	Slankaus kablelio skaičius	Rekomenduojama kaina pardavimui.	Ne
\$preke["charakteristikos"]	Rinkinys	Prekės charakteristikos, detaliau apibūdinančios prekę.	Ne
\$preke["nuotraukos"]	Rinkinys	Prekės nuotraukos.	Ne

Visi kintamieji gaunami tiesiogiai iš XML dokumento elementų, bet du iš jų, tai yra, charakteristikos ir nuotraukos, dažniausiai reikalauja duomenų surinkimo iš kelių elementų, tam naudojamos funkcijos getChr() ir getFoto(), kuriomis naudojantis nuskaitomi ir sugrupuojami visi reikalingi elementai ir viskas įrašoma į masyvą.

Surinkus visą prekės informaciją, į atmintį išsaugojamas prekės numeris, tam, kad vėliau būtų galima patikrinti, kurios prekės dingo iš XML failo. Tada duomenys išsaugojami į tam skirtą kintamąjį \$duomenys_atnaujinimui, pagal kurį bus atnaujinama informacija.

Atlikus visus veiksmus, iškviečiamas atnaujinimo modulio metodas checkPreke(), pagal kurį atnaujinamos prekės tarpinėje duomenų bazėje. Šis metodas plačiau aprašytas 2.7. skyriuje.

```
getElementsByTagName("prekes_elementas");
```



29 pav. Prekių nuskaitymo ciklas

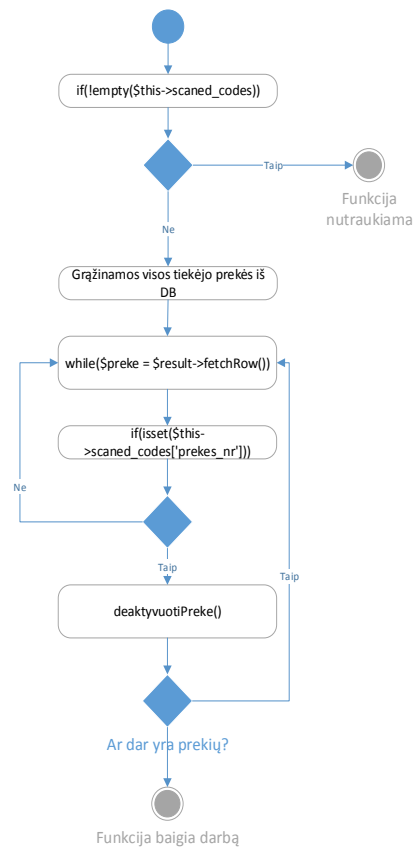
6 lentelė. Metodų aprašymų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
addScanedCodes()	Metodas, kuris PHP atmintyje išsaugo nuskaitytų prekių identifikacinius numerius.	Nuskaitytos prekės identifikacinis numeris	-
checkPreke()	Metodas patikrinantis duomenų bazėje prekės informaciją ir nusprendžiantis, ar reikia įrašyti naują prekę, ar reikia	Nuskaitytų prekių informacijos rinkinys (\$preke)	-

	atnaujinti prekės duomenis, ar nereikia nieko atlikti su informacija.		
--	---	--	--

2.5.2. Nerastų prekių deaktivavimo algoritmas

Jei tiekėjas išjungė prekę iš XML dokumento ir niekur nepateikia prekių, kurios yra išimtos, apsaugoti nuo duomenų neatitikimo tarp tiekėjo informacinės sistemos ir el. parduotuvės sistemos yra sukurtas algoritmas (žr. 30 pav.). Nuskaitant prekes iš tiekėjo XML failo, po kiekvienos prekės patikrinimo, yra išsaugojamas prekės kodas tam skirtame masyve \$scanned_codes. Po visų prekių nuskaitymo yra vykdoma funkcija pateikta 31 pav.



30 pav. Nerastų prekių XML faile deaktivavimo algoritmas

Funkcija pradeda darbą tikrindama, ar prekių kodų masyvas nėra tuščias. Jei tiekėjas klaidos atveju pateiktų tuščią ar nekorektišką XML failą, jo visos prekės deaktyvuotųsi. Ši sąlyga yra apsaugojimui nuo to. Jei prekės kodų masyvas nebuvo tuščias, iš duomenų bazės grąžinamos visos tiekėjo prekės. Ir po vieną tikrinamos while ciklu. Tikrinimas vyksta tokiu būdu:

1. Patikrinama ar grąžinta prekė yra scanned_codes masyve.
2. Jei ši prekė masyve, tai yra grąžinama sekanti prekė.
3. Jei ši prekė nebuvo rasta masyve, duomenų bazėje prekės likutis nustatomas lygus 0.

Kai nebelieka grąžinamų prekių, funkcija baigia darbą.

7 lentelė. Nerastų prekių XML faile deaktyvavimo algoritmo metodų aprašų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
deaktyvuotiPreke()	Metodas nustatantis prekės kiekį lygų 0, duomenų bazėje.	Prekė numeris, tiekėjo identifikacinis kodas.	-

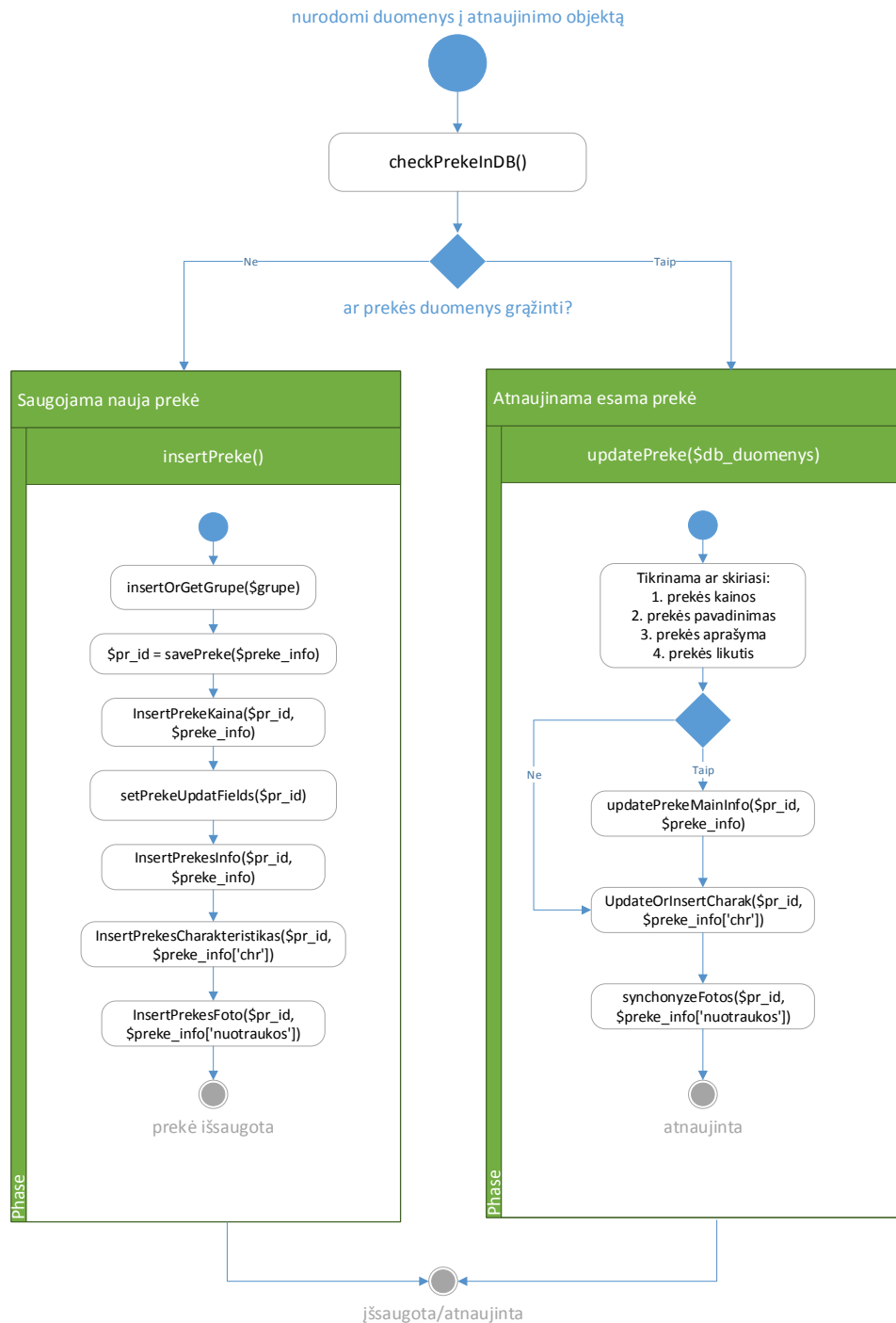
2.6. Atnaujinimo modulis

Kad prekės atsirastų ar atsinaujintų tarpinės sistemos duomenų bazėje, sukurtas atnaujinimo modulio algoritmas, kuris nusprendžia, kada reikia pakeisti prekės informaciją tarpinėje sistemoje, o kada reikia integruoti naują prekę į šią sistemą.

Algoritmas pradedamas bandant grąžinti prekės informaciją iš duomenų bazės pagal duomenis nurodytus į atnaujinimo modulio objektą. Jei prekė rasta yra grąžinama informacija, kuri pateikta lentelėje (žr. 8 lentelė.).

8 lentelė. Grąžinami prekės duomenys iš duomenų bazės

DB lentelė	Kintamasis	Tipas	Aprašymas
preke	pid	Sveikas skaičius	Prekės identifikacinis numeris iš tarpinės sistemos duomenų bazės
preke	pavadinimas	Tekstinis	Prekės pavadinimas tarpinėje duomenų bazėje
grupe	grupe_pav	Tekstinis	Grupės pavadinimas tarpinėje duomenų bazėje
preke	pre_kiekis	Sveikas skaičius	Prekės kiekis tarpinėje duomenų bazėje
preke	del_date	Data	Jei prekė buvo ištrinta, prekės ištrynimo data
preke	gamintojas	Tekstinis	Prekės gamintojas tarpinėje duomenų bazėje
preke_info	aprasymas	Tekstinis	Prekės aprašymas tarpinėje duomenų bazėje
preke_kaina	savikaina	Slankaus kablelio skaičius	Prekės savikaina
preke_kaina	rrp_kaina	Slankaus kablelio skaičius	Rekomenduojama pardavimo kaina el. parduotuvei
nuotrauka	nuotraukos	Tekstinis	Visos prekės nuotraukų nuorodos grąžinamos vienoje eilutėje, atskirtos simboliais: [+]



31 pav. Atnaujinimo modulio algoritmo schema

Sprendimas, ar nuskaityti prekės duomenys iš XML failo turi būti atnaujinami ar įrašyti, daromas viena sąlyga (žr. 31 pav.), kuri tikrina, ar yra grąžinti duomenys (žr. 8 lentelė.). Jei prekė rasta duomenų bazėje ir jos informacija skiriasi nuo duomenų XML dokumente, prekė atnaujinama, kitu atveju - įrašoma nauja prekė.

9 lentelė. Prekės sukūrimo metodų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
insertPreke()	Metodas, skirtas įrašyti naujai prekei.	Nuskaitytos prekės XML informacijos rinkinys	-

insertOrGetGrupe()	Metodas, patikrinantis ar nurodytas prekės grupės pavadinimas jau yra duomenų bazėje, nesamos kategorijos įrašomos į tarpinės sistemos duomenų bazę.	Prekės kategorijos pavadinimas	Grupės identifikacinis numeris tarpinėje duomenų bazėje.
savePreke()	Sukuriamas pagrindinis prekės įrašas, prie kurio rišasi visa kita papildoma informacija.	Prekės informacijos rinkinys (pavadinimas, prekės identifikacinis numeris, gamintojo pavadinimas, grupės ID, prekės kiekis, tiekėjo identifikacinis numeris)	Sukurta prekės ID.
insertPrekeKaina()	Sukuriamas prekės kainos įrašas.	Prekės ID, prekės savikaina, rekomenduojama prekės kaina	-
setPrekeUpdateField()	Sukuriamas įrašas, kuriame saugoma kurią prekės informaciją galima atnaujinti. (pagal nutylėjimą, vėliau vadybininkas visa tai gali koreguoti).	Prekės ID	-

10 lentelė. Prekės atnaujinimo metodų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
updatePreke()	Metodas, skirtas atnaujinti prekės duomenis.	Nuskaitytos prekės XML informacijos rinkinys, prekės informacijos iš duomenų bazės rinkinys.	-
updatePrekeMainInfo()	Jeigu tarpinėje duomenų bazėje randamas informacijos	Prekės informacijos rinkinys iš XML failo.	-

	neatitikimas su informacija prekės informacija XML faile, išskviečiama ši funkcija, kuri atnaujina visus prekės duomenis.		
UpdateOrInsertCharak()	Tam, kad neapkrauti duomenų bazės papildomais informacijos sulyginimais, kiekvienai nuskaitytai prekei atnaujinamos charakteristikos nepriklausomai ar jos sutampa ar ne.	Prekės charakteristikų rinkinys.	-
synchronizeFotos()	Metodas, sulyginantis prekės duomenų bazėje esantį nuotraukų rinkinį su nuskaitytos XML nuotraukų informacijos rinkiniu, ir sulyginantis duomenis.	Tarpinės duomenų bazės nuotraukų rinkinys ir prekės XML failo nuotraukų rinkinys.	-

2.7. Duomenų sinchronizavimo modulis

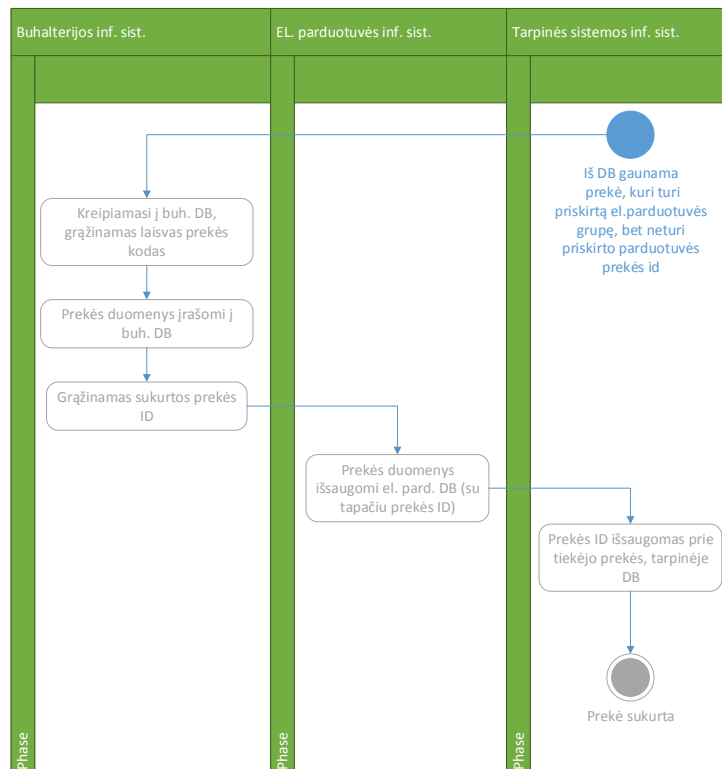
Elektroninės parduotuvės duomenų sinchronizavimui su tarpinės sistemos duomenimis, suprojektuotas duomenų sinchronizavimo modulis, kuris susideda iš tokių algoritmų:

1. Prekių kainos ir likučio atnaujinimo algoritmas.
2. Prekių aprašymo atnaujinimo algoritmas.
3. Prekių nuotraukų atnaujinimo algoritmas.

Kita prekės informacija, tokia kaip prekės vienetai, gamintojas ir prekės pavadinimas, automatiškai nebekoreguojami dėl buhalterinės programos apribojimų, nors, esant poreikiui, tokie algoritmai gali būti sukurti.

2.7.1. Prekės įrašymas į el. parduotuvės informacinę sistemą

Pradinį darbą, kad prekės atsidurtų el. parduotuvėje, atlieka vadybininkas. Jis nurodo, kurias prekes ar prekių grupes įkelti į el. parduotuvės atitinkamas prekių kategorijas. Algoritmas, kaip prekė įrašoma į parduotuvės DB, pateiktas 32 pav.



32 pav. Prekės įrašymo į el. parduotuvę algoritmas

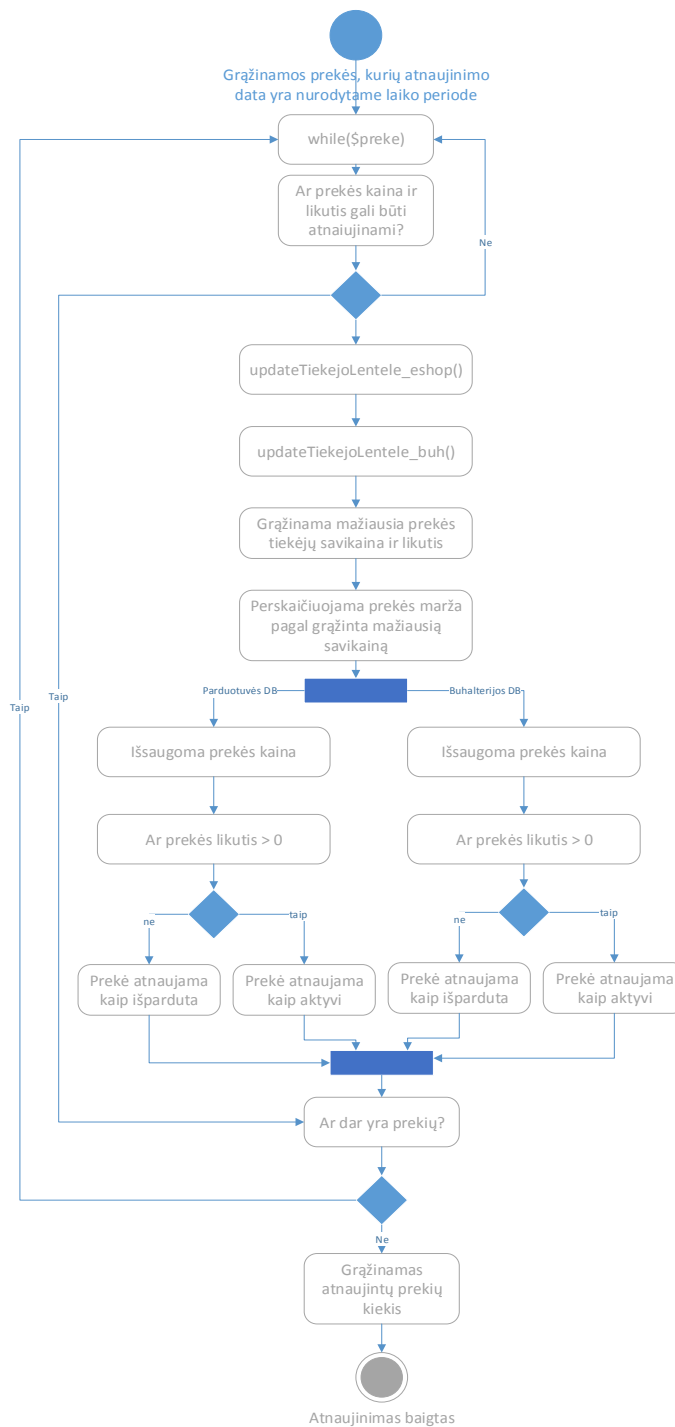
Algoritmas pradamas gražinant prekes iš duomenų bazės, kurios turi priskirtą el. parduotuvės kategorijos ID, bet neturi priskirto prekės ID iš el. parduotuvės duomenų bazės. Visų pirma kreipiamasi į buhalterijos DB ir gražinamas laisvas prekės kodas, tada į tą pačią DB įrašoma visa prekės informacija ir, atlikus šį veiksma, gražinamas sukurtos prekės ID buhalterijos DB.

Nuosekliais veiksmais prekės informacija su tuo pačiu prekės ID buhalterinei DB įrašoma į el. parduotuvės duomenų bazę.

Atlikus visus veiksmus, sėkmingai sukurtos prekės ID išsaugomas tiekėjo prekės profilio informacijoje, tarpinės sistemos duomenų bazėje.

2.7.2. Prekės kainos ir likučio atnaujinimas

Algoritmas, pavaizduotas 33 pav. skirtas tarpinės duomenų bazės prekių kainoms suvienodinti su internetinės parduotuvės prekių kainomis. Taip pat, pagal prekės likutį tiekėjo sandėlyje, nustatyti prekės aktyvumo būseną el. parduotuvėje.



33 pav. Prekės atnaujinimas

Algoritmas gali būti naudojamas įvairiais laiko tarpais, taip kaip bus nurodyta programuotojo nustatymuose. Galimas valandinis atnaujinimas ar, dažniau naudojamas, kartą į parą (naktinis atnaujinimas, kai parduotuvė mažiau apkrauta) atnaujinimas ir t.t.

Metodas pradedamas kreipiantis į duomenų bazę ir gražinant prekes, kurios buvo atnaujintos tarpinėje sistemoje nurodytame laiko tarpe. Visų pirma yra patikrinama, ar prekei nėra nustatytas dabar vykdomas kainos ir likučio atnaujinimo draudimas. Jei draudimas galioja, ši prekė tiesiog praleidžiama ir pereinama prie kitos prekės, kitu atveju, gražintos prekės savikaina kartu su prekės identifikavimo kodu tiekėjo informacinėje sistemoje, išsaugoma (jei jau egzistuoja atnaujinama) el. parduotuvės ir buhalterijos duomenų bazėse.

El. parduotuvės ir buhalterinės duomenų bazės yra suprojektuotos taip, jog viena prekė gali tūrėti n tiekėjų, o prekės marža turi būti paskaičiuojama pagal pigiausią tiekėją su likučiu, kuris didesnis nei 0. Jei visi tiekėjai neturi sandėlyje konkrečios prekės, kainos marža apskaičiuojama pagal pigiausią kainą su likučiu 0. Taigi, atsižvelgiant į minėtas taisykles, po kainos atnaujinimo parduotuvės informacinėje sistemoje, kreipiamasi į duomenų bazę gauti pigiausią prekės kainą ir atitinkamą likutį tiekėjo sandėlyje.

Pagal grąžintas kainas, likutį ir antkainio procentą yra apskaičiuojama kainos marža, nusprendžiama, kokia būsena bus nustatyta prekėje (išparduota/aktyvi), ir visa tai išsaugoma parduotuvės informacinėje sistemoje.

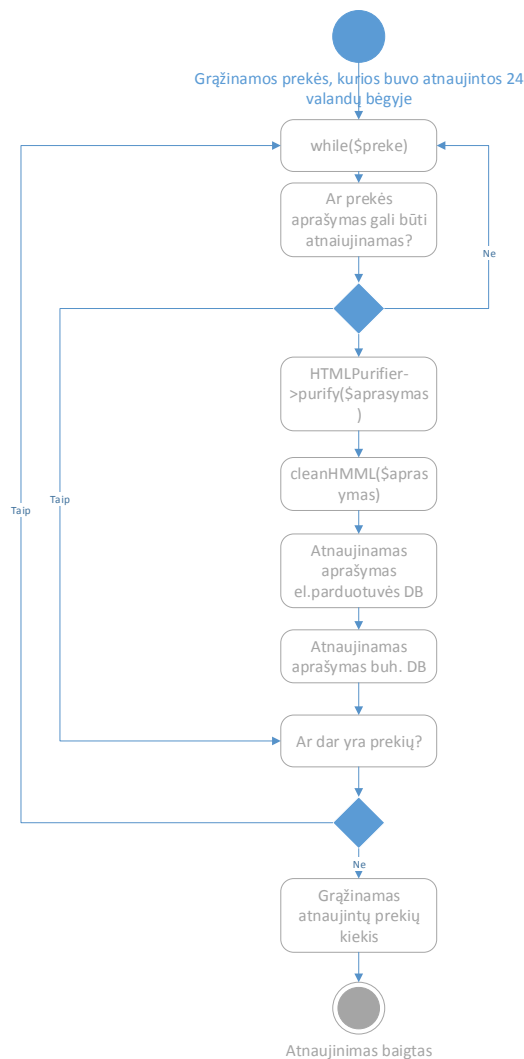
Cikle neberadus daugiau prekių, algoritmas grąžina, kiek prekių buvo atnaujinta. Vėliau šis skaičius gali būti saugomas statistikos informacijai kaupti.

11 lentelė. Metodų lentelė

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
updateTiekejoLentele_esh op()	Metodas išsaugantis prekės tiekėjo kainų informaciją el. parduotuvės duomenų bazėje	Prekės savikaina, rekomenduojama pardavimo kaina, likutis sandėlyje bei prekės tiekėjo identifikacinis kodas	-
updateTiekejoLentele_bu h()	Metodas išsaugantis prekės tiekėjo kainų informaciją buhalterijos duomenų bazėje	Prekės savikaina, rekomenduojama pardavimo kaina, likutis sandėlyje bei prekės tiekėjo identifikacinis kodas	-

2.7.3. Prekės aprašymo atnaujinimas

Siekiant suvienodinti prekių aprašymus tarpinėje duomenų bazėje su elektroninės parduotuvės informacine sistema, sudarytas šis algoritmas (žr. 34 pav.)



34 pav. Prekės aprašymo atnaujinimo algoritmas

Kaip ir kiti duomenų sinchronizavimo ir atnaujinimo algoritmai, jo aktyvavimo laikas yra nustatomas iš serverio.

Aktyvavęsis algoritmas darbą pradeda grąžindamas prekes, kurios atnaujintos 24-rių valandų bėgyje, iš tarpinės duomenų bazės. Vykdamas ciklą, kiekviena prekė yra patikrinama ar nėra nustatytas draudimas atnaujinti šios tiekėjo prekės aprašymą į elektroninės parduotuvės informacinę sistemą, jei toks draudimas galioja, šios prekės atnaujinimas praleidžiamas ir einama prie sekančios prekės.

Prekės aprašymo atnaujinimo draudimui negaliojant, visų pirma, naudojantis įskiepiu HTMLPurifier[17], patikrinama ar aprašymo HTML struktūra yra tinkama, ar visi elementai uždaryti tvarkingai.

Sekantis žingsnis yra HTML išvalymas nuo kenkėjiško turinio, tokio kaip:

1. JavaScript veikimo algoritmų.
2. Nuorodų į kitus puslapius.
3. *iframe* HTML elementų, kurie gali vaizduoti elektroninei parduotuvei nenaudingą turinį.

Atlikus prekės aprašymo HTML ratifikavimą ir išvalymą nuo kenkėjiško kodo, tiekėjo prekės aprašymas yra išsaugomas el. parduotuvės informacinėje sistemoje.

Baigus ciklą yra grąžinamas atnaujintų prekių kiekis, kuris gali būti naudojamas statistikos kaupimui.

12 lentelė. *Metodų lentelė*

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
HTMLPurifier->purify(\$aprašymas)	Metodas naudojantis HTMLPurifier įskiepi, kuris sunormalizuoja HTML kodą.	Tiekėjo prekės aprašymas	Pakoreguotas aprašymas
cleanHTML()	Metodas išvalantis HTML prekės aprašymą nuo galimo kenkėjiško turinio.	Tiekėjo prekės aprašymas	Pakoreguotas aprašymas

2.7.4. Prekės nuotraukų atnaujinimas

Nuotraukų atnaujinimo algoritmas skirtas suvienodinti prekės vizualią informaciją tarpinėje sistemoje ir el. parduotuvės informacinėje sistemoje. Šis algoritmas pateiktas 13 lentelė. 35 pav.

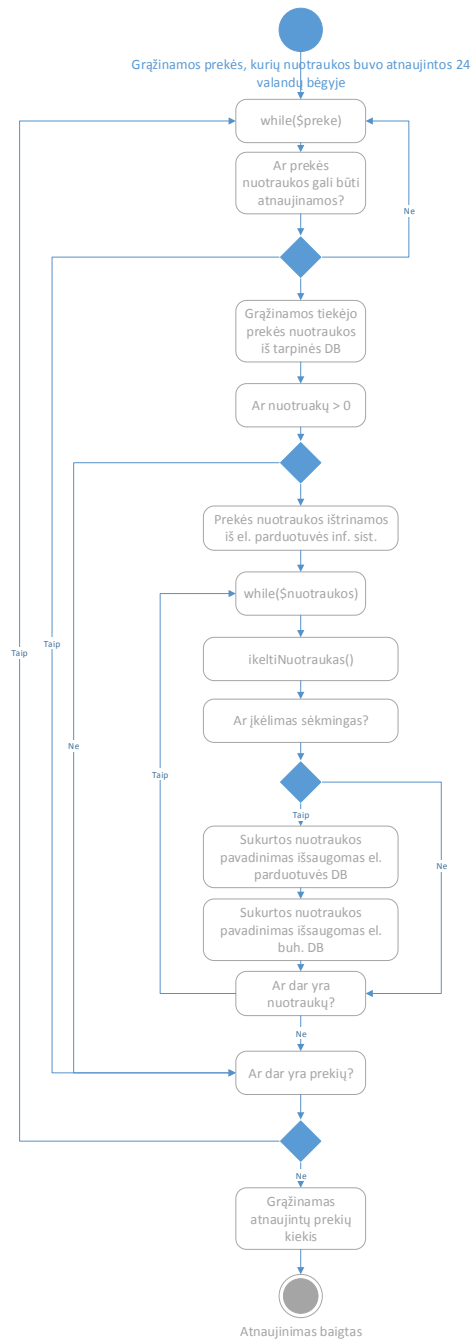
Algoritmas, kaip ir kiti, pradedamas grąžinant atnaujintas prekes iš tarpinės duomenų bazės ir naudojantis ciklu prekę po prekės atnaujinant. Kaip ir prieš tai minėtuose sinchronizavimo algoritmuose, kiekviena prekė tikrinama ar jai nėra nustatytas draudimas atnaujinti tiekėjo nuotraukas.

Jei tokio draudimo nėra, kreipiamasi į tarpinę sistemą ir grąžinamos prekės nuotraukos. Grąžintų nuotraukų kiekis yra patikrinamas, jei kiekis didesnis už 0, nusprendžiama, kad prekės nuotraukas galima įkelti, kitu atveju - pereinama prie kitos prekės.

Kiekviena nuotrauka yra sukuriama paeiliui ciklo pagalba. Įvykdžius metodą ikeltiNuotraukas(), tikrinama ar nebuvo grąžintas klaidos pranešimas, prekės nuotraukos pavadinimas, esantis FTP saugykloje, išsaugomas ir el. parduotuvės informacinės sistemos duomenų bazėje. Platesnis metodo aprašymas pateiktas lentelėje (žr. 13 lentelė.)

13 lentelė. *Metodų aprašų lentelė*

Metodo pavadinimas	Paskirtis	Įvesties parametrai	Išvesties parametrai
ikeltiNuotraukas()	Metodas, nuotraukas sumažinantis/padidinant iki reikiamo dydžio ir išsauganti jas į FTP saugyklą.	Tiekėjo prekės nuotraukos nuoroda	Klaidos pranešimas arba sukurtos FTP saugykloje nuotraukos pavadinimas.



35 pav. Prekės nuotraukų atnaujinimo algoritmas

2.8. Skyriaus išvados

Suprojektavus tarpinę sistemą, buvo išspręstos pagrindinės automatinio duomenų sinchronizavimo problemos elektroninės parduotuvės tarpinėje sistemoje.

Projektuojant sistemą, susipažinta su keliais tiekėjų XML failais. Įvairūs jų formatai leido padaryti išvadas, kokios struktūros tarpinės sistemos duomenų bazės reikia. Diskutuojant su užsakovu, nuspręsta iki kokių apimčių įmanomas prekių profilių plėtimas, dėl šios priežasties buvo išnuomotas pakankamai galingas serveris, kuris galėtų greitai apdoroti didelius duomenų kiekius.

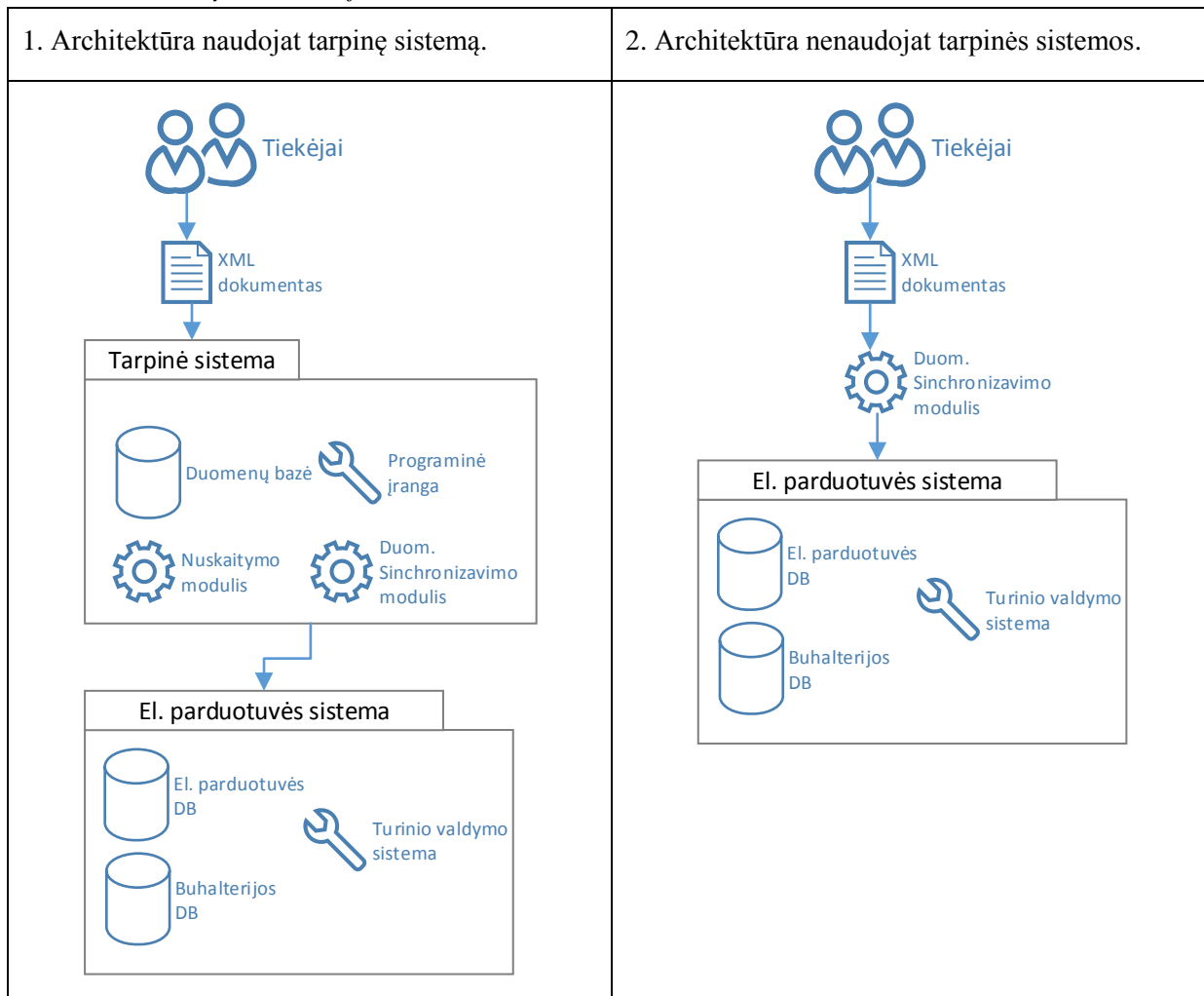
Projektuojant duomenų sinchronizavimo modulį, pastebėta, jog vieningo algoritmo visam prekės profilio sinchronizavimui padaryti negalima, todėl ši modulis išskirstytas į kelis smulkesnius algoritmus, ko pasėkoje turėtų būti sutaupoma laiko ir serverio resursų.

PREKIŲ APRAŠŲ SINCHRONIZACIJOS MODELIO TYRIMAS

2.9. Tyrimo tikslai ir apimtis

Testavimo metu siekiama iširti tarpinės sistemos el. parduotuvėje funkcionalumą, efektyvumą ir panaudojamumą. Tyrimas atliekamas panaudojant skirtingas duomenų apdorojimo ir mainų architektūras, kurios pateiktos 14 lentelė.

14 lentelė. Tyrimo situacijos



Pirmoje situacijoje bus testuojami tyrimo kriterijai suprojektuotai tarpinei sistemai, kurios struktūra aptarta 2-ame skyriuje.

Antroje situacijoje bus testuojami tyrimo kriterijai be suprojektuotos tarpinės sistemos. Taip bus iširtas aiškus tarpinės sistemos naudingumas elektroninėje parduotuvėje.

2.10. Tyrimo techninė ir programinė aplinka

Testavimas bus atliekamas tame pačiame serveryje, kuriame ir bus įdiegta tarpinė sistema.

El. parduotuvės serverio duomenys:

- Procesorius: 19.20 GHz (8 x 2.40 GHz)
- Atmintis: 24576 MB + 12288 MB SWAP
- Disko vieta: 200 GB RAID (našumas 200 IOPS)
- Pralaidumas: 100 Mbps (užsienio sutankinimas 1:1 iki 1024 GB)

- Operacinė sistema: CentOS Linux 5.3
- Duomenų bazė: MySql 5.4
- PHP 5.3.14

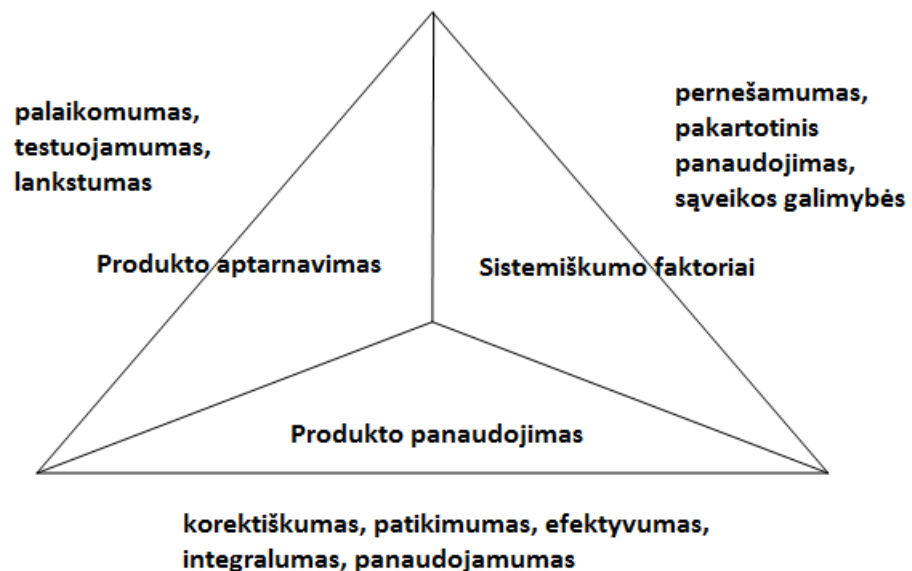
Sistemos apkrovimui stebėti bus naudojami šie įrankiai:

- Serverių nuomos tiekėjo pateikiami serverio apkrautumo statistikos grafikai [20]
- Parduotuvės lankomumo statistikos įrankis Google Analytics [21]

Sistemos algoritmų testavime taip pat bus naudojama PHP programavimo kalba, su ja bus rašomi tyrimo algoritmai skaitinėms ir laikinėms reikšmėms gauti.

2.11. Testavimo kriterijai

Šiai tarpinei sistemai ištestuoti produkto naudingumą buvo pasirinktas James A. McCal kokybės trikampio metodas, kuris pateiktas 36 pav.



36 pav. James A. McCal kokybės trikampis [19]

McCal kokybės trikampis susideda iš trijų pagrindinių kategorijų: produkto panaudojimas, produkto aptarnavimas ir sistemiškumo faktoriai. Šio visos kategorijos savyje turi jas nusakančius kriterijus.

Norint ištirti suprojektuotos sistemos naudingumą, bus remiamasi šiais McCal kokybės trikampio kriterijais [19]:

1. Korektiškumas (sistemos veikimas) – šis kriterijus apima sistemos atitikimą specifikacijai ir užsakovo lūkesčių išpildymą. (savybes ar įsirašo į atitinkamus laukus)
2. Patikimumas (sistemos veikimas) – nusako kaip programinė įranga gali atlikti jai pavestas funkcijas ar neatlikti nepavestų funkcijų.
3. Efektyvumas (sistemos našumas) – apibūdinamas, kaip minimalūs techniniai resursai sistemai padeda funkcionuoti ir kiek jie panaudojami.

4. Integralumas (duomenų vientisumas) – apibūdinama, kaip tiksliai duomenys gali integruotis į sistemą.
5. Lankstumas (duomenų vientisumas) – tai funkcionuojančios sistemos funkcionalumui pakeisti reikalingos pastangos.
6. Panaudojamumas (sistemos veikimas, sistemos našumas) – tai kriterijus, nusakantis programinės įrangos panaudojimo galimybes.

Naudojant šią metodiką, bus tiriami kriterijai, atskleidžiantys sukurtos sistemos savybes. Šios savybės bus suklasifikuotos į tris kategorijas, vienas iš jų tai produkto panaudojimas, kuris įvardijamas kaip produkto veikimas, neįtraukiant integralumo ir efektyvumo kriterijų. Kita kategorija tai sistemos našumas, kurį apibūdina efektyvumo savybė. Ir trečia kategorija tai duomenų vientisumas, kurį apibūdina tokie McCal trikampio kriterijai, kaip lankstumas ir vientisumas. Į tiriamuosius kriterijus yra neįtraukiami sistemiškumo faktoriai, kadangi jie tiria galimybes panaudoti sistemą kitose sistemose, o šio atvejo nėra numatyta tarpinės sistemos projekte.

2.12. Sistemos veikimo tyrimo rezultatai

Kadangi vienas pagrindinių tarpinės sistemos uždavinių yra prekės profilių duomenų apdorojimas ir tikslingas jų patalpinimas į el. parduotuvės sistemą, šiam veikimui iširti buvo pasitelkti šie kriterijai:

Korektiškumas – tai savybė, kuri nusakoma klaidos tikimybe įrašant prekių profilius, kai nenaudojama tarpinė sistema (žr. 14 lentelė. , 2 atvejis).

Patikimumas – patikimumas šioje sistemoje sprendžiamas prekės profilio aspektu, t.y. ar prekės profilio apraše numatyta kategorija leis jai įsirašyti į tinkamą el. parduotuvės sistemos kategorijų medžio šaką.

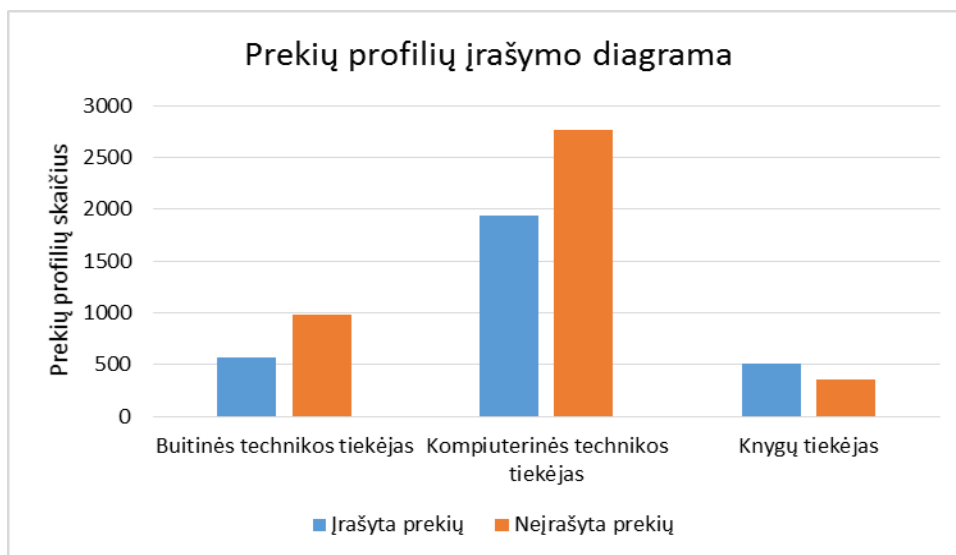
Šio tyrimo tikslas – atskleisti, jog nenaudojant tarpinės sistemos ir prekių kategorijų susiejimui nepanaudojus žmogiškųjų išteklių, išlieka gan didelė tikimybė, kad prekių profiliai nepateks į el. parduotuvės duomenų bazę ar pateks ne į reikiamą prekių grupę.

Tyrimui atlikti buvo pasirinkti 3 tiekėjų XML failai ir, pasitelkiant nuskaitymo modulį, nuskaitomi prekės profiliai. Nuskaityti prekės profiliai pagal pateiktas tiekėjų prekių kategorijas buvo bandomi įrašyti į duomenų bazę (žr. 23 pav.)

Tyrimas pradedamas surenkant prekės informaciją iš tiekėjo XML failo ir patikrinant, ar yra nurodyta prekės grupė. Jei ne - nutraukiami visi veiksmai su preke, kitu atveju - ieškomas grupės atitikmuo. Atitikmens neradus, prekė neįrašoma į el. parduotuvės sistemą, o atitikmenį radus yra įrašoma nauja prekė ir ji yra priskiriama rastai prekių grupei el. parduotuvės sistemoje.

Tyrimas bus atliekamas su trimis tiekėjais, kurie pateikia skirtingas prekių kategorijas:

1. Buitinės technikos tiekėjas
2. Kompiuterinės technikos tiekėjas
3. Knygų tiekėjas



37 pav. Prekių profilių įrašymo diagrama

Tyrimo duomenys kaip prekių profiliai įsirašė ar neįsirašė į el. parduotuvės duomenų bazę pateikti diagramoje (žr. 39 pav.). Visuose tiekėjų XML failuose buvo pateiktos prekių grupės, todėl dalis prekių neįsirašė į duomenų bazę tik dėl prekių grupių neatitikimų. Apžvelgus rezultatus, gauta, jog tikimybė, kad prekė neįsirašys į duomenų bazę yra daugiau nei **54 %**.

Tačiau, priklausomai nuo tiekėjo, šis neįsirašymo procentas yra stipriai įtakojamas pačio tiekėjo. Tai priklauso nuo prekės profilio pateikimo sudėtingumo, kaip pateikta yra prekės profilio kategorija. Dėl to klaidos tikimybė gali svyruoti nuo 41 % iki 64 %.

Dėl tokios didelės klaidos tikimybės verslui tikrai nepalanku turėti tik pusę tiekėjo pateikiamos produkcijos tik tam, kad ji integruotųsi automatiškai.

Duomenų, jog likusių prekių profilių patikimo automatinio įrašymo tikimybė išlieka labai nedidelė, patikslinimui bus atliekamas tiekėjo grupės atitikimo el. parduotuvės grupei tyrimas. Kadangi kai kurios tiekėjų grupės gali neatitikti el. parduotuvės grupės reikšmės.

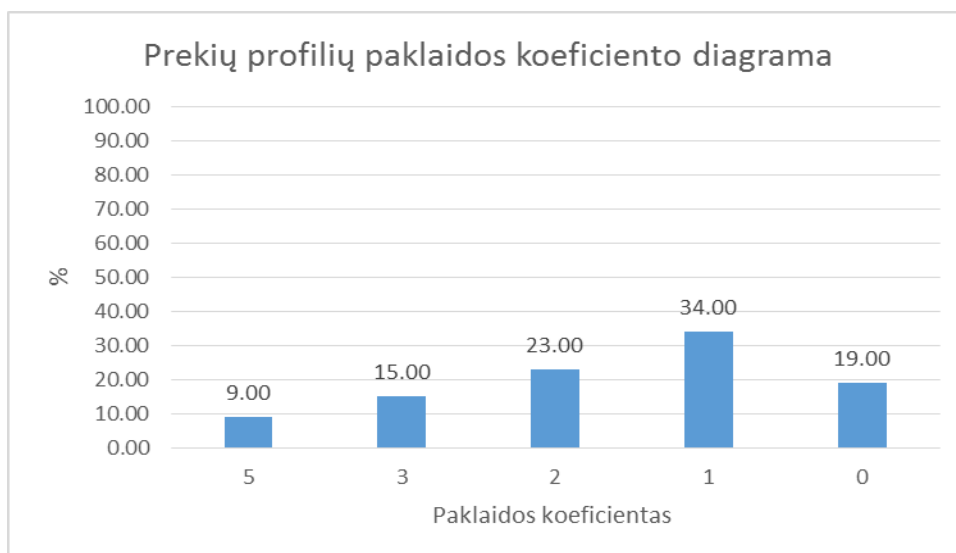
Prekės aprašo įrašymo į kategorijų medį paklaidos koeficientas įvertinamas pagal sukurtą metodiką, kuri pateikiama lentelėje (žr. 15 lentelė.). Iš lentelės matyti, kad įrašoma tiekėjo prekės grupė ir šiai grupei atitinkantis kelias iki reikiamos el. parduotuvės prekių grupės. Prie el. parduotuvės grupės lygio, kuris atitinka reikiamą kategorijos šaką, įrašoma paklaida lygi 0, o už jos einančioms grupėms surašomi paklaidos koeficientai, kurie yra apskaičiuojami – vienas el. parduotuvės grupės medžio lygis plus 1 balas. Jei prekės profilio aprašo grupė skiriasi šaknine kategorija, paklaidos koeficientui skiriami penki balai.

15 lentelė. Paklaidos koeficiento apskaičiavimo lentelė

Prekės pavadinimas	Tiekėjo prekės grupė	El. parduotuvės kelias iki grupės				Paklaidos koeficientas
		1 lygis	2 lygis	3 lygis	4 lygis	
Samsung Galaxy Note 10.1	Kompiuterinė technika	Elektronika	Kompiuterinė technika	Kompiuteriai	Planšetiniai kompiuteriai	
<i>Paklaidos dydis</i>		3	2	1	0	2
Pačiūžos FLOICE dydis 41	Pačiūžos	Laisvalaikio prekės	Sporto prekės	Žiemos sportas	Pačiūžos	
<i>Paklaidos dydis</i>		3	2	1	0	0

Vėdinimo aerodinamika	Santehnika	Knygos ir leidiniai	Santehnika			
		Santehnika				
<i>Paklaidos dydis</i>		5	0			5

Iš lentelės (žr. 15 lentelė.) matoma, kad prekės aprašo kategorija atitinka el. parduotuvės kategorijos medžio šaką, koeficientas lygus 0, tai reiškia, jog prekė įsirašė į reikiamą el. parduotuvės prekių grupę, kitais atvejais paklaida svyruoja nuo 1 iki 5. Kuo paklaidos koeficientas aukštesnis, tuo prekės aprašo įrašymo nuokrypis į el. parduotuvės reikiamą kategoriją yra didesnis. Tokiais atvejais panaudojama tarpinė sistemoje esanti programinė įranga (žr. 27 pav.).



38 pav. Prekių profilių paklaidos koeficiento diagrama

Ištyrus 100 prekių, kurios priklauso 5 tiekėjams, gauti duomenys, kiek procentų prekių surinko didesnę nei 0 paklaidos koeficientą, o kurios prekės įsirašė į el. parduotuvės duomenų bazę korektiškai. Tyrimo rezultatų diagrama pateikta 40 pav. Iš jos matoma, jog tik 19 % prekių pateko į reikiamas kategorijos šakas, 72% prekių pateko į netinkamas kategorijos medžio šakas, bet išliko reikiamoje šakninėje kategorijoje, o 9% prekių atsidūrė net kitoje šakninės kategorijos šakoje.

Apibendrinus korektiškumo ir patikimumo tyrimų rezultatus galima teigti, jog tikimybė, kad prekė įsirašys į duomenų bazę, vidutiniškai yra 46%, bet iš šios tikimybės dar galima išskirti patikimumo kriterijų, kuris pagal gautuosius duomenimis yra tik 19%, todėl dalis prekių gali įsirašyti netinkamoje kategorijų šakoje, dėl to klientui sudėtingėja prekės paieška.

Aprašyti problemai spręsti tarpinėje sistemoje yra sukurta programinė įranga (žr. 2.4 skyrių), kuri skirta prekės grupei susieti su el. parduotuvėje taikomomis prekių grupėmis, ar išskaidyti jas į tiksles kelias prekių grupes. Pavyzdžiui, tiekėjas gali savo sistemoje mobiliuosius telefonus ir jų priedus saugoti toje pačioje prekių grupėje, o el. parduotuvės sistemoje prekių aprašai gali būti išskaidyti į dvi kategorijas (mobilieji telefonai, mobiliųjų telefonų priedai).

2.13. Sistemos našumo tyrimo rezultatai

Ištirti sistemos našumui bus atliekami du tyrimai:

1. Užklausų skaičius – kiek tarpinė sistema sugeneruoja užklausų į duomenų bazę atliekant prekių aprašų atranką, atnaujinimą, įrašymą ir šalinimą .
2. Sistemos apkrautumas – sistemos savybė, kuria nusakoma, kiek resursų naudoja sistema atskirais atvejais.

Įvertinant užklausų skaičių lyginami keli algoritmai, kurie kreipiasi į duomenų bazę. Skaičiuojama kiek kartų į duomenų bazę kreipiasi tarpinė sistema norit atlikti algoritmą, ir kiek kartų į duomenų bazę kreipiasi metodas be tarpinės duomenų bazės. Tokiu būdu bus galima procentine išraiška įvertinti kaip skiriasi suprojektuotas sistemos duomenų bazės apkrautumas su tiesioginio atnaujinimo algoritmu.

Šiame tyrime naudojamas prekės profilis XML pavidale, kuris pateiktas 2 priede.

Algoritmai, kurie pritaikomi įvertinant užklausų skaičių:

1. Prekės įrašymas į el. parduotuvės informacinę sistemą – tyrimas vyks įrašant prekę naudojantis tarpine sistema, vėliau išvalius duomenis ir pakartojus įrašymą be tarpinės DB.
2. Prekės kainos ir likučio atnaujinimas – tyrimas vyks pakeitus XML faile esančias kainas.
3. Prekės aprašymo atnaujinimas – tyrimas vyks atlikus korekcijas prekės aprašyme.
4. Prekės nuotraukų atnaujinimas – nuotraukų atnaujinimas vyks, abi XML faile pateiktas nuotraukas pakeitus visiškai kitomis, ir pakeitus nuorodas į jas.
5. Prekės charakteristikų atnaujinimas – charakteristikų atnaujinimas bus tiriamas įrašyti prekei pakeitus dvi iš 5 charakteristikų reikšmių ir 3 iš jų ištrynus.

Visų pirma sukuriamas prekės profilis, kuris pateikiamas 2 priede. Po to, atitinkamai keičiant duomenis XML faile, vykdomas bent vienas iš minėtų algoritmų. Kiekvienas iš šių algoritmų sugeneruoja tam tikrą užklausų kiekį prekės profiliui atnaujinti ar įrašyti. Grįžimui į pradinę situaciją ir bandymui vėl įrašyti prekę be tarpinės sistemos, el. parduotuvėje pašalinami visi šios prekės duomenys. Užklausų skaičius į duomenų bazę skaičiuojamas taip: vienas kreipimasis į vieną duomenų bazę lygus 1.

Tyrimo rezultatai peikiami lentelėje (žr. 16 lentelė.)

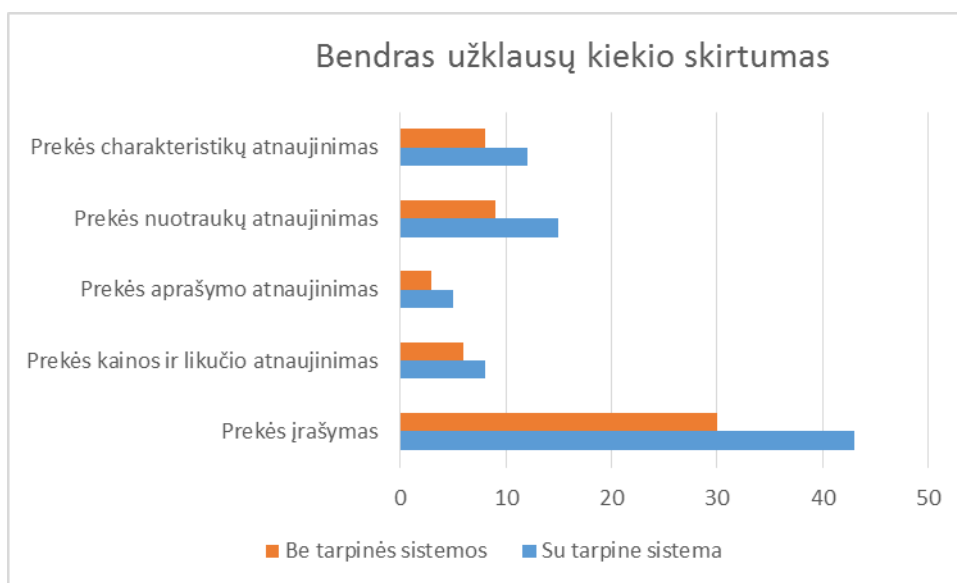
16 lentelė. Užklausų skaičiaus tyrimo duomenys

Tiriamas veiksmas	Užklausos tipas	Užklausų skaičius duomenų bazėje, naudojant tarpinę duomenų bazę			Užklausų skaičius duomenų bazėje, nenaudojant tarpinės duomenų bazės	
		Tarpinė Sistema	Produkcijos sistema		Produkcijos sistema	
			El.parduotuvės DB	Buhalterijos DB	El.parduotuvės DB	Buhalterijos DB
Prekės įrašymas	Atranka	2	0	0	0	2
	Įrašymas	23	11	7	21	7
	Atnaujinimas	0	0	0	0	0
	Šalinimas	0	0	0	0	0
Prekės kainos ir likučio atnaujinimas	Atranka	2	1	0	2	0
	Įrašymas	0	0	0	0	0
	Atnaujinimas	1	2	2	2	2
	Šalinimas	0	0	0	0	0
Prekės aprašymo	Atranka	2	0	0	1	0
	Įrašymas	0	0	0	0	0

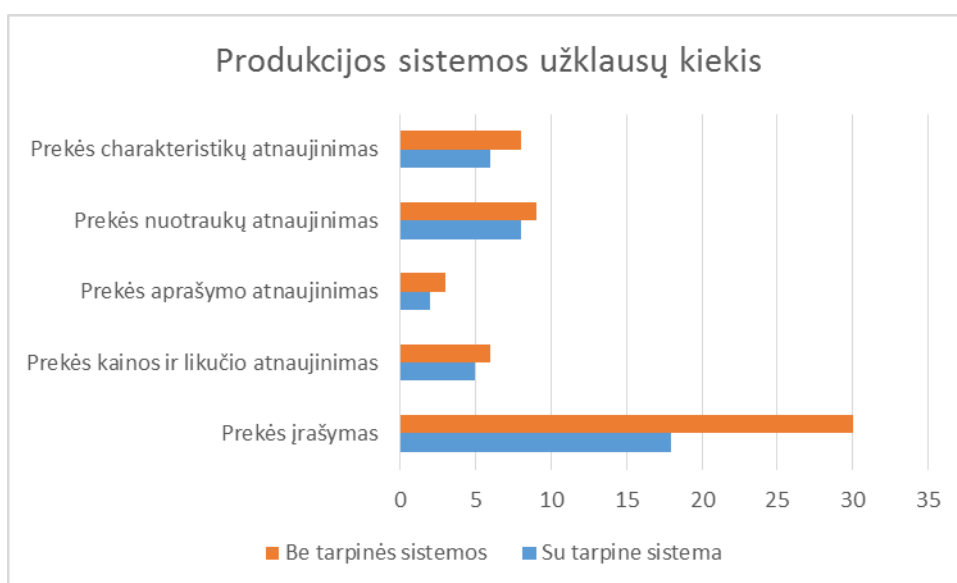
atnaujini mas	Atnaujinimas	1	1	1	1	1
	Šalinimas	0	0	0	0	0
Prekės nuotraukų atnaujini mas	Atranka	2	0	0	1	0
	Įrašymas	2	2	2	2	2
	Atnaujinimas	1	0	0	0	0
	Šalinimas	2	2	2	2	2
Prekės charakteri stikų atnaujini mas	Atranka	1	0	0	3	0
	Įrašymas	0	0	0	0	0
	Atnaujinimas	2	3	0	2	0
	Šalinimas	3	3	0	3	0

Lentelėje (žr. 16 lentelė.) pateikiami užklausų kiekiai naudojant ir nenaudojant tarpinės sistemos (žr. 14 lentelė.), kuriuose fiksuojamas užklausų skaičius į jų duomenų bazes.

Gauti užklausų kiekiai sumuojami ir apskaičiuojama procentinė dalis nuo algoritmo ir jame vykdomų veiksmų (žr. 16 lentelė.).



39 pav. Bendras užklausų kiekis susumavus ir tarpinę sistemą



40 pav. Produkcijos sistemos užklausų kiekis

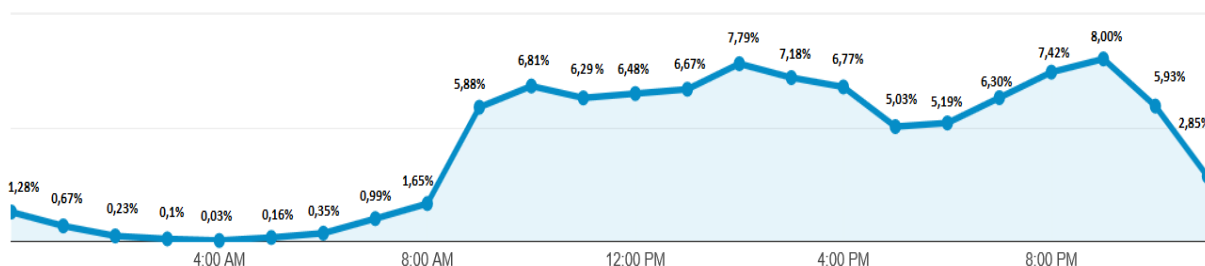
Atlikus tyrimą, iš gautų duomenų (žr. 39 pav. ir 40 pav.) matoma, jog bendras užklausų kiekis naudojant sistemą su tarpine sistema padidėja. Apskaičiavus, kiekvienai iš funkcijų, algoritmas su tarpine duomenų baze, vidutiniškai generuoja apie 33.71% daugiau užklausų. Dauguma tų užklausų yra labai panašios į tas, kurios atliekamos be tarpinės duomenų bazės, taigi, galima teigti, jog bendrai ši sistema reikalauja daugiau resursų. Bet patikslinus duomenis ir palyginus produkcijos sistemos skaičius, apskaičiuota, jog į produkcijos sistemą kreipiamasi vidutiniškai 36.5 % mažiau, taip el. parduotuvės duomenų bazės tampa mažiau apkrautos ir gali sklandžiau veikti.

Diagramose (žr. 39 pav. ir 40 pav.) matomas skirtumas, kiek užklausų atliekant veiksmus su preke sugeneruojama susumavus bendrą užklausų kiekį ir kiek užklausų sugeneruota tik į produkcijos sistemą.

Produkcijos sistemos užklausų kiekis sumažėja naudojant tarpinę sistemą, tai matoma 40 pav. Kiekvienoje iš funkcijų užklausų kiekis sumažėja dėl šių pagrindinių aspektų:

- Patikrinimas, ar prekė yra duomenų bazėje, vyksta tarpinėje sistemoje, ne el. parduotuvėje.
- Prekės nuotraukas tarpinėje sistemoje galima sulyginti su tiekėjo duomenimis. Dėl to jos gali atsinaujinti arba gali nekisti. Produkcijos sistemos nuotraukų pavadinimai skiriasi nuo tiekėjų, dėl šios priežasties, nenaudojant tarpinės sistemos, norit nuotraukas išlaikyti atnaujintas, kas naktį būtina ištrinti esamas nuotraukas ir iš naujo įkelti esamas tiekėjo prekės nuotraukas, dėl šios priežasties gaunamas perteklinis duomenų naudojimas.

Apkrautos sistemos apibrėžimo įvedimui būtina išsiaiškinti, kada sistemoje yra didžiausias lankomumas, bei kada sistema naudoja daugiausiai resursų.



41 pav. El. parduotuvės lankomumo diagrama

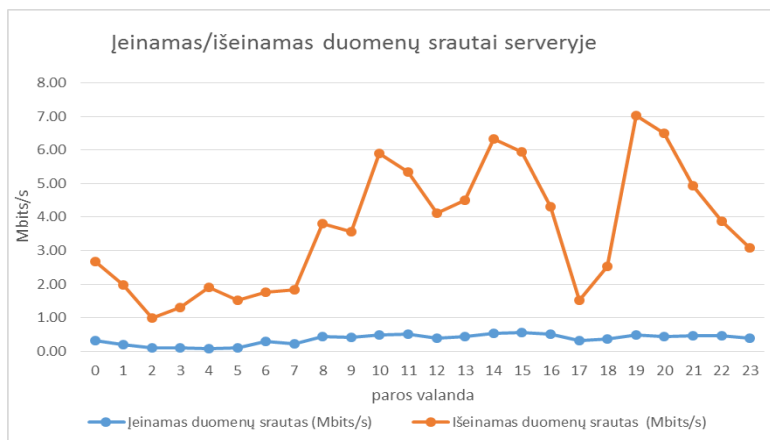
Iš lankomumo statistikos sistemos Google Analytics paimta vidutinė mėnesio paros lankomumo procentinė diagrama (žr. 41 pav.). Procentai rodo, kiek vidutiniškai procentų paros lankomumo tą valandą susilaukė el. parduotuvė. Iš šios diagramos (žr. 41 pav.) matoma, jog didžiausio lankytojų srauto el. parduotuvė sulaukia nuo 9 valandos ryto iki 10 valandos vakaro.

Tolimesniam tyrimui yra įvedami du kriterijai:

1. Nakties metas (ne piko valandos) – savybė, kai paros laikas yra nuo 10 val. vakaro iki 9 val. ryto. Šiuo atveju el. parduotuvės lankomumas yra mažesnis nei 5,88%.

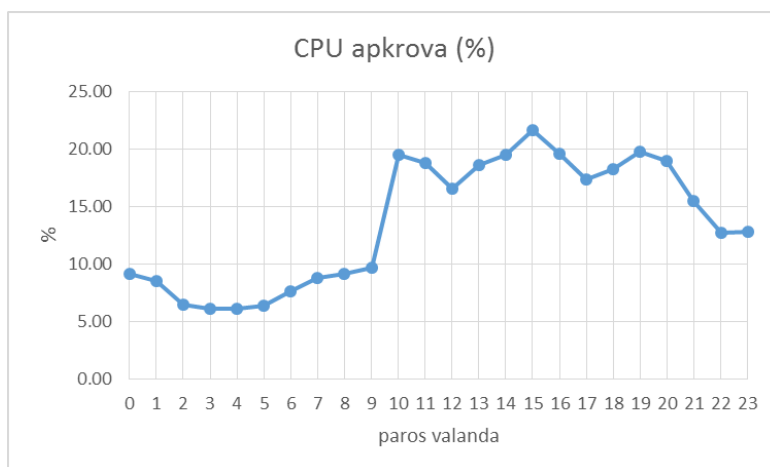
2. Dienos metas (piko valandos) – savybė, kai paros laikas yra nuo 9 val. ryto iki 10 val. vakaro. Šiuo atveju el. parduotuvės lankomumas yra didesnis nei 5,88%.

Tiriant serverio naudojamų resursų duomenis, buvo susumuoti savaitės serverio duomenų rezultatai ir vidurkiai pagal paros valandas suvesti į diagramas (žr. 42 pav. 43 pav. 44 pav.)



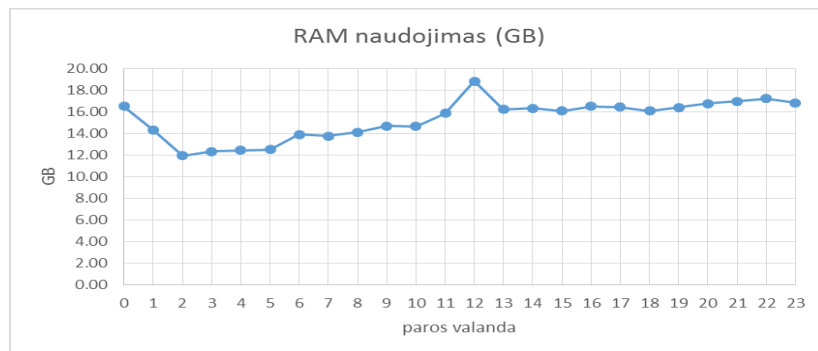
42 pav. Įeinamų/išeinamų duomenų srautų diagrama

Serveryje bendras leistinas srauto pralaidumas yra 100 Mbits/s. Išnaudojamo duomenų srauto serveryje diagrama yra pateikta 42 pav. Iš jos matoma, kad nakties metu, kai el. parduotuvė yra mažiausiai lankoma klientų (pagal 41 pav.), šis srautas yra išnaudojamas nuo 1,33% iki 2,98%. Piko valandomis, kai el. parduotuvė lankoma daugiau (pagal 41 pav.), duomenų srauto išnaudojimas pakyla iki 8%.



43 pav. Procesoriaus apkrovos diagrama

Išnagrinėjus CPU serverio apkrovą, taip pat pastebima, jog nakties (ne piko) metu, serverio CPU apkrova mažesnė ir svyruoja nuo 9.64% iki 12,73%, o dienos (piko) metu, serverio CPU apkrautumas yra nuo 15,47% iki 19,48%.



44 pav. RAM naudojimo serveryje diagrama

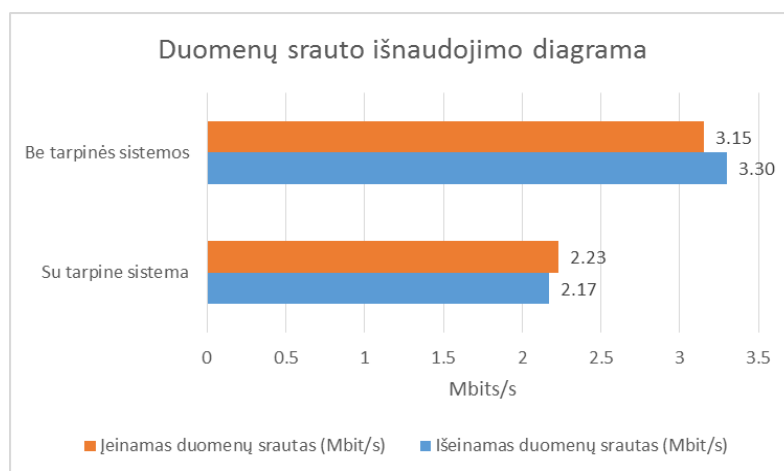
Serveryje naudojamos RAM atminties diagrama paros bėgyje pateikiama 44 pav. Viso serveryje RAM atminties yra 24 GB. Svyravimai tarp piko ir ne piko valandų yra pastebimi. Nakties metu, kai el. parduotuvė lankoma mažiau (žr. 41 pav.), serverio RAM atminties naudojimas svyruoja nuo 49.87% iki 59.71%, tuo tarpu piko metu RAM naudojimas svyruoja nuo 59.71% iki 78.42%.

Išnagrinėjus el. parduotuvės lankomumą ir serverio apkrautumo statistiką, nuspręsta, jog tarpinės sistemos atnaujinimo ir duomenų sinchronizavimo moduliai turėtų veikti ne piko valandomis, nuo 10 valandos vakaro iki 6 valandos ryto, tam, kad duomenų pokytis būtų mažiausiai pastebimas lankytoji, ir veikiantys algoritmai netūrėtų įtakos sistemos veikimui.

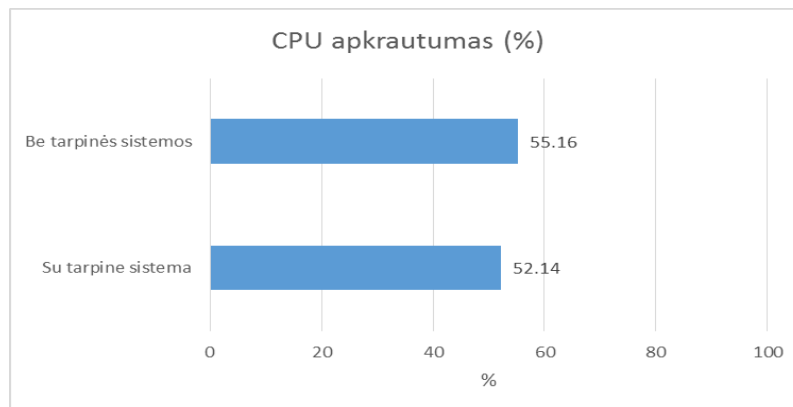
Toliau bus tiriama, kaip el. parduotuvės serveris naudoja savo resursus naudojant tarpinės sistemos modelį ir jo nenaudojant. Tyrimo metu bus bandoma į el. parduotuvės duomenų bazę įrašyti 100-to prekių profilių paketą iš tarpinės sistemos ir palyginti serverio duomenis kai 100 prekių profilių bus įrašoma nuskaitant XML failą ir šias prekes įrašant tiesiai į el. parduotuvės sistemą. Kadangi buvo nuspręsta, kad atnaujinimas turi vykti ne piko valandomis, todėl abu atvejai bus tiriami ne piko valandomis.

Žemiau pateiktos diagramos (žr. 45 pav. 46 pav. 47 pav.) iliustruoja tyrimo rezultatus, kai buvo bandoma importuoti 100-tą prekių naudojant tarpinę sistemą ir jos nenaudojant. Atlikus tyrimą buvo fiksuojamas ir šio importo vykdymo laikas:

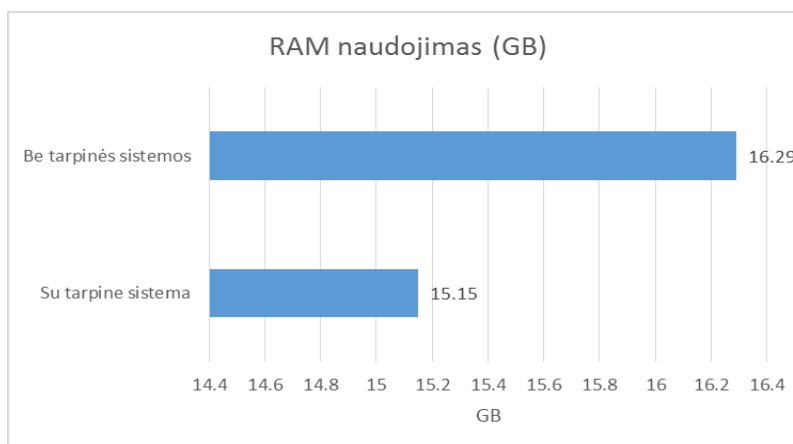
- Su tarpine sistema vykdymo laikas buvo 5 min. 34 s.
- Be tarpinės sistemos vykdymo laikas siekė 9 min. 13 s.



45 pav. Duomenų srauto tyrimo metu diagrama



46 pav. CPU apkrautumo tyrimo metu diagrama



47 pav. RAM naudojimo tyrimo metu diagrama

Išnagrinėjus serverio apkrovimo diagramas (žr. 45 pav. 46 pav. 47 pav.), matoma, jog naudojant tarpinę sistemą ar jos nenaudojant, serverio resursų naudojimas gerokai padidėja.

Atliekant tyrimą ne piko valandomis su tarpine sistema serverio resursai kito taip:

- Įeinamų duomenų srautas padidėjo nuo 0.21 Mbits/s iki 2.23 Mbits/s.
- CPU apkrova padidėjo nuo 7.8% iki 52.14%.
- RAM naudojimas išaugo nuo 13.65 GB iki 15.15 GB.

Atliekant tyrimą ne piko valandomis be tarpinės sistemos serverio resursai kito taip:

- Įeinamų duomenų srautas padidėjo nuo 0.21 Mbits/s iki 3.15 Mbits/s.
- CPU apkrova padidėjo nuo 7.8% iki 55.16%.
- RAM naudojimas išaugo nuo 13.65 GB iki 16.29 GB.

Apžvelgus serverio naudojimo resursus su ir be tarpinės sistemos, matoma, jog atliekant duomenų importavimą su tarpine sistema apkrautumas yra vidutiniškai 13.89% mažesnis nei be tarpinės sistemos. Taip yra todėl, jog duomenų importas su tarpine sistema vyko iš duomenų bazės. Be tarpinės sistemos duomenų importavimas vyko tiesiai iš XML failo, dėl to algoritmas naudojo papildomai daugiau resursų.

Iš CPU apkrautumo diagramos matoma, jog CPU ir vienu ir kitu atveju apkraunamas gan daug, bet lyginant kiek laiko vyko algoritmai su tarpine sistema, serverio apkrautumas trunka 60,4% trumpiau nei algoritmas be tarpinės sistemos.

Apibendrinant galima teigti, jog prekių profilių atnaujinimas turi vykti nakties (ne piko) metu, kadangi el. parduotuvės lankomumas yra mažesnis nei 5.88%, todėl informacijos pokyčius klientai pastebės mažiau. Taip pat iš serverio rodmenų matoma, jog algoritmas nemažai apkrauna serverį, todėl, apsitraudžiant nuo galimų sistemos sutrikimų, atnaujinimo ir sinchronizavimo, moduliai turi vykti tada, kai serveris apkrautas mažiau.

2.14. Duomenų vientisumo tyrimo rezultatai

Sistemos duomenų vientisumo tyrimui buvo pasitelkti šie kriterijai:

Lankstumas – tai kaip tarpinė sistema pritaikyta lanksčiai sukurti/šalinti naujus prekės profilio duomenis.

Integralumas – šis kriterijus remiasi prekės profilio pilnos informacijos įrašymu į duomenų bazę, tai yra savybės reikšmių prieš ir po importavimo sutapimas.

Šiame tyrime suderinti du kriterijai, tai yra sistemos lankstumas ir duomenų integralumas. Tyrimas bus atliekamas pasinaudojus 2 priede pateiktu XML prekės profiliu. Šis prekės profilis bus įrašomas į tarpinę sistemą naudojantis XML nuskaitymo moduliui. Vėliau, keičiant šio XML profilio duomenis ir stebint pokyčius duomenų bazėje, bus fiksuojami integruotų duomenų rezultatai. Tam, kad būtų įvertintas profilio savybių lankstumas, įvesties duomenų kiekis turi būti liginamas su išvesties duomenų kiekiu.

Tyrimo duomenys pateikti 17 lentelė. yra apibūdinami taip:

1. Nuskaitymo modulis įrašo prekę pateiktą 2 priede.
2. Iš tos pačios prekės profilio XML faile pašalinama viena savybė ir pakoreguojamos dvi savybių reikšmės.
3. Įvestoje prekėje pridedamos trys savybės, kurios jau yra tarpinėje duomenų bazėje.
4. Įvestoje prekėje pridedama neegzistuojanti savybė.
5. Pakeičiamas prekės kodas ir paliekamos 4 prekės savybės.

17 lentelė. Tyrimo įvesties savybės

Tyrimo atvejis	Įvesties savybės					
	Prekių skaičius		Savybių skaičius			
	Esamų (I ₁)	Naujų (I ₂)	Nuskaityta savybių	Atnaujinamų (I ₃)	Naujų Egzistuojančių (I _{4,1})	Neegzistuojančių (I _{4,2})
1.	0	1	5	0	0	5
2.	1	0	4	2	0	0
3.	1	0	7	0	3	0
4.	1	0	8	0	0	1
5.	0	1	4	0	2	2

18 lentelė. Tyrimo išvesties duomenys

Tyrimo atvejis	Rezultatas po duomenų importavimo					
	Prekių skaičius		Savybių skaičius			Pašalinta savybės reikšmė
	Atnaujintų (O ₁)	Sukurtų (O ₂)	Atnaujinta reikšmė (O ₃)	Priskirta nauja reikšmė (O ₄)	Sukurta nauja savybė (O ₅)	
1.	0	1	0	5	5	0
2.	1	0	2	0	0	1
3.	1	0	0	3	0	0
4.	1	0	0	1	1	0
5.	0	1	0	4	2	0

Iš tyrimo rezultatų matoma, jog duomenų bazė prisitaiko prie naujų prekių profilių duomenų, jei prekės profilis dar neegzistuoja sistemoje, jis integruojasi su visomis naujomis savybėmis į tarpinę duomenų bazę. Jei prekės profilis jau egzistuoja, jo nebeįrašo, bet patikrina visas galimas prekės savybes. Matoma, kad prekės profilis duomenų bazėje visiškai lankstus pokyčiams.

Tyrimo duomenys (žr. 17 lentelė. ir 18 lentelė.) yra naudingi išsiaiškinimui ar duomenų vientisumas, t.y. integralumo kriterijus, yra tinkamas. Šis kriterijus bus išskaidytas į du duomenis:

1. Duomenų kiekio vientisumas.
2. Duomenų reikšmių vientisumas.

Duomenų kiekio vientisumas bus apskaičiuojamas gaunant procentinių santykių vidurkį tarp įvedamų prekės savybių kiekio ir atitinkamų šių savybių veiksmų kiekio.

Duomenų reikšmių vientisumas bus apskaičiuojamas gaunant procentinių santykių vidurkį, tarp nuskaitytų savybių kiekio ir atitinkančių duomenų kiekio.

Matematinė formulės išraiška (1):

$$K = \frac{\sum_{i=1}^5 O_i}{\sum_{i=1}^5 I_i} \cdot 100, \text{ kur } I_i > 0 \quad (1)$$

$$I_5 = I_4 = I_{4,1} + I_{4,2}$$

19 lentelė. Integralumo tyrimo rezultatai

Testinis atvejis	Duomenų kiekio vientisumas	Duomenų reikšmių vientisumas
1.	100%	100%
2.	100%	100%
3.	100%	100%
4.	100%	100%
5.	100%	100%

Iš 19 lentelė. tyrimo rezultatų matoma, jog visų tyrimo atvejų duomenų vientisumo vidurkis yra 100%, todėl, galima daryti išvadą, jog sistema veikia tinkamai ir visos prekių profilių savybės integruojasi į sistemą reikiama tvarka.

2.15. Panaudojamumo tyrimas

Panaudojamumas – sistemos aspektu apibrėžiama kaip savybė leidžianti mažesnę prekės aprašo nepilnumo tikimybę, kai nenaudojama šiame darbe siūloma programinė įranga.

Tarpinė sistema kaip vieną iš komponentų turi programinę įrangą, kuri skirta el. parduotuvės vadybininkui. Naudojantis šia programine įranga, vadybininkas gali atlikti tokius veiksmus:

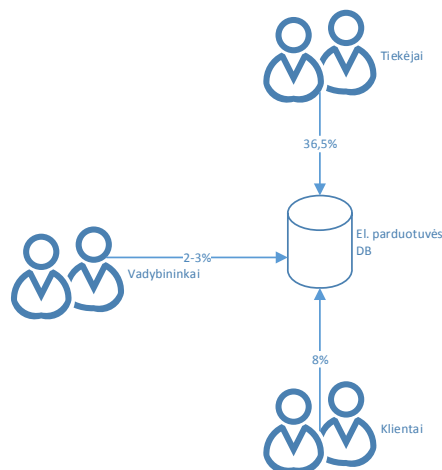
1. Susieti tiekėjo prekių grupes ar pačias prekes su el. parduotuvės prekių grupėmis.
2. Susieti tiekėjo prekes su el. parduotuvės prekėmis.
3. Administruoti, kurią tiekėjo prekės informaciją atnaujinti.
4. Administruoti valiutų kursus, pagal kuriuos apskaičiuojamos prekės kainos užsienio tiekėjams.

Programinės įrangos naudingumas 1 ir 2 punktui, jog prekės įsirašo patikimiau naudojant tarpinę sistemą ir jos programinę įrangą, jau įrodytas 2.12 skyriuje.

Programinės įrangos funkcija - administruoti, kurią prekės informaciją atnaujinti reikalinga tokiais atvejais, kai:

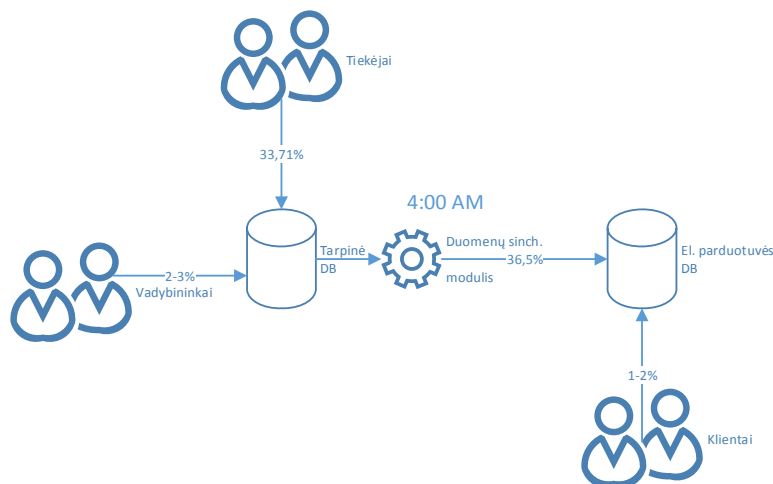
- Vienoje el. parduotuvės prekėje yra susieta kelių skirtingų tiekėjų prekės – kai vieną el. parduotuvės prekę atitinka keli skirtingi tiekėjai, ji tampa susieta su keliais prekių profiliais, taip neparinkus kurio tiekėjo ir kuri prekės informacija turi būti pagrindinė, duomenys gali pradėti maišytis, vieną parą rodyti vieną aprašymą, kitą parą - kitą, tas pats gali atsitikti su nuotraukomis ar charakteristikomis.
- Tiekėjas pateikia produktų nuotraukas su vandens ženklais – kai kurie tiekėjai gali pateikti savo prekes su jų įmonės vandens ženklu, tokiu būdu pirmą kartą įkėlus prekes vadybininkas turi susitvarkyti nuotraukas, kad jos nebūtų su šiais vandens ženklais, po nuotraukų sutvarkymo yra svarbu, jog jos kitą kartą neatsinaujintu ir vėl neapkrautų darbu vadybininko.
- Tiekėjas pateikia prekės profilio aprašymą ir kitą tekstinę informaciją ne lietuvių kalbą – panaši situacija kaip su vandens ženklais yra ir ši, kai kurie tiekėjai yra iš užsienio ir prekių profilių aprašus pateikia savo gimtąja kalba (anglų, lenkų, čekų, estų ir kt.), šioje situacijoje taip pat pirmą kartą prekės yra sukliamos automatinio būdu, bet vadybininkas turi susitvarkyti užsienio kalba pateiktą tekstinę informaciją, po pakeitimo ši informacija nebegali kisti, todėl reikalinga išjungti tekstinės informacijos atnaujinimą.

Paskutinė programinės įrangos funkcija – užsienio valiutų administravimas. Duomenų bazėje yra saugomi užsienio tiekėjų valiutų kursai lyginant su litu, taip kaskart nuskaitymo modulis tiekėjo valiutos kursą gali gražintis iš duomenų bazės ir automatiškai konvertuoti tiekėjo valiutą į litus.



48 pav. El. parduotuvės apkrautumas nenaudojant tarpinės sistemos

Nenaudojant tarpinės sistemos, el. parduotuvės duomenų bazės apkrautumas pateiktas 48 pav. Šiame pavyzdyje pateikta, jog tiekėjų atnaujinimas apkrauna duomenų bazę iki 36.5% pagal tyrimo duomenis pateiktus 3.5 skyriuje. Taip pat skyriuje 2.12 pateikiami duomenys, jog, nenaudojant tarpinės sistemos duomenų importavimo, patikimumas siekia vos 19%, dėl to vadybininkas darbo metu, kai išeinamų duomenų srautas siekia iki 8% (žr. 45 pav.), turi koreguoti prekės profilio įrašymo vietą el. parduotuvės prekių grupių medyje.



49 pav. El. parduotuvės apkrautumas naudojant tarpinę sistemą

Naudojant tarpinę sistemą duomenų apkrautumas tenka tarpinės sistemos duomenų bazei. Vadybininkui dienos metu nereikia koreguoti prekės informacijos el. parduotuvėje, visą darbą jis atlieka su tarpinės sistemos duomenų baze, taip užtikrinama prekės profilio vieta el. parduotuvės prekių grupių medyje. Prekės atnaujinamos tik 4 valandą ryto, kai klientų išeinamų duomenų srautas siekia vos 1-2% (žr. 49 pav.), taip el. parduotuvės duomenų bazę apkraunama mažiau.

2.16. Skyriaus išvados

Ištyrus sistemos veikimą – korektiškumą ir patikimumą, gautas rezultatas, jog prekės profiliai į duomenų bazę be tarpinės sistemos gali įsirašyti tik 46% tikimybe (korektiškumas), o šių integruotų prekių profilių tinkamo įsirašymo tikimybė į el. parduotuvės kategorijų medį siekia vos 19% (patikimumas). Dėl šios priežasties tarpinė sistema yra pranašesnė, kadangi prieš prekių profiliais

įsirašant į el. parduotuvės duomenų bazę, vadybininkas susieja reikiamas tiekėjo prekes su reikiamomis el. parduotuvės grupėmis ir tikimybė, kad prekė įsirašys tinkamai, sieks iki 99% (nes visada išlieka žmogiškasis klaidos faktorius).

Sistemos našumo tyrimas įvertintas užklausų kiekiu ir serverio apkrautumu. Apibendrinus rezultatus, tarpinė sistema generuoja 33.71% daugiau užklausų į duomenų bases, bet elektroninės parduotuvės duomenų bazė su tarpine sistema apkraunama 36.5% mažiau. Taip pat iš serverio apkrautumo duomenų pastebėta, jog prekių įrašymas su tarpine sistema serverį apkrauna vidutiniškai 13,89% mažiau. Iš šių rezultatų galima teigti, jog tarpinė sistema efektyviau išnaudoja ir duomenų bazės ir serverio resursus.

Apibendrinus duomenų vientisumo rezultatus, gauta, jog tarpinės sistemos duomenų bazė yra lanksti, kadangi tyrimo duomenyse matoma, kad išvesties duomenų kiekis sutampa su išvesties duomenų kiekiu. Taip pat sistemos integralumas įvertintas 100%, kadangi tyrimo metu tikrintos įvesties reikšmės sutapo su duomenų bazėje išsaugotomis reikšmėmis. Dėl to galima teigti, jog sistemos duomenų vientisumas tinkamas ir sistema veikia tinkamai.

DARBO IŠVADOS

1. Išanalizavus elektroninės parduotuvės struktūrą ir jos elementus, matoma, jog pagrindinis el. verslo elementas yra produkto profilis. Jo išsamumas ir pilnumas leidžia klientui pateikti detalią produkto informaciją. Bet, norit turėti didelių prekių profilių asortimentą ir nenorint didinti žmogiškųjų išteklių, būtina automatizuoti produktų profilių informacijos sinchronizavimą su produktų tiekėjų sistema.

2. Išanalizavus galimas prekių profilių sinchronizavimo problemas el. parduotuvės sistemoje, buvo pasiūlytas ir suprojektuotas prekių profilių sinchronizavimo metodas naudojant tarpinę sistemą. Projektuojant pastebėta, jog norit sukurti XML nuskaitymo algoritmą, kuris sinchronizuotu prekių profilių informaciją be programuotojo įsikišimo, el. parduotuvė turėtų įvesti savo XML struktūros standartus, kitu atveju - norint prisitaikyti prie bet kokio tiekėjo teikiamo XML failo, būtina kurti atskirus nuskaitymo modulius šiems failams.

3. Tyrimo metu suprojektuotas, prekių profilių sinchronizavimo metodas naudojantis tarpine sistema, buvo lyginamas su metodu prekių profilių informaciją sinchronizuojančiu be tarpinės sistemos. Šio tyrimo metu buvo pastebėta, jog naudojant tarpinę sistemą serverio ištekliai yra paskirstomi laiko tarpais, kai sistema yra neapkrauta, tokiu būdu išvengiama problemų el. parduotuvės serveryje bei galimų prekių profilių duomenų vientisumo sutrikimų. Todėl, buvo padaryta išvada, jog prekių profilių sinchronizavimas naudojant tarpinę sistemą yra tikslingesnis nei naudojant tiesioginį algoritmą.

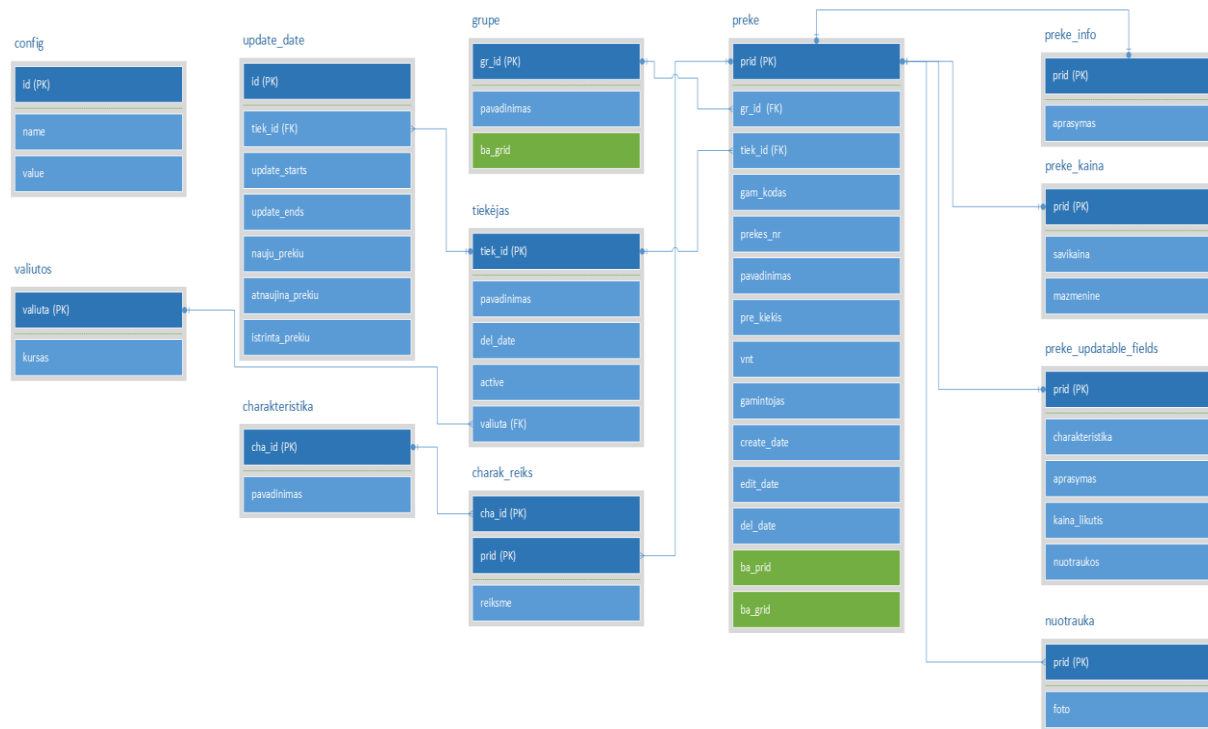
LITERATŪRA

1. Qiming Chen, Umesh Dayal, Meichun Hsu ir Martin Griss, *Dynamic-Agents, Workflow and XML for E-Commerce Automation* [interaktyvus] – [žiūrėta 2012-12-27]. Prieiga per internetą: <http://martin.griss.com/pubs/ECWEB_final2000.pdf>
2. *Extensible Markup Language (XML)* [žiūrėta 2012-12-27]. Prieiga per internetą: <<http://www.w3.org/XML/>>
3. Joe Tekli, Richard Chbeir ir Kokou Yetongnon, *An overview on XML similarity: background, current trends and future directions*, 2009 m. [interaktyvus] – [žiūrėta 2013-12-27]. Prieiga per internetą: <<http://disi.unitn.it/~p2p/RelatedWork/Matching/2313.pdf>>
4. *Chapter 4 SOAP: Simple Object Access Protocol*, 2008 m. [interaktyvus] – [žiūrėta 2012-12-27]. Prieiga per internetą: <http://www.cs.colorado.edu/~kena/classes/7818/f08/lectures/lecture_3_soap.pdf>
5. *e-business Process Automation* [interaktyvus] – [žiūrėta 2012-12-27]. Prieiga per internetą: <<ftp://ftp.software.ibm.com/software/ts/mqseries/library/whitepapers/ebusinesspa.pdf>>
6. Costin Bădică, Maria Ganzha ir Marcin Paprzycki, *Developing a Model Agent-based E-commerce System*, 2007 m. [interaktyvus] – [žiūrėta 2012-12-27]. Prieiga per internetą: <http://link.springer.com/chapter/10.1007%2F978-3-540-37017-8_26?LI=true>
7. Kęstutis Stankevičius, *REST architektūrinio stiliaus palyginimas su SOAP, įgyvendinant šiuolaikines interneto paslaugas*, 2013m., ISSN 2029-2341 – [žiūrėta 2014-03-27]. Prieiga per internetą: <<http://www.mla.vgtu.lt/index.php/mla/article/download/mla.2013.16/pdf> >
8. Hatem Hamad, Motaz Saad ir Ramzi Abed Performance, *Evaluation of RESTful Web Services for Mobile Devices*, 2010 m. [interaktyvus] – [žiūrėta 2013-11-20]. Prieiga per internetą: <http://www.iajet.org/iajet_files/vol.1/no.3/Performance%20Evaluation%20of%20RESTful%20Web%20Services%20for%20Mobile%20Devices.pdf>
9. R. L. Costello, *Building Web Services the REST Way. xFront*. [žiūrėta 2013-11-20]. Prieiga per internetą: <<http://www.xfront.com/REST-Web-Services.html>>
10. *Chapter 2: PHP and Simple Application Programming Interface for XML*, 2010 m. – [žiūrėta 2014-03-27]. Prieiga per internetą: <<http://www.beknowledge.com/wp-content/uploads/2010/09/464.pdf>>
11. Kiril Griahev, *Pusiau struktūrizuoto internetinio puslapio duomenų įrašų stebėjimo modelis* (2013 m.)
12. Robert Richards, *Pro PHP XML and Web Services*. 2006m. ISBN: 1590596331
13. *DOM versus SAX* [interaktyvus] – [žiūrėta 2014-03-27]. Prieiga per internetą: <http://pic.dhe.ibm.com/infocenter/tpfhelp/current/index.jsp?topic=%2Fcom.ibm.ztpf-ztpfdf.doc_put.cur%2Fgtpx1%2Fdomsax.html>
14. *Introduction to Distributed Systems*, 2005m. [interaktyvus] – [žiūrėta 2014-02-26] Prieiga per internetą: <http://courses.cs.washington.edu/courses/cse490h/07sp/readings/dist_systems.pdf>

15. Sara Cohen, *Generating XML Structure Using Examples and Constraints*, 2008 m. – [žiūrėta 2013-10-15] <http://www.vldb.org/pvldb/1/1453910.pdf>
16. Gintarė Jurgelionytė, *Elektroninės prekybos plėtros modelis*, 2010 m. – [žiūrėta 2014-03-10] Prieiga per internetą: <http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2010~D_20100615_131620-20840/DS.005.0.02.ETD>
17. *HTML Purifier Standards-Compliant HTML Filtering*, [interaktyvus] – [žiūrėta – 2014-03-15] Prieiga per internetą: <<http://htmlpurifier.org>>
18. Renata Janošcová, *Evaluation of software quality*, 2011 m. – [žiūrėta – 2014-03-15] Prieiga per internetą: <http://www.ondrejsimpach.ic.cz/publikace/konference_mezinarodni/IMEA2012/IMEA2012_Sbornik.pdf#page=28>
19. Hughes B., Cotterell. M, *Software Project management*, 1999 m. McGraw-Hill Publishing Company.
20. *Domenai, domenu registravimas - UAB "Interneto vizija"* [interaktyvus] – [žiūrėta – 2014-03-15] Prieiga per internetą: <<http://iv.lt>>
21. *Oficiali „Google Analytics“ svetainė – žiniatinklio analizė ir ataskaitų teikimas – Google Analytics* - [žiūrėta – 2014-03-15] Prieiga per internetą: <<http://www.google.com/analytics>>
22. REESE, G. *Database Programming with JDBC and Java*. 2000. 328 p. ISBN: 1565926161.
23. Nurzhan Nurseitov, Michael Paulson, Randall Reynolds ir Clemente Izurieta, *Comparison of JSON and XML Data Interchange Formats: A Case Study*, 2005m. [interaktyvus] – [žiūrėta 2014-04-26] Prieiga per internetą: < <http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf>>

1 priedas. Duomenų bazės fizinė schema

Schemoje (žr. 50 pav.) pavaizduota tarpinės sistemos duomenų bazės struktūra, tai yra: lentelės, jų savybės ir sąryšiai. Žalia spalva išskirti trys laukeliai *ba_grid* (*grupės id e-parduotuvės duomenų bazėse*) ir *ba_prid* (*prekės id e-parduotuvės duomenų bazėse*). Naudojant šiuos elementus, duomenų sinchronizavimo modulis įrašo nesamus prekių profilius arba sinchronizuoja informaciją su e-parduotuvės duomenų bazėmis.



50 pav. Tarpinės sistemos duomenų bazės struktūra

Lentelių detalizavimas

Pavadinimas	preke
Aprašymas	Skirta pagrindinei tiekėjo prekės informacijai saugoti.
Struktūra	prid: Integer(10), required, PK, AI gr_id: Integer(10), required tiek_id: Integer(10), required gam_kodas: varchar(100) prekes_nr: varchar(100), required pavadinimas: varchar(500), required pre_kiekis: decimal(10,4), required vnt: varchar(50) gamintojas: varchar(100) create_date: date_time, required edit_date: date_time del_date: date_time ba_prid: Integer(10) ba_grid: Integer(10)
Apribojimai	Įrašas turi būtinai turėti: tiekėjo id (tiek_id), kad būtų galima atskirti kuriam tiekėjui prekė priklauso, grupės id (gr_id), kad būtų galima susieti prekes su tiekėjo grupėmis, prekės numerį (pagal jį identifikuojamas prekės unikalumas), pavadinimą, sukūrimo datą ir sandėlio likutį. Visi kiti laukeliai užpildomi pasirinktinai.
Sąsaja	n:1 su lentele „grupe“ 1:1 su lentele „preke_info“ 1:1 su lentele „preke_kaina“

	1:1 su lentele „preke_updatable_fields“ 1:n su lentele „nuotrauka“ n:1 su lentele „tiekejas“ 1:n su lentele „charak_reiks“
<i>Resursai</i>	-
<i>Skaičiavimai</i>	-

<i>Pavadinimas</i>	preke_info
<i>Aprašymas</i>	Skirta prekės aprašymui saugoti.
<i>Struktūra</i>	prid: Integer(10), required, PK aprasymas: Bigtext
<i>Apribojimai</i>	-
<i>Šąsaja</i>	1:1 su lentele „preke“
<i>Resursai</i>	-
<i>Skaičiavimai</i>	-

<i>Pavadinimas</i>	preke_kaina
<i>Aprašymas</i>	Skirta prekės kainoms saugoti.
<i>Struktūra</i>	prid: Integer(10), required, PK savikaina: Decimal(10,4), required mazmenine: Decimal(10,4)
<i>Apribojimai</i>	Prekė būtinai turi turėti savikainą, ji yra saugoma su PVM.
<i>Šąsaja</i>	1:1 su lentele „preke“
<i>Resursai</i>	-
<i>Skaičiavimai</i>	Savikaina*PVM(21%)

<i>Pavadinimas</i>	preke_updatable_fields
<i>Aprašymas</i>	Skirta saugoti būsenas (true/false), kurias prekės informacijos dalis galima atnaujinti.
<i>Struktūra</i>	prid: Integer(10), required, PK charakteristika: TinyInt(), required kaina_likutis: TinyInt(), required aprasymas: TinyInt(), required nuotraukos: TinyInt(), required
<i>Apribojimai</i>	Kiekvienas prekės įrašas turi turėti parametrus, kurie nustatomi įkėlimo momentu automatiškai, vėliau juos gali koreguoti vadybininkas.
<i>Šąsaja</i>	1:1 su lentele „preke“
<i>Resursai</i>	-
<i>Skaičiavimai</i>	-

<i>Pavadinimas</i>	nuotrauka
<i>Aprašymas</i>	Skirta prekės nuotraukų nuorodoms saugoti.
<i>Struktūra</i>	prid: Integer(10), required, PK foto: Varchar(755) ba_grid: Integer(10)
<i>Apribojimai</i>	-
<i>Šąsaja</i>	1:n su lentele „preke“
<i>Resursai</i>	-
<i>Skaičiavimai</i>	-

<i>Pavadinimas</i>	grupe
<i>Aprašymas</i>	Skirta tiekėjų grupėms saugoti.
<i>Struktūra</i>	gr_id: Integer(10), required, PK, AI pavadinimas: Varchar(755), required

Apribojimai	Grupės pavadinimas yra privalomas, iki 755 utf_8 koduotės simbolių.
Sąsaja	1:n su lentele „preke“
Resursai	-
Skaičiavimai	-

Pavadinimas	tiekejās
Aprašymas	Skirta esamų tiekėjų informacijai ir parametrų saugoti.
Struktūra	tiek_id: Integer(10), required, PK pavadinimas: Varchar(755), required del_date: date active: TinyInt valiuta: varchar(10), required
Apribojimai	Privalomi laukai - pavadinimas (utf_8, 755 simbolių) ir valiuta (10 simbolių, tiekėjo pateikiama prekės kainos valiuta).
Sąsaja	1:n su lentele „preke“ 1:n su lentele „update_date“ n:1 su lentele „tiekejās“
Resursai	-
Skaičiavimai	-

Pavadinimas	charak_reiks
Aprašymas	Skirta prekės charakteristikų reikšmėms saugoti.
Struktūra	prid: Integer(10), required, PK cha_id: Integer(10), required, PK reiksme: Varchar(755)
Apribojimai	Įrašant prekės charakteristiką, reikšmė privaloma (utf_8, 755 simbolių).
Sąsaja	n:1 su lentele „preke“ n:1 su lentele „charakteristika“
Resursai	-
Skaičiavimai	-

Pavadinimas	charakteristika
Aprašymas	Skirta charakteristikų pavadinimams saugoti.
Struktūra	cha_id: Integer(10), required, PK, AI pavadinimas: Varchar(755), required
Apribojimai	Charakteristikos pavadinimas privalomas (utf_8, 755 simbolių).
Sąsaja	1:n su lentele „charak_reiks“
Resursai	-
Skaičiavimai	-

Pavadinimas	update_date
Aprašymas	Skirta kaupti prekių atnaujinimo/sukūrimo/pašalinimo statistikai.
Struktūra	id: Integer(10), required, PK, AI tiek_id: Integer(10), required update_starts: date_time update_ends: date_time nauju_prekiu: Integer(11) atnaujintu_prekiu: Integer(11) istrintu_prekiu: Integer(11)
Apribojimai	Tiekėjo id (tiek_id) privalomas.
Sąsaja	n:1 su lentele „tiekejās“
Resursai	-
Skaičiavimai	-

Pavadinimas	valiuta
Aprašymas	Skirta saugoti valiutų kursams lito atžvilgiu.
Struktūra	valiuta: Varchar(10), required, PK kursas: Decimal (10,4), required
Apribojimai	Sukuriant valiutos kurso įrašą, kurso skaitinė vertė privaloma.
Šąsaja	1:n su lentele „tiekejas“
Resursai	-
Skaičiavimai	-

Pavadinimas	config
Aprašymas	Skirta saugoti tarpinės sistemos parametrus.
Struktūra	id: Integer(10), required, PK, AI name: Varchar (50), required value: Varchar(255), required
Apribojimai	Sukuriant tarpinės sistemos parametą, pavadinimas ir reikšmė yra privalomi elementai.
Šąsaja	-
Resursai	-
Skaičiavimai	-

2 priedas. Prekės profilis XML pavidalu

```

<Product>
  <prekes_nr>719003</prekes_nr>
  <vienetai>vnt</vienetai>
  <Pavadinimas>Hidroforas WILO HMP 304 EM</Pavadinimas>
  <Kaina>2000</Kaina>
  <mazmenine_kaina>2200</mazmenine_kaina>
  <sandelyje>0</sandelyje>
  <gamintojas>WILO</gamintojas>
  <kategorija>Hidroforai</kategorija>
  <pics>
    <pic>http://www.jaukurai.lt/out/pictures/1/719003_p1.png</pic>
    <pic>http://www.jaukurai.lt/out/pictures/2/719003_p2.png</pic>
  </pics>
  <Aprasymas><![CDATA[<p>Konstrukcija</p>
    <ul>
      <li>Nesavisiurbis vandentiekio įrenginys. </li></ul>
    <p>Naudojimas: ....</p>
    <ul>
  ]>
  <Aprasymas>
  <Attributes>
  <Attribute title="Galingumas, W">550</Attribute>
  <Attribute title="Tūris, l">50</Attribute>
  <Attribute title="El. tinklo jungtis, V">1~230 V/50 Hz </Attribute>
  <Attribute title="Prekė">Hidroforas</Attribute>
  <Attribute title="Gamintojas">WILO</Attribute>
  </Attributes>
</Product>

```