KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMATIKA

KAROLIS ČERNIAUSKAS

VIZUALINIS ŠABLONO SUDARYMAS INFORMACIJOS GAVIMUI IŠ INTERNETINIŲ ŠALTINIŲ

Magistro darbas

Vadovas

prof. R. Butleris

Konsultantas

dokt. Edvinas Šinkevičius

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMATIKA

KAROLIS ČERNIAUSKAS

VIZUALINIS ŠABLONO SUDARYMAS INFORMACIJOS GAVIMUI IŠ INTERNETINIŲ ŠALTINIŲ

Magistro darbas

Vadovas

prof. R. Butleris

Parašas: Data:

Konsultantas

dokt. Edvinas Šinkevičius

Recenzentas

dr. A.Riškus

Parašas: Data:

Studentas

K.Černiauskas

Parašas: Data:

KAUNAS, 2013

AUTORIŲ GARANTINIS RAŠTAS
DĖL PATEIKIAMO KŪRINIO

20.. - ..... - ..... d.
Kaunas

Autoriai,_____
(vardas, pavardė)

_____,

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūrinys)_____
(kūrinio pavadinimas)
_____

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

_____	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)
_____	_____
(vardas, pavardė)	(parašas)

TURINYS

Summary

The amount of data in the Web is increasing exponentially. Humans by themselves are not able to cope with the increasing demand of various data every day. In our every day lives this manifests in struggle to schedule everything as tightly as possible and absorb as much information as possible in as little time as possible. To aid in the increasing demand of data, computers have been used almost as much increasingly well. However, current state of the Web (which provides us with almost all everyday data) is not perfect for usage by automated computers. The Internet mostly provides us with data in text or image format that computers can merely display, but they can not comprehend the meaning of it.

This situation calls for a solution which involves semanticizing current text-only Internet, i.e. giving meaning to the text, which computers could understand and manipulate as well as humans do now. This paper investigates the means to approach the lack of machine understandable text on the Web and suggests a tool which can aid humans in semanticizing it.

1. ĮVADAS

Duomenų gausa internete didėja eksponentiškai. Dėl esančios ir vis augančios duomenų gausos internete pasaulinis kompiuterių tinklas tapo vertingiausiu žinių šaltiniu. Deja, informacija internete nėra pateikta mašinoms suprantama kalba. Žinios arba informacija internete yra pateikta grynų neapdorotų duomenų pavidalu. Šie duomenys yra sutverti žmonėms skaityti ir suprasti, o ne kompiuterinėms programoms prasmingai jais manipuliuoti. Norint tuos duomenis apdoroti ir paversti naudinga informacija, reikia vis daugiau išteklių ir darbo jėgos. Tai daryti rankiniu būdu yra nepraktiška, todėl žiniatinklio technologijos yra vystomos automatizavimo linkme. Dėl šių technologijų tampa vis lengviau itin greitai surasti ieškomą informaciją žiniatinklyje. Paieškos sistemos tobulėja – dabar jos geba nuspėti vartotojo užklausos kontekstą ir pateikti tikslų atsakymą. Tačiau ne visa informacija žiniatinklio tekstuose yra vienprasmiška ir suprantama paieškos programai. Ši problema sprendžiama semantizuojant tinklapių duomenis, tai yra pildant tinklapius papildoma informacija apie juose vaizduojamus duomenis. Šiame darbe yra analizuojami metodai padedantys semantizuoti dabartinį žiniatinklį. Analizuojami šiuo metu įgyvendinta programinė įranga, kuri vienu ar kitu būdu eksploatuoja semantizavimo metodus. Darbe tiriamos technologijos, kurių pagalba galima realizuoti patobulintą žiniatinklio semantizavimo įrankį. Paskutinėje darbo dalyje pateikiamas praktinis darbo projektas ir jo realizacijos detalės.

1.1. Darbo tikslas

Šiame darbe atliekamas galimų informacijos gavybos iš interneto šaltinių būdų tyrimas. Siekiama sukurti įrankį padėsiantį semantizuoti interneto šaltinius ir priartinti juos prie trečiosios kartos žiniatinkliui būdingos struktūros (WEB 3.0).

Darbo uždaviniai:

1. Ištirti metodus, kuriais yra siekiama semantizuoti esamus informacijos šaltinius internete.
2. Suprojektuoti įrankį, kurio pagalba daug techninių ir informacinių technologijų žinių neturintis asmuo galėtų prisidėti prie informacijos semantizavimo.
3. Įgyvendinti minėto įrankio prototipą, kuris įgalintų vartotoją kurti tinklapių analizės šablonus vizualiniu būdu (naudojantis grafine sąsaja).

2. TINKLAPIŲ SEMANTIZAVIMO METODŲ ANALIZĖ

Semantinis tinklas yra duomenų tinklas. Dabartiniame internete kol kas daugiausia informacijos yra pateikta teksto pavidalu įvairiuose dokumentuose, o kiekviena duomenis naudojanti programinė įranga laiko juos sau paprastai nesidalindama jais su kitokiu tikslu dirbančiomis programomis. Semantinio tinklo sąjūdžio vizija yra išplėsti dabartinio interneto principus nuo dokumentų iki žinių (duomenų). Duomenys turėtų būti prieinami naudojantis bendra interneto architektūra, pavyzdžiui, URI; duomenys turėtų sietis tarpusavyje taip, kaip dabar tarpusavyje siejasi dokumentai arba jų dalys. Į šią viziją taip pat įeina kūrimas bendro pagrindo, kuris leistų duomenimis dalintis ir būti naudojamiems tarp skirtingos programinės įrangos ar skirtingų žmonių bendruomenių. Turi būti galima apdoroti duomenis žmonėms, o taip pat kompiuteriams įvairiems tikslams, įskaitant atskleidžiant naujus ryšius tarp įvairių duomenų fragmentų.[16]

Norint pasiekti aukščiau išdėstytus tikslus svarbiausia yra apibrėžti sąryšius tarp duomenų (resursų) internete. Tai yra panašu į nuorodas naudojamas interneto tinklapiuose. Šios nuorodos susieja du dokumentus tarpusavyje. Svarbiausias skirtumas tarp šių nuorodų ir sąryšių tarp duomenų yra tas, kad semantiniame tinkle ši sąsaja gali būti sudaroma tarp bet kokių dviejų resursų; nėra sąvokos „dabartinis“ puslapis. Kitas svarbus skirtumas yra tas, kad semantiniame

tinkle pati nuoroda turi savo pavadinimą, kurio reikšmė yra nusakyta iš anksto. Tuo tarpu tradiciniame internete nuorodos (ryšiai) tarp dokumentų pavadinimų neturi ir jų prasmė yra vartotojo suprantama pagal kontekstą. Šių ryšių aprašymai leidžia automatizuotus mainus duomenimis. RDF (angl. „Resource Description Framework“ - liet. „resursų aprašymo pagrindas“) yra viena fundamentalių semantinio tinklo sudedamųjų dalių, kuri pateikia formalų duomenų mainų apibrėžimą. Papildomos sudedamosios dalys remiasi RDF pagrindu; keli kitų semantinio tinklo sudedamųjų dalių pavyzdžiai yra:

- Įrankiai. skirti užklausti informaciją aprašytą minėtais ryšiais (pvz., SPARQL).
- Įrankiai, skirti tiksliau klasifikuoti ir apibrėžti ryšius, o taip pat resursus, kuriuos ryšiai jungia. Tai užtikrina sėkmingą tarpusavio sąveiką ir sudėtingus automatizuotus veiksmus. Vieną kartą aprašius ryšį tarp dviejų resursų, jis gali būti naudojamas daugelyje sričių be poreikio kiekvieną kartą perrašyti sąvokas (RDF aprašus ir kt.).
- Galimi įrankiai, kurių pagalba aprašomi ryšiai tarp resursų ir jų ryšių. Pavyzdžiui, jei ryšys sieja asmenį ir jo elektroninio pašto adresą, galima aprašyti taisykles, pagal kurias šis adresas yra unikalus žmogui. Šio lygmens įrankiai gali užtikrinti geresnę sąveiką, gali aptikti nesuderinamumus ir rasti naujus ryšius.
- Metodai, skirti išgauti duomenims iš tradicinių informacijos šaltinių ir susieti su jais, kad būtų užtikrintas duomenų suderinamumas tarp skirtingų informacijos šaltinių. Tokių metodų pavyzdžiai yra GRDDL, RDFa, POWDER.

Semantinio tinklo vystymas įtakojo ir dirbtinio intelekto sistemų vystymąsi. Dirbtinio intelekto sistemose yra reikalaujama dirbti su nepilnais duomenimis, kurie yra gauti iš skirtingų šaltinių naudojantis URI (universaliais resursų identifikatoriais – angl. „Uniform Resource Identifier“).

Informacijos gavybos iš interneto (angl. „Web mining“) technologijos yra tinkamas sprendimas norint gauti reikalingą informaciją. Automatizuotam informacijos gavimui iš interneto išteklių yra naudojamos programos vadinamos interneto robotais arba vorais (angl. „web crawler“). Daug svetainių naudoja interneto robotus informacijos gavimui iš interneto. Itin populiariai tokie vorai naudojami paieškos sistemų. Jose šios programos naršydamos internetą kiekvienai svetainei sukuria raktinių žodžių aibę, kuri apibūdina konkrečią svetainę. Tai vadinama indeksavimu. Daugėjant svetainių publikuojančių įvairius straipsnius, indeksavimas taip pat vis plačiau naudojamas periodinę informaciją pateikiančiuose interneto portaluose. Interneto robotai aplankomuose puslapiuose raktinių žodžių ieško pagal iš anksto sudarytus šablonus. Interneto robotai taip pat naudojami informacijos gavybai vykdyti. Norint paimti iš konkretaus šaltinio informaciją, visų pirma reikia žinoti, kur tiksliai ta informacija yra, o taip pat dažniausiai reikia ją bent dalinai suprasti, t.y., žinoti kokias sąvokas ar savybes apibrėžia skaitomas tekstas. Žmonėms šis darbas yra intuityvus ir nereikalauja jokių specialių pastangų, nes žmonės įgyja duomenims suprasti reikalingą žinių bazę su patirtimi. Tuo tarpu kompiuterinės programos paprastai neturi šios žinių bazės, todėl yra naudojami metaduomenys (duomenys apie duomenis), kurie padeda kompiuteriams „suprasti“ informaciją. Vienas didžiausių tokių metaduomenų standartų kūrimo judėjimų yra W3 konsorciumo (W3C) remiamas interneto semantizavimas. Minėti interneto semantizavimo įrankiai yra remiami W3C. [17] Šiam darbui yra aktualu semantikos kūrimo principai (metodai) ir programinė įranga, kuri naudoja šiuos principus informacijai gauti, apdoroti ir pateikti vartotojo pageidaujamu formatu.

2.1. Tinklapių semantizavimo metodai

RDF formatu galima puikiai aprašyti ryšius tarp įvairių objektų ar abstrakčių sąvokų, nesvarbu, ar jie yra pasiekiami tinklu ar ne. Norint semantizuoti internete esančius informacijos šaltinius, pavyzdžiui, įvairius tinklapius, kurie yra pateikti HTML kalboje, reikia juose pateikiamą informaciją sužymėti kompiuteriams suprantamomis žymėmis. Tik sužymėjus dokumentus

semantinėmis (prasminėmis) žymomis kompiuteriai gali skaityti juose esančią informaciją. Tuo remiasi interneto semantizavimo procesai, kuriuos plėtoja W3C. RDF nėra skirtas susieti paties HTML dokumento teksto su jame parašytų sąvokų prasme – jis tik nusako ryšius tarp įvairių sąvokų ar dokumentų. Norint automatizuoti tinklo semantizavimą yra reikalingas įrankis, kuris išgautų sąvokas iš teksto ir pateiktų jas RDF tripletus kuriančiai programai. Tai yra viena anksčiau minėtų semantinio tinklo sudedamųjų dalių. Vienas šio darbo antrinių tikslų yra išnagrinėti tokių įrankių veikimą. Yra keli RDF sudarymo iš HTML dokumentų metodai. [17]

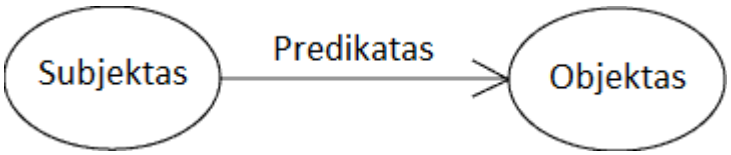
2.1.1. Resursų aprašymo sistema (RDF)

RDF – tai standartinis duomenų apsikeitimo internete modelis. RDF leidžia duomenų sujungimą net tada, kai jų struktūros skiriasi. Šis modelis palaiko duomenų struktūrų kaitą be poreikio keisti duomenis vartojančią programinę įrangą. RDF praplečia nuorodų struktūrą internete naudojant URI įvardinti patį ryšį tarp resursų, o taip pat įvardinti abu nuorodos galus. Tai paprastai vadinama tripletu. Tripleto grafo struktūra pateikta Pav. 1. Naudojant šį paprastą modelį galima maišyti, platinti ir dalintis struktūrizuotais arba pusiau struktūrizuotais duomenimis tarp skirtingų taikomųjų programų. Ši nuorodų struktūra suformuoja kryptingą, sužymėtą grafą, kuriame briaunos vaizduoja ryšį tarp dviejų resursų, kurie vaizduojami grafo viršūnėmis. RDF tikslas yra įgyvendinti paprastą būdą kurti teiginiams apie interneto resursus, pavyzdžiui, tinklapius. Įsivaizduokime, kad norime paskelbti, jog asmuo vardu Petras Petraitis sukūrė konkretų tinklapį. Tiesioginis būdas tam padaryti natūralioje kalboje, pavyzdžiui, lietuvių būtų paprastas teigiamasis sakinys:

http://www.pavyzdys.lt turi kūrėją, kuris yra Petras Petraitis

Šiame teiginyje galima pastebėti, jog norint apibūdinti tam tikro daikto savybę(-es) turi būti būdai įvardinti dalykus:

- daiktą, kurį apibūdina teiginys (šiuo atveju - tinklapis);
- savybę, kurią teiginys apibūdina (šiuo atveju tai yra kūrėjas);
- dalyką, kuris teiginyje yra laikomas to daikto savybės reikšme (čia įvardijamas kūrėjas).



1 pav. RDF tripleto grafo struktūra

Pavyzdiniame teiginyje identifikuoti tinklapį yra naudojamas jo URL (universalus resurso adresas). Taip pat žodis „kūrėjas“ naudojamas nusakyti savybę, o žodžiai „Petras Petraitis“ nusako daiktą (asmenį), kuris yra savybės reikšmė. Kitos šio tinklapio savybės gali būti apibūdinamos kitais panašios struktūros lietuvių kalbos sakiniais naudojant URL tinklapiui vienareikšmiškai nusakyti ir žodžius ar jų junginius nusakyti savybes ir jų reikšmes. RDF yra pagrįstas mintimi, kad daiktai, kurie yra apibūdinami, turi savybes, kurios turi reikšmes, ir kad resursai gali būti apibūdinami rašant panašius į čia nagrinėtą pavyzdį teiginius, kurie nurodo savybes ir jų reikšmes. RDF naudoja specifinę terminologiją, kuomet yra kalbama apie įvarias teiginių dalis. Teiginio dalis, kuri identifikuoja daiktą, apie kurį yra teiginys, vadinama subjektu.

Dalis, kuri nusako subjekto savybę (pvz,. kūrėją) yra vadinama predikatu, o dalis, kuri nusako predikato reikšmę yra vadinama objektu. Taigi nagrinėjant pavyzdį

http://www.pavyzdys.lt turi kūrėją, kuris yra Petras Petraitis

teiginio subjektas yra „http://www.pavyzdys.lt“, predikatas yra „kūrėjas“, o objektas yra „Petras Petraitis“. Natūrali kalba (pvz., lietuvių) yra tinkama bendrauti žmonėms, tačiau RDF yra skirtas kurti teiginiams, kuriuos galėtų apdoroti mašinos. Norint paversti tokius teiginius tokiais, kokius galėtų apdoroti kompiuteriai, reikia:

- kompiuteriams skaitomų identifikatorių sistemos, kuri padėtų nusakyti subjektą, predikatą ir objektą be rizikos, kad jie būtų supainiojami su panašiais žodžiais, kurie yra vartojami internete kitoje vietoje, kitame kontekste;
- kompiuteriams suprantamos kalbos, kuria šie teiginiai galėtų būti atvaizduojami ir jais keičiamasi tarp kompiuterių.

Dabartinė interneto struktūra yra palanki abiems šioms sąlygoms. Kaip minėta aukščiau, internete jau yra naudojamas unikalus, vienareikšmiškas resurso adresas (URL). Pavyzdyje URL buvo panaudotas kaip adresas identifikuoti tinklapiui, kurį sukūrė Petras Petraitis. URL yra simbolių eilutė, kuri identifikuoja interneto resursą nurodydama būdą jį pasiekti (praktiškai tai yra vieta tinkle). Tačiau pasaulyje yra poreikis aprašyti informaciją apie daiktus, kurie priešingai nei tinklapiai neturi vietos tinkle ar URL. RDF koncepcijoje vietoje URL yra naudojama bendresnė identifikatoriaus forma – URI (universalus resurso identifikatorius). URI gali būti naudojamas identifikuoti tinkle esančius resursus – lygiai kaip URL, tačiau URI pagalba galima vienareikšmiškai nusakyti objektus, kurie nėra prijungti prie pasaulinio tinklo – žmones, įmones, knygas bibliotekoje ir kitus fizinius objektus. Taip pat URI gali identifikuoti ne tik fizinius ar tinkle esančius objektus – URI pagalba taip pat galima vienareikšmiškai nusakyti abstrakčias sąvokas, kurios neturi fizinio pavidalo, pavyzdžiui, „kūrėjo“ sąvoka. Į URI gali įeiti unikodo simboliai, o tai leidžia naudoti daugelį kalbų aprašant resursų identifikatorius. Dėl URI lankstumo RDF gali apibūdinti praktiškai viską ir visus ryšius.

Teiginių formulavimui RDF formate yra naudojama XML. Ši kalba buvo sukurta, kad visiems leistų sukurti savo dokumento formatą ir kurti dokumentus remiantis šiuo formatu. RDF naudoja specialią XML kalbą, žinomą kaip RDF/XML, RDF informacijos atvaizdavimui ir jos keitimuisi tarp kompiuterių.

Kodo pavyzdys 1

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

      <contact:Person
rdf:about="http://www.w3.org/People/EM/contact#me">
        <contact:fullName>Petras Petraitis</contact:fullName>
        <contact:mailbox rdf:resource="mailto:pp@ktu.edu"/>
        <contact:personalTitle>Prof.</contact:personalTitle>
      </contact:Person>

</rdf:RDF>
```

Kodo pavyzdyje 1 pateiktas tipiškas RDF/XML naudojimas. Čia yra aprašomas asmuo vardu Petras Petraitis, yra pasakoma, kad jis turi elektroninio pašto dėžutę adresu pp@ktu.edu, ir

jo titulas (šiuo atveju mokslinis laipsnis) yra „Prof.“. Asmens savybių pavadinimų aibė „contact“ yra aprašyta adresu http://www.w3.org/2000/10/swap/pim/contact#.[18]

2.1.2. Gleaning Resource Descriptions from Dialects of Languages

GRDDL – angliškai „Gleaning Resource Descriptions from Dialects of Languages“ – metodas gauti RDF duomenis iš XML dokumentų, o tiksliau XHTML tinklapių. Autoriai gali apibrėžti dokumentų transformacijos algoritmus, kurie paprastai nurodomi naudojant „link“ elementą dokumento „head“ (antraščių) srityje. Paties XHTML dokumento elementai yra papildomi atributais, kurie nurodo elemente vaizduojamo teksto prasmę. [19]

Kodo pavyzdys 2

```
...
<head profile="http://www.w3.org/2003/g/data-view">
    <title>Jono dienotvarkė</title>
    <link rel="transformation"
href="http://www.w3.org/2002/12/cal/glean-hcal"/>
...
```

Kodo pavyzdyje 2 pateikta galimo dokumento palaikančio GRDDL sintaksę ištrauka. „Link“ elemente nurodoma, kad skaitant dokumentą turi būti naudojama transformacija saugoma resurse adresu „http://www.w3.org/2002/12/cal/glean-hcal“. Taip pat ištraukoje matomoje dokumento antraštės dalyje nurodoma, kad dokumentas palaiko GRDDL sintaksę. Tai nusako elemento „head“ atributas „profile“ turintis reikšmę „http://www.w3.org/2003/g/data-view“. Šis „profile“ atributas pasako, kad dokumento „skaitytojas“ (programa), turi ieškoti dokumente elementų su atributu „rel“ turinčiu reikšmę „transformation“ ir naudoti visas rastas nuorodas ieškant transformaciją aprašančių dokumentų. Transformacijų dokumentuose saugoma informacija, kaip iš skaitomo XHTML dokumento paimti reikalingus duomenis. Šiuo atveju adresu „http://www.w3.org/2002/12/cal/glean-hcal“ saugomame dokumente nurodytos įvairių virtualaus kalendoriaus informacijos laukų buvimo vietos trinklapyje. Pavyzdžiui, kalendoriuje žymimo įvykio pradžios data bus korektiškai nuskaityta, kai elementas, kuriame ji saugoma, turės atributą „class“ su reikšme, į kurią įeina „dtstart“ eilutė. Tai yra nusakoma transformacijos dokumente eilute:

```
<xsl:with-param name="class">dtstart</xsl:with-param>
```

Programa, skaitydama dokumentą, suras visus elementus, kurie turi atributą „class“, į kurio reikšmę įeina žodis „dtstart“ ir sukurs RDF tripletą, kuris subjektą, kuris čia yra aprašomas įvykis, susies su objektu, kuris yra elemente saugoma reikšmė, predikatu „dtstart“.

2.1.3. RDFa

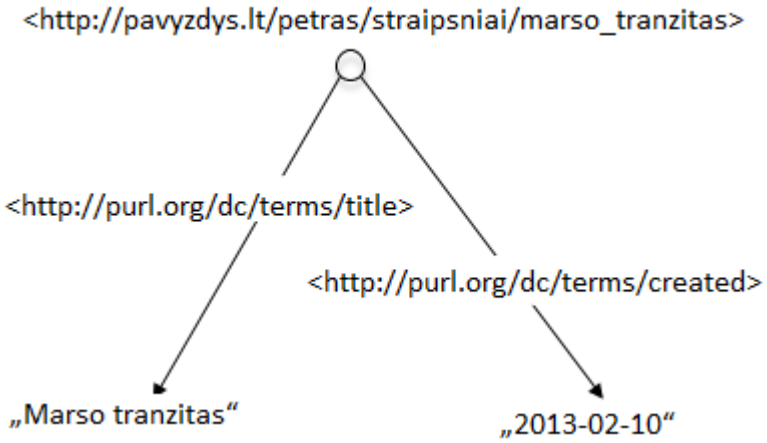
RDFa – angliškai „Resource Description Framework in attributes“ – tai resursų aprašymo sistema atributuose. Tai yra W3C rekomendacija, kuri prideda atributų lygmens plėtinius HTML, XHTML ir įvairiems XML pagrindu sukurtiems dokumentų tipams. Šių plėtinių pagalba galima XHTML dokumentus papildyti RDF subjekto-predikato-objekto metaduomenis. RDFa taip pat leidžia suderinamai programinei įrangai išgauti RDF tripletus iš minėtų tipų dokumentų, kurie yra papildyti šiais metaduomenimis. [20] Naudojant RDFa metodą, kompiuterių skaitomi dokumentai yra sudaromi sekančiu būdu. Sakykime, turime dokumentą (tinklaraščio puslapį), kuriame yra parašytas straipsnis su pavadinimu ir paskelbimo data:

Kodo pavyzdys 3

```
. . .
```

```
  <h2 property="http://purl.org/dc/terms/title">Marso
tranzitas</h2>
  <p>Data: <span property="http://purl.org/dc/terms/created">2013-
02-10</span></p>
...
```

Kodo pavyzdys 3 sudaro duomenų struktūrą pavaizduotą Pav. 2. Tinklaraščio straipsnio adresas yra „http://pavyzdys.lt/petras/straipsniai/marso_tranzitas“, todėl gaunami du RDF tripletai, kuriuose subjektą „http://pavyzdys.lt/petras/straipsniai/marso_tranzitas“ apibūdina predikatai nurodantys pavadinimą ir parašymo datą.



2 pav. Pavyzdinė RDFa duomenų struktūra

RDF sprendimas naudoti URL nusakyti predikatams remiamas tuo, kad URL yra unikalūs ir vienareikšmiški. Kai visi žodyno terminai yra apibrėžti pagal URL, jų prasmės yra tik už vieno paspaudimo. Tai leidžia visiems – ir žmonėms, ir mašinoms – sekti nuorodą, kad išsiaiškintų, ką reiškia konkretus terminas. Naudojant URL nusakyti konkrečiam ryšiui, pavyzdžiui, „http://purl.org/dc/terms/created“, ir žmonės, ir mašinos gali suprasti, kad šis adresas vienareikšmiškai nusako „datą, kada resursas buvo sukurtas“, šiuo atveju tai tinklaraščio straipsnis. Tai ypatingai palengvina sprendžiant, ar skirtinguose dokumentuose pavartotas terminas reiškia tą patį. [20] Schema.org pateikia populiariausių naudojamų atributų apibrėžimų žodynus, kurie gali būti naudojami su RDFa 1.1 Lite versija, tačiau formatas leidžia naudoti savo sukurtus žodynus („http://www.data-vocabulary.org/“).

2.1.4. Microdata – mikroduomenys

Microdata yra HTML specifikacija, naudojama įterpti semantinę informaciją į tinklapio turinį. Tai veikia panašiu principu kaip ir ankstesni aptarti metodai. Interneto robotai, naršydami tinklapius, gali gauti metaduomenis (mikroduomenis) iš papildomų HTML atributų. Informacijai pateikti Microdata naudoja papildomus žodynus ir pavadinimo-reikšmės poras. [21] Microdata gali naudoti tuos pačius Schema.org žodynus duomenims aprašyti, tačiau, kaip ir RDFa atveju, formatas leidžia kurti savo žodynus. Microdata modelyje minėtos pavadinimo-reikšmės poros yra grupuojamos į grupes, kurios gali turėti tipą ar kelis tipus, kiekvienas pavadinimas gali turėti vieną arba daugiau reikšmių, o kiekviena reikšmė yra simbolių eilutė arba kita grupė sudaryta iš pavadinimo-reikšmės porų. Keli naudojami žodynai yra aprašyti kaip dalis Microdata standarto, o taip pat standartiškai yra aprašyti kelių populiarių globalių savybių pavadinimai, kurie gali būti naudojami visiems aprašomiems dalykams. Žodynai nurodo aprašomų dalykų tipus. Pavyzdžiui, yra aprašytas tipas „vevent“ žymintis įvykį. Šis tipas yra paremtas žodynu, kuris aprašytas iCalendar internetinio kalendoriaus specifikacijoje RFC2445. [22] Įvykį aprašantys metaduomenys į HTML galėtų būti įterpiami taip, kaip parodyta Kodo pavyzdys 4.

Kodo pavyzdys 4

```
<body item="vevent">
 ...
 <h1 itemprop="summary">Įvykio santrauka</h1>
 ...
 <time itemprop="dtstart" datetime="2009-05-10T13:15:00Z">Gegužės 10 d.
13:15</time>
 (iki <time itemprop="dtend" datetime="2009-05-
10T21:00:00Z">21:00</time>)
 ...
 <a href="http://pavyzdys.lt/ivykis/0510"
    rel="bookmark" itemprop="url">Nuoroda</a>
 ...
 <p>Vieta: <span itemprop="location">Kaunas</span></p>
 ...
 <meta itemprop="description" content="pavyzdys.lt">
</body>
```

Kodo pavyzdys 4 yra sudaroma HTML struktūra, kurios hiearchijos viršuje yra nurodomas, aprašomo objekto tipas, kuris šiuo atveju yra įvykis. Tai padaroma atributu „item“, kuriam priskiriama reikšmė „vevent“. Tai reiškia, kad elemente, kuris turi šį atributą ir šią reikšmę, visi esantys elementai gali aprašyti įvykio savybes. Pačios savybės yra nurodomos papildomais HTML atributais, pvz., „itemprop“ ir „datetime“, o jų reikšmės yra imamos pagal žodyną, t.y. panašiai kaip naudojantis GRDDL tranformacija. Pavyzdžiui, tipo „dtstart“ reikšmė yra imama iš to paties elemento atributo „datetime“. Kiti reikšmių tipai gali būti aprašomi kitomis taisyklėmis. Elementų audio, embed, iframe, img, source ir video elementų reikšmės yra absoliutūs URL, kurie išgaunami iš atributo „src“ reikšmės, arba tuščia eilutė, jei išgaunant URL įvyksta klaida arba atributas „src“ tuščias.[21]

2.1.5. Website Parse Template

GRDDL, RDFa ir ypač Microdata dokumentų semantizavimo ir gavimo metodai remiasi metaduomenų įterpimu į HTML dokumentus. Tai yra priklausoma nuo dokumento autoriaus, o jis ne visada tuo pasirūpina. Norint semantizuoti HTML dokumentus, kuriuose nėra iš anksto aprašyti metaduomenys, reikalingas kitos šalies (ne autoriaus) mechanizmas. Vienas toks sprendimas yra organizacijos OMFICA („Open Market For Internet Content Accessibility“) sukurta tinklapių skaitymo šablonų (Website Parse Template – WPT) specifikacija. WPT yra XML pagrindu sukurtas formatas, kuris leidžia aprašyti tinklapių HTML struktūrą. WPT interneto robotams leidžia sugeneruoti semantinio tinklo RDF šablono aprašomiems tinklapiams. WPT susideda iš tokių dalių:

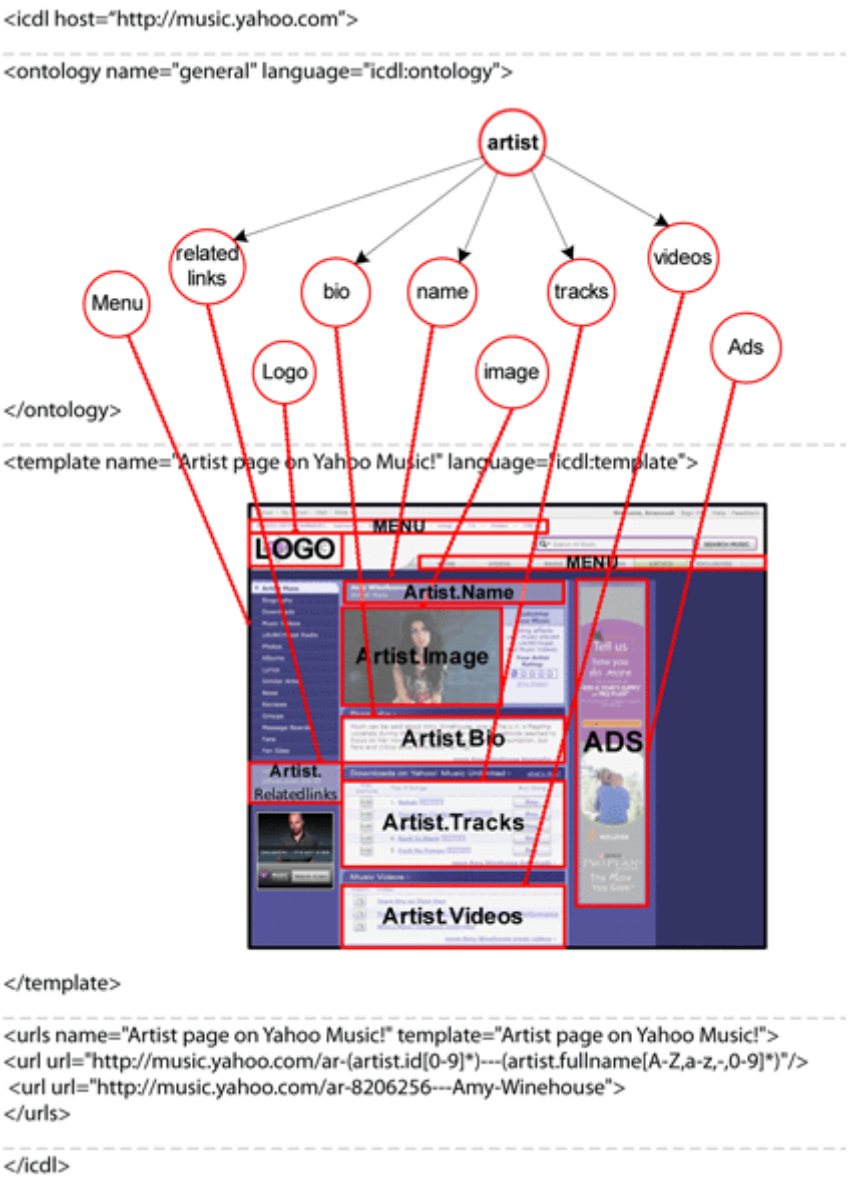
- **Ontologijos**, kurioje aprašomos sąvokos ir ryšiai, naudojami tinklapyje. Aprašant ontologijas yra nurodomi visi objektai, apie kuriuos informacija pateikta svetainėje.

- **Puslapių šablonų**, kuriuose aprašyti visi svetainės viduje esantys puslapiai, kurie yra panašūs turiniu ir struktūra. Šablonuose pateikti HTML elementų identifikatoriai (ID) ir nuorodos į ontologijos sąvokas.
- **Nuorodų (URL)**, kuriose pateikiami nuorodų šablonai, kurie atitinka panašius tinklapius aprašytus tinklapių šablonais.

Tinklapių sintaksinės analizės šablonai prasideda elementu <icdl> ir baigiasi elementu </icdl>. Vienas šablonas visada atitinka tą pačią svetainę, tačiau viena svetainė gali turėti keletą šablonų aprašančių jos HTML struktūrą. Reikalaujama nurodyti svetainės pirminį adresą elemente <icdl>:

```
<icdl host=http://www.pavyzdys.lt>
```

Vizualinis WPT atvaizdavimas pateiktas pav. 3. Čia yra sudarytas šablonas portalo „Yahoo! Music“ puslapiui, kuriame pateikta informacija apie muzikos atlikėją. Pirmiausia WPT šablone nurodomas pirminis tinklapio adresas. Po adreso aprašoma ontologija – visi naudingos informacijos laukų ryšiai. Galiausiai surašomos nuorodų šablonai arba pačios nuorodos, rodančios į kitus panašią informaciją (tokia pat struktūra) pateikiančius svetainės tinklapius. [2]



3 pav. Vizualinis WPT vaizdas

2.1.5.1. Ontologija

Ontologija – tai modelis naudojamas gauti ir atvaizduoti žinias. Ji aprašo tam tikros sferos sąvokas ir ryšius tarp jų. Taip pat ontologija leidžia samprotavimą, tai yra padaryti išvadą, gauti papildomos informacijos, kuri nėra tiesiogiai nusakoma. Ontologijos naudojamos dirbtinio intelekto, programinės įrangos inžinerijos ir biomedicinos srityse aplinkos ar jos dalies žinių atvaizdavimui. Dabar ontologijos naudojamos semantiniam žiniatinkliui formuoti. Šioje srityje tinklapių turinys yra papildoas semantine informacija. Ontologijos suteikia geresnį suderinamumą įvairiuose tinklapiuose esančiai informacijai. [2]

Ontologijose nurodomi visi pavadinimai bruožų, kuriais gali būti aprašytas tas objektas. Pavyzdžiui, kuriant ontologiją puslapiui, kuriame pateikta informacija apie muzikinius kūrinius, puslapio ontologija galėtų atrodyti taip:

Kodo pavyzdys 5

```
<ontology name="bendra" language="icdl:ontology">
          <concept name="atlikejas">
                    <inherit concept="asmuo"></inherit>
                    <has object="vardas"></has>
                    <has object="dainos_pavadinimas"></has>
                    <has object="nuotrauka"></has>
                    <has object="bio"></has>
                    <has object="id"></has>
                    <has object="pilnas_vardas"></has>
          </concept>
          <concept name="Logo"></concept>
          <concept name="meniu"></concept>
          <concept name="reklama"></concept>
</ontology>
```

Kiekvienos naujos sąvokos aprašymas prasideda elementu <concept> ir baigiasi </concept>. Elementas <inherit> žymi paveldimumo ryšį tarp dviejų sąvokų, o <has> elementas žymi sąvokai priskiriamus atributus. Visos aprašytos sąvokos turi bazinius atributus – objekto identifikatorių (id), naudojamą interneto robotų koordinuoti to paties objekto atrobutus naudojamus skirtinguose puslapiuose viename tinklapyje. Yra keletas formato specifikacijoje aprašytų sąvokų, kurios būdingos visiems tinklapiams:

„Menu“ – navigacijos juosta / meniu;

„Logo“ – dizaino elementas / logotipas;

„Content“ – elementas, kuriame yra tekstinis turinys;

„Advertisement“ – reklamos juostos;

„External link“ – elementas, kuriame yra išorinės nuorodos.

Ontologijos taip, kaip naudojamos šiandien, turi svarbių trūkumų, kurie dar nėra išspręsti, o tai trukdo platesniam jų naudojimui. Pirmiausia, ontologijos neapima visų modeliavimo sąvokų reikalingų apibrėžti pasauliui. Pavyzdžiui įsivaizduokime asmenį, apie kurį informacija pateikiama keliuose šaltiniuose skirtinguose kontekstuose: šeimos, darbo ir sporto klubo. Kaip turėtų atrodyti jo savybės kaip tėvo, darbuotojo ar krepšinio žaidėjo? Kaip dėl savybių, kurios naudojamos visuose kontekstuose, pavyzdžiui, vardas ar gimimo data? Šiame pavyzdyje

susiduriame su dviem skirtingais abstrakcijos lygiais. Vienas identifikuoja žmogų, o kitas priklauso nuo konteksto, kuriame tas žmogus minimas. [2]

Ontologijos kalbos šiandien atsiranda kaip de facto standartas žiniatinklio semantikai apibūdinti. Viena svarbiausių ontologijos aprašymo kalbų šiuo metu yra „Interneto ontologijos kalba“ (angl. Web Ontology Language), kitaip – OWL, kuri yra standartizuota ir rekomenduojama pasaulinio žiniatinklio konsorciumo – W3C. Ši kalba yra orientuota į bendrą žiniatinklio struktūrą ir konkrečiai semantinio žiniatinklio svetaines. Ji sukurta RDF pagrindu. Šiuo metu naujausia OWL versija yra OWL 2. OWL 2 ontologijos aprašo klases, savybes, objektus ir duomenų reikšmes ir visa tai saugo semantinio žiniatinklio dokumentų pavidalu. OWL 2 ontologijos gali būti suderinamos su informacija suformuota pagal RDF. [2]

2.1.5.2. Puslapių šablonai

Šablonų skyriuje yra keletas šablonų aprašančių panašių puslapių grupes. Kiekvienas šablonas aprašo vieną grupę panašią struktūrą turinčių tinklapių. HTML elementų Xpath nuorodos arba elementų ID naudojami susieti struktūrizuotą turinį su aprašančiomis sąvokomis. Šablono aprašas pradedamas atidaromuoju elementu <template> ir baigiamas elementu </template>. Elemente <template> yra nurodomas šablono pavadinimas ir kalba, naudojama šablono aprašyme. Šablono pavadinimas gali būti bet kokia simbolių seka, o šablono aprašo kalba turi būti viena iš palaikomų kalbų, pavyzdžiui: „icdl:template“, „rdf“, „unl:expression“. Šablono pavyzdys „Yahoo! Music“ svetainei pateiktas Kodo pavyzdys 6.

Kodo pavyzdys 6

```
<template name="Atlikejas Yahoo! Music tinklapyje"
language="icdl:template">
<html_tag tagid="yent-uhdr" content="Menu"/>
 <html_tag xpath="/html/body/div[2]/div/div/div[3]/div/a/span"
content="Logo"/>
 <html_tag xpath="/html/body/div/div" content="reklama"/>
 <html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[2]/td/div/
h1" content="atlikejas.vardas"/>
 <html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[3]/td/tabl
e/tbody/tr/td/img" content="atlikejas.nuotrauka"/>
 <html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[7]/td"
content="atlikejas.bio" reference="Atlikejo Bio"/>
 <container
container_xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[
10]/td/table">

          <repeatable_block
block_xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[10]/
td/table/tbody/tr/td">
 <html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[10]/td/tab
le/tbody/tr/td" content="atlikejas.daina"/>
          </repeatable_block>
</container>
</template>

<template name="Atlikejo Bio" language="icdl:template">
```

```
<html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[2]/td/div/
h1" content="atlikejas.vardas"/>
 <html_tag
xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[7]/td"
content="atlikejas.bio"/>
 </template>
```

WPT šablone informacija gali būti aprašoma hierarchine struktūra nurodant parametre „reference“ atitinkamo šablono pavadinimą. Taip interneto robotai gali atpažinti tą patį objektą skirtinguose puslapiuose toje pačioje svetainėje. [2]

2.1.5.3. Nuorodos (URL)

Šioje WPT dalyje pateikiami nuorodų šablonai, atitinkantys grupes puslapių turinčių panašią struktūrą į aprašytąją puslapių šablonų dalyje. WPT nuorodų pavyzdys pateiktas Kodo pavyzdys 7.

Kodo pavyzdys 7

```
<urls name="Atlikėjas Yahoo! Music" template="Atlikejas Yahoo! Music
tinklapyje">
 <url url="http://music.yahoo.com/ar-8206256---Amy-Winehouse"/>
 <url url="http://music.yahoo.com/ar-(atlikejas.id[[0-9]*])---
(atlikejas.pilnas_vardas[[a-z,a-z,-,0-9]*])"/>
</urls>

<urls name="Atlikejo biografija" template="Atlikejo Bio">
 <url url="http://music.yahoo.com/ar-8206256-bio--Amy-Winehouse"/>
 <url url="http://music.yahoo.com/ar-(atlikejas.id[[0-9]*])-bio--
(atlikejas.pilnas_vardas[[a-z,a-z,-,0-9]*])"/>
</urls>
```

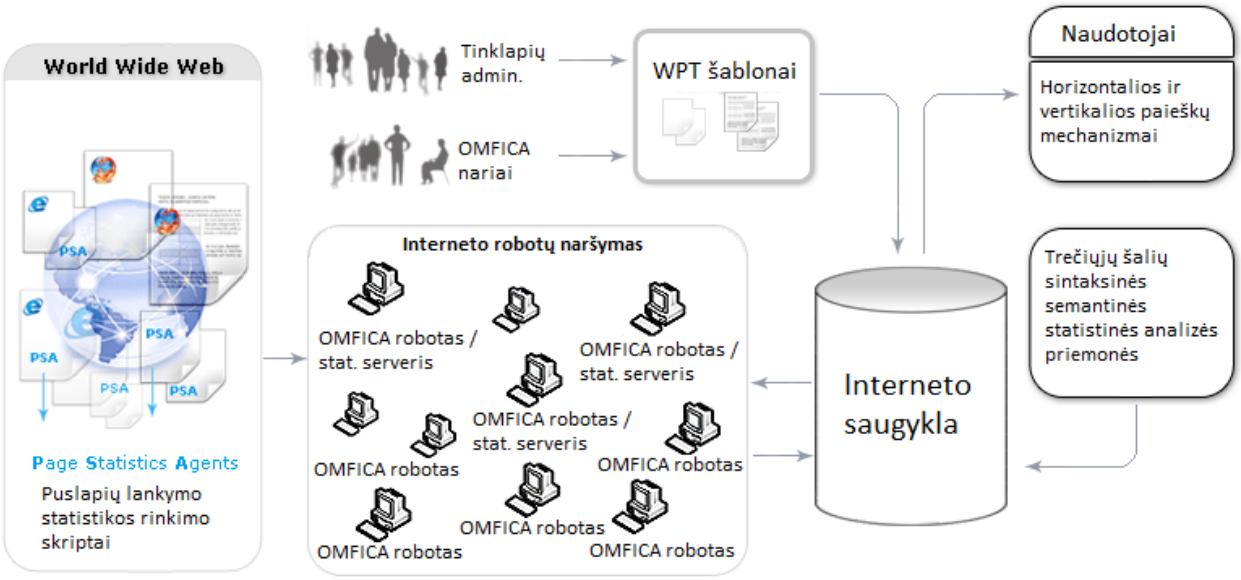
Pavyzdyje pateikti nuorodų pavyzdžiai, atitinkantys jų šablonus, URL šablonai aprašomi RegEx sintakse. Naudojamos nuorodų aprašymuose sąvokos tokios kaip „id“ arba „pilnas_vardas“ turi būti aprašomos ontologijų dalyje. [2]

2.1.6. Interneto robotų naršymas

Organizacija OMFICA vysto savo ir trečiųjų šalių technologijas, padedančias kurti atvirą interneto saugyklą (angl. „Open Web Repository“) – interneto semantinių duomenų bazę, kurią galėtų naudoti visi. OMFICA vykdoma veikla gali būti logiškai suskirstyta į grupes:

- Interneto naršymas robotais ir jų surinktos informacijos talpinimas interneto saugykloje.
- Tinklapių naršymo statistikos valdymas.
- Pastovus WPT šablonų saugyklos pildymas.
- Kasdienių atnaujinimų generavimas pavidalu failų, kurie gali būti parsisiunčiami FTP protokolu.

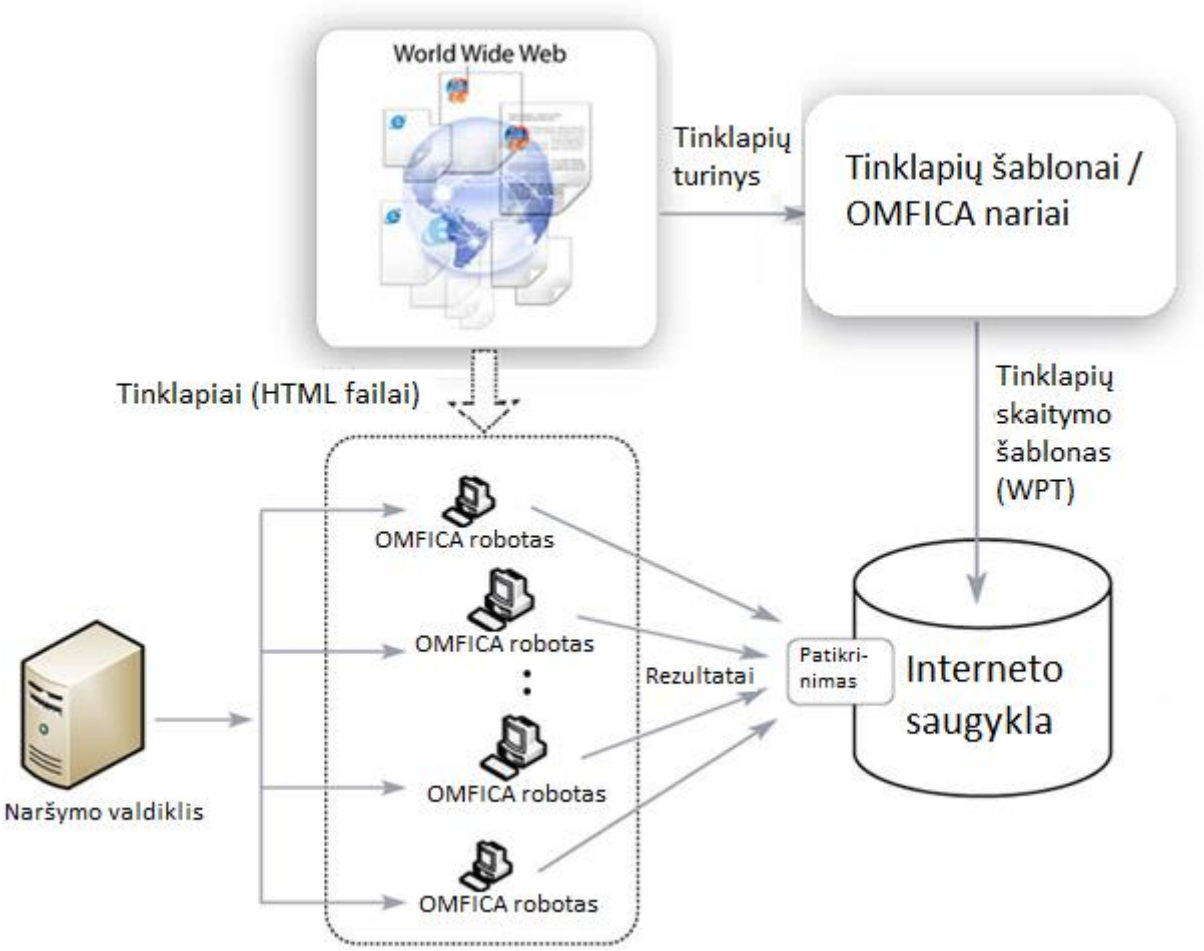
OMFICA leidžia interneto vartotojams, kompanijoms, tinklapių administratoriams, tekstą analizuojančioms šalims įsitraukti į atviros interneto saugyklos kūrimą.



4 pav. Duomenų judėjimo schema OMFICA veikloje

OMFICA veiklos schema pavaizduota Pav. 4. Ji susideda iš tokių dalių:

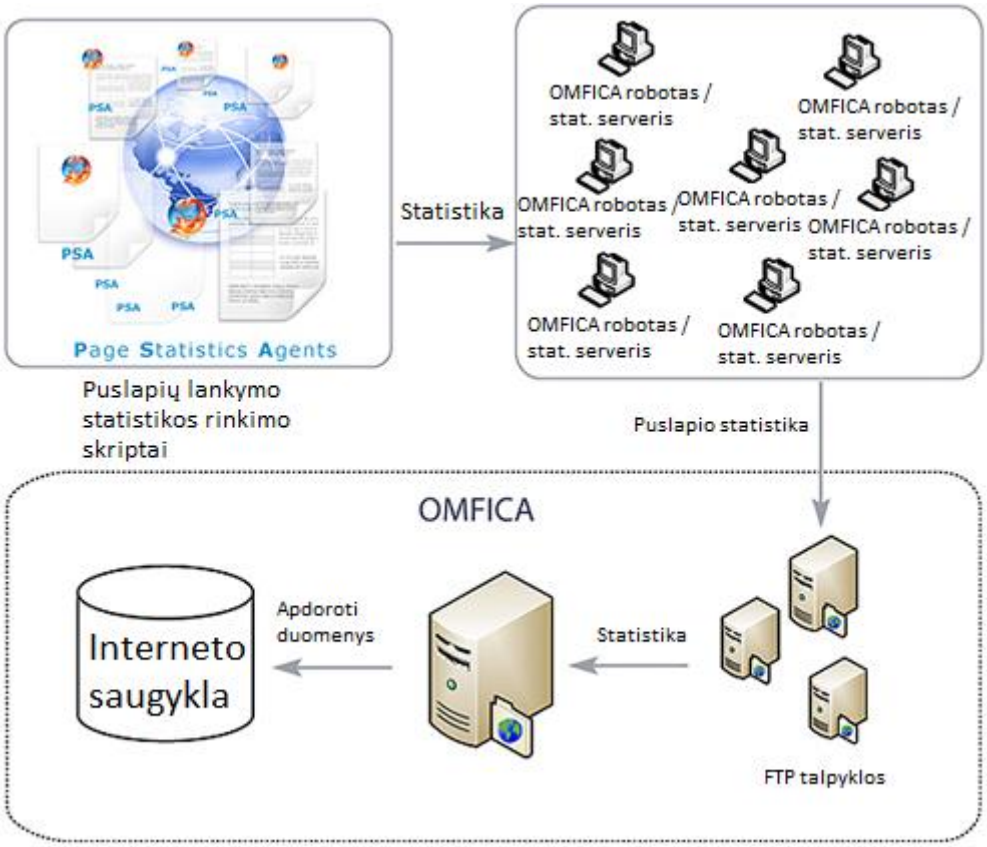
- **JavaScript skriptai**, kurie, patalpinti įvairiuose tinklapiuose, teikia tinklapių lankymo statistiką į informacijos saugyklą.
- **Interneto robotai.** Robotai – specialios programos, automatiškai naršančios internetą ir renkančios informaciją. Jie įdiegti savanorių kompiuteriuose, iš kurių veikdami siunčia surinktą informaciją interneto saugyklos serveriams.
- **Integruota duomenų saugykla.** Saugo tinklapių struktūrų aprašymus, jų turinių kopijas, lankymo statistiką ir kt.
- **Trečiųjų šalių technologijos**, padedančios kurti OMFICA duomenų saugyklą.
- **Kompanijos ir individai**, patikėti su priėjimu prie duomenų bazės.
- **WPT šablonai**. Tai XML pagrindu sukurtas formatas, saugantis tinklapių HTML struktūrą.



5 pav. OMFICA interneto naršymas

OMFICA interneto robotų naršymo schema pateikta pav. 5. OMFICA robotų interneto naršymas veikia paskirstytos sistemos principu. Daug pavienių programų veikia, kad sukurtų interneto saugyklą. Pagrindinė interneto robotų naršymo idėja yra įgalinti interneto vartotojus, kompanijas, tinklapių administratorius, tinklapių kūrėjus ir teksto analizavimo programas bendradarbiauti interneto saugyklos turinio kūrime, pakartotinai automatizuotai naršant interneto tinklapius. Visame internete yra gausu informacijos ir jos daugėja su kiekviena minute, todėl neįmanoma išnaršyti viso interneto vienam asmeniui ar net kompanijai. Atviro interneto naršymas robotais sukuria terpę naudoti sinchronizacijos protokolus, kad naršymas būtų decentralizuotas ir paralelizuotas. Naršymas robotais vykdomas per kliento/serverio sistemą, kurioje naršymo valdiklis veikia kaip užduotis skiriantis komponentas. Jis skiria užduotis daugeliui atviro kodo interneto robotų, kurie yra įdiegti savanorių kompiuteriuose (5 pav.). Interneto naršymo valdiklis pagal nario identifikacinį numerį (ID) ir pagal roboto ID pasirenka naršymo robotus, kuriems siunčia sąrašus adresų tinklapių, kurie turi būti išnaršyti. Tinklapių adresų sąrašai gali būti periodiškai sujaukiami, kad būtų išvengta galimo rezultatų klastojimo. OMFICA robotai naudoja tik laisvus kompiuterių CPU resursus ir iš eilės lanko svetaines ir jas skaito pagal WPT šablonus. Naršymo rezultatai yra dvigubai sutikrinami valdiklio ir po to išsaugomi interneto saugykloje. Naršymo valdiklis tikrinimo metu gali uždrausti piktnaudžiaujančius kopmiuterių resursus pateikusius savanorius. Interneto saugykla yra periodiškai skenuojama ir duomenys yra kopijuojami į archyvų failus, kurie yra kategorizuojami pagal valandinius, dieninius, savaitinius ir mėnesinius atnaujinimus. Šie failų katalogai turi specialias priėjimų teises kompanijoms ir individams – OMFICA sistemos naudotojams.

Interneto naršymo statistika yra viešų interneto tinklapių naudojimo statistikos duomenų rinkinys. Šie duomenys yra išanalizuoti ir apjungti. OMFICA tinklapių statistikos rinkimo schema pavaizduota 6 pav.



6 pav. OMFICA tinklapių lankymo statistikos rinkimo schema

Interneto naršymo statistikos rinkimas vykdomas trimis pagrindinėmis priemonėmis:

- **Puslapio statistikos agentas.** Tai yra JavaScript kodas, kuris renka tinklapio naudojimo ir naršymo statistiką ir siunčia duomenis nurodytam puslapio statistikos serveriui. Šie agentai yra įterpti į tinklapio kodą. Kai puslapis užkraunamas, jie seka vartotojo veiksmus – pelės paspaudimus, turinio peržiūros laiką ir kt.
- **Puslapio statistikos serveris.** Tai yra viena programa ar kelių serverio programų rinkinys, kuris yra įdiegtas projekto savanorių kompiuteriuose arba serveriuose, kurie atsakingi už duomenų surinkimą iš puslapių statistikos agentų ir tų duomenų siuntimą į FTP serverius valdomus OMFICA. Kiekvienas puslapio statistikos serveris yra autorizuotas prisijungti prie konkrečios FTP talpyklos direktorijos.
- **Puslapių statistikos apdorojimo įranga.** Tai yra kelių serverių sistema, kuri yra atsakinga už tinklapių vartojimo/naršymo statistikos apjungimą ir analizę. Puslapių statistikos apdorojimo įranga periodiškai skenuoja puslapių statistikos serverių direktorijas FTP talpyklose ir apdoroja duomenis iš interneto saugyklos, kad duomenimis galėtų pasinaudoti sistemos vartotojai.

Interneto saugykloje laikomi duomenys:

- Puslapių skaitymo šablonai (WPT).
- Išgauti tinklapių duomenys.
- Tinklapių skaitymo rezultatas.
- Tinklapių žemėlapiai.

- HTML skriptai.
- Tinklapių vidinės ir išorinės nuorodos.
- Trečiųjų šalių sugeneruoti tinklapių RDF dokumentai.

Tinklapių duomenys yra tinklapių, kuriuos aplankė OMFICA interneto robotai, turinys. Jie saugomi XML formatu (Kodo pavyzdys 8).

Kodo pavyzdys 8. Tinklapio turinys.

```
<Description about="http://www.pavyzdys.lt">
<omfica:contentText>
          Lorem ipsum dolor
</omfica:contentText>
...
</Description>
```

Tinklapių skaitymo rezultatas yra XML pagrindu sukurtas formatas, kuris saugo struktūrizuotus duomenis apie konkretų tinklapį. Šiuos duomenis išgauna OMFICA interneto robotai vadovaudamiesi WPT šablonų ontologijos skiltimi ir puslapio šablonų skiltimi. Kodo pavyzdys 9 rodo, kaip galėtų atrodyti tinklapio „http://music.yahoo.com/ar-8206256---Amy-Winehouse“ skaitymo rezultatas pagal WPT.

Kodo pavyzdys 9. RDF aprašas.

```
<?xml version="1.0"?>
<RDF
xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:omfica="http://omfica.org/rdfnamespace/music/schema/">
<Description about="http://music.yahoo.com/ar-8206256---Amy-
Winehouse">
<omfica:id>8206256</omfica:id>
<omfica:fullname>Amy-Winehouse</omfica:fullname>
</Description>
</RDF>
```

Tinklapių žemėlapiai yra XML pagrindu sukurtas formatas apibrėžtas http://www.sitemaps.org/.

Tinklapio vidinės ir išorinės nuorodos. yra XML pagrindu sukurtas formatas, kuriuo saugomos vidinės ir išorinės nuorodos iš tinklapio.

Kodo pavyzdys 10. Vidinės ir išorinės tinklapio nuorodos.

```
<Description
about="http://www.omfica.org/npo_data_repository.php">
<omfica:links>
http://www.omfica.org/npo_website_template.php
http://www.omfica.org/npo_open_web_crawling.php
...
</omfica:links>
<omfica:externalLinks>
http://www.sitemaps.org
http://www.w3.org/2001/sw
...
</omfica:externalLinks>
</Description>
```

Kodo pavyzdys 10 pateiktas XML, aprašantis puslapio „http://www.omfica.org/npo_data_repository.php“ vidines ir išorines nuorodas.

Trečiųjų šalių sugeneruoti tinklapių RDF dokumentai yra XML dokumentai, kuriuose saugoma:

- Raktiniai tinklapio objektai ir žymės.
- Sematiniai interneto duomenys susiję su tinklapiu. Šie duomenys pateikiami RDF tripletų pavidalu.

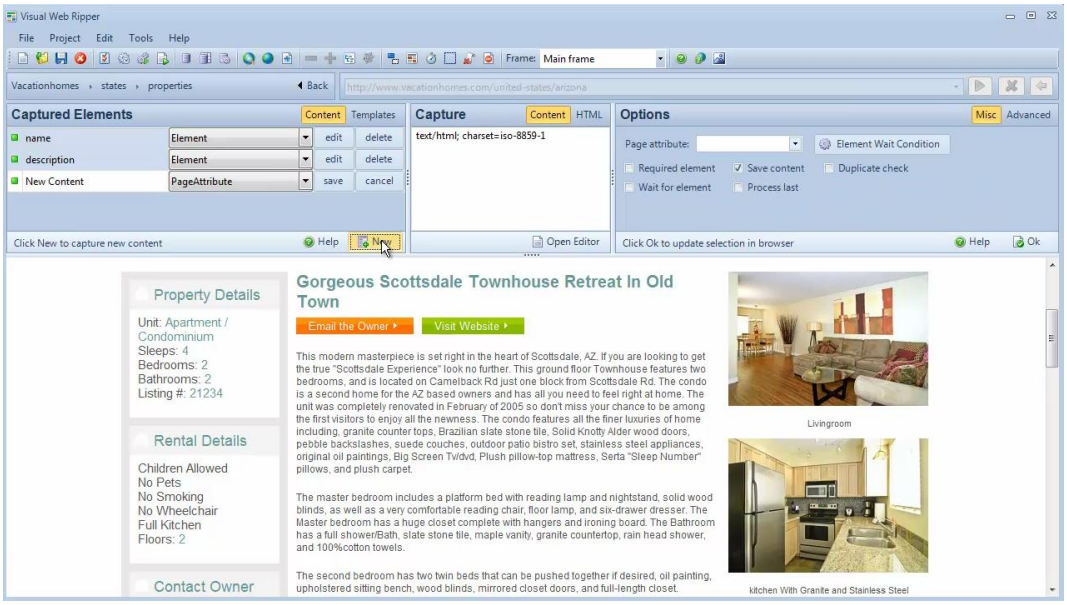
2.1.7. Tinklapių semantizavimo metodų palyginimas

Visi aptarti tinklapių semantizavimo metodai veikia panašiu pagrindiniu principu – žodyne aprašytų reikšmių susiejimu su HTML dokumente vaizduojamomis reikšmėmis. RDFa, GRDDL, Microdata remiasi metaduomenimis pačiame HTML dokumente, todėl šie tinklapių semantizavimo būdai yra tiesiogiai susieti su duomenų atvaizdavimu. Norint semantizuoti tinklapį naudojant šiuos metodus, dokumentų autoriai patys turi pasirūpinti vaizduojamų duomenų aprašymu semantiniais duomenimis. Tokia situacija nėra palanki kitoms šalims, kurios taip pat yra suinteresuotos tinklapių semantizavimu. Tokių šalių pavyzdys būtų interneto paieškos sistemos, nes jų kūrėjams yra aktualu suteikti tinklapiuose vaizduojamiems duomenims prasmę, kad jų kuriama paieškos sistema galėtų suprasti sklaidomuose tinklapiuose vaizduojamą informaciją ir jos kontekstą, tokiu būdu pateikdama tikslius paieškos rezultatus. OMFICA sukurtas tinklapių skaitymo šablono (WPT) metodas remiasi metaduomenų saugojimu atskiruose dokumentuose, kurie nėra susiję su duomenų vaizdavimu, todėl, palengvinus WPT ir analogiškų įrankių plėtojimą, galima daug prisidėti prie internete saugomos informacijos semantizavimo. Dėl šios priežasties šiame darbe pasirinkta suprojektuoti įrankį, padėsiantį generuoti tinklapių WPT šablonus.

2.2. Tinklapių šablonų sudarymo įrankių apžvalga

2.2.1. Visual Web Ripper

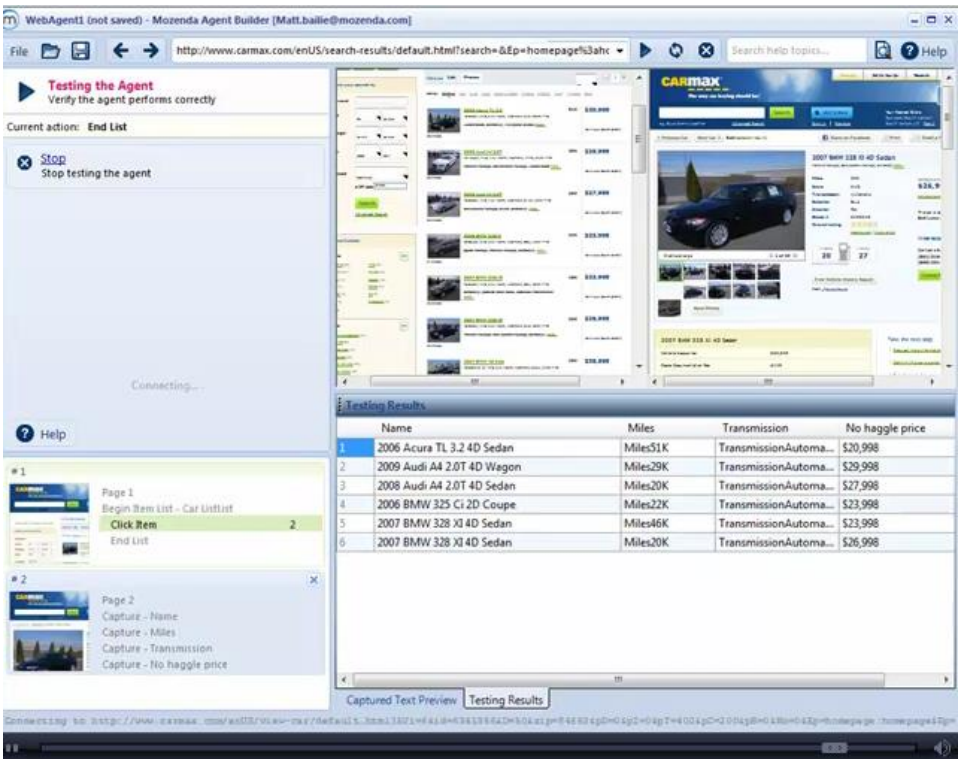
„Visual Web Ripper“ – tai taikomoji programa veikianti MS Windows aplinkoje. Ji skirta automatizuotam informacijos gavimui iš interneto. Šios programos pagalba galima sudaryti šabloną, pagal kurį ji po to surenka duomenis iš pageidaujamo tinklapio. Taigi „Visual Web Ripper“ yra ir interneto robotas, automatizuotai renkantis informaciją (žr. Pav. 7). „Visual Web Ripper“ nėra multiplatforminė programa, o tai gali kelti problemų taikant ją didesniame projekte ir ypač, jei jame dalyvauja daug informaciją renkančių žmonių. Lyginant su internetine aplikacija nepatogu yra ir tai, kad norint dirbti programą reikia įdiegti kompiuteryje. Taip pat programa nesuteikia gautiems duomenims prasmės, t.y. čia nėra naudojami ontologijų aprašai. Į vieną raportą programa gali rinkti duomenis tik iš vieno šaltinio.[5]



7 pav. Visual Web Ripper

2.2.2. Mozenda

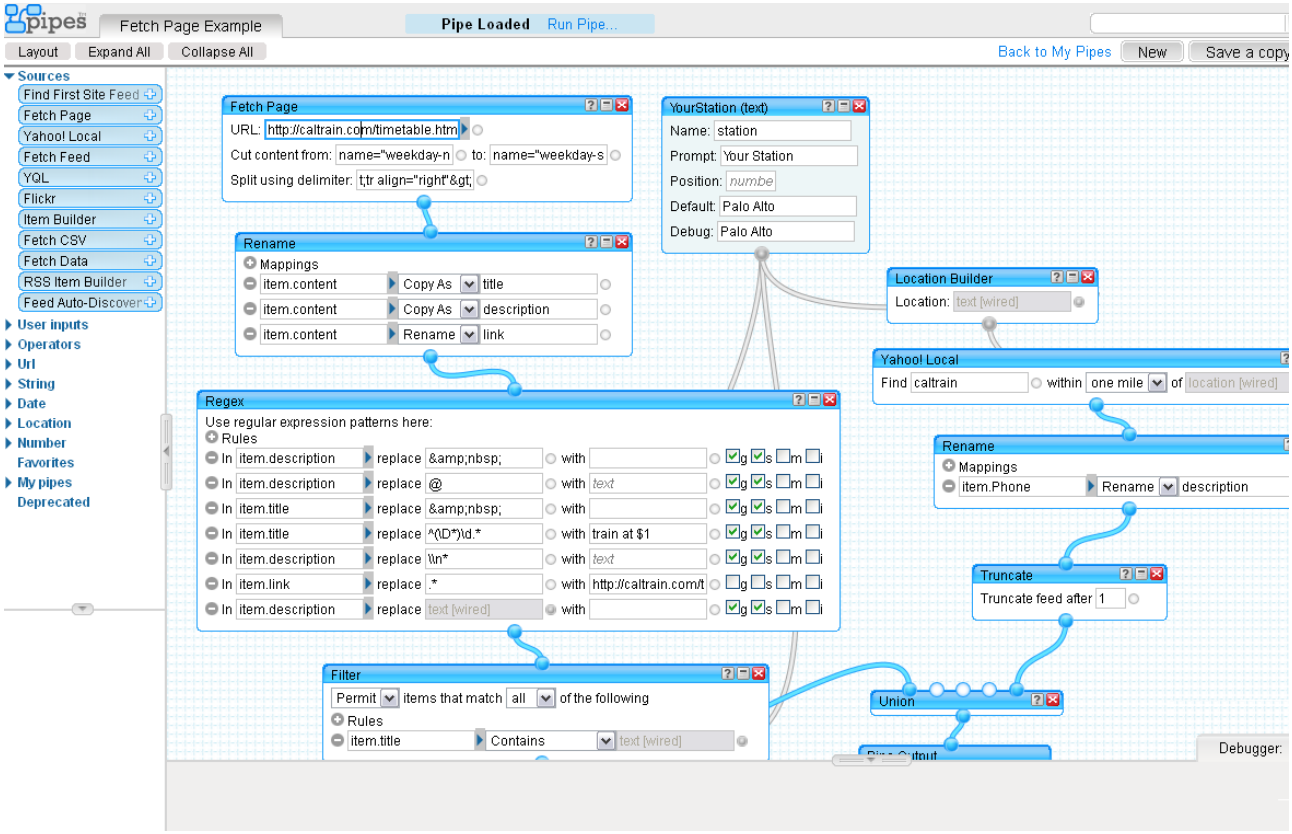
Kaip ir „Visual Web Ripper“, „Mozenda“ yra taikomoji programa. Šis įrankis taip pat leidžia grafinės sąsajos pagalba sudaryti interneto puslapio šabloną nurodant pageidaujamus laukus su informacija. Nurodžius šablonus, programa veikia kaip interneto robotas automatiškai renkantis informaciją (žr. Pav. 8). „Mozenda“ programa turi tokius pat trūkumus kaip ir „Visual Web Ripper“ – ji taip pat nėra multiplatforminė, ją reikia įdiegti kompiuteryje ir ji nenaudoja ontologijų aprašų. Į vieną raportą programa gali rinkti duomenis tik iš vieno šaltinio. [6]



8 pav. Mozenda

2.2.3. Yahoo! Pipes

Tai yra internetinis įrankis skirtas informacijos gavimui struktūrizavimui. Jis pateikia grafinę sąsają, kurios pagalba galima surinkti įvairius duomenų tipus, kaip XML, HTML, RSS, CSV, Json ir kt., ir juos atvaizduoti. Aplikacija leidžia apdoroti duomenis pagal pasirinktus kriterijus norint gauti tik reikalingą informaciją. Tokie kriterijai gali būti apibrėžti filtro ar kitų taisyklių pagalba. „Yahoo! Pipes“ programa susideda iš įrankių bibliotekos, modeliavimo srities ir derinimo paprogramės (angl. „debugger“). Žiūrėti Pav. 9. Ši programa neturi suderinamumo su skirtingomis operacinėmis sistemomis problemos, nes yra pilnai naudojama per interneto naršyklę. Tačiau kaip ir anksčiau nagrinėti įrankiai, šis nesieja gautų duomenų su ontologijomis. „Yahoo! Pipes“ nėra patogus sudarinėti tinklapių šablonus, nes jis nesiūlo grafinės sąsajos elementų pozicijoms nurodyti. [7]



9 pav. Yahoo! Pipes

2.2.4. Numatytas įgyvendinti įrankis

Šiame darbe projektuojama programa bus internetinė aplikacija. Ji pateiks grafinę sąsają, kurios pagalba sudaroma reikiamos interneto svetainės duomenų ontologija. Šio įrankio grafinės sąsajos pagalba duomenys, esantys svetainėje, susiejami su aprašytomis ontologijomis ir automatiškai sudaromas WPT šablonas, kuris naudojamas žinių gavybai iš tos interneto svetainės.

2.2.5. Analogiškų įrankių analizės rezultatai

Lygindami apžvegtą interneto semantizavimo programinę įrangą, galime išskirti tris parametrus, pagal kuriuos galime ją vertinti:

- Pasiekiamumas vartotojui (ar tai internetinė aplikacija);
- Ontologijų naudojimas (duomenų įprasminimas);
- Vizualaus turinio šablono sudarymo galimybė.

1 lentelėje pateiktas programų palyginimas. Visual Web Ripper ir Mozenda programos yra labai panašios, jos yra programos diegiamos kompiuteryje ir priklausomos nuo operacinės sistemos, o taip pat nė viena iš jų nepalaiko ontologijų, tačiau turi grafinę sąsają, kuria patogiai galima kurti tinklapio šabloną. Yahoo! Pipes programa yra internetinė aplikacija, o tai reiškia, kad ja naudotis patogiai gali bet kokią OS turintys vartotojai, tačiau ji nepalaiko ontologijų ir šablonai nėra sudaromi vizualiai matant apdorojamą tinklapį. Šiame darbe projektuojamas įrankis pasižymės visais šiais pliusais, kurių neturi šiuo metu prieinami įrankiai.

1 lentelė. Įrankių palyginimas.

Kriterijus / Programa	Internetinė aplikacija	Ontologijų naudojimas	Vizualus turinio šablono sudarymas
Visual Web Ripper	✘	✘	✓
Mozenda	✘	✘	✓
Yahoo! Pipes	✓	✘	✘
Projektuojamas įrankis	✓	✓	✓

2.3. Išvados

1. Ontologijos yra svarbi semantinio interneto dalis, padedanti struktūrizuoti duomenis.
2. Duomenų struktūrizavimui naudojami automatiniai įrankiai veikiantys pagal vartotojų aprašytas taisykles.
3. Darbe projektuojama programa išsiskiria iš analogiškų įrankių savo funkcionalumu. Žr. Lentelė 1.

3. VIZUALINIO ĮRANKIO KŪRIMO TECHNOLOGIJŲ ANALIZĖ

Pageidautina, kad tinklapių šablonų sudarymo programa galėtų naudotis kuo daugiau vartotojų. Tam programa turi būti pasiekiama nesunkiai. Lengviausią programos pasiekiamumą suteikia žiniatinklio technologijos. Per internetą prieinami resursai gali būti pasiekiami iš bet kurios vietos, o taip pat naudojant platų fizinių įrenginių ir programinės įrangos asortimentą. Interneto puslapiai ir dauguma interneto aplikacijų dažniausiai nereikalauja specialios programinės įrangos, kuri dar nebūtų įdiegta. Jos taip pat dauguma yra nepriklausomos nuo įrenginio operacinės sistemos ar platformos.

3.1. Technologijos

3.1.1. Microsoft Silverlight karkasas

MS (Microsoft) Silverlight yra įrankis, skirtas kurti interneto aplikacijoms. Tai yra nemokamas įrankis, palaikomas visose populiariausiose platformose – Windows, Mac, Linux - ir naršyklėse – Firefox, Google Chrome, Internet Explorer, ne oficialiai Opera. [8] Linux operacinėje sistemoje Silverlight technologija palaikoma per realiazaciją Moonlight. [9]

Silverlight yra programinės įrangos sistema (angl. framework) leidžiantis kurti „turtingas“ interneto taikomasias programas (angl. Rich Internet Application – RIA). Šio karkaso savybės ir tikslai yra panašūs į Adobe Flash technologiją. Ankstesnės MS Silverlight versijos buvo skirtos vaizdo ir garso pristatymo internetu problemoms, tačiau vėlesnės versijos taip pat įgalina programinės įrangos kūrėjus kurti įmantrias interaktyvias grafines sąsajas. Leidžia nesunkiai pasiekti dokumento HTML ir JavaScript elementus.

Šioje technologijoje naudojamas XML kalbos plėtinys – XAML – grafinei sąsajai kurti. Programavimas gali būti atliekamas bet kuria .NET Framework palaikoma kalba.

MS Silverlight veikia kaip interneto naršyklės priedas. Dėl to, interneto puslapiai naudojantys Silverlight komponentus yra prieinami kur kas mažesniam vartotojų skaičiui, nei tinklapiai nereikalaujantys, kad naršyklė turėtų įdiegtų papildomų priedų.

3.1.2. JavaFX platforma

JavaFX taip pat (kaip anksčiau minėtas MS Silverlight) pradėtas vystyti kaip „turtingų“ interneto aplikacijų (RIA) kūrimo įrankis, tačiau šiuo metu išleista versija įgalina kurti programinę įrangą darbastaliui, interneto naršyklėms ir mobiliesiems įrenginiams. JavaFX palaikomas Windows, Mac OS X ir Linux sistemose.

JavaFX technologijoje grafinė sąsaja aprašoma XML kalbos plėtiniu – FXML. Progarmuojama tik Java kalba. JavaFX:

- palaiko visą Java API;
- palaiko CSS stilius;
- programoje leidžia vaizduoti HTML;
- leidžia nesunkiai pasiekti dokumento HTML ir JavaScript elementus;

Jei programa yra internetinė, JavaFX kaip ir MS Silverlight reikalauja, kad interneto naršyklė turėtų įdiegtą atitinkamą priedą. Tai vėlgi mažina aplikacijų prienamumą interneto vartotojams, tačiau, lyginant su MS Silverlight, Java yra plačiau naudojama, todėl tikimybė, kad kompiuteryje jau yra įdiegtas reikalingas priedas, yra didesnė. [12][13]

JavaFX yra atviro kodo programinės įrangos sistema, vystomas projekte OpenJFX. [11]

3.1.3. Adobe Flash platforma

Adobe Flash yra multimedijos platforma, naudojama interneto puslapiuose pateikti animaciją ir vaizdo ir garso klipus, o taip pat suteikti jiems interaktyvumą. Vėlesnės versijos įgalina programuotojus šį įrankį naudoti RIA kūrimui. O taip pat Adobe sukūrė multiplatforminį programavimo karkasą, kuris leidžia kurti RIA, kuriose kombinuojama HTML, JavaScript, Flash ir Flex technologijos. [14]

Flash aplikacijos programuojamos ActionScript kalba, o grafinė sąsaja aprašoma MXML kalba – XML kalbos plėtiniu. Leidžia nesunkiai pasiekti dokumento HTML ir JavaScript elementus.

Adobe Flash naudojančios programos reikalauja, kad įrenginyje būtų įdiegtas atitinkamas priedas. Šiuo metu Flash technologija yra itin plačiai paplitusi, todėl dažniausiai ši sąlyga problelų nesukelia. [13]

3.1.4. HTML / JavaScript kalbos

HTML – tai pagrindinė kalba, kuria yra parašyta kiekvieno tinklapio struktūra. Kartu su HTML kalba tinklapių išvaizda aprašoma CSS kalba, kuri naudojama nurodyti kiekvieno HTML dokumento elemento ar jų grupės stilių, spalvas, poziciją ir kt. Kadangi HTML leidžia aprašyti tik statinę tinklapio struktūrą, dinaminiams elementams naudojama JavaScript kalba. Ši kalba leidžia pasiekti RIA funkcionalumą analogišką kitoms nagrinėtoms technologijoms.

HTML + CSS + JavaScript derinys yra palaikomas beveik visų naršyklių nenaudojant papildomų trečiųjų šalių priedų. Ši technologija nepriklauso nuo operacinės sistemos. Tai leidžia kurti internetines aplikacijas nesunkiai prieinamas visiems besinaudojantiems vienu iš daugelio įrengimų, gebančių vaizduoti interneto puslapių turinį.

3.1.5. Technologijų palyginimas

Bandant pasirinkti technologiją, kuri bus naudojama projektui įgyvendinti, reikia atsižvelgti į tokius programos vertinimo kriterijus:

- pasiekiamumas vartotojui;
- sudėtingumas programos dalių serverio ir kliento pusėse;
- technologijos galimybės;

Technologijas pagal jų suteikiamą programos pasiekiamumą vartotojui galima palyginti atsižvelgus į tai, kiek platformų ir įrenginų tipų gali leisti programą. Programa taikoma daugiausia asmeniniams kompiuteriams, todėl lyginamos populiariausios asmeninių kompiuterių naršyklių ir programavimo įrankių naujausios versijos rašymo metu.

2 lentelė. Naršyklių palaikomos technologijos.

Naršyklė / Technologija	Internet Explorer	Mozilla Firefox	Google Chrome	Opera	Safari	Nereikalauja priedo	Paplitimas*
MS Silverlight	✓	✓	✓	O	✓	✘	68.26%
JavaFX	✓	✓	✓	✓	✓	✘	68.70%
Adobe Flash / Flex	✓	✓	✓	✓	✓	✘	95.66%
HTML / JavaScript	✓	✓	✓	✓	✓	✓	100%

*Paplitimas rodo, kuri dalis interneto vartotojų turėjo sistemoje įdiegtą reikalingą priedą 2012 metų balandį. [15]

O – neoficialiai;
✓ – palaiko/taip;
✘ – nepalaiko/ne;

Kaip matyti iš lentelės 2, visos nagrinėjamos technologijos yra palaikomos visose populiariausiose naršyklėse. Lyginimas sutampa atlikus tyrimą operacinėse sistemose Windows 7, Mac OS X 10.5, Ubuntu.

Stulpelis „paplitimas“ rodo, kad Adobe Flash platforma yra labiausiai paplitusi iš trijų platformų, kurios reikalauja įdiegto papildinio ir, kad lenkia kitas technologijas virš 25% visų vartotojų. Tačiau HTML su JavaScript technologija pagal paplitimą yra kur kas pranašesnė už kitas, nes tai yra pagrindinė interneto puslapių naudojama technologija, kurią palaiko visos naršyklės be papildomų priedų.

Lyginant technologijas pagal sudėtingumą, reikia įvertinti programuotojų patirtį ir technologijos mokymosi slenkstį.

Vertinant technologijų galimybes reikia atsižvelgti, kaip jos padeda įgyvendinti projekto užduoties sprendimą. Kliento pusėje inklapių šablonų sudarymo programoje yra poreikis manipuliuoti HTML elementais, kurie yra analizuojamame tinklapyje. Tai daryti tiesiogiai leidžia tik JavaScript kalba. Serverio pusėje programa turi vykdyti duomenų mainus tarp programos sąsajos (kliento dalies) ir duomenų bazės. Tai gali būti vykdoma tik naudojant serverio pusės programavimą.

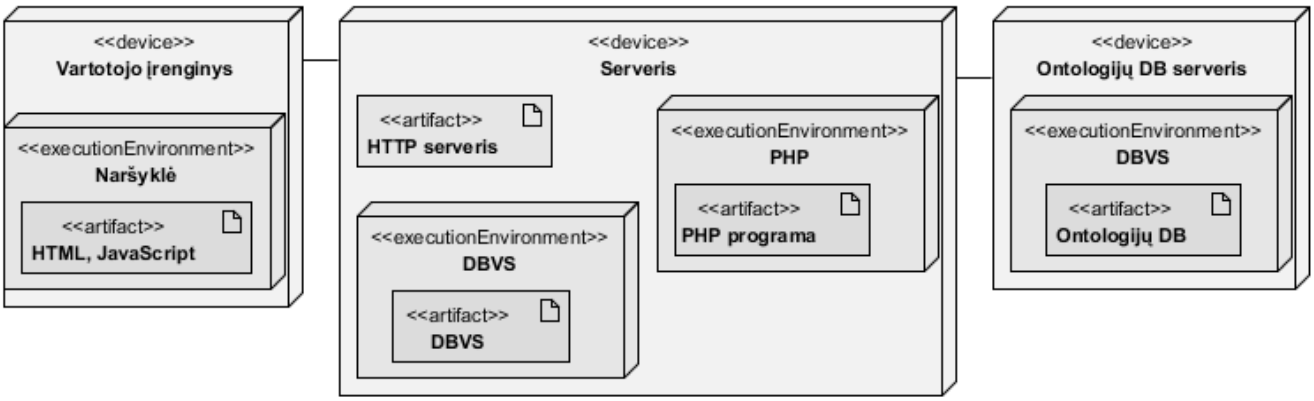
3.2. Darbo technologijų analizės išvados

Siekiant padaryti programą pasiekiamą kuo daugiau vartotojų, reikia vengti priklausomybės nuo vartotojo programinės įrangos, kadangi skirtingi vartotojai gali turėti labai platų asortimentą įvairios programinės įrangos, kuri veikia įvairiose platformose. Todėl projektuojamas tinklapių šablonų sudarymo įrankis turėtų būti kuriamas naudojant gryną HTML ir JavaScript kalbas. Šios technologijos yra standartizuotos ir plačiai naudojamos visame internete.

4. ŠABLONŲ SUDARYMO ĮRANKIO KONCEPCINIAI MODELIAI

4.1. Projektas

Šablonų sudarymo sistemos struktūrinė schema pateikta Pav. 10. Sistema susideda iš ontologijų duomenų bazės, kurioje yra saugomos visos naudojamos ontologijos. Sistemoje duomenys yra apdorojami PHP programos, kuri dirba internetiniame serveryje. Jame taip pat veikia duomenų bazių valdymo sistema, kurioje laikoma tinklapių šablonų ir vartotojų duomenų bazė. Čia dirba HTTP serveris, kuris atsako į užklausas iš sistemos vartotojo įrenginiuose dirbančių interneto naršyklių. Naršyklės atlieka visas grafinės sąsajos operacijas apdorodamos HTML ir JavaScript kalbas.



10 pav. Sistemos struktūra

Sistemos tikslas yra leisti vartotojui sukurti tinklapio WPT šabloną naudojantis grafine sąsaja. Pirmiausia vartotojas tinklapio pagalba gali sukurti ontologiją, pagal kurią vėliau bus sudarinėjamas šablonas.

4.2. Funkciniai reikalavimai tinklapių šablonų sudarymo sistemai

4.2.1. Sistemos naudojimo sekos diagrama

Sistemos veiksmų sekos diagrama pateikta 12 pav. Vartotojas prisijungęs prie sistemos gali užkrauti norimą analizuoti tinklapį ir naudodamasis valdymo sąsaja sudarinėti tinklapio šabloną naršyklėje. Kai vartotojas patvirtina šabloną, jis pagal raktinius žodžius yra susiejamas su atitinkama ontologija, kuri saugoma ontologijų duomenų bazėje. Po to šablonas išsaugomas šablonų duomenų bazėje.

4.2.2. Duomenų bazė

Suprojektuota duomenų bazė susideda iš dviejų lentelių :

1. **ontologies** – saugo ontologijas. Ontologija saugoma teksto XML formatu, kurio kiekvienas įrašas identifikuojamas pavadinimo lauku. Lentelėje 3 pateikti duomenų bazės lentelės „ontologies“ laukų paaiškinimai.

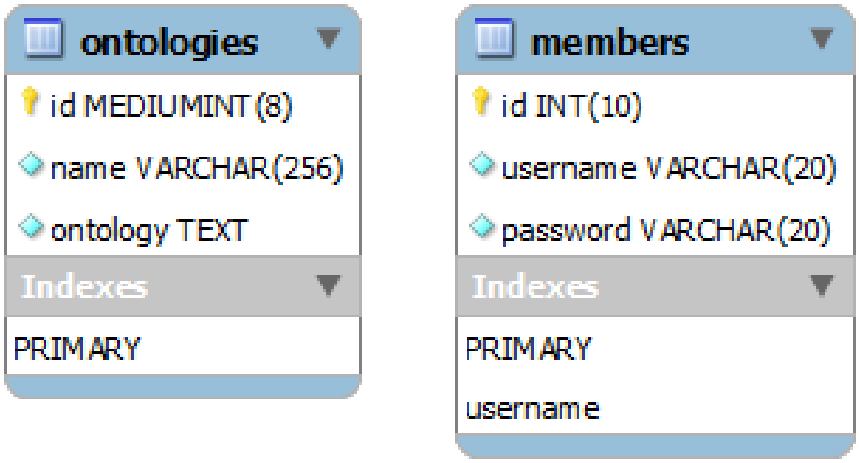
3 lentelė. Duomenų bazės lentelės „ontologies“ laukų paaiškinimai

Lauko pavadinimas	Tipas	Paskirtias
id	sveikas skaičius	unikalus lentelės įrašo identifikatorius
name	simbolių eilutė (256)	saugo unikalų ontologijos pavadinimą
ontology	tekstas	saugo ontologijos XML tekstą

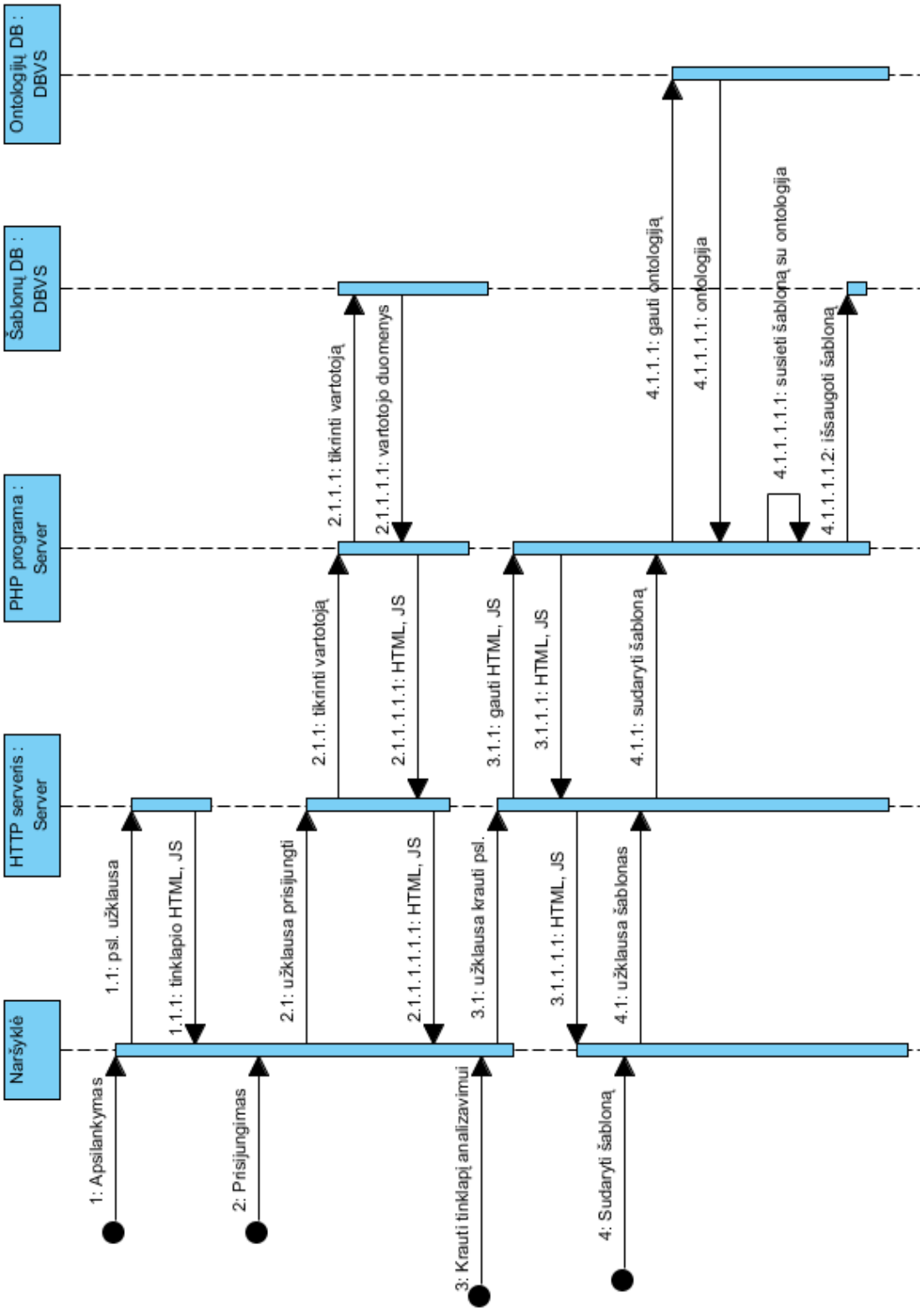
2. **members** – saugo informaciją apie vartotojus. Naudojama saugiam vartotojų prisijungimui.

4 lentelė. Duomenų bazės lentelės „members“ laukų paaiškinimai

Lauko pavadinimas	Tipas	Paskirtias
id	sveikas skaičius	unikalus lentelės įrašo identifikatorius
username	simbolių eilutė (20)	saugo unikalų vartotojo vardą
password	simbolių eilutė (20)	saugo vartotojo slaptažodį



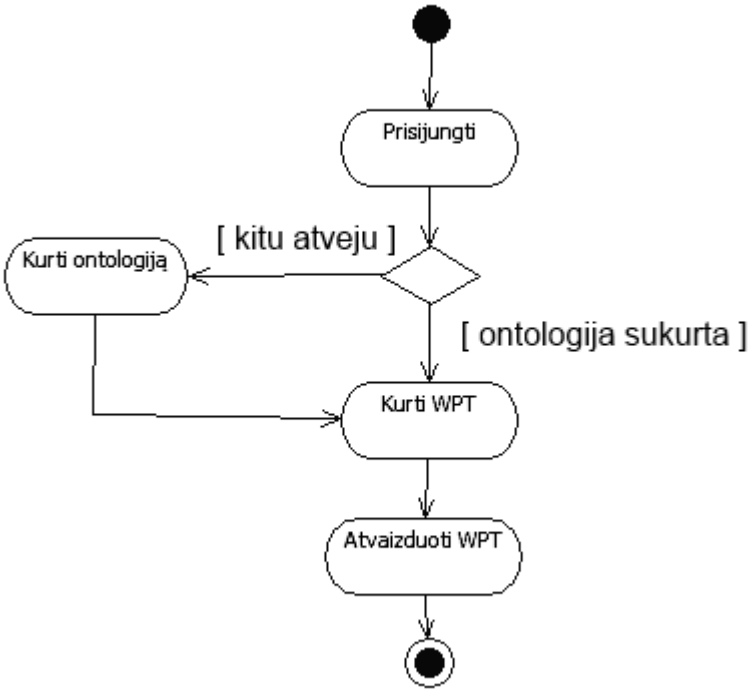
11 pav. Sistemos duomenų bazė



12 pav. Sistemos veiksmų sekos diagrama

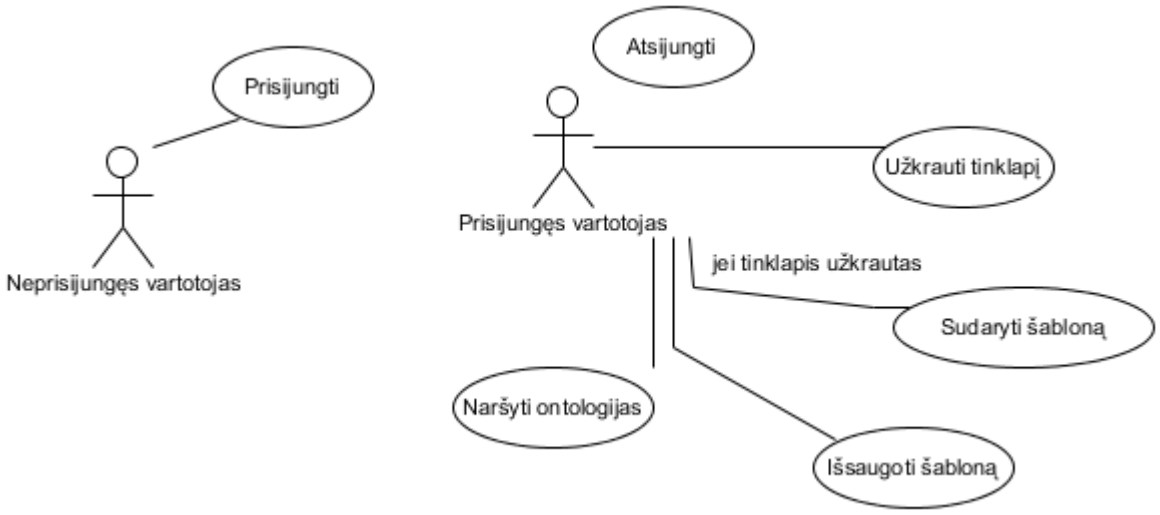
4.2.3. Sistemos vartotojų funkcijos

Tinklapių šablonų sudarymo sistemos veiklos diagrama pateikta 13 pav. Prisijungęs vartotojas gali kurti ontologiją, jei jo norima naudoti ontologija nėra sukurta. Kai ontologija yra išsaugota sistemoje, vartotojas gali pradėti WPT kūrimą. Sukūręs WPT, sistemos naudotojas gali gauti sukurtą WPT XML tekstą tolesniam naudojimui.



13 pav. Sistemos veiklos diagrama.

Sistemos vartotojų funkcijų panaudos atvejų diagrama pavaizduota pav. 14. Sistemos vartotojas, norintis prieiti prie sistemos ir jos duomenų, būtinai turi prie jos prisijungti.



14 pav. Sistemos vartotojų funkcijos

Žemiau (5 lentelė) pateikti sistemos vartotojų panaudos atvejų diagramos paaiškinimai.

5 lentelė. Sistemos vartotojų panaudos atvejų diagramos paaiškinimai.

Panaudos atvejis	**Prisijungti**
Aktoriai	Neprisijungę vartotojai
Išankstinės sąlygos	Vartotojas turi būti neprisijungęs, atsidaręs tinklapį
Įvykis sukeliantis atvejį	Vartotojas suveda prisijungimo duomenis ir paspaudžia „prisijungti“
Sąlygos po panaudos	Vartotojas įgyja teisę naudotis sistema
Panaudos atvejis	**Atsijungti**
Aktoriai	Prisijungęs vartotojas
Išankstinės sąlygos	Vartotojas prisijungęs
Įvykis sukeliantis atvejį	Vartotojas naudoja sąsają funkcijai išvkiesti
Sąlygos po panaudos	Vartotojas nebeturi teisės naudotis sistema
Panaudos atvejis	**Užkrauti tinklapį**
Aktoriai	Prisijungęs vartotojas
Išankstinės sąlygos	Vartotojas turi būti prisijungęs
Įvykis sukeliantis atvejį	Naudojama grafinė sąsaja funkcijai iškviesti
Sąlygos po panaudos	Ekrane vaizduojamas ir į atmintį įrašomas analizuojamo tinklapio tekstas
Panaudos atvejis	**Sudaryti šabloną**
Aktoriai	Prisijungęs vartotojas
Išankstinės sąlygos	Užkrautas tinklapis
Įvykis sukeliantis atvejį	Vartotojas naudoja grafinę sąsają funkcijai iškviesti.
Sąlygos po panaudos	Vartotojo kompiuterio atmintyje saugomas tinklapio šablonas
Panaudos atvejis	**Išsaugoti šabloną**
Aktoriai	Prisijungęs vartotojas
Išankstinės sąlygos	Vartotojas turi būti prisijungęs
Įvykis sukeliantis atvejį	Naudojama grafinė sąsaja funkcijai iškviesti
Sąlygos po panaudos	Šablonas įrašomas į duomenų bazę
Panaudos atvejis	**Naršyti ontologijas**
Aktoriai	Prisijungęs vartotojas
Išankstinės sąlygos	Vartotojas turi būti prisijungęs
Įvykis sukeliantis atvejį	Naudojama grafinė sąsaja funkcijai iškviesti
Sąlygos po panaudos	Nekinta

4.3. Nefunkciniai reikalavimai tinklapių šablonų sudarymo programai

Projektuojama programa turi pasižymėti tam tikromis savybėmis, kurios įgalintų vartotoją naudotis sistema greitai, patogiai ir saugiai.

Visų pirma, visi duomenys, kuriuos pateikia programos naudotojas, turi būti nepasiekiami pašaliniams asmenims, kurie neturi teisių tuos duomenis valdyti ar peržiūrėti. Vartotojų slaptažodžiai saugomi užkoduoti.

Šablonų sudarymo programa yra internetinė aplikacija, kuri ne darbo metu nelaiko vartotojo įrengimo atmintyje jokios informacijos. Programa turi būti suderinama su naujausiomis populiariausių naršyklių versijomis.

Ji turi turėti išplėtimo galimybę. Esant poreikiui, programa gali būti papildoma naujais moduliais ar funkcijomis.

Programos vartotojo sąsaja turi būti intuityvi. Visos programos funkcijos pasiekiamos per sąsają.

Programos atsako į vartotojo komandas laikas turi būti pakankamai trumpas, kad nesukeltų vartotojui diskomforto naudojantis programa.

Programinė įranga turi būti atspari klaidingai vartotojų įvesčiai. Taip pat programa turi teisingai susidoroti su nekorektiškais duomenimis, kurie laikomi duomenų bazėje. Įrankis neturi netikėtai nutraukti darbą.

Atsižvelgiant į planuojamas naudoti technologijas galime kelti techninius reikalavimus sistemai. Projektuojama sistema yra tinklapis, todėl jai talpinti yra reikalingas serveris. Norint sistema naudotis, taip pat reikalingas kompiuteris, kuris gali būti tas pats, kuriame tinklapis yra patalpintas, nors paprastai tai būna kita mašina.

Techniniai programinės ir aparatūrinės įrangos reikalavimai yra tokie:

1. Serveris:
    1) IBM PC;
    2) HTTP serverinė programinė įranga, palaikanti PHP 5.3.25. Pavyzdžiui, Apache HTTP serveris arba IIS su PHP.
    3) PHP 5.3.25;
    4) Techniniai aparatūrinės įrangos reikalavimai priklauso nuo pasirinktos serverio programinės įrangos.

2. Kliento kompiuteris:
    1) IBM PC;
    2) Pelė;
    3) Reikalinga moderni naršyklė gebanti apdoroti JavaScript – Google Chrome 27, Mozilla Firefox 21, Safari 6, Opera 12.15, Internet Explorer 9 ar naujesnės naršyklių versijos.

5. VIZUALINIO ŠABLONO SUDARYMO ĮRANKIO PROTOTIPAS

Projektas realizuotas naudojant PHP, HTML, JavaScript kalbas. Sąsajai aprašyti pasitelkta CSS kalba.

Projekte naudojamos trečiųjų šalių bibliotekos:

1) JavaScript jQuery 1.9.1 biblioteka.
2) vkbeautify.0.99.00.beta biblioteka.

Projektui įgyvendinti naudojama programinė įranga:

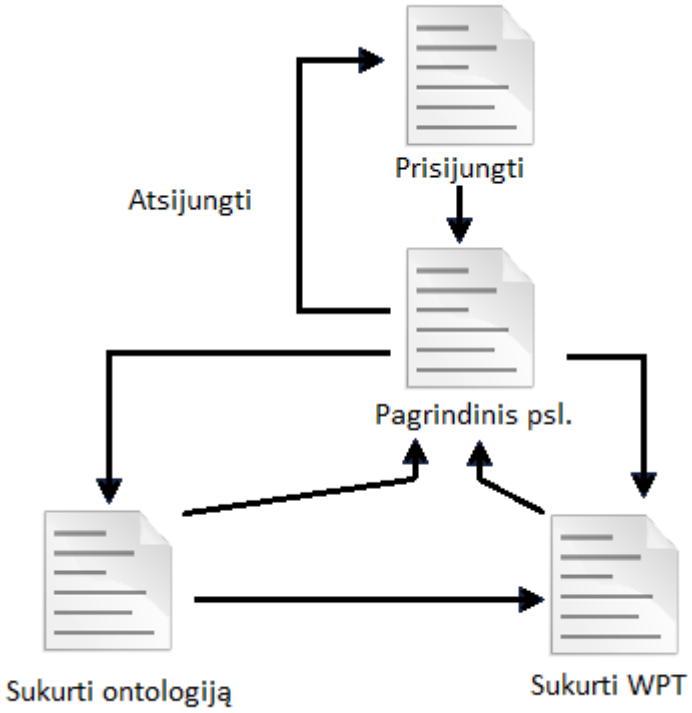
1) Windows 7 Ultimate SP 1 64bit;
2) Apache HTTP 2.2.22 serveris;
3) PHP 5.4.8;
4) MySQL 5.5.28 serveris.

Eksperimentui naudojama techninė įranga - IBM PC:

1) Intel Core I3-2120 CPU;
2) RAM 8 GB.

5.1. Įgyvendintas programos prototipas

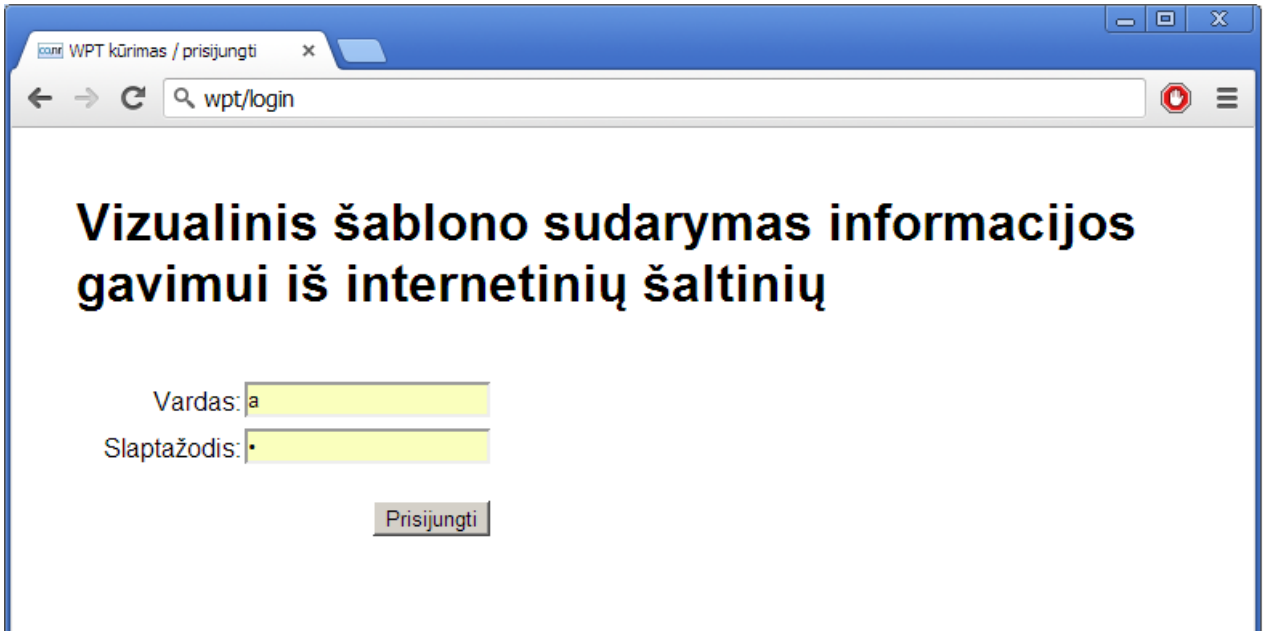
Sukurto tinklapio navigacijos schema pavaizduota 15 pav. Atėjęs į tinklapį naudotojas negali pasiekti jokio puslapio, išskyrus autorizacijos puslapį. Prisijungęs (autorizavęsis sistemoje) vartotojas patenka į pagrindinį puslapį, iš kurio gali naviguoti į ontologijų kūrimo puslapį arba WPT šablono kūrimo puslapį. Taip pat iš pagrindinio puslapio prisijungęs vartotojas gali atsijungti nuo sistemos. Iš ontologijos kūrimo puslapio vartotojas gali grįžti į pagrindinį puslapį arba pereiti prie WPT šablono kūrimo. Iš WPT šablono kūrimo puslapio vartotojas gali grįžti į pagrindinį puslapį.



15 pav. Tinklapio navigacijos schema

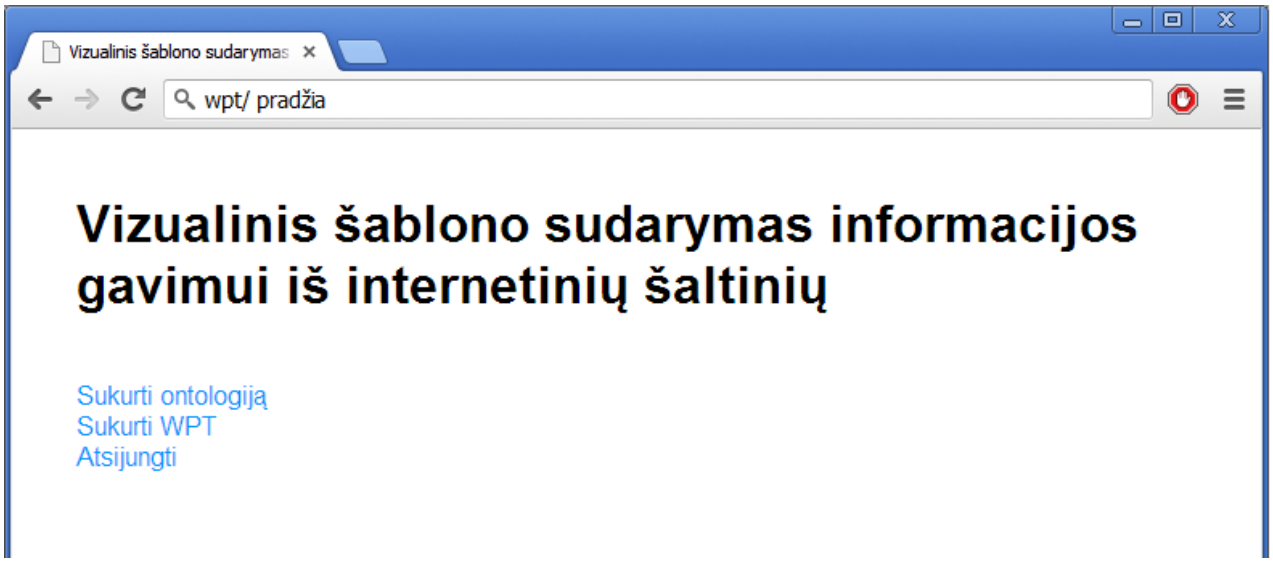
5.2. Tinklapių šablono sudarymo programos veikimo metodika

Kadangi programa parašyta internetinės aplikacijos principu, programos diegti vartotojo kompiuteryje nereikia. Pakanka naršykle kreiptis adresu HTTP serverio, kuriame įdiegta aplikacija.



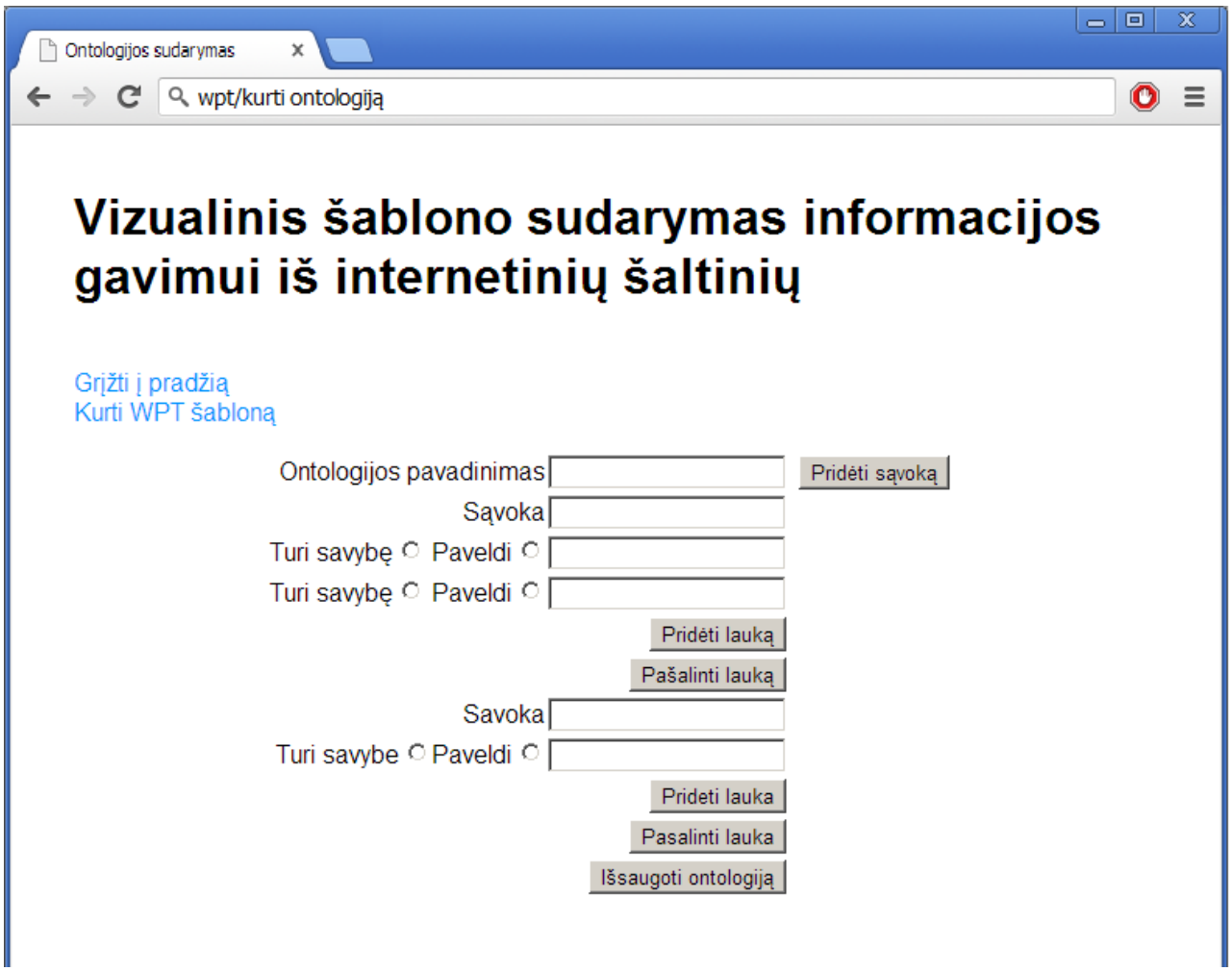
16 pav. Prisijungimo puslapis

Jei vartotojas nėra prisijungęs ir bando pasiekti bet kurį svetainės puslapį, lankytojas nukreipiamas į prisijungimo puslapį, kurio ekranvaizdis pateiktas 16 pav. Prisijungęs vartotojas patenka į pagrindinį svetainės puslapį. Pradinio tinklapio puslapio vaizdas pateiktas 17 pav. Iš čia lankytojas gali rinktis sukurti ontologiją arba sukurti WPT šabloną. Taip pat iš šio pradinio puslapio galima atsijungti nuo sistemos.



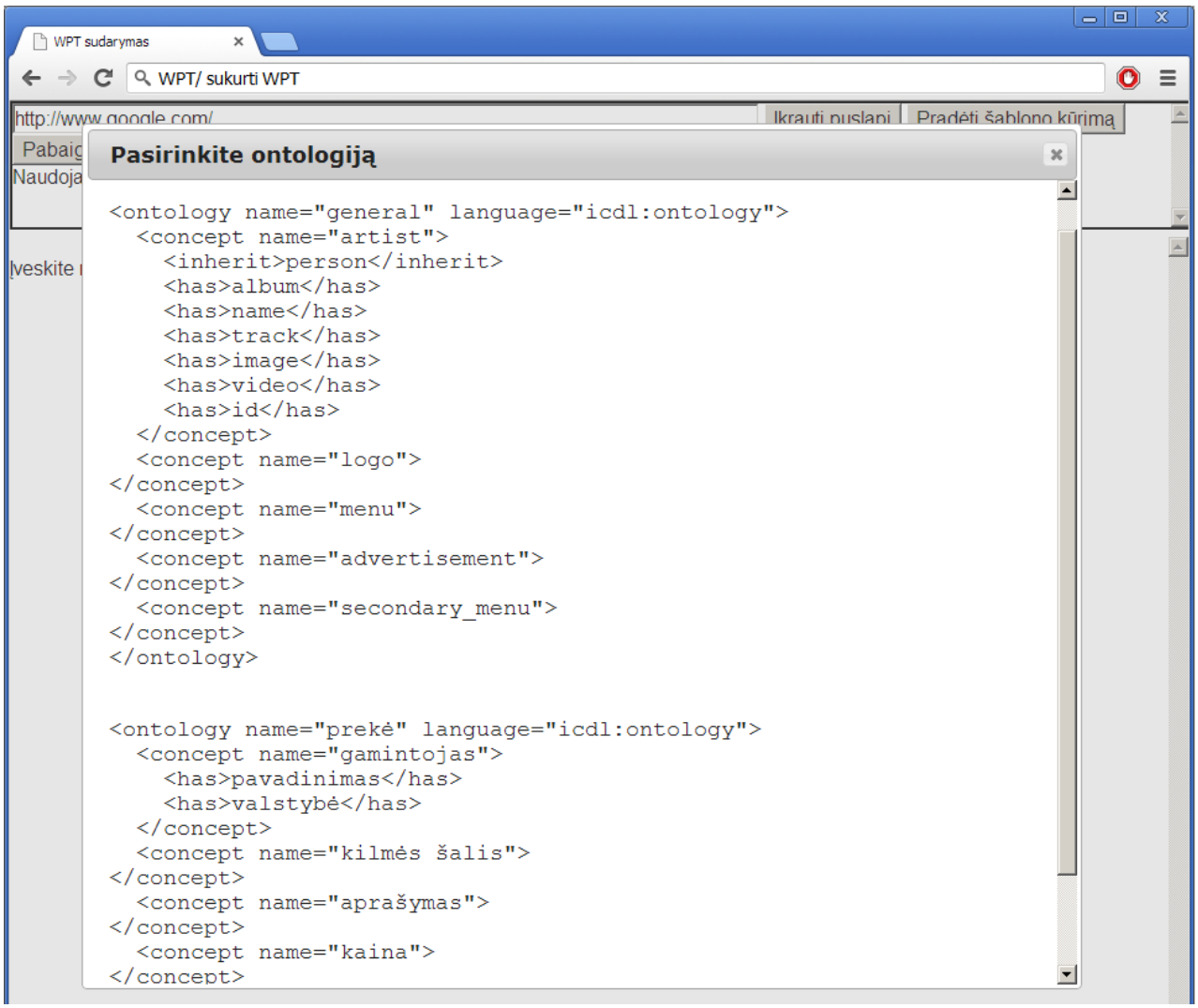
17 pav. Pradinis tinklapio puslapis

Ontologijos kūrimo puslapyje vartotojui pateikiama forma, kurios pagalba galima pridėti ontologijos objektus ir jų savybes. Ontologijos kūrimo puslapio vaizdas pateiktas 18 pav. Taip pat čia yra meniu, leidžiantis grįžti į pradinį puslapį arba pereiti prie WPT šablono komponavimo.



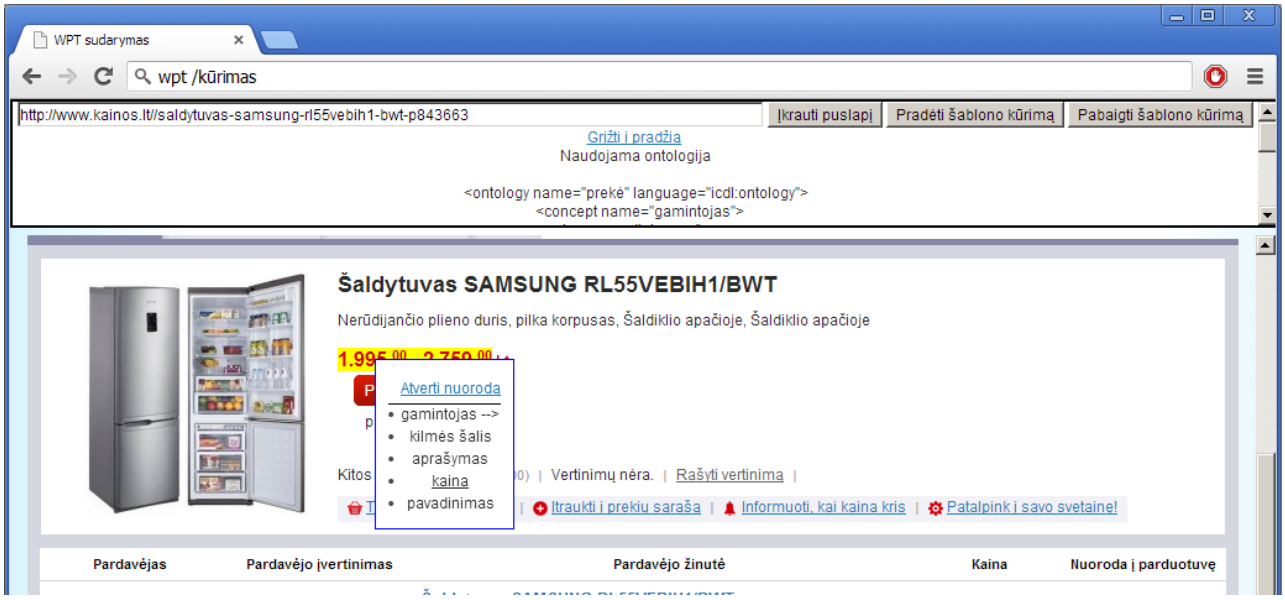
18 pav. Ontologijos kūrimo puslapis

Apsilankius WPT šablono kūrimo puslapyje. Vartotojui yra pateikiamas visų sistemoje sukurtų ontologijų sąrašas, iš kurio prašoma pasirinkti ontologiją, kuri bus naudojama kuriant šabloną. Ontologijos pasirinkimo dialogo vaizdas pateiktas 19 pav. Pasirinkus ontologiją, ji yra parodoma sąsajos viršuje, pagal ją sukuriamas kontekstinis meniu, kurio pagalba analizuojamo tinklapio elementams priskiriamos ontologijos reikšmės.



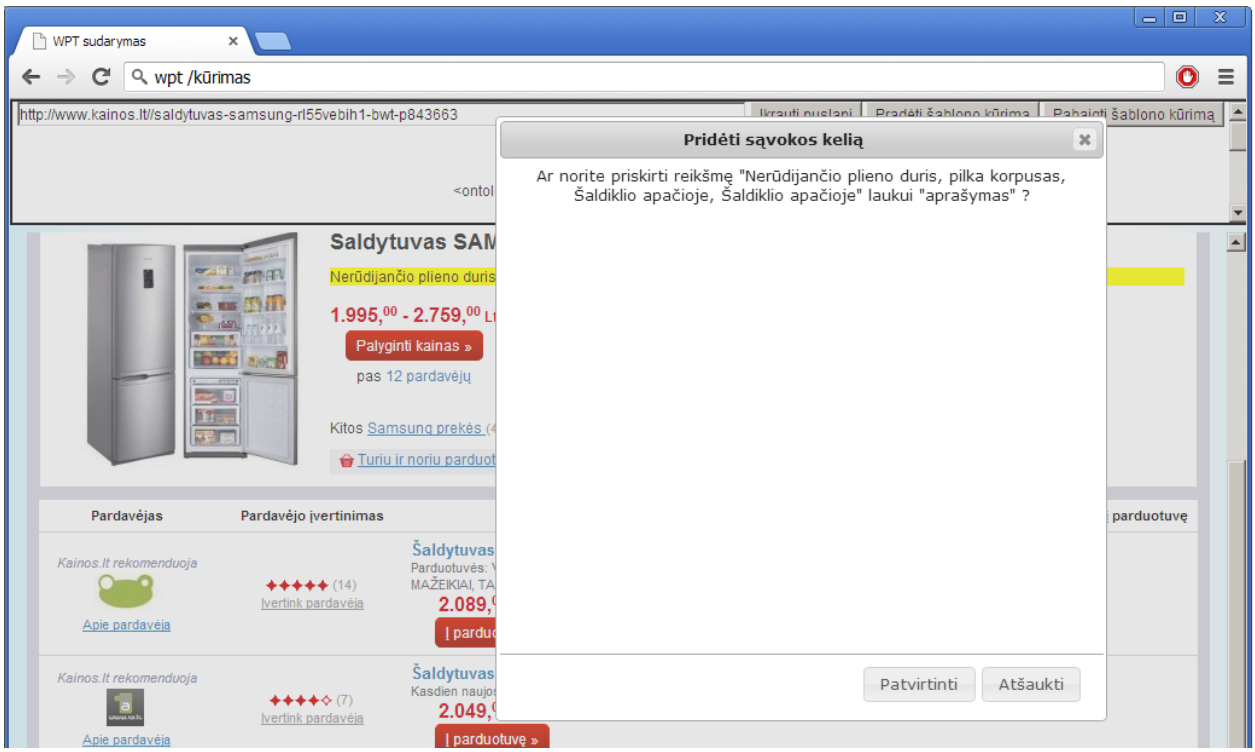
19 pav. Ontologijos pasirinkimas

Pasirinkus tinklapį galima pradėti kurti šabloną. Kai šablono kūrimas pradedamas, užkrautame tinklapyje veikia kontekstinis meniu, kurio pagalba galima susieti ontologijos laukus su informacija, vaizduojama tinklapyje. Meniu pavyzdys pateiktas 20 pav.



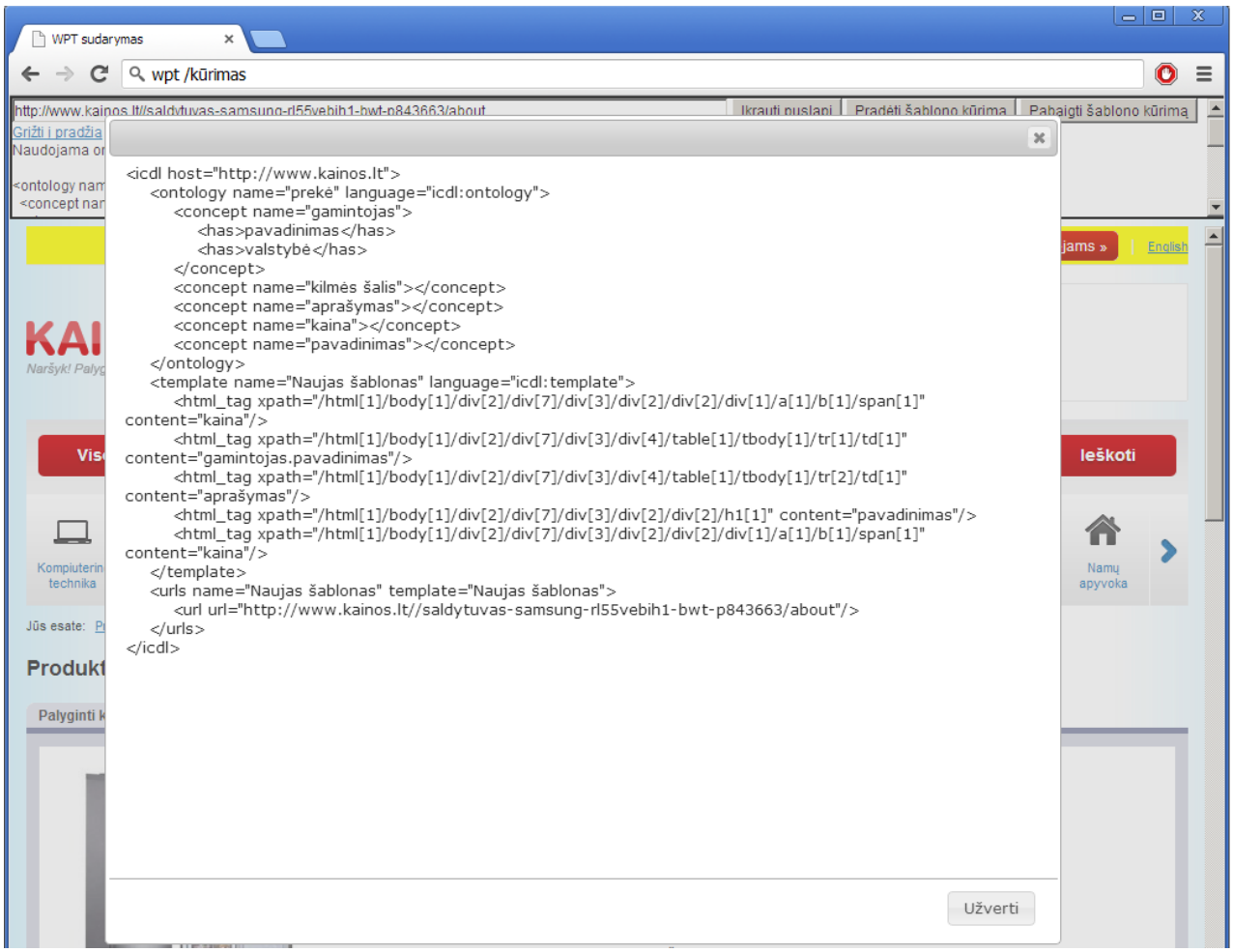
20 pav. WPT kūrimo meniu

Pasirenkant lauko reikšmę, programa užklausia patvirtinimo, kurio vaizdas pateiktas 21 pav.



21 pav. Patvirtinimas susiejant reikšmę su ontologijos lauku

Pabaigus kurti šabloną, spaudžiamas mygtukas „Pabaigti šablono kūrimą“. Tada vartotojui yra pateikiamas sugeneruotas WPT šablonas. Šablono pateikimo dialogas parodytas 22 pav.



22 pav. Šablono pateikimo dialogas

6. REZULTATAI

Sukurta sistema testuojama tikrinant jos atitikimą reikalavimams. Pagrindiniai reikalavimai jos funkcionalumui yra:

1. Programa turi gebėti kurti ontologiją.
2. Programa turi gebėti panaudoti sukurtą ontologiją WPT šablono kūrimui.

Vykdomas ontologijos kūrimas. Suprojektuoto ir įgyvendinto įrankio pagalba sukurta prekės ontologija pateikta Kodo pavyzdys 11.

Kodo pavyzdys 11. Sukurta ontologija.

```
<ontology name="prekė" language="icdl:ontology">
    <concept name="gamintojas">
        <has>pavadinimas</has>
        <has>valstybė</has>
    </concept>
    <concept name="kilmės šalis"></concept>
    <concept name="aprašymas"></concept>
    <concept name="kaina"></concept>
    <concept name="pavadinimas"></concept>
</ontology>
```

Sukurta ontologija atitinka WPT šablone naudojamoms ontologijoms keliamus reikalavimus, todėl ją galima naudoti WPT šablono kūrimui.

Vykdomas WPT šablono kūrimas. Testavimui naudojamas tinklapis „http://www.kainos.lt//saldytuvas-samsung-rl55vebih1-bwt-p843663/about“ pateikiantis informaciją apie Šaldytuvą SAMSUNG RL55VEBIH1/BWT. Atlikus WPT šablono kūrimo veiksmus susiejant keturis informacijos laukus su atitinkamais ontologijos laukais, gaunamas WPT šablonas pateiktas Kodo pavyzdys 12.

Kodo pavyzdys 12. Sugeneruotas WPT šablonas.

```
<icdl host="http://www.kainos.lt">
    <ontology name="prekė" language="icdl:ontology">
        <concept name="gamintojas">
            <has>pavadinimas</has>
            <has>valstybė</has>
        </concept>
        <concept name="kilmės šalis"></concept>
        <concept name="aprašymas"></concept>
        <concept name="kaina"></concept>
        <concept name="pavadinimas"></concept>
    </ontology>
    <template name="Naujas šablonas" language="icdl:template">
        <html_tag
xpath="/html[1]/body[1]/div[2]/div[7]/div[3]/div[4]/table[1]/tbody[1]
/tr[1]/td[1]" content="gamintojas.pavadinimas"/>
        <html_tag
xpath="/html[1]/body[1]/div[2]/div[7]/div[3]/div[4]/table[1]/tbody[1]
/tr[2]/td[1]" content="aprašymas"/>
        <html_tag
xpath="/html[1]/body[1]/div[2]/div[7]/div[3]/div[2]/div[2]/h1[1]"
content="pavadinimas"/>
        <html_tag
xpath="/html[1]/body[1]/div[2]/div[7]/div[3]/div[2]/div[2]/div[1]/a[1
]/b[1]/span[1]" content="kaina"/>
    </template>
    <urls name="Naujas šablonas" template="Naujas šablonas">
        <url url="http://www.kainos.lt//saldytuvas-samsung-
rl55vebih1-bwt-p843663/about"/>
    </urls>
</icdl>
```

Matome, kad sugeneruotas WPT šablonas atitinka organizacijos OMFICA nustatytą specifikaciją. Su papildoma programine įranga, kurią naudoja OMFICA, šablonas gali būti naudojamas sudaryti tinklapio puslapio RDF schemai.

6.1. Išvados

Realizuotas vizualinis tinklapių šablono sudarymo įrankis geba sudaryti semantinį tinklapio šabloną, kuris gali būti naudojamas interneto robotų informacijos ir metaduomenų gavybai. Įrankis pasižymi tuo, kad jo naudojimas nereikalauja papildomos programinės įrangos ir yra pasiekiamas internetu.

7. IŠVADOS

1. Atlikus žiniatinklio semantizavimo metodų analizę buvo nustatyta problemos sritis. Šiuo metu palyginti maža dalis informacijos yra suprantama mašinoms, nes informacija yra pritaikyta skaityti žmonėms, kai tuo tarpu kompiuteriai reikalauja griežtesnės informacijos struktūros – semantinės informacijos.
2. Esamų semantizavimo įrankių analizė leido susidaryti pagrindą interneto semantizavimo įrankio projekto kūrimui. Buvo projektuojamas įrankis, kuriuo naudotis galėtų kuo daugiau suinteresuotų žmonių, todėl pasirinkta įgyvendinti grafinę sąsają turintį įrankį, kuris būtų pasiekiamas internetu, nereikalaujant papildomos programinės įrangos.
3. Sukurtas grafinio įrankio prototipas, kurio pagalba galima kurti interneto svetainių puslapius semantizuojančius šablonus.
4. Sukurtas įrankis yra daug vartotojų palaikanti sistema, pasiekiama internetu, todėl gali būti naudojama iš daugelio interneto ryšį turinčių įrenginių.
5. Suprojektuotas ir įgyvendintas prototipas galėtų būti toliau plėtojamas pridedant naują funkcionalumą. Pavyzdžiui, gilesniam informacijos semantizavimui reikia susieti naudojamas ontologijas su standartizuotomis ir šiuo metu viešai naudojamomis ontologijomis.

8. LITERATŪROS ŠALTINIAI

[1] Großer Beleg, „Ontology Composition using a Role Modeling Approach“, 2007. Prieiga per internetą: http://mp.binaervarianz.de/ontology_composition_gb.pdf

[2] „Website Parse Template“ [žiūrėta 2011-10-15]. Prieiga per internetą: http://www.omfica.org/npo_website_template.php

[3] „A Good Role Model for Ontologies: Collaborations“, 2010. Prieiga per internetą: http://www.igi-global.com/viewtitlesample.aspx?id=39044

[4] OWL 2 apžvalga [žiūrėta 2011-10-22]. Prieiga per internetą: http://www.w3.org/TR/owl2-overview/

[5] „Web harvesting demonstration video“ [žiūrėta 2011-11-09]. Prieiga per internetą: http://www.visualwebripper.com/Demonstrations/Popup.aspx?video=IntroVideoHD&x=1280&y=720

[6] „Web data scraping videos“ [žiūrėta 2011-11-10]. Prieiga per internetą: http://www.mozenda.com/video01-overview

[7] „Yahoo! Pipes“ programa [žiūrėta 2011-11-20]. Prieiga per internetą: http://pipes.yahoo.com/pipes/pipe.edit

[8] Microsoft Silverlight Frequently Asked Questions [žiūrėta 2012-05-20]. Prieiga per internetą: http://www.microsoft.com/silverlight/what-is-silverlight/#sys-req

[9] Moonlight [žiūrėta 2012-05-22]. Prieiga per internetą: http://www.mono-project.com/Moonlight

[10] JavaFX Frequently Asked Questions [žiūrėta 2012-05-22]. Prieiga per internetą: http://www.oracle.com/technetwork/java/javafx/overview/faq-1446554.html

[11] OpenJFX projektas [žiūrėta 2012-05-22]. Prieiga per internetą: http://openjdk.java.net/projects/openjfx/

[12] Java naudojimo statistika [žiūrėta 2012-06-10]. Prieiga per internetą: http://w3techs.com/technologies/details/pl-java/all/all

[13] Kliento pusės programavimo kalbų naudojimo statistika [žiūrėta 2012-06-10]. Prieiga per internetą: http://w3techs.com/technologies/overview/client_side_language/all

[14] Adobe Flex [žiūrėta 2012-12-11]. Prieiga per internetą: http://www.adobe.com/products/flex.html

[15] Flash, Java, SilverLight technologijų palaikymo statistika [žiūrėta 2012-12-10]. Prieiga per internetą: http://www.statowl.com/custom_ria_market_penetration.php

[16] Tim Berners-Lee, James Hendler, Ora Lassila, „The Semantic Web“, 2001-05-17 [žiūrėta 2013-04-20]. Prieiga per internetą: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[17] W3C dažnai užduodami klausimai (DUK) apie semantinį tinklą [žiūrėta 2013-04-20]. Prieiga per internetą: http://www.w3.org/RDF/FAQ

[18] RDF pradžiamokslis – W3C rekomendacija. „RDF Primer“, 2004-02-10 [žiūrėta 2013-05-10]. Prieiga per internetą: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[19] GRDDL pradžiamokslis – W3C rekomendacija „GRDDL Primer“, 2007-09-11 [žiūrėta 2013-05-10]. Prieiga per internetą: http://www.w3.org/TR/2007/REC-grddl-20070911/

[20] RDFa 1.1 pradžiamokslis – W3C rekomendacija „RDFa 1.1 Primer“, 2012-06-07 [žiūrėta 2013-05-10]. Prieiga per internetą: http://www.w3.org/TR/2012/NOTE-rdfa-primer-20120607/

[21] WHATWG Microdata standartas „Microdata“, 2013-05-05 [žiūrėta 2013-05-10]. Prieiga per internetą: http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html#microdata

[22] F.Dawson, Lotus, D.Stenerson, Microsoft, „Internet Calendaring and Scheduling Core Object Specification (iCalendar)“, 1998-11 [žiūrėta 2013-05-11]. Prieiga per internetą: http://www.ietf.org/rfc/rfc2445.txt

9. Priedai

Lentelių sąrašas

Paveikslų sąrašas