KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Jurgis Gleixner

Saityno informacijos išrinkimo metodų tyrimas

Magistro darbas

Darbo vadovas
doc. dr. T. Blažauskas
2011-05-25

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Jurgis Gleixner

Saityno informacijos išrinkimo metodų tyrimas

Magistro darbas

Recenzentas
doc. dr. Rita Butkienė
2011-05-25

Vadovas
doc. dr. T. Blažauskas
2011-05-25

Atliko
IFM-9/2 gr. stud.
Jurgis Gleixner
2011-05-25

Kaunas, 2011

Evaluation of information extraction methods from World Wide Web

Summury

Nowadays the amount of information in internet is increasing very fast. It becomes a difficult and time consuming work to find required information. Not many websites offer a possibility to filter information in more complex ways. The solution of this problem is an information extraction system, which collects information from websites and transforms it into a more flexible form (XML, CSV, DB), where complex filters and data manipulations can be applied.

In this work we analyze methods to automatically extract information from websites in simple and interactive way. This work is more focused on structural pattern based information extraction systems. We introduce such a system and compare its functionality with other similar systems. Precision is one of the most important attributes of such systems, so we analyze ways to increase it.

Turinys

1 Įvadas

Nuolat didėjant įvairių informacinių, el. parduotuvių, skelbimų ir kitokio pobūdžio internetinių svetainių kiekiui, tampa vis sunkiau surasti ir atsirinkti reikalingą informaciją. Ypač daug laiko reikalauja periodinis informacijos ieškojimas, nes kiekvienas najuai atsiradęs ieškomos informacijos šaltinis prailgina visą paiešką. Dar sunkaiu yra todėl, kad ne visos internetinės svetainės teikia vienodas ir geras galimybes atsirinkti norimą informaciją, o kai kuriose iš jų tokių galimybių praktiškai nėra. Pavyzdžiui ieškodami skelbimų turime peržiūrėti daugybę interneto svetainių. Kai kuriuose iš jų galime naudotis paieška (jei ji yra), kad atsirinktume tai ko mums reikia, kitose turime papraščiausiai peržiūrėti visus neseniai atsiradusius skelbimus. Norėdami įsygti prekę, galime ieškoti jos el. parduotuvėse, tačiau kadangi jų yra labai daug ir vis daugėja, labai sunku yra rasti optimalų variantą ir tuo pačiu tai reikalauja daug laiko.

Todėl atsiranda poreikis automatizuoti informacijos išrinkimą iš daugelio internetinių svetainių ir pateikti ją vartotojui patogioje formoje, taip kad jis galėtų išrinktą informaciją filtruoti, palyginti ir t. t. nebeieškant jos daugybėje skirtingų inrneto svetainių.

Tarkim vartotojas eBay puslapyje ieško mp3 grotuvo, kuris parduodamas aukcionu, iki aukciono pabaigos liko mažiau negu viena diena ir jo kaina 20-40 Eurų. Ši svetainė, kaip ir dauguma kitų svetainių neturi tokių plačių paieškos galimybių, todėl joje negalėtų būti atlikta tokia paieška, tačiau turint išrinktą informaciją patogesniame formate (duomenų bazėj, XML), tokia paieška nebūtų sudėtinga [1]. Taip pat vartotojas sąrašą galėtų apjungti su kitų aukcionų puslapių prekėmis.

Apibendrinant, pagrindiniai informacijos išrinkimo sistemos privalumai yra šie:

- Informacijos paieška yra automatizuota, todėl vartotojas sutaupo daug laiko.
- Vartotojas gauna tik tą informaciją, kuri jam yra aktuali. Neaktuali ir perteklinė informacija yra atmetama informacijos išrinkimo sistemos.
- Kadangi informacija saugoma lanksčiais formatais, vartotojas turi galimybę atlikti sudėtingesnes informacijos paieškas.

Šio darbo tikslas yra išanalizuoti informacijos išrinkimo metodus, ištirti juos įvairiais aspektais.

2 Informacijos išrinkimo metodų analizė

2.1 Tyrimo sritis

Pagrindinė tyrimo sritis – informacijos išrinkimas iš interneto saityno, informacijos išrinkimo procesas ir metodai. Informacijos išrinkimo procesą galima išskaidyti į kelis etapus, kurie gali skirtis priklausomai nuo pasirinkto informacijos išrinkimo metodo. Bendru atveju informacijos išrinkimo procesas yra ciklinis ir gali būti aprašytas šiais žingsniais [2,3]:

1. Šablono/paieškos kriterijų sudarymas
2. Puslapio analizė
3. Vidinių nuorodų išrinkimas
4. Informacijos atpažinimas
5. Informacijos ištraukimas ir saugojimas

Šablono sudarymas – tai paieškos kriterijų nustatymas. Priklausomai nuo metodo, tai gali būti interaktyvus vartotojo ieškomų duomenų nurodymas, paieškos teksto įvedimas ir panašiai.

Sekantys žingsniai sudaro ciklą. Šiuose etapuose analizuojamas puslapis, išrenkamos nuorodos į kitus vidinius puslapius (kitų ciklų metu informacija išrenkama iš rastų vidinių puslapių), remiantis paieškos kriterijais randami informacijos blokai, iš kurių ištraukiama ir paruošiama pateikimui informacija.

2.2 Informacijos išrinkimo metodai

Kaip pavaizduota žemiau esančiame paveiksle, informacijos išrinkimo metodus galima sugrupuoti į dvi grupes: šabloninius ir nešabloninius. Šabloninių metodų informacijos paieška yra paremta šablonais, tai yra išrenkama ta informacija, kuri atitinka tam tikrą šabloną [4,5]. Šablonai dažniausiai sudaromi vartotojo pagalba interaktyviu būdu, tačiau taip pat taikomi metodai automatiniam struktūrizuotos informacijos šablonų sudarymui [6]. Nešabloniniai paieškos metodai paieškai naudoja kitus būdus, pvz. kalbos analizę, vizualią puslapio išvaizdą, t. y. tai kaip puslapį mato vartotojas [7].

Pav. 1 Informacijos išrinkimo metodų grupavimas

2.2.1 Tekstiniais šablonais paremtas informacijos išrinkimas

Tekstiniais šablonais paremtas informacijos išrinkimas yra vienas iš paprastesnių išrinkimo metodų. Jis analizuoja puslapį ne kaip HTML struktūrą, o kaip vientisą tekstą. Šablonas, paprasčiausiu atveju, yra teksto ar HTML kodo dalys, tarp kurių yra ieškoma informacija [8]. Tarkim turime tokią kodo dalį iš kurioje reikia surasti kainą:

```
<span>Kaina: 100Lt</span>
```

Vienas iš galimų šablonų (užrašytas kaip reguliari išraiška) galėtų būti „Kaina: (.*)Lt“, kuris reikštų, kad ieškoma informacija yra tarp teksto „Kaina: “ ir „Lt“ [9,10]. Šio metodo tikslumas labai priklauso nuo sudaryto šablono. Kuo mažiau teksto įtrauksime į šabloną, tuo didesnė tikimybė išrinkti nepageidaujamą informaciją, tačiau jei į šabloną bus įtraukta per mažai teksto, informacijos išrinkimas bus nelankstus pokyčiams, tai yra mažas pokytis puslapyje gali lemti tai, kad informacija nebus surasta. Dėl šių priežasčių šablonų sudarymas yra sudėtingas uždavinys, kuris dažnai rinkoje egzistuojančiuose produktuose paliekamas spręsti vartotojui.

Kadangi šis metodas dažniausiai neanalizuoja HTML struktūros, jis gali būti taikomas ir su kitokio formato failais.

Kaikurios sistemos naudoja pakopinius šablonus ir veikimo principu primena struktūriniais šablonais paremtus metodus [11,12]. Naudojant pakopinius šablonus informacija ieškoma keliais etapais. Pavyzdžiui ieškodami žodžio „tekstas“ tokiame HTML faile galėtume sudaryti tris pakopas.

```
<body>
        <div>....</div>
        <div>
                <span>tekstas</span>
        </div>
</body>
```

Pirmiausia rastume HTML kodo dalį esančią tarp ‚<body‘ ir ‚</body>‘, tada šioje dalyje ieškotume kodo esančio tarp ‚<div‘ ir ‚ ‘. Paskutiniame žingsnyje visuose praeito žingsnio rezultatuose ieškotume teksto esančio tarp ‚<span‘ ir ‚‘.

2.2.2 Struktūriniais šablonais paremtas informacijos išrinkimas

Šis informacijos išrinkimo metodas ieško informacijos atsižvelgiant į HTML struktūrą. Paprastai iš pradžių sudaromas puslaipio medis, kuriame ieškoma šabloną atitinkančių fragmentų [13,14]. Yra įvairių šablonų sudarymo būdų, nuo kurių priklauso sistemos parametrai, kaip informacijos išrinkimo tikslumas bei sparta.

Pavyzdžiui jei turime tokį HTML kodą, iš kurio norime išrinkti tam tikrus sąrašo paveiksliukus ir jų pavadinimus:

```
<body>
        <div><img src="paveiksliukas.jpg">...</div>
        <div>
                <ul>
                <li><span>pavadinimas a</span><img src="a.jpg" alt="" /></li>
                <li><span>pavadinimas b</span><img src="b.jpg" alt="" /></li>
                </ul>
        </div>
        <ul>
                <li>tekstas</li>
                <li>tekstas</li>
        </ul>
...
</body>
```

Pirmame žingsnyje (dažniausiai interaktyviai) sudaromas ieškomos informacijos šablonas. Bendru atveju šablonas yra kelias HTML medyje su tam tikrais apribojimais [15]. Šiuo atveju šablonas galėtų būti toks:

body->div->ul->li->span
body->div->ul->li->img

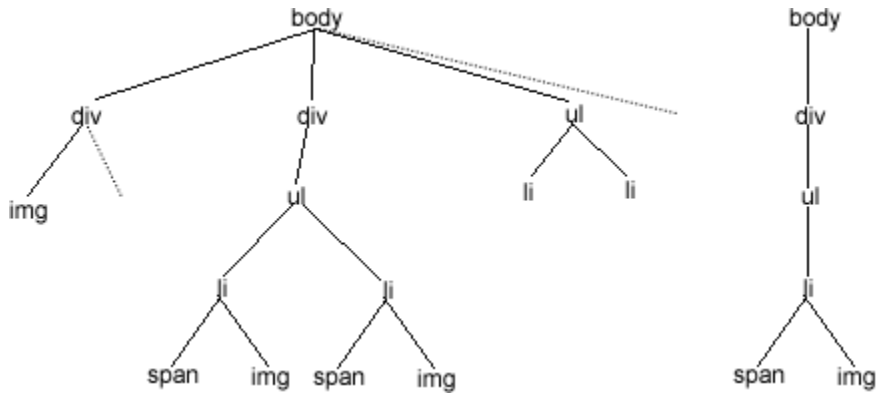
Kadangi pavadinimas ir paveiksliukas sudaro vieną informacijos vienetą, pravartu apjungti šablonus (sudaryti bendrą šabloną) [16]. Radus šabloną atitinkantį elementą, jame būtų ieškoma atitinkama informacija.

body->div->ul->li
span
li

Kai kuriais atvejais, norint padidinti informacijos išrinkimo tikslumą, galima įsivesti tam tikrus apribojimus. Apribojimais galėtų būti elementų numeris medyje.

body->div[2]->ul->li

Tokiu būdu padidinsim išrinkimo tikslumą, nes nebus išrenkama informacija, atsitiktinai atitikusi šabloną, iš kitų ‚div' elementų. Tuo pačiu sumažinsim atsparumą pokyčiams. Įsiterpus vienam ‚div' elementui į pradžią, ieškoma informacija nebus rasta.
Puslapio analizės metu sudaromas HTML medis. 2 pav. pavaizduota sudaryta struktūros dalis ir paieškos šablonas.



Pav. 2 Sudarytas HTML medis ir paieškos šablonas

Toliau pagal šabloną šioje struktūroje pažingsniui ieškoma informacija. Pradedant nuo viršūnės lyginame ją su šablono pirmuoju elementu. Atitikus elementams einame gylyn ir

lyginame visus atitikusio elemento vaikus su sekančiu šablono elementu atsižvelgdami į apribojimus.

2.2.3 Puslapio išvaizda paremtas informacijos išrinkimas

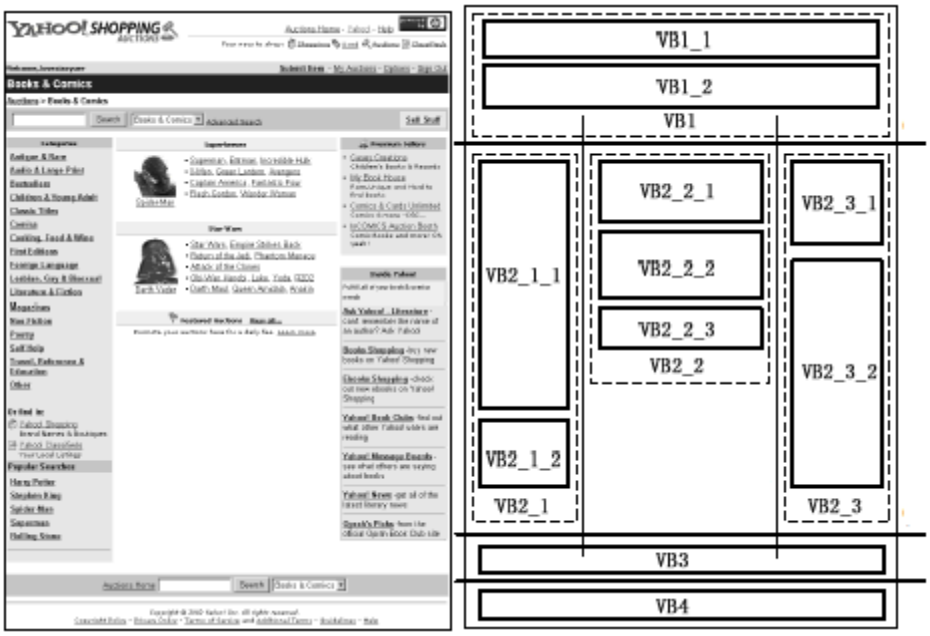
Puslapio išvaizda paremtas informacijos išrinkimas analizuoja puslapį pagal tai, kaip jį mato vartotojas. Kadangi interneto puslapiai yra sudaryti iš daug blokų, iš kurių daugumoje nėra naudingos informacijos (reklamų blokai, navigacijos blokas, logotipas ir t. t.), šio metodo pagrindinis uždavinys – atpažinti blokus, kuriuose yra naudinga informacija. Šį uždavinį galima padalinti į du etapus [17,18]:

1. Blokų atpažinimas
2. Naudingų blokų radimas

Pirmame etape analizuojamas puslapis ir nustatomi puslapio blokai remiantis šiais atributais [18]:

- HTML elementų pozicija, dydis
- Elementų fonas, rėmelis
- Skyrikliai (paveiksliukai, linijos)
- Elementų turinys

Kaip pavaizduota 3 pav. remiantis aukščiau išvardintais atributais, puslapis buvo padalintas į stambesnius ir smulkesnius blokus.



Pav. 3 HTML puslapis ir jame atpažinti blokai

Antrame etape apskaičiuojamas šių blokų tinkamumas, t. y. tikimybė, kad bloke yra naudinga informacija. Ši tikimybė apskaičiuojama remiantis įvairiais parametrais (bloko pozicija, turiniu ir panašiai) [18]. Šiuo atveju išrenkamas blokas su numeriu VB2_2.

Kadangi šis informacijos išrinkimo metodas pats nustato informacijos vietą puslapyje, šablono sudarymas vartotojui tampa nebereikalingas. Nors šis metodas labiau tinkamas nestruktūrizuotai informacijai išrinkti, tą galime padaryti analizuodami nustatyto bloko vidinius blokus. Pateiktame pavyzdyje blokai VB2_2_1, VB2_2_2 ir VB2_2_3 gali būti informacijos vienetais. Šių blokų analizės metu turėtų būti atmetamas paskutinis blokas (VB2_2_3), kadangi jo struktūra ryškiai skiriasi nuo likusių dviejų. Šiam etape vidinių blokų struktūros analizė turi būti atliekama atsižvelgiant ir į kitus svetainės puslapius [18].

2.2.4 Kalbos analize paremtas informacijos išrinkimas

Šis metodas išrenka informaciją iš teksto, remdamasis sakinių, žodžių analize. Žodžiai analizuojami pasitelkiant [19]:

- Žodyną
- Pavadinimų, vardų, vietovardžių sąrašus

- Šablonus, pagal kuriuos nustatoma žodžio reikšmė

Žodyno paskirtis – atpažinti kalbos dalis, kurių pagalba vėliau analizuojamas sakinys. Šiuo atveju žodynas yra ne tik žodžių sąrašas, bet ir ryšiai tarp žodžių, sinoniminiai žodžiai.

Vardų, vietovardžių ir kitų pavadinimų sąrašų pagalba nustatomi žodžiai, neesantys žodyne. Platesni sąrašai lemia didesnį sistemos informacijos išrinkimo tikslumą.

Kai kurių žodžių reikšmės ar paskirtis gali būti atpažinta lyginant jį su įvairiais šablonais [20]. Tokius būdu gali būti atpažintos datos, kainos, įmonių pavadinimai ir kitų sričių žodžiai. Pavyzdžiui, žmonių vardus dažnai sudaro arba du žodžiai prasidedantys didžiąja raide arba didžioji raidė, po kurios seka taškas ir antras žodis iš didžiosios raidės, taigi pagal tai galima sudaryti tokus šablonus vardų atpažinimui (dėl paprastumo neįtrauktos lietuviškos raidės).

```
[A-Z]\. [A-Z][a-z]*
[A-Z][a-z]* [A-Z][a-z]*
```

Įmonių pavadinimai dažnai naudojami kartu su santrumpom „AB“, „UAB“, „VŠĮ“, todėl radę tokią santrumpą, galime teigti, kad po jos einantis žodis ar žodžiai yra įmonės pavadinimas. Analogiškai sudaromi šablonai įvairiems datų formatams, kainoms ir kitiems žodžiams, kurie gali būti atpažinti pagal šabloną.

Paieškos etape tekstas suskaidomas į sakinius, kurie skaidomi į smulkesnius fragmentus. Remiantis žodynu randamos žodžių kalbos dalys (daiktavardis, veiksmažodis, būdvardis...), randami vietovardžiai, pavadinimai, kiekvienam žodžiui bandoma pritaikyti kurį nors šabloną [20,21]. Toliau seka sakinių, sintaksės analizė, kurios metu nustatomos sakinio dalys, tai yra:

- Veiksnys (kas?)
- Tarinys (ką daro?)
- Papildiniai (ką? ko? kam?)
- Pažyminiai (koks?)
- Vietos bei laiko aplinkybės (kur? kada?)

Išanalizavus sakinius, nustatomi ryšiai tarp jų. Kadangi objektai gali būti pavadinti skirtingais pavadinimais, sakiniuose ieškome daiktavardžių ir sinonimų žodyno pagalba tikriname jų atitikimus ankstesniuose sakiniuose [19,21]. Pavyzdžiui, viename sakinyje galėtų būti paminėtas žmogaus vardas, kuris būtų atpažintas šablono pagalba, tačiau sekančiame sakinyje šis vardas galėtų būti pakeistas į žodį „asmuo“, „pilietis“ ar kokį nors kitą sinonimą. Norint padidinti šio etapo tikslumą, labai svarbu yra turėti didelį kiekį ryšių tarp žodžių.

Išanalizavus sakinius ir jų ryšius belieka atrinkti ir suformuoti naudingą informaciją. Naudinga informacija gali būti nurodoma įvairiais būdais. Abstrakčiausias būdas būtų nurodyti šabloną:

Veiksnys tarinys (papildinys) (aplinkybė)

Pagal šį šabloną būtų atrenkami visi faktai (tai būtų praktiškai visa informacija). Toliau galima siaurinti šablonus. Tarkim mus domina faktai apie konkretų asmenį. Tokiu atveju detalizuojam veiksnį:

Veiksnys[„Pavardė“] tarinys (papildinys) (aplinkybė)

Analogiškai gali būti apribotas tarinys arba aplinkybės. Pavyzdžiui galima ieškoti įvykių įvykusių tam tikroj šalį ar tam tikrą dieną.

2.3 Informacijos atmetimas

Dėl šio metodo veikimo principo, jis tik iš dalies galėtų būti priskirtas prie informacijos išrinkimo metodų, tačiau jis galėtų būti kombinuojamas su kitais metodais informacijos išrinkimo tikslumui padidinti.

Šio metodo esmė - atpažinti ir atmesti viską, kas nėra informacija, t. y. vartotojo navigacija, baneriai ir panašiai. Idealiu atveju iš puslapio turėtų likti tik blokas, kuriame yra naudinga informacija [22].

Atmetimas gali būti realizuotas analizuojant HTML struktūrą ir atmetinėjant jos elementus. Elementų išdėstymas skirtinguose interneto puslapiuose yra daugmaž panašus. Pavyzdžiui viršuj dažnai būna blokas su logotipu ir navigacija. Tame pačiame bloke arba šone prisijungimo arba paieškos forma, kurią galima atpažinti iš laukų bei mygtuko. Daugumoje puslapių yra viena arba keletas navigacijos juostų. Jos gali būti atpažintos iš daugelio viena po kitos sekančių nuorodų į vidinius svetainės puslapius. Reklamos gali būti atpažintos iš paveiksliukų arba teksto su išorinėm nuorodom.

Iš dalies šis metodas panašus į puslapio išvaizdos analize paremtą išrinkimo metodą, tačiau remiantis vien tik turinio alanlize, pasiekiamas nedidelis tikslumas.

2.4 *Informacijos išrinkimo sistemos ir paieškos sistemos*

Dažnai informacijos išrinkimo sistemos painiojamos su paieškos sistemomis, pvz. su „Google“. Šių sistemų paskirtis nėra ta pati. Informacijos išrinkimo sistemos pagrindinė paskirtis yra ne ieškoti informacijos interneto puslapiuose, o išrinkti nurodytą informaciją iš internetos svetainės (vėliau paieška gali būti atlikta rastuose rezultatuose). Ji tinkama struktūrizuotai informacijai išrinkti (pvz. naujienoms, prekėms ir t. t.). Kitas didelis šių sistemų skirtumas yra rezultatuose ir jų manipuliavimo galimybėse. Visi pagrindiniai šių sistemų skirtumai pateikti lentelėje.

	Informacijos išrinkimo sistema	**Paieškos sistema („Google“ ir kitos)**
Veikimo principas	Ieškoma tik nurodytuose puslapiuose. Atrenkama tik reikalinga informacija, atmetant visą kitą.	Analizuojami visi interneto puslapiai, indeksuojama visa informacija, vėliau atliekant paiešką euristiniais metodais atrenkami puslapiai, kurie labiausiai atitinka vartotojo užklausą
Užklausos forma	Vartotojas sudaro norimos informacijos šabloną, pagal kurį bus išrenkama visa tos svetainės informacija	Vartotojas įveda žodžius, pagal kuriuos paieška atliekama visose svetainėse
Rezultatas	Išrinkta informacija	Nuorodos į puslapius, atitikusius paieškos užklausą
Rezultato formatas	Lankstus formatas, gali būti XML, saugojimas duomenų bazėje	HTML
Manipuliacija rezultatais	Gali būti vykdomos sudėtingos paieškos	Labai ribota

Lentelė 1 Informacijos išrinkimo sistemų ir paieškos sistemų palyginimas

2.5 Informacijos išrinkimo metodų analizės Išvados

Analizės metu buvo išskirti keturi plačiau naudojami informacijos išrinkimo metodai. Remiantis jų privalumais bei trūkumais buvo nuspręsta naudoti struktūriniais šablonais paremtą informacijos išrinkimo metodą. Šiuo metodu pasiekiamas didžiausias išrinkimo tikslumas, be to galimas automatizuotas šablonų sudarymas.

Panašiu principu veikiantis yra tekstiniais šablonais paremtas informacijos išrinkimo metodas. Dauguma šį metodą naudojančių programų šablono sudarymą palieka vartotojui, o tai reikalauja specifinių žinių ir mažina tikslumą, nors ir šis metodas yra spartesnis ir paprastesnis kūrimo atžvilgiu, jis buvo atmestas dėl mažesnio tikslumo.

Kalbos analizės metodai yra labai priklausomi nuo turimų duomenų kiekio ir nagrinėjamo teksto. Bendru atveju pasiekiamas mažas tikslumas. Šie metodai nėra tinkami struktūrizuotai informacijai išrinkti, tačiau kai kurie jų elementai gali būti kombinuojami su kitais informacijos išrinkimo metodais (pvz. datų, skaičių atpažinimas).

Puslapio išvaizda paremtas metodas pasiekia pakankamai gerą tikslumą, tačiau šio metodo rezultatai yra informacijos blokas ir jo subblokai. Norint interaktyviai sudaryti informacijos šablonus ir tiksliai nurodyti reikalinga informaciją, jis turi būti kombinuojamas su kitais metodais.

3 Projektinė dalis

Šiame skyriuje trumpai aptariami projekto tikslai, sprendžiami uždaviniai bei priimti techniniai sprendimai. Vėliau apžvelgiamas bendras sistemos architektūros vaizdas, tuo pačiu pateikiamos esminės sistemos architektūrą paaiškinančios diagramos, pateikiamas sistemos duomenų modelis ir aptariama projekto išeiga.

Skyrius yra akcentuotas į pagrindines sistemos dalis prie, kurių buvo dirbta: grafinį šablonų sudarymą, bei informacijos išrinkimą.

3.1 Projekto tikslas, paskirtis

3.1.1 Sistemos tikslas

Projekto tikslas – suteikti vartotojams galimybę paprastai ir greitai gauti informaciją, teikiamą įvairiose interneto svetainėse. Projektuojama sistema turėtų automatiškai išrinkti vartotojo nurodytą informaciją iš skirtingų svetainių ir pateikti ją patogiu būdu. Sistema gali būti naudojama asmeniniais tikslais (pvz. darbo skelbimų išrinkimui iš įvairių skelbimų puslapių), bei komerciniais tikslais (pvz. įmonės konkurentų produktų kainų stebėjimui ir t. t.).

Taip pat labai svarbu yra stebėti informacijos pokyčius, t. y. programai periodiškai kas tam tikrą laiko tarpą išrenkant informaciją, turi aiškiai matytis, kurie duomenys yra nauji, kurie liko nepakitę ir kuriuose atsirado pokyčiai (pvz sumažėjo ar padidėjo kaina).

3.1.2 Projekto kūrimo pagrindas (pagrindimas)

Nuolat didėjant įvairių informacinių, el. parduotuvių, skelbimų ir kitokio pobūdžio internetinių svetainių kiekiui, tampa vis sunkiau surasti ir atsirinkti reikalingą informaciją. Ypač daug laiko reikalauja periodinis informacijos ieškojimas, nes kiekvienas najuai atsiradęs ieškomos informacijos šaltinis prailgina visą paiešką. Dar sunkaiu yra todėl, kad ne visos internetinės svetainės teikia vienodas ir geras galimybęs atsirinkti norimą informaciją, o kai kuriose iš jų tokių galimybių praktiškai nėra. Pavyzdžiui ieškodami skelbimų turime peržiūrėti daugybę interneto svetainių. Kai kuriuose iš jų galime naudotis paieška (jei ji yra), kad atsirinktume tai ko mums reikia, kitose turime paprasčiausiai peržiūrėti visus neseniai atsiradusius skelbimus.

3.2 Reikalavimų specifikacija

3.2.1 Įpareigojantys apribojimai

3.2.1.1 Apribojimai sprendimui

Darbas vyksta kliento-serverio režimu. Visi skaičiavimai yra atliekami serverio pusėje. Sistema yra realizuota ant Apache serverio su Linux operacine sistema. Svetainės veikimas yra nepriklausomas nuo vartotojo operacinės sistemos. Vartotojui pakanka turėti Interneto naršyklę.

3.2.1.2 Diegimo aplinka

Sistema diegiama servery.

Operacinė sistema: Unix

Serverio valdymo įranga Apache

3.2.1.3 Bendradarbiaujančios sistemos

Sistema veiks Unix operacinėje sistemoje. Serverį valdys Apache įranga. Taip pat turi būti įdiegtas PHP modulis. Duomenys bus talpinami MySQL duomenų bazė

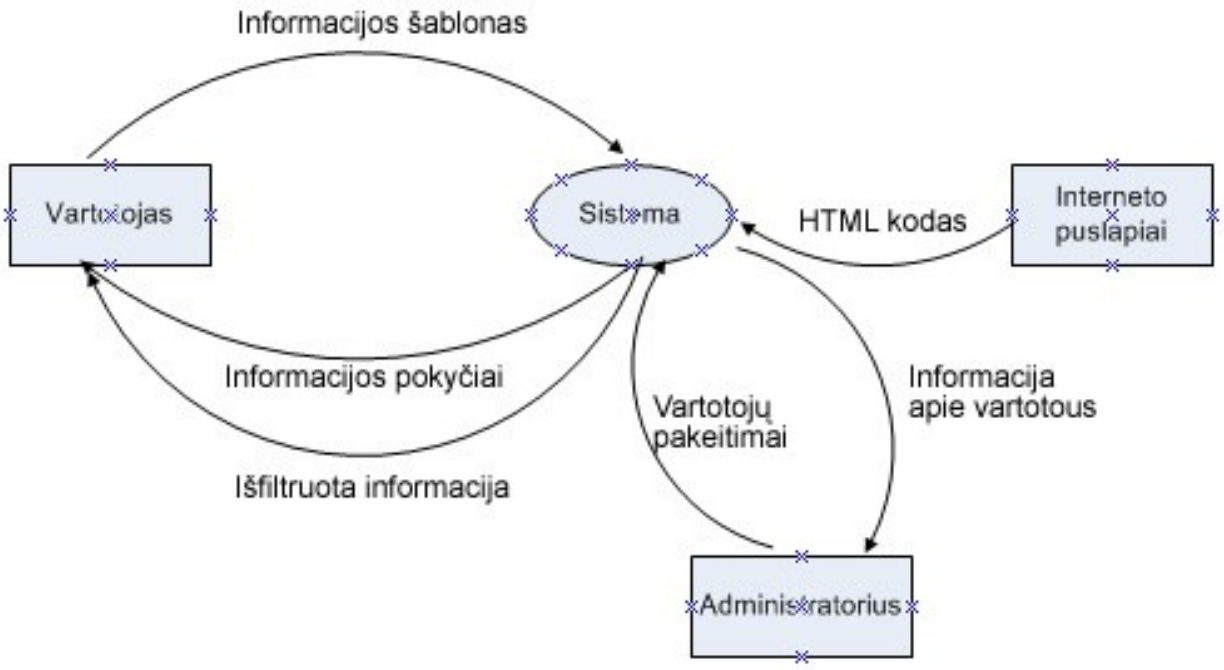
3.2.1.4 Numatoma darbo vietos aplinka

Sistema veiks serveryje, o kliento pusėje reikalinga bus tik naršyklė, todėl darbo vietos aplinka, vartotojo naudojama operacinė sistema kuriamai sistemai įtakos neturi.

3.2.2 Veiklos sfera

3.2.2.1 Veiklos kontekstas

Žemiau esančiame paveiksle ir lentelėje matome sistemos dalyvius ir pagrindinius duomenų srautus tarp jų. Vartotojas nurodo informacijos šabloną ir rezultate gauna informaciją ir jos istoriją. Administratorius mato vartotojų duomenis ir gali atlikti vartotojų pakeitimus. Pagal vartotojų nurodytus šablonus iš atitinkamų interneto puslapių gaunamas HTML kodas.



Pav. 4 Sistemos veiklos kontekstas

3.2.2.2 Veiklos padalinimas

Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Vartotojas nurodo informacijos šabloną	Informacijos šablonas
2	Vartotojas gauna išrinktą informaciją	Informacija
3	Vartotojas atsirenka informaciją	Išfiltruota informacija
4	Rodomi informacijos pokyčiai	Informacijos pokyčiai
5	Sistema išrenka informaciją	Išrinkta informacija
6	Sistema nustato pokyčius	Nustatyti pokyčiai
7	Sistema informuoja vartotoją apie pokyčius	Pranešimas
8	Administratorius redaguoja vartotojus	Vartotojų pakeitimai

Lentelė 2 Sistemos veiklos padalinimas

3.2.3 Produkto veiklos sfera

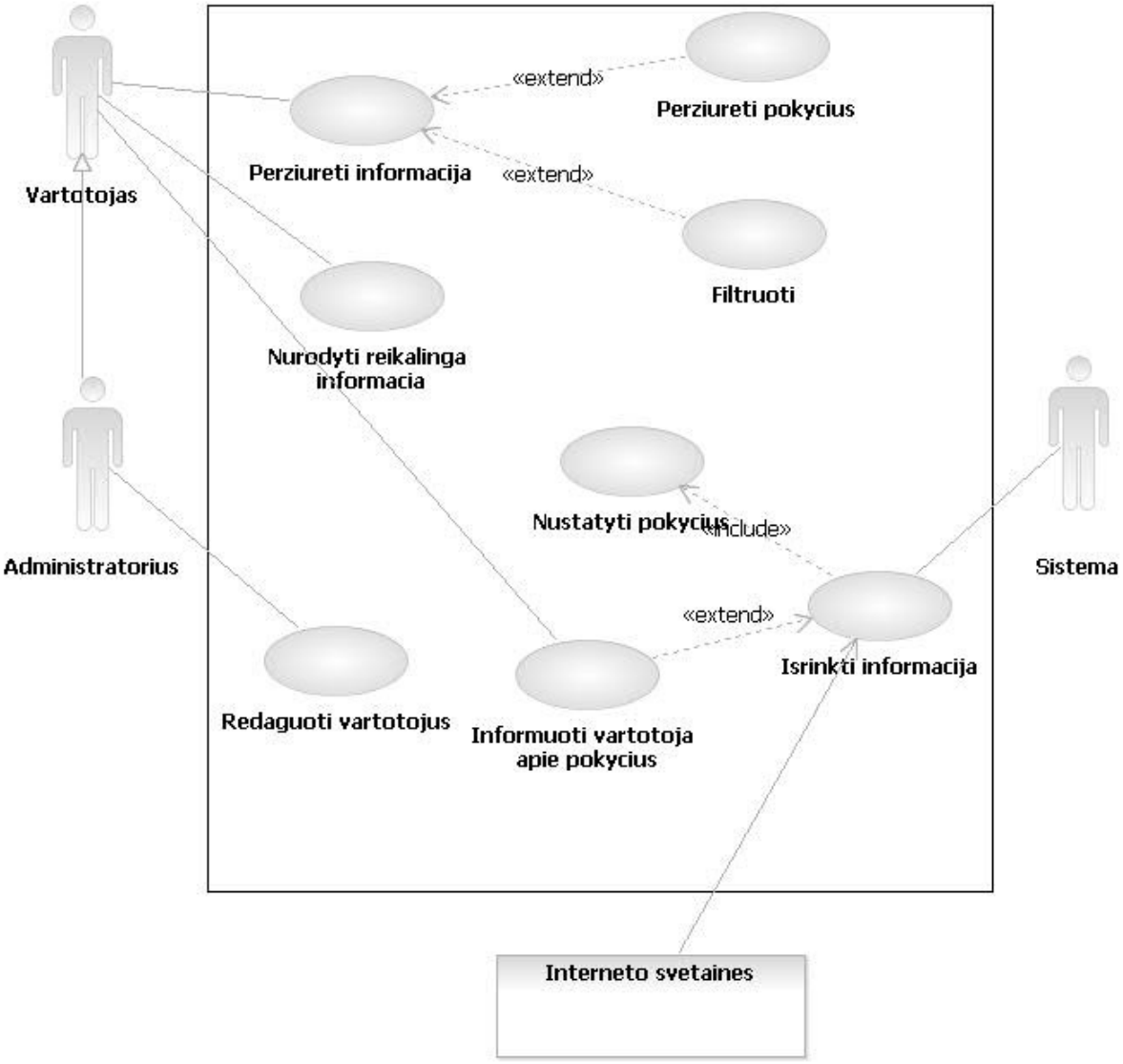
3.2.3.1 Sistemos ribos

Žemiau pavaizduotame paveiksle pavaizduotos sistemos ribos ir panaudos atvejai. Pagrindiniai sistemos aktoriai:

Vartotojas – sudaro šablonus, peržiūri informacija

Administratorius – turi vartotojo funkcijas ir gali redaguoti vartotojus

Sistema – pagrindinė funkcija – informacijos išrinkimas



Pav. 5 Sistemos ribos. Panaudos atvejai

3.2.3.2 Panaudojimo atvejų sąrašas

1. Nurodyti reikalingą informaciją

Tikslas: vienareikšmiškai nurodyta vartotojui reikalinga informacija

Aktoriai: vartotojas

Nefunkciniai reikalavimai: informacijos nurodymas turėtu būti patogus ir aiškiai suprantamas vartotojui.

2. Peržiūrėti informaciją

Tikslas: Pateikti vartotojui informaciją patogioje formoje

Aktoriai: Vartotojas

Nefunkciniai reikalavimai: patogus informacijos atvaizdavimas atsižvelgiant į tai, kad jos gali būti dideli kiekiai

3. Filtruoti

Tikslas: Suteikti vartotojui iš visos išrinktos informacijos jam atsirinkti svarbiausius duomenis

Aktoriai: Vartotojas

Nefunkciniai reikalavimai: patogus filtras pagal įvairius laukus

4. Peržiūrėti pokyčius

Tikslas: parodyti vartotojui naujai atsiradusią ir pakitusią informaciją

Aktoriai: Vartotojas

Nefunkciniai reikalavimai: Kadangi informacijos pokyčiai sistemoje yra labia svarbūs, vartotojui jie turi būti aiškiai matomi.

5. Išrinkti informaciją

Tikslas: Teisingai išrinkti vartotojo nurodytą informaciją.

Aktoriai: Sistema

Nefunkciniai reikalavimai: informacija išrenkama teisingai atsižvelgiant į galimus smulkius struktūrinius pokyčius

6. Nustatyti pokyčius

Tikslas: Nustatyti kuri informacija pakitus ar nauja

Aktoriai: Sistema

Nefunkciniai reikalavimai: nustatomi tam tikrų atributų pokyčiai, bei naujai atsiradusi informacija

7. Informuoti vartotoją apie pokyčius

Tikslas: Pranešti vartotojui apie informacijos pokyčius

Aktoriai: Sistema

Nefunkciniai reikalavimai: vartotojui pranešama apie pokyčius jam tai nurodžius. Pranešimas vyksta el. paštu arba kitokiais būdais.

8. Redaguoti vartotojus

Tikslas: pakeisti vartotojų duomenis, šalinti vartotous

Aktoriai: Administratorius

Nefunkciniai reikalavimai: nėra

3.2.4 Reikalavimai sistemos išvaizdai (Look and feel)

Intuityvi ir lengvai suprantama vartotojo sąsaja

3.2.5 Reikalavimai panaudojamumui (Usability)

Sistemą galima naudotis neturint HTML žinių

3.2.6 Reikalavimai sistemos priežiūrai (Maintainability and portability)

Sistema turi veikti nepriklausomai nuo serverio, kuriame bus įdiegta (gali būti perkelta į kitą serverį), taip pat nesvarbi serverio operacinė sistema.

3.2.7 Reikalavimai saugumui (Security)

Vartotojai gali matyti tik savo informaciją. Kitų vartotojų informacija prieinama tik administratoriui.

3.3 Architektūros specifikacija

3.3.1 Statinis sistemos vaizdas

Sistemą galima suskaidyti į 4 komponentus (žr. 6 pav.):

1. Vartotojai

Tai vartotojo valdymo komponentas, teikiantis standartines vartotojų valdymo funkcijas (pvz. registracija, prisijungimas ir panašiai)

2. Informacija

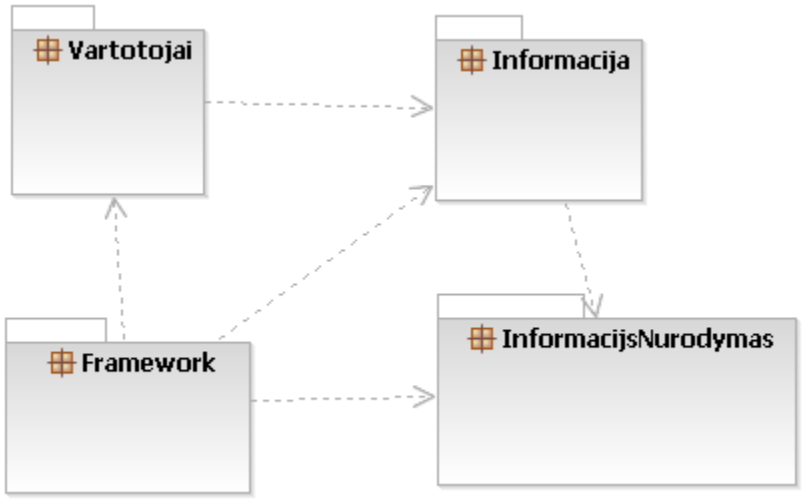
Tai pagrindinis komponentas skirtas automatiniam informacijos išrinkimui

3. Informacijos nurodymas

Šis komponentas skirtas grafinei vartotojo sąsajai, šablonų sudarymui pagal vartotojo nurodymus.

4. Framework

Šis komponentas yra sistema į kurią bus integruojama kuriama sistema. Kadangi šis komponentas yra tik sistemos karkasas ir yra nesusijęs su sistemos funkcionalumu, jis nebus toliau detalizuojamas



Pav. 6 Paketų diagrama

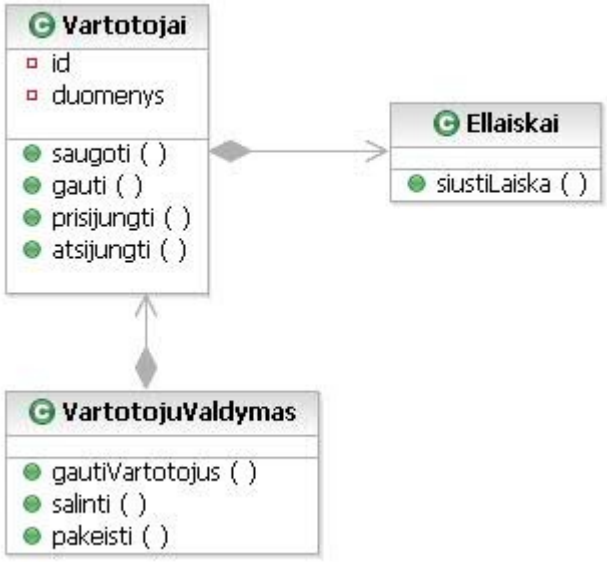
Paketas “Vartotojai”

Šiame paveikslėlyje pavaizduotos paketo „Vartotojai“ klasės ir jų pagrindiniai metodai bei atributai. Klasių paskirtis:

Vartotojai – vartotojo veiksmams (prisijungimas, registracija)

Ellaiskai – klasė el. laiškų siuntimui

VartotojuValdymas – klasė veiksmams su vartotojais



Pav. 7 Paketo "Vartotojai" klasių diagrama

Paketas “Informacija”

Žemiau pateikiama paketo “Informacija” klasių diagrama. Į diagramą įtraukti pagrindiniai klasių metodai. Klasių paskirtis:

Informacija – informacijos atvaizdavimas, filtravimas

Isrinkimas – informacijos išrinkimas, pokyčių nustatymas tarp esamų duomenų ir naujai išrinktų.

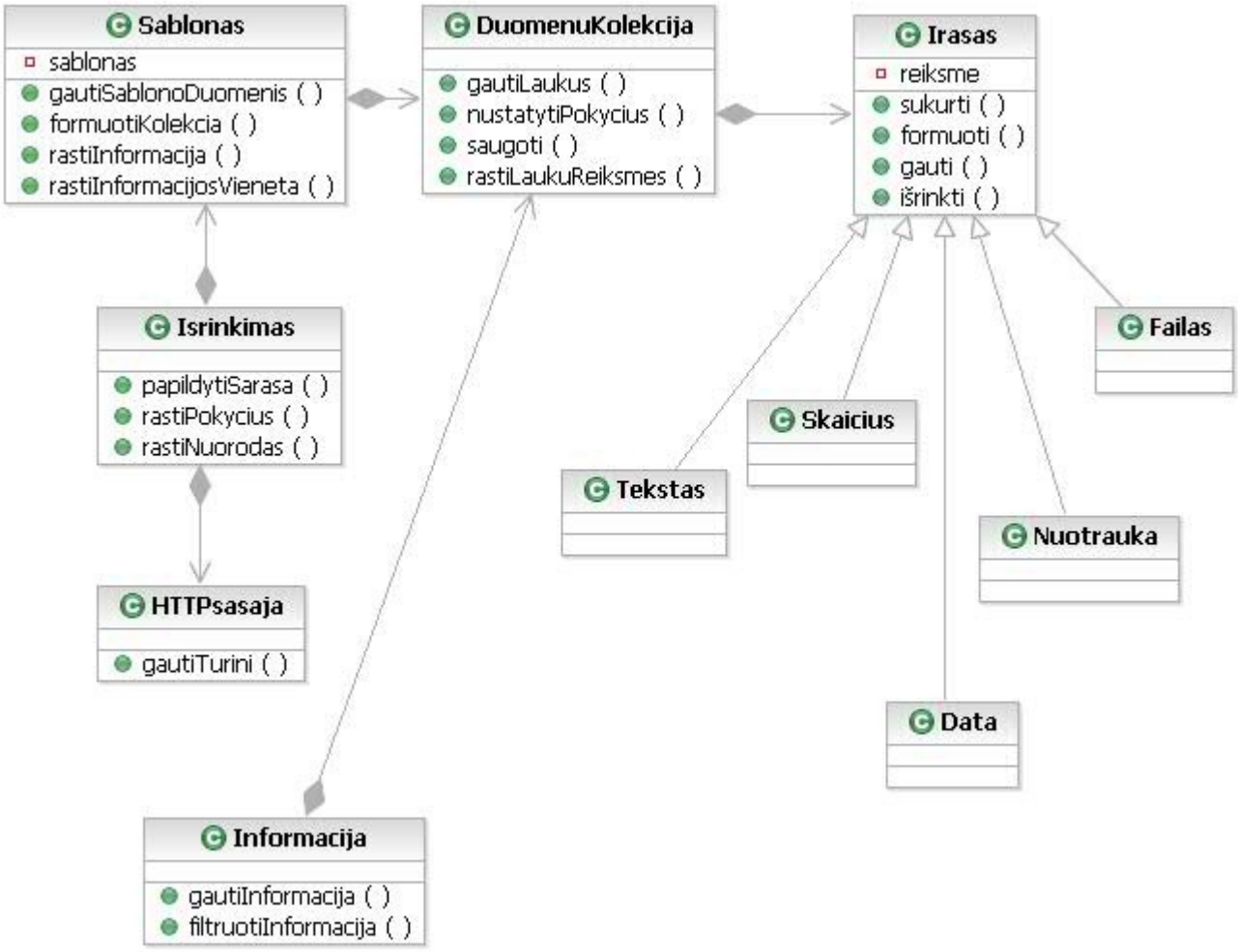
Sablonas – šablonų sudarymas, šablonų valdymas

HTTPSasaja – sąsajos tarp sistemos ir interneto svetainių realizacijai.

DuomenuKolekcija – veiksmams su vienu informacijos vienetu (viena eilute)

Irasas – veiksmams su vienu lauku.

Nuotrauka, Failas, Skaicius, Tekstas, Data – klasės paveldinčios klasės “Irasas” funkcionalumą, skirtos veiksmams su atitinkamo tipo duomenimis.



Pav. 8 Paketo "Informacija" klasių diagrama

Paketas “InformacijosNurodymas”

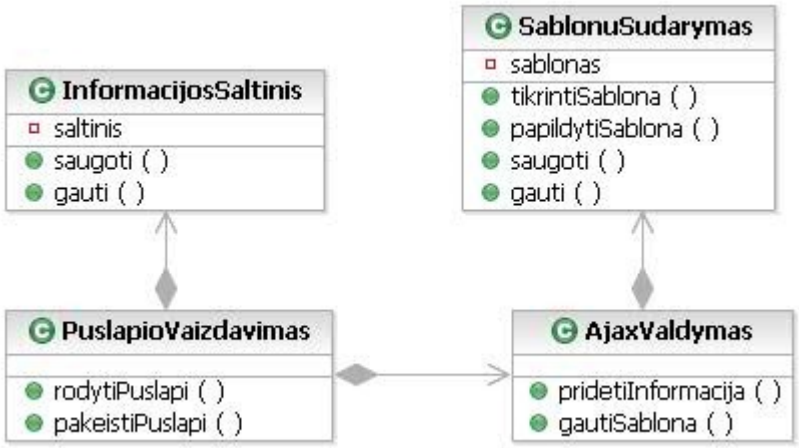
Šis paketas skirtas informacijos nurodymui. Diagramoje pateiktos pagrindinės paketo detalės. Paketo klasių paskirtis:

InformacijosSaltinis – veiksmai su informacijos šaltiniais (saugojimas, išrinkimas, tikrinimas)

PuslapioVaizdavimas – puslapio paruošimas interaktyviam šablono sudarymui

AjaxValdymas – interaktyviam sąsajai sudarant šabloną

SablonuSudarymas – funkcijos šablonui sudaryti



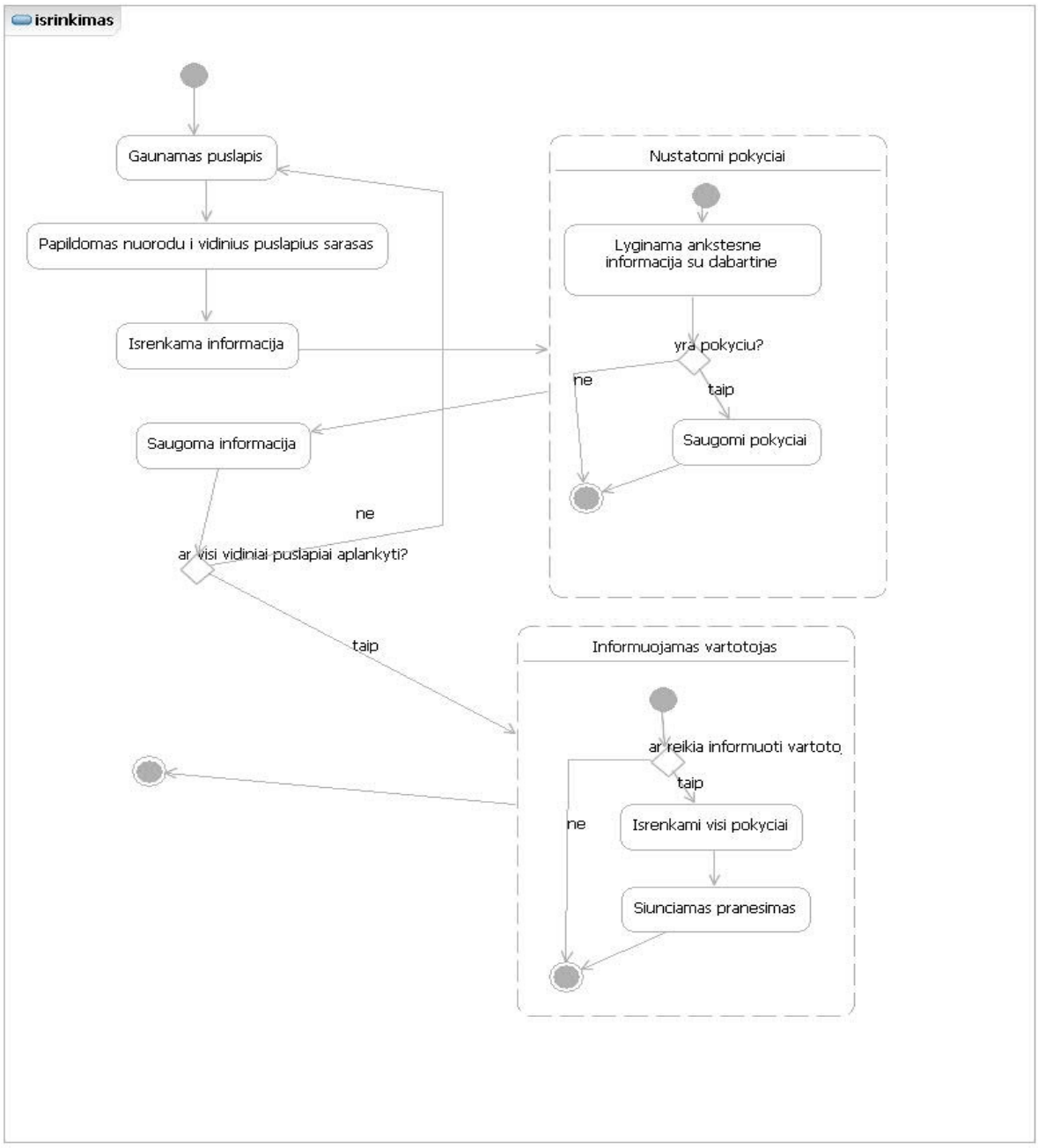
Pav. 9 Paketo "InformacijosNurodymas" klasių diagrama

3.3.2 Dinaminis sistemos vaizdas

Šiame skyriuje pateikiamos sistemos veiklos bei sekų diagramos. Šiame skyriuje pateiktose diagramose pavaizduotos tik pagrindinės sistemos funkcijos ir veiksmai.

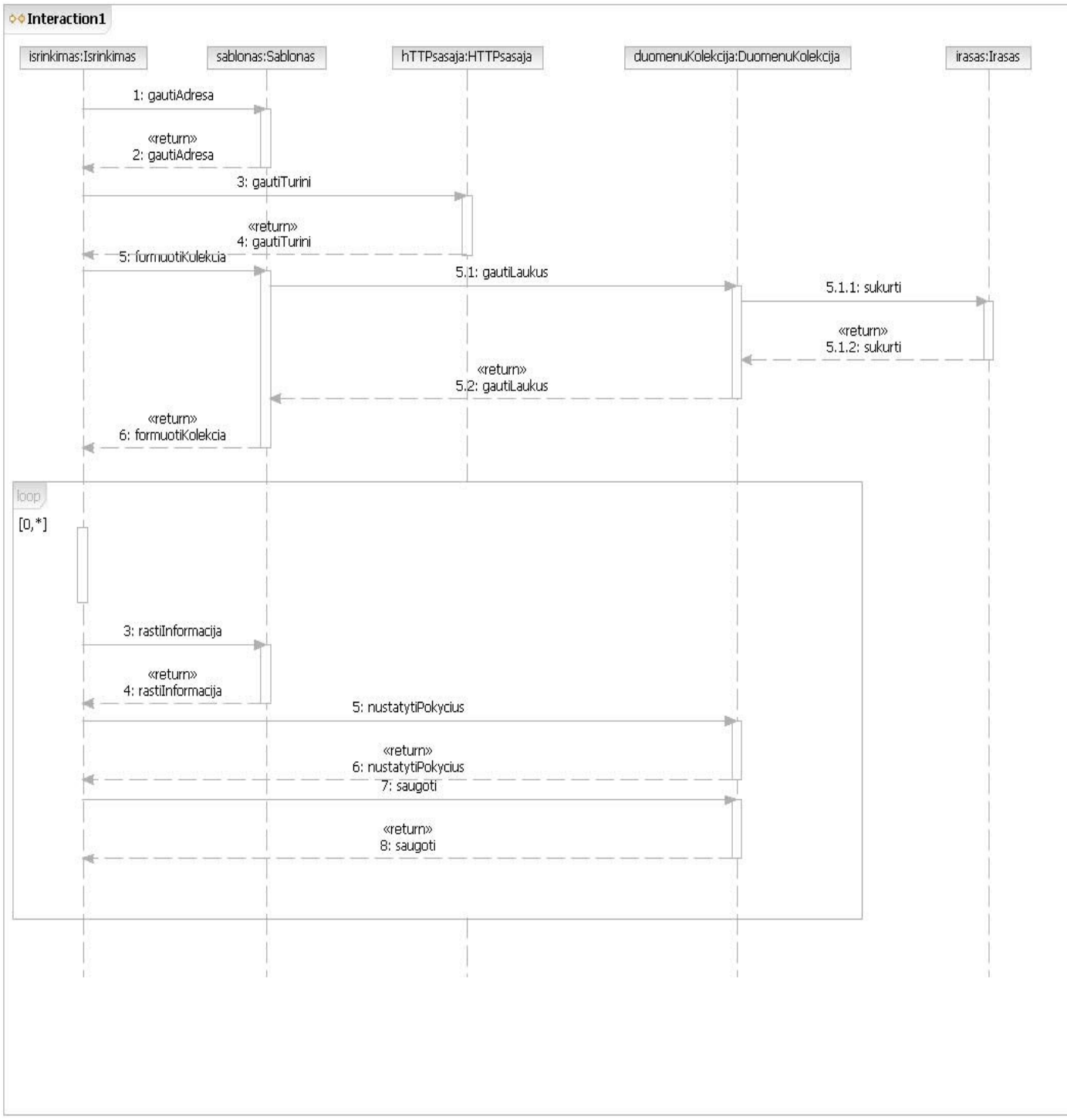
3.3.2.1 Informacijos išrinkimas

Šioje diagramoje pateiktas informacijos išrinkimo procesas. Po puslapio atsisiuntimo sudaromas puslapio struktūros medis, tuo pačiu randamos nuorodos į kitus vidinius puslapius ir pildomas nuorodų sąrašas. Sudarius puslapio struktūrą pradedama informacijos paieška. Radus informaciją, tikrinama ar tokios informacijos dar neturime, jei turime, nustatoma ar ji pakitus ir pokyčio atveju saugoma istorija. Ciklas kartojamas tol, kol nepatikriname viso sąrašo nuorodų.



Pav. 10 Informacijos išrinkimo veiklos diagrama

Sekančiame paveiksle labiau detalizuotas informacijos išrinkimo paieškos procesas. Diagrama pavaizduota išrinkimui iš vieno puslapio. Gavus puslapio turinį ir sudarius jo struktūrą, sudaroma šablono struktūra pagal kurią atliekama paieška. Su kiekviena rasta informacijos eilute atliekamas pokyčių patikrinimas ir išsaugoma informacija.



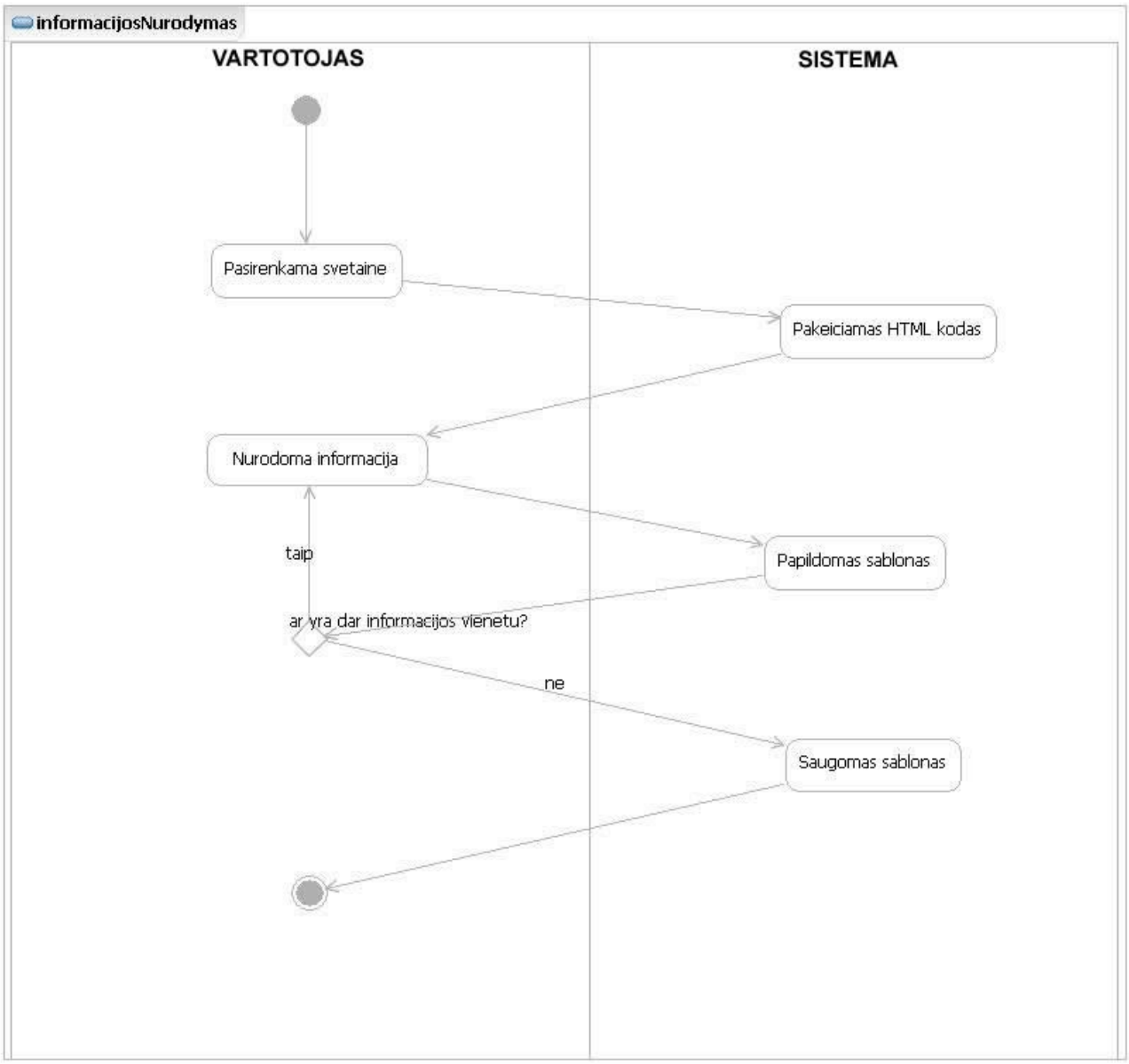
Pav. 11 Informacijos išrinkimo sekų diagrama

3.3.2.2 Informacijos nurodymas

Informacijos nurodymo etape vartotojas įveda svetainės adresą, iš kurios nori išrinkti informaciją (pav. 12). Tada sistema atsisiunčia nurodytos svetainės HTML kodą ir jį atitinkamai pakeičia. Atliekami tokie pakeitimai:

- Pašalinamas visas JavaScript kodas. Tai atliekama tam, kad vartotojas puslapį matytų tokį, kokį jį matys sistema (nepakeista JavaScript kodo). Kita priežastis yra tai, kad šablono sudarymui naudojamos bibliotekos gali būti nesuderinamos su puslapyje esančiu JavaScript kodu.
- Įdedami JavaScript failai, reikalingi šablono nurodymui.
- Įdedamas stiliaus failas, šablono nurodymo elementų išvaizdai.
- Pašalinami „onload“ atributai. Kadangi išimamas JavaScript kodas, šie atributai gali sukelti klaidas, nes gali būti kviečiama neegzistuojanti fumkcija.
- Įdedama „base“ žymė (jeigu jos nėra) su „href“ atributu, kuris yra nuoroda į tą svetainę. Kadangi puslapiuose paveiksliukų ir failų keliai dažniausiai yra relatyvus, neesant base atributui naršyklė laikys sistemos adresą pagrindiniu ir paveiksliukai bus nerasti.

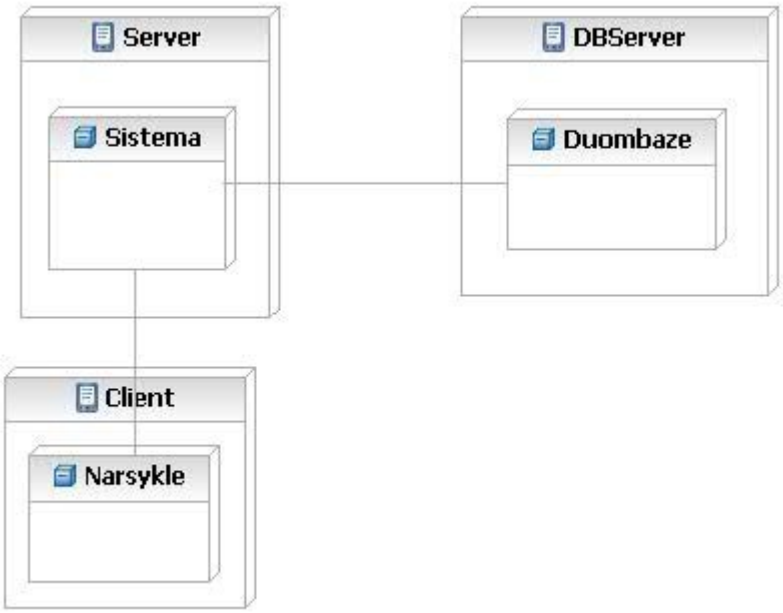
Sekančiame žingsnyje vartotojas nurodo informaciją. Užvedus pelę ant HTML elemento puslapyje ji įrėminama, taip parodant vartotojui, kuri informacija bus pažymėta. Paspaudus ant informacijos pele ir suvedus atributus pridedamas šablono laukas. Tai kartojama tol kol pažymimi visi laukai, tada šablonas išsaugomas.



Pav. 12 Informacijos nurodymo veiklos diagrama

3.3.3 Išdėstymo vaizdas

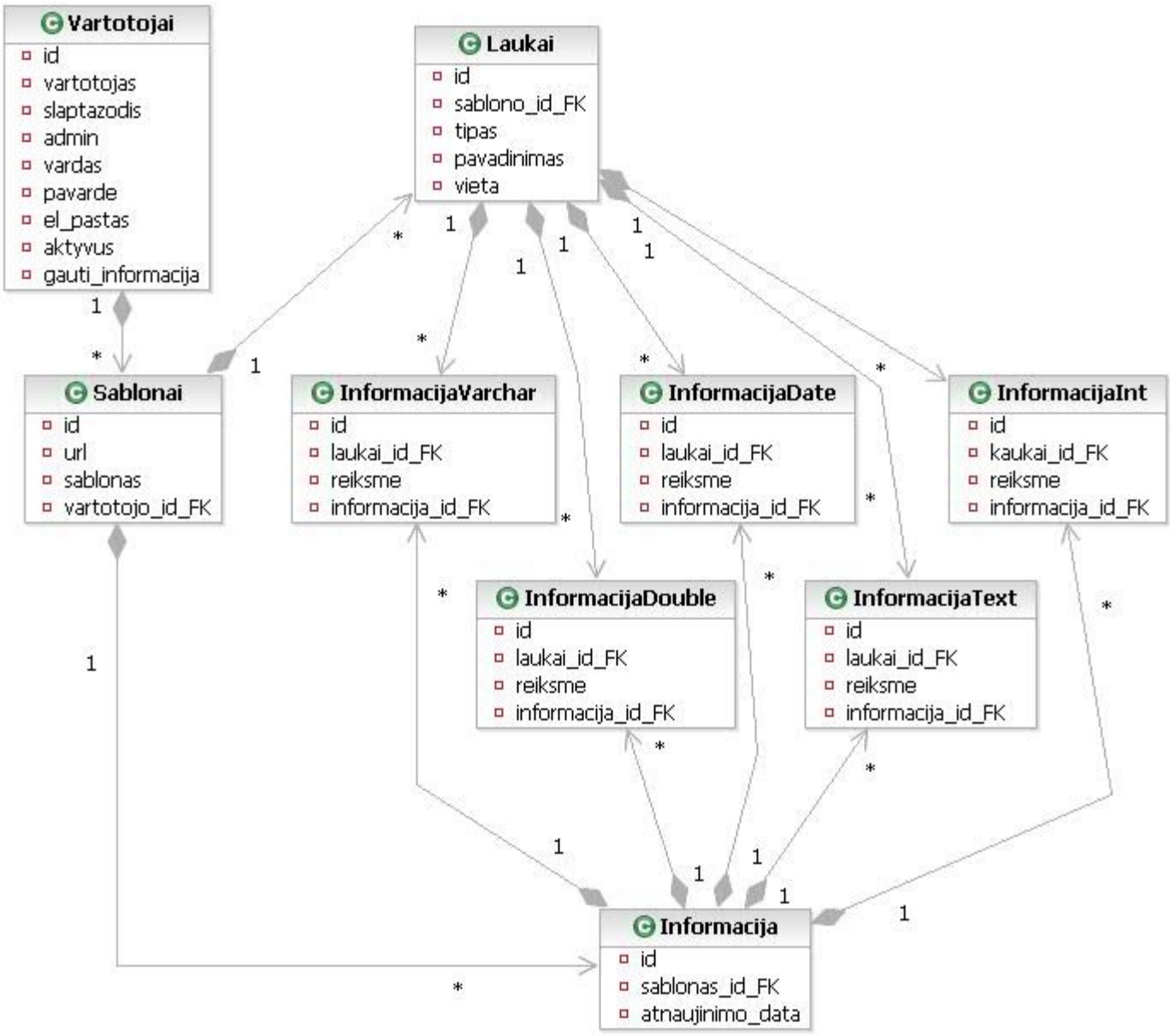
Visa sistemos logika bus patalpinta viename serveryje. Duomenų bazė esant poreikiui gali būti iškelta į atskirą serverį, tačiau dėl našumo rekomenduotina ją laikyti kartu su sistemos logika. Kliento kompiuteryje reikalinga tik naršyklė. Esant poreikiui (esant per dideliam serverio apkrautumui), padarius darbalaukio vartotojo sąsają, sistemos logika galėtų būti perkelta į vartotojo kompiuterį. Taip pat vartotojo kompiuteryje turėtų būti įdiegta DBVS arba realizuota sąsaja tarp vartotojo ir duomenų bazės esančios serveryje.



Pav. 14 Išdėstymo vaizdas

3.3.4 Duomenų vaizdas

Išrenkamos informacijos vienetas gali būti sudarytas iš įvairių ir įvairaus kiekio atributų. Pavyzdžiui jei išrenkama informacija yra darbo skelbimai, informacijos vienetas bus vienas darbo skelbimas (įrašas apie jį bus saugomas lentelėj ‘Informacija’), jo atributai galėtų būti įdėjimo data, aprašymas, kas įdėjo, užmokestis ir panašiai. Kiekvienas šių atributų būtų saugomas atitinkamoj lentelėj, priklausomai nuo jo tipo (data – lentelėj ‘InformacijaDate’, aprašymas – lentelėj ‘InformacijaText’, kas idėjo skelbimą – lentelėj ‘InformacijaVarchar’). Laukų pavadinimai (data, aprašymas…) ir tipai saugomi lentelėj Laukai. Taip pat šioje lentelėje saugoma šablono dalis (šio lauko vieta šablone), o pats šablonas saugomas lentelėj ‘Sablonai’. Duomenų bazės lentelių paskirtis detaliau aprašyta 3 lentelėje.



Pav. 15 Duomenų bazės schema

Lentelė	Aprašymas
Vartotojai	Saugomi vartotojo duomenys, jo teisės (administratorius ar paprastas vartotojas), parinktys, bei statusas
Sablonai	Saugomos svetainių nuorodos ir juose sudaryti informacijos šablonai.
Laukai	Saugomi informacijos laukai (jų pavadinimai ir tipai)
Informacija	Saugomi įrašai apie informacijos vienetus
InformacijaVarchar	Saugomi varchar tipo (iki 255 simbolių) informacijos vieneto atributai
InformacijaInt	Saugomi integer tipo informacijos vieneto atributai
InformacijaDouble	Saugomi double tipo informacijos vieneto atributai
InformacijaDate	Saugomi date tipo informacijos vieneto atributai
InformacijaText	Saugomi text tipo (ilgesni tekstai) informacijos vieneto atributai

Lentelė 3 Duomenų bazės lentelių aprašas

3.3.5 Kokybė

Sistema veikia serverio pusėj. Kliento naudojama operacinė sistema ir naršyklė neturi įtakos sistemai.

Informacijos išrinkimo komponentas veikia beveik nepriklausomai nuo kitų sistemos komponentų, todėl esant poreikiui (taupant serverio resursus), atlikus papildymus jis galėtų būti perkeltas iš serverio į kliento kompiuterį, tačiau tai apsunkintų sistemos pakeitimus.

Architektūra leidžia lengvai praplėsti išrenkamos informacijos tipus.

Atvaizdavimas atskirtas nuo logikos. Tai leidžia lengvai keisti vartotojo sąsają.

4 Struktūriniais šablonais paremto informacijos išrinkimo metodo tyrimas

Pagrindinis faktorius lemiantis informacijos išrinkimo sistemų kokybę yra tikslumas. Šis faktorius susideda iš dviejų sudedamųjų dalių, kurios yra:

- Kiek informacijos buvo surasta – šį parametrą galime apskaičiuoti procentais padalinę rastos informacijos kiekį iš visos svetainėje esančios informacijos kiekio.
- Kiek klaidingos informacijos buvo surasta – šį parametrą apskaičiuosime padalinę neteisingą informaciją iš visos surastos informacijos kiekio.

Analizuojant puslapių struktūrą, buvo padaryta išvada, kad pagrindiniai netikslumo šaltiniai yra šie:

- JavaScrip kodas
- Kintanti HTML struktūra
- HTML standartų neatitinkantys interneto puslapiai
- Informacija pasiekiama tik per paiešką (nėra nuorodų)

4.1 Netikslumai dėl JavaScript kodo, AJAX technologijų

Kai kuriais atvejais informacija gali būti gaunama arba modifikuojama JavaScript technologijų pagalba. Kadangi sistemoje nėra integruotas JavaScript interpretavimo mechanizmas, tokiais atvejais informacija yra nerandama arba nepilna (netiksli).

Kartais puslapių turinys gaunamas naudojant AJAX technologijas. Tokiais atvejais nuorodos į puslapius su šiuo turiniu gali neegzistuoti.

4.2 Netikslumai dėl kintančios puslapių HTML struktūros

Dažnai puslapių struktūros nėra visiškai identiškos. Kartais įterpiami papildomi HTML elementai (pvz. reklamos), arba kai kurie blokai tam tikruose puslapiuose nerodomi. Tarkim sudarytas šablonas:

body->div[2]->ul->li

Čia ieškome ‚body' elemente antro ‚div' elemento, tačiau jeigu į pradžią įsiterptų dar vienas ‚div' blokas, tada blokas su informacija persislinktų per vieną poziciją ir informacija nebūtų rasta. Ta pati problema iškiltų jeigu pradingtų pirmasis blokas.

Vienas iš problemos sprendimo būdų galėtų būti šablono sąlygų ignoravimas, t. y. galėtume ieškoti ne tik antrame bloke, o visuose. Šis sprendimas padidina informacijos radimo tikslumą, tačiau kyla pavojus išrinkti taip pat ir neteisingą informaciją, nes tokiu atveju išplečiamos paieškos ribos. Analizuojant įvairių puslapių struktūrą pastebėta, kad informacijos šabloną atsitiktinai atitinkantys duomenys pasitaiko pakankamai dažnai, ypač jeigu šablono sudarymo metu nurodoma mažai laukų.

Tikslesnis variantas yra atsižvelgiant į HTML žymių poziciją medyje, skaičiuotį jų nuokrypį nuo šablono. Kiekvienas medžio elementas turėtų būti papildytas dviem parametrais:

Elemento nuokrypis: $k_e = (|mPos – sPos|) / d$

d – elemento gylis medyje

mPos – pozicija medyje

sPos – pozicija pagal šabloną

Minimalus šakos nuokrypis: $k_{min} = MIN(k_{1min}+k_{1e}, k_{2min}+k_{2e}, \ldots)$

k_{1min}, k_{2min},… - elemento vaikų minimalūs nuokrypiai

Jeigu medžio dalyje nėra kelio, atitinkančio šabloną, tada : $k_{min} = \infty$

Kuo mažesnis elemento minimalus šakos nuokrypis, tuo labiau atitinka šabloną kažkuri elemento šaka.

Elementai medžio pradžioje dažniausiai formuoja puslapio struktūrą, todėl nuokrypis šioje vietoje turi būti vertinamas labiau negu nuokrypis giliai medyje. Čia neatitikimas šablonui gali reikšti, kad elementas yra kitame puslapio bloke (pvz. stulpelyje arba apačioje, o ne pagrindiniam bloke, kuriame yra informacija). Dėl šios priežasties nuokrypis dalinamas iš gylio (kuo giliau elementas, tuo mažiau reikšmingas nuokrypis).

Tikslumui padidinti į šabloną reikėtų įvesti ne tik elemento poziciją, bet ir ‚id‘ atributą (jeigu jis yra). Radus elementą su vienodu ‚id‘ atributų, galima teigti, kad šis elementas yra tikrai tas, kurio ieškoma, net jeigu jis yra ne toj pozicijoj, kurioj turėtų būti. Tokiu atveju elemento $k_e = 0$.

Apskaičiavę nuokrypius pasirenkame kelią medyje, kuris mažiausiai nukrypęs nuo šablono, tačiau jo nuokrypis neviršija tam tikros ribos. Ši riba turėtų priklausyti nuo šių parametrų:

- Vartotojo nurodytų privalomų laukų kiekis. Kuo daugiau privalomų laukų nurodyta, tuo mažesnė tikimybė, kad rasime kitą, elementą, kuris turės tokius pat laukus, tačiau nebus pageidaujamas. Pavyzdžiui jeigu vartotojas nurodė paieškai paveiksliukus, kurie yra sąraše (‚li‘ žymėse), yra didelė tikimybė, kad atsitiktinai rasime nepageidaujamus elementus su tokiu turiniu. Tačiau jei buvo nurodyta ieškoti sąrašo elementų, kuriuose yra paveiksliukas, ‚strong‘, ‚span‘ ir ‚a‘ žymės, ši tikimybė labai sumažėja.
- Šablono medžio gylis. Kuo mažesnis šablono medžio gylis, tuo mažesnė tikimybė, kad kurioj nors vietoj bus įterpta HTML žymė. Šis parametras turi žymiai mažesnę įtaką negu pirmasis.

Taigi bendru atveju šią ribą galima būtų apskaičiuoti taip:

$r = l * k_l + d * k_d$

r – maksimali riba, kurią viršijus, informacija atmetama

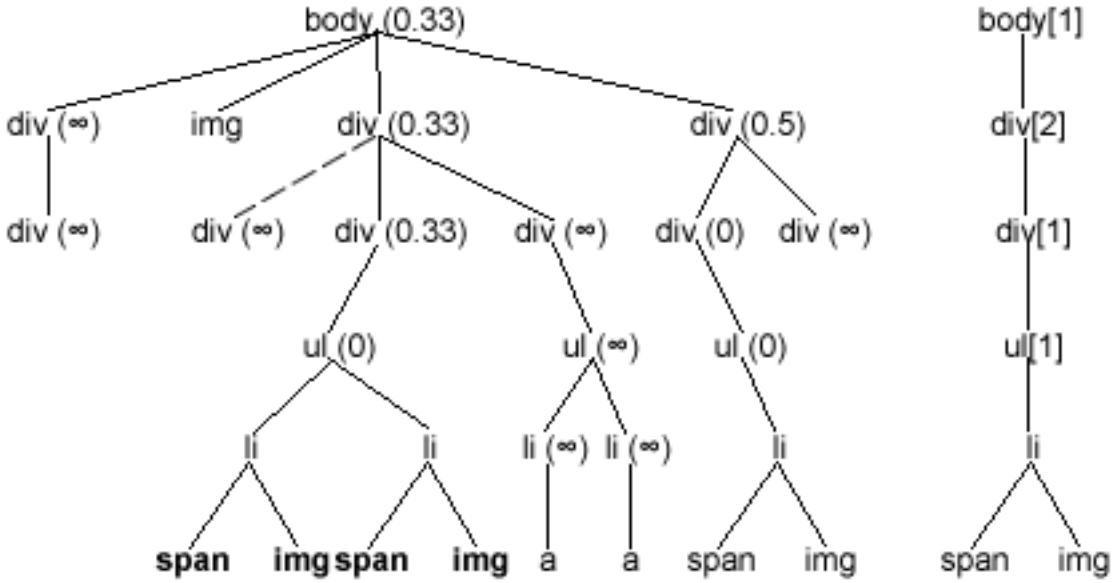
l – privalomų laukų kiekis šablone

d – šablono medžio gylis

k_l – konstanta, nusakanti laukių kiekio įtaką ribai

k_d - konstanta nusakanti šablono medžio gylio įtaką ribai

Žemiau esančiame paveikslėlyje pavaizduotas sudarytas puslapio medis ir paieškos šablonas. Skliaustuose užrašytas elementų nuokrypis ($k_{min}+k_e$), pagal kurį nustatomas informacijos tinkamumas. Punktyrine linija pavaizduota įsiterpusi HTML žymė, vartotojo nurodyta informacija pastorinta.



Pav. 16 HTML medžio šakų nuokrypio nuo šablono skaičiavimas

4.3 HTML klaidų įtaka puslapiuose

Puslapio neatikimas HTML standartams įtakos turi ne visais atvejais. Klaidas iššaukia tik grubūs standartų pažeidimai:

- Trūksta pradžios arba pabaigos HTML žymių
- Neteisingai užrašytos HTML žymės (trūksta ‚<‘ arba ‚>‘ simbolių)

Pirmuoju atveju jeigu klaida kartojasi visuose puslapiuose, ji greičiausiai neįtakos tikslumo. Šablono sudarymo metu ir paieškos metu tokios klaidos bus interpretuojamos vienodai (nors interpretacija ir nebus teisinga, tačiau šablonas atitiks struktūros dalis). Problema kyla jei klaida aptinkama ne visuose puslapiuose. Sekančiame pavyzdyje, trūksta ‚div' žymės pabaigos.

```
<body>
        <div>
                tekstas
                <img src="..." alt="" />
</body>
```

Ši klaida gali būti interpretuojama įvairiais būdais. Sistema laikytų, kad žymės pabaiga yra prieš ‚body' žymės pabaigą.

```
<body>
        <div>
                tekstas
                <img src="..." alt="" />
        </div>
</body>
```

Galimas variantas yra kad žymė turi baigtis prieš paveiksliuką ir paveiksliukas nėra ‚div' elemento viduj.

```
<body>
        <div>
                tekstas
        </div>
        <img src="..." alt="" />
</body>
```

Tokiu atveju sistema sudarytų neteisingą struktūrą, kas galėtų lemti informacijos paieškos klaidas.

Tuo atveju jei neteisingai užrašytos HTML žymės, dalis teksto gali būti palaikyta elemento atributais arba žymės dalimi, neradus žymės pradžios simbolio, žymė gali būti palaikyta tekstu. Šios klaidos pasitaiko labai retai.

4.4 Paieškos formos ir informacijos nepasiekiamumas

Kai kuriais atvejais svetainėje gali nebūti nuorodų į visus informacijos puslapius, ji pasiekiama tik užpildžius paieškos formą. Sistemoje vartotojas šių parametrų nenurodo (nurodo tik informacijos vietą). Realizuoti automatinę informacijos paiešką ir iš rezultatų išrinkti ieškomą informaciją sudėtinga, nes:

- Negalima nustatyti ar yra informacijos, kuri pasiekiama tik per paiešką
- Svetainėje gali būti daug formų, bendru atveju neaišku, kuri iš jų yra paieškos forma
- Nėra žinoma, kokiais duomenimis užpildyti paieškos laukus

Vienintelis būdas išrinkti informaciją iš tokių svetainių, būtų sistemos funkcionalumo išplėtimas. Vartotojas turėtų nurodyti paieškos formą ir surašyti visų laukų reikšmių sąrašą arba rėžius ir kitimo žingsnį. Tokiu atveju galėtų būti automatiškai atliktos visos imanomos paieškos kombinacijos ir išrinkta informacija iš rezultatų. Šis pakeitimas sistemoje nėra realizuotas.

5 Sistemų tikslumo ir funkcionalumo eksperimentinis tyrimas

5.1 Sistemų tikslumo ir jį įtakojančių faktorių eksperimentinis tyrimas

5.1.1 Tyrimo metodas ir apribojimai

Sistemos tikslumui, bei jį įtakojantiems faktoriams nustatyti pasirinkta 30 internetinių svetainių. Kad būtų galima patikrinti rezultatus, tyrimui įvestas puslapių kiekio ribojimas svetainėj. Sistema nutraukia paiešką po 15 puslapių. Šablone buvo nurodomi mažiausiai du atributai.

Laikoma, kad informacija iš svetainės išrinkta tiksliai, jeigu:

- Peržiūrėtuose puslapiuose esančios informacijos kiekis sutampa su išrinktos informacijos eilučių kiekiu

- Rasti visi nurodyti atributai
- Neišrinkta netinkama informacija

5.1.2 Tikslumo tyrimo rezultatai

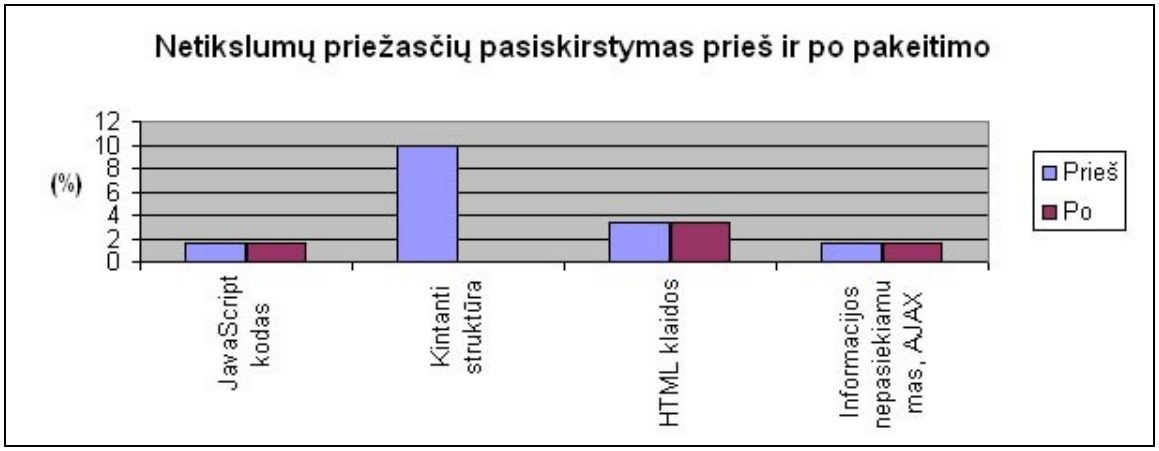
Žemiau esančioje lentelėje pateikti eksperimentinio tyrimo rezultatai. Eksperimentas buvo atliktas du kartus: prieš realizuojant pakeitimus (žr. 4.2) ir po to. Lentelėje pateiktas puslapių kiekis, kuriuose informacija buvo išrinkta netiksliai, taip pat priežastis, dėl kurios tai įvyko.

Svetainė	Klaidos ir netikslumai (psl.)		Pastaba
	Prieš pakeitimus	Po pakeitimų	
http://www.topocentras.lt	2	0	Kintanti HTML struktūra
http://www.skelbiu.lt	0	0	
http://www.darbo.lt	0	0	
http://www.skelbimai.lt	0	0	
http://www.alfa.lt	0	0	
http://www.norwegian.no	-	-	Nerastos nuorodos. Informacija pasiekiama per paiešką ir naudojant AJAX technologijas
http://www.ebay.com	0	0	
http://www.travel-oasis.lt	0	0	
http://www.ktu.lt	0	0	
http://www.islosk.lt	0	0	
http://www.darbo.lt	0	0	
http://www.parduodu-perku.lt	0	0	
http://www.pigu.lt	5	0	Kintanti HTML struktūra
http://www.elix.lt	0	0	

http://www.varle.lt	1	1	HTML klaidos
http://www.anglu-lietuviu.com	0	0	
http://www.ovao.lt	0	0	
http://www.demotyvacija.lt	0	0	
http://www.links4u.lt	0	0	
http://www.shopping.com	0	0	
http://www.cvbankas.lt	0	0	
http://www.mediashop.lt	2	0	Kintanti HTML struktūra
http://skelbiuosi.lt/	0	0	
http://www.paslaugos24.lt	0	0	
http://parduodu.net	0	0	
https://www.geradovana.lt	0	0	
http://jobs.com	0	0	
http://www.cv.lt	0	0	
http://www.kinoteka.lt/	0	0	
http://www.obuolys.lt	0	0	

Lentelė 4 Tikslumo tyrimo rezultatai

Žemiau esančioje diagramoje pavaizduota šių keturių faktorių įtaka informacijos išrinkimo tikslumui. Matome, kad beveik 10% netiksliai išrinktos (dažniausiai nerastos) informacijos lėmė kintanti HTML struktūra. Po pakeitimų realizacijos šios klaidos buvo eliminuotos. JavaScript kodas, AJAX technologijos ir informacijos nepasiekiamumas sudaro sumoj 3.2% netikslumų. Dėl šių klaidų atsiradę netikslumai dažniausiai pastebimi iškart, nes praktiškai nerandama jokios informacijos. Dėl kintančios struktūros arba HTML klaidų puslapyje atsiradę netikslumai, esant dideliam informacijos kiekiui pastebimi sunkiau, kadangi tai dažniausiai įtakoja tik mažą dalį svetainės puslapių.



Pav. 17 Galimų problemų įtaka informacijos išrinkimo tikslumui

5.1.3 Tikslumo tyrimo išvados

Atlikus tyrima ir išanalizavus rezultatus nustatyta, kad informacijos išrinkimo tikslumą labiausiai lemia HTML struktūros pokyčiai. Tai yra jei puslapyje įterpiamas papildomas elementas, gali kisti informacijos vieta HTML medyje, todėl būtina paieškos algoritmą padaryti atsparesnį pokyčiams.

Realizavus pakeitimus ir atlikus pakartojus eksperimentą, rezultatai parodė 10% didesnį tikslumą.

5.2 Sistemų funkcionalumo tyrimas

Šiame skyriuje bus palyginamo informacijos išrinkimo sistemos funkcionalumo požiūriu, bei analizuojami informacijos išrinkimo metodai ir jų techninės charakteristikos.

Funkcionalumo palyginimui pasirinktos šios plačiau naudojamos sistemos:

- Yahoo pipes
- Mozenda
- Web Info Extractor

Žemiau esančioje lentelėje pateikti tyrimo rezultatai

	Sukurta sistema	Yahoo pipes	Mozenda	Web Info Extractor
Sistemos tipas	internetinė	internetinė	Darbastalio sistema	Darbastalio sistema
Šablonų sudarymas	Šablonų sudarymas grafinis, puslapyje pele paspaudžiant ant informacijos.	Sistemos sudarymas grafinis, tačiau labai sudėtingas. Informacijos vieta nurodoma įrašant rėžius (nuo kur iki kur išrinkti informaciją)	Grafinis šablonų sudarymas	Šablonų sudarymas grafinis, informacijos vieta nurodoma pele
Rezultatų peržiūra sistemoj	Galimybė filtruoti ir rikiuoti, rodomos nuorodos į puslapį, istorija.	Filtrai ir rikiavimas nurodomi tik sistemos sudarymo metu	Galimybė tik rikiuoti	Galimybė rikiuoti informaciją, paieška nerealizuota
Paieška visoj svetainėj	Nurodžius šabloną automatiškai randamos nuorodos į kitus puslapius, paieška vyksta visoj svetainėj. Yra galimybė riboti paiešką	Plačios galimybės sudarinėjant nuorodas, taip pat galima paieška visoj svetainėj	Realizuota dalinai. Reikia nurodyti visas nuorodas arba nuorodos tekstą.	Sudarant šabloną reikia nurodyti nuorodų vietą, kuriuose taip pat norėsim ieškoti. Problemų kylą dėl puslapiavimo kai rodomi ne visi puslapiai
Svetainių apjungimas	Galima apjungti informaciją, jei sutampa laukai ir jų tipai	Galima apjungti rastą informaciją iš daugelio šaltinių	Nėra	Nėra
Rezultatų modifikavimas išrinkimo metu	Nėra	Plačios modifikavimo galimybės, galimos įvairios filtrų kombinacijos	Nėra	Modifikavimas naudojant reguliarias išraiškas
Rekursyvi paieška	Nėra	Realizuojama labai sudėtingai	Yra	Yra
Išrenkamų duomenų formatai	Tekstas ir paveiksliukai (saugomos tik paveiksliuku nuorodos)	Tekstas	Tekstas, paveiksliukų nuorodos	Tekstas, paveiksliukai, failai, html žymių atributų reikšmės
Informacijos istorija, pokyčiai	Pagal unikalius laukus nustatomi pokyčiai, saugoma istorija	Nėra	Nėra	Pakitusi informacija saugoma kaip naujas įrašas.
Eksportas	CSV, XML	CSV, JSON, RSS, PHP (serializuotas masyvas)	XML	CSV, tekstinis failas, HTML, DB failai

Lentelė 5 Sistemų funkcionalumo įvertinimas

Apibendrinant, visus šių sistemų tyrimo duomenis galima sugrupuoti į keturias pagrindines grupes:

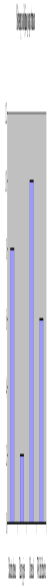
- Sistemos valdymo paprastumas
- Informacijos išrinkimo galimybės
- Informacijos peržiūra sistemoje
- Veikimo aplinka ir sistemos kaina

Kiekvienai grupei buvo nustatyti vertinimo kriterijai. Jei sistema pilnai tenkina kriterijų, skiriami 2 balai. Jei kriterijus tenkinamas iš dalies, skiriamas 1 balas, jei netenkinamas – 0.

5.2.1 Sistemos valdymo paprastumas

Žemiau pateiktoje diagramoje pavaizduotas sistemų valdymo paprastumo įvertinimas. Įvertinimo metrikos buvo pasirinktos pagal vartotojo atliekamus žingsnius šablono sudarymo metu.

- Galimybės interaktyviai nurodyti reikalingą informaciją ieškomam puslapy
- Ar vartotojas privalo turėti HTML ar kitų specifinių žinių
- Kitų svetainės puslapių nurodymas
- Išrenkamų duomenų atvaizdavimas šablono sudarymo metu (ar vartotojas mato, kokie duomenys bus išrinkti?)
- Šablono testavimo ir koregavimo galimybės



Pav. 18 Sistemų naudojimo paprastumo įvertinimas

5.2.2 Informacijos išrinkimo galimybės

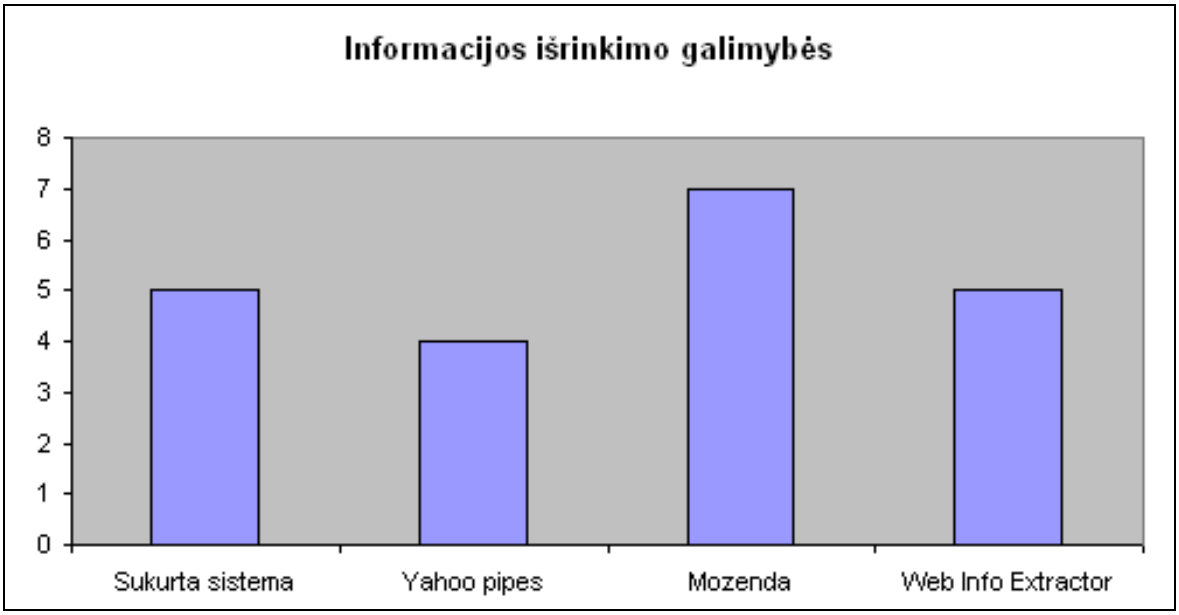
Automatinis informacijos išrinkimas įgauna prasmę jei sistema išrenka informacija ne iš vieno puslapio, o iš visos svetainės arba iš visų nurodytų puslapių, todėl svarbiausia paieškos galimybių vertinimo metrika yra automatinis vidinių puslapių radimas arba bent galimybė nurodyti, kurioje vietoje yra nuorodos į kitus puslapius. Kadangi svetainėje puslapių gali būti labia daug, antrasis variantas yra žymiai ekfektyvesnis laiko atžvilgiu, tačiau tirtose sistemose buvo susidurta tam tikromis problemomis.

Taip pat svarbi funkcija yra rekursyvi paieška, kuri leistų ieškoti informacijos išsidėsčiusios per kelis puslapius.

Kiti svarbūs vertinimo kriterijai yra:

- Pokyčių nustatymas ir dublikuotos informacijos eliminavimas
- Rezultatų transformacija. Pavyzdžiui jeigu viename HTML lauke turėtume miesto pavadinimą ir prieš jį einantį žodį “miestas: “, tada rezultatuose šis žodis atsikartotų visame duomenų stulpelyje, todėl būtu patogu turėti galimybę šį žodį išfiltruoti.
- Papildomos funkcijos (pavienių elementų prijungimas prie sąrašo, pelės paspaudimų imitacija, laukų pildymas).

Pav. 18 pavaizduoti informacijos išrinkimo galimybių tyrimo rezultatai. “Yahoo Pipes“ yra labai plačių galimybių sistema. Ši sistema labiau koncentruota į informacijos išrinkimą iš labiau struktūrizuotų formatų (pvz. RSS). Nors ir kitomis savo galimybėmis ji pranoksta kitas tirtas sistemas, tačiau tyrimo metu buvo atsižvelgta tik į struktūrizuotos informacijos paieškos galimybes HTML puslapiuose.



Pav. 19 Informacijos išrinkimo galimybių įvertinimas

5.2.3 Rezultatų peržiūra

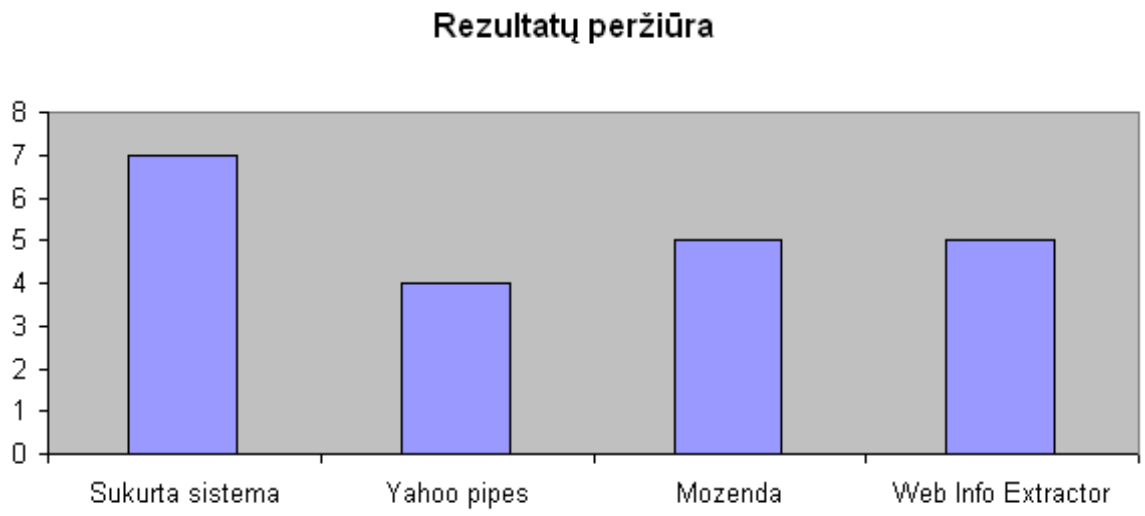
Šiame skyriuje tiriamas išrinktos informacijos pateikimas sistemoje. Kadangi išrinktos informacijos kiekis gali būti labai didelis, pagrindinis vertinimo kriterijus yra filtravimo galimybės. Filtrą iš dalies atstoja eksporto galimybė, nes išeksportavimus duomenis juos filtruoti galima panaudojant kitus įrankius, tačiau filtravimo galimybės toje pačioje sistemoje yra žymiai patogesnės.

Be filtravimo ir eksporto dar buvo atsižvelgiama į šiuos kriterijus:

- Rikiavimas
- Paveikslėlių peržiūra (dažnai rodomos tik nuorodos į jios)
- Papildomos informacijos pateikimas (nuoroda į puslapį, kuriame buvo rasta informacija, išrinkimo data, numeris)

„Yahoo Pipes“ turi plačias filtravimo, bei kitas duomenų manipuliavimo galimybes, tačiau jos taikomos ne peržiūros metu, o sudarinėjant informacijos išrinkimo

mechanizmą. Žemiau pateiktoje diagramoje pavaizduotos rezultatų peržiūros galimybės sistemose.



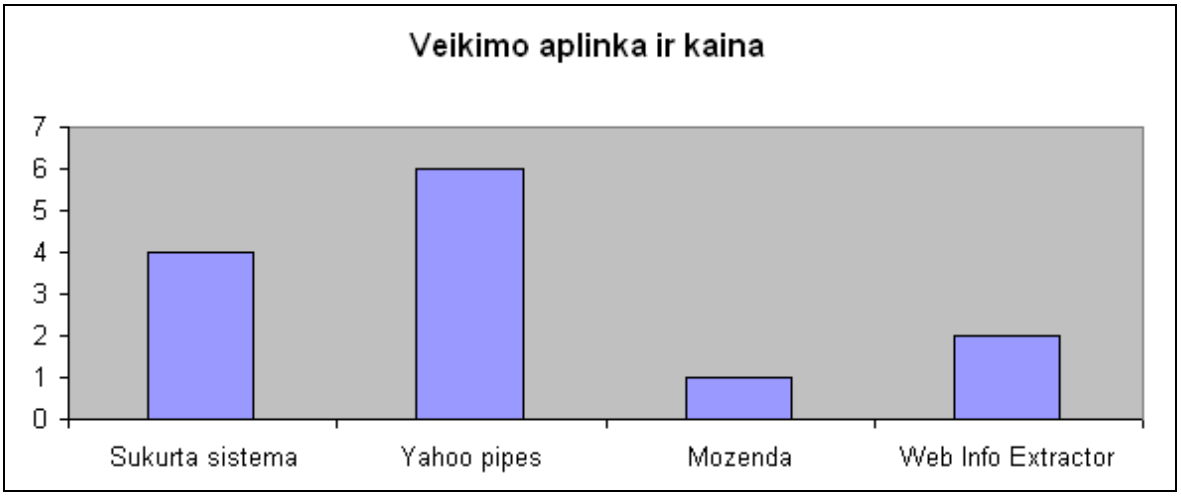
Pav. 20 Rezultatų peržiūros galimybių sistemoje įvertinimas

5.2.4 Veikimo aplinka ir sistemos kaina

Svarbūs sistemos atributai yra jos veikimo aplinka ir kaina. “Mozenda” ir “Web Info Extractor” yra darbastalio sistemos ir veikia tik Windows operacinėje sistemoje. Likusios dvi yra internetinės ir nepriklausomos nuo kliento operacinės sistemos. “Yahoo Pipes” taip pat veikia ir su senomis naršyklėmis (pvz. IE6). Iš tirtų sistemų “Mozenda” yra mokama, tačiau suteikiamas mėnesio bandomasis laikotarpis. Sekančioje diagramoje pavaizduotas apibendrintas veikimo aplinkos ir kainos įvertinimas.

Apibendrintai šio tyrimo metrikos yra:

- Nepriklausoma nuo OS
- Nepriklausoma nuo naršyklės
- Nemokama



Pav. 21 Sistemų veikimo aplinkos ir kainos įvertinimas

5.2.5 Sistemų funkcionalumo tyrimo išvados

Šiame skyriuje buvo tirtas sistemų funkcionalumas iš vartotojo perspektyvos. Buvo išskirtos keturios pagrindinės tyrimo dalys, kurių pirmos trys yra vartotojo atliekami žingsniai pradedant šablono sudarymu ir baigiant išrinktų rezultatų peržiūra. "Yahoo pipes" yra universalaus pobūdžio sistema, bet buvo tirtas tik funkcionalumas susijęs su struktūrizuotos informacijos išrinkimu.

"Mozenda" turi patogią vartotojo sąsają ir daug papildomų naudingų informacijos išrinkimo funkcijų, tačiau taip pat kaip ir kitose dviejose sistemose, čia nenustatomi informacijos pokyčiai išrinkinėjant informaciją antrą kartą, taip pat nėra galimybės filtruoti išrinktų rezultatų, kas kai kuriais atvejais labia naudinga vartotojui.

Darbo metu sukurtoj sistemoj yra galimybė stebėti pokyčius, tačiau trūksta kai kurių informacijos išrinkimo funkcijų (rekursyvios paieškos, rezultatų transformavimo). Taip pat čia realizuoti rezultatų filtrai.

"Web Info Extractor" teikia įvairias rezultatų transformavimo galimybes, tačiau jų naudojimas reikalauja specifinių žinių.

Nors ir "Yahoo pipes" funkcionalumas yra labia platus, didžiajai grupei vartotoju šios funkcijos dėl didelio sistemos sudėtingumo gali likti nepanaudojamas. Su šia sistema iš principo galima išrinkinėti struktūrizuotą informaciją, tačiau šiam tikslui ji nėra labia tinkama.

6 Išvados

Automatinis informacijos išrinkimas iš interneto svetainių yra aktualus, nes tai leidžia atmesti perteklinę informaciją, manipulioti informacija, taikyti sudėtingus filtrus, stebėti informacijos kaitą, bei peržiūrėti informaciją patogiu būdu.

Darbo metu buvo analizuojami informacijos išrinkimo metodai, jų veikimo principai ir ypatumai. Darbe išanalizuoti šie metodai (žr. 2.2 skyrių):

- Tekstinių šablonų metodai
- Struktūrinių šablonų metodai
- Puslapio išvaizda paremti metodai
- Kalbos analizės metodai

Išanalizavus informacijos išrinkimo metodus buvo sukurta informaciją išrenkanti sistema, paremta struktūriniais šablonais ir atliktas jos tyrimas ir tobulinimas siekiant padidinti tikslumą.

Tyrimo metu buvo analizuoti struktūriniais šablonais paremtos informacijos išrinkimo sistemos tikslumą įtakojantys faktoriai (žr. 4 skyrių). Buvo analizuotos problemos ir galimi sprendimo būdai. Nustatyta, kad didžiausią įtaką turi:

- HTML struktūros pokyčiai
- HTML klaidos
- JavaScript kodas / AJAX technologijos
- Informacijos neprieinamumas per nuorodas

Eksperimentinis tyrimas parodė, kad didžiausią įtaką tikslumui turi HTML struktūros pokyčiai (žr. 5.1). Ši problema buvo išanalizuota, pasiūlytas ir realizuotas sistemos patobulinimas. Eksperimento rezultatų analizė parodė teigiamą patobulinimo įtaką sistemos tikslumui (žr 5.1.2 skyrių).

Remiantis tyrimo rezultatais bei literatūros analize, nustatyta, kad didžiausias tikslumas pasiekiamas naudojant struktūriniais šablonais paremtą informacijos išrinkimo metodą. Tekstinių metodų tikslumas labai priklauso nuo šablonų sudarymo metodų ir kai kuriais atvejais gali būti labai didelis. Puslapio išvaizda bei kalbos analize paremti metodai yra mažiau tikslesni, nes yra žymiai daugiau faktorių, įtakojančių jų tikslumą. Nors puslapio išvaizda paremtų metodų tikslumas nėra mažas, šie metodai mažiau tinkami struktūrizuotai informacijai išrinkti. Kalbos analizės metodų tikslumas yra vienas

iš mažiausiu. Šiem metodams labai didele įtaką turi rašybos, skirybos klaidos, perkeltine prasme panaudoti žodžiai, taip pat žodyno, ryšių tarp žodžių, sąrašų dydis, bei daugelis kitų dalykų.

Darbo metu buvo tirtos analogiškos sistemos funkcionalumo požiūriu („Yahoo pipes“, „Mozenda“ ir „Web Info Extractor“) (žr. 5.2 skyrių). „Yahoo Pipes“ nėra specializuota struktūrizuotos informacijos išrinkimui. Ji turi labai plačias galimybes, tačiau yra labai sudėtinga. Eksperimentinio tyrimo rezultatai parodė, kad „Mozenda“ turi pakankamai daug informacijos išrinkimo galimybių ir yra labai patogi naudotis, bet yra mokama. Šioje ir kitose sistemose trūksta informacijos pokyčių nustatymo galimybės, bei rezultatų filtravimo.

7 Literatūra

[1] Chia-Hui Chang; Mohammed Kayed; Moheb Ramzy Girgis; Khaled Shaalan, "A Survey of Web Information Extraction Systems", [Žiūrėta 2009 10 20], prieiga per internetą
http://www.csie.ncu.edu.tw/~chia/pub/iesurvey2006.pdf

[2] Oren Etzioni; Michael Cafarella; Doug Downey; Stanley Kok; AnaMaria Popescu; Tal Shaked; Stephen Soderland; Daniel S. Weld, "WebScale Information Extraction in KnowItAll", [Žiūrėta 2009 10 21], prieiga per internetą
http://turing.cs.washington.edu/papers/www-paper.pdf

[3] Mahmoud Shaker; Hamidah Ibrahim; Aida Mustapha; Lili Nurliyana Abdullah, "Information Extraction from Hypertext Mark-Up Language Web Pages", [Žiūrėta 2009 11 13] prieiga per internetą
http://www.scipub.org/fulltext/jcs/jcs58596-607.pdf

[4] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu Tal Shaked, Stephen Soderland, Daniel S.Weld, and Alexander Yates, "Methods for Domain-Independent Information Extraction from the Web", prieiga per internetą
http://www.cs.washington.edu/homes/soderlan/AAAI04.pdf

[5] Sergey Brin, "Extracting Patterns and Relations from the World Wide Web", [Žiūrėta 2009 10 20], prieiga per internetą
http://bolek.ii.pw.edu.pl/~gawrysia/WEDT/brin.pdf

[6] Chia-Hui Chang, " IEPAD: Information Extraction Based on Pattern Discovery", [žiūrėta 2011 04 26], prieiga per internetą
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.1532&rep=rep1&type=pdf

[7] Claire Cardie, „Empirical Methods in Information Extraction", [žiūrėta 2011 04 20] prieiga per internetą
http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=D7A02CD401DD28A69C724030960A6445?doi=10.1.1.20.8120&rep=rep1&type=pdf

[8] Doug Downey; Oren Etzioni; Stephen Soderland; Daniel S. Weld, "Learning Text Patterns for Web Information Extraction and Assessment", [Žiūrėta 2009 10 21], prieiga per internetą
http://www.cs.washington.edu/homes/weld/papers/DowneyATEM04.pdf

[9] Maxime Crochemore; Wojciech Ryller, "Text Algorithms", psl 152-170

[10] Dan Gusfield, "Algorithms on strings, trees, and sequences– computer science and computational biology", psl 64-72

[11] Fabio Ciravegna, "(LP)2 an Adaptive Algorithm for Information Extraction from Web-related Texts", [Žiūrėta 2009 11 13], prieiga per internetą

http://www.dcs.shef.ac.uk/~fabio/paperi/Atem01.pdf

[12] Justin Park; Denilson Barbosa, “Adaptive Record Extraction From Web Pages”, [Žiūrėta 2009 11 13], prieiga per internetą
http://www2007.org/posters/poster1012.pdf

[13] Steve O’Neil, “HTML Document Structure”, [Žiūrėta 2009 10 20], prieiga per internetą
http://faculty.ksu.edu.sa/mmowafy/Documents/CS236/Tutorials/2-Document_Structure.pdf

[14] Man I Lam1; Zhiguo Gong1; Maybin Muyeba, “A method for Web Information Extraction”, [Žiūrėta 2009 10 21], prieiga per internetą
http://www.sftw.umac.mo/~fstzgg/apweb2008.pdf

[15] Robert Baumgartner, Sergio Flesca, Georg Gottlob, “Visual Web Information Extraction with Lixto“, [žiūrėta 2011 04 22], prieiga per internetą
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.6892&rep=rep1&type=pdf

[16] Georg Gottlob; Christoph Koch, “Logic-based Web Information Extraction”, [Žiūrėta 2009 11 13], prieiga per internetą
http://www.cs.toronto.edu/~libkin/dbtheory/georg.pdf

[17] Shian-Hua Lin, Jan-Ming Ho, “ Discovering Informative Content Blocks from Web Documents“, [žiūrėta 2011 04 26], prieiga per internetą
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.7133&rep=rep1&type=pdf

[18] Deng Cai1, Shipeng Yu, Wei-Ying Ma, “Extracting Content Structure for Web Pages based on Visual Representation“, [žiūrėta 2011 04 26], prieiga per internetą
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.3169&rep=rep1&type=pdf

[19] Theodore W. Hong; Keith L. Clark, “Using Grammatical Inference to Automate Information Extraction from the Web”, [Žiūrėta 2009 11 15], prieiga per internetą
http://www.cl.cam.ac.uk/~twh25/academic/papers/ecml.pdf

[20] Fabio Ciravegna and Yorick Wilks, “Designing Adaptive Information Extraction for the Semantic Web in Amilcare”, [Žiūrėta 2009 11 13], prieiga per internetą
http://www.dcs.shef.ac.uk/~fabio/paperi/AmilcareAnnotation.pdf

[21] Georgios Petasis; Vangelis Karkaletsis; Constantine D. Spyropoulos, “Cross-lingual Information Extraction from Web pages”, [Žiūrėta 2009 11 15], prieiga per internetą
http://www.ellogon.org/2004_site/documents/RANLP-CameraReady.pdf

[22] Ralph Grishman, “Information Extraction: Techniques and Challenges “, [žiūrėta 2011 04 26], prieiga per internetą
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.6923&rep=rep1&type=pdf

8 Terminų ir santrumpų žodynas

Reguliari išraiška – teksto šabloną atitinkanti taisyklė, užrašyta kaip teksto eilutė, riamiantis tam tikrom taisyklėm.

Kalbos dalys - žodžių klasės, skiriamos pagal reikšmės, sintaksinių ryšių ir morfologinių požymių bendrumą (daiktavardis, veiksmažodis, būdvardis, ...).

Sakinio dalys - žodžiai, sakinyje atsakantys į tam tikrus klausimus (veiksnys – kas? tarinys – ka veikia? papildinys – ką? kam? ko? vietos aplinkybė – kur? laiko aplinkybė – kada?)

HTML - tai standartizuota kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete.

HTML žymė – pagrindinis HTML kalbos vienetas.

JavaScript – programavimo kalba, dažniausiai naudojama internetinių puslapių interaktyvumo realizacijai.

AJAX – programavimo metodai internetinių puslapių interaktyvumui padidinti (asinchroninis JavaScript ir XML programavimas)

XML - bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.

RSS - XML failu formatu šeima internetiniam duomenu rinkimui iš naujienu portalu ir tinklarašciu.

CSV – duomenų saugojimo standartas,

JSON – JavaScript objektų notacija