VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis magistro darbas

Tinklapių medinės struktūros radimas, išgavimas ir vizualizacija

Semi-supervised semi-interactive methods to identify, extract and visualize navigational menus from Web sites

Atliko: 2 kurso, 9 grupės studentas
Gytis Sušinskas (parašas)

Darbo vadovas:
Dr. Artūras Mažeika (parašas)

Vilnius

2012

TURINYS

LENTELĖS

PAVEIKSLAI

ANOTACIJA

Baigiamajame magistro darbe pateikti praktiniai sprendimai, leidžiantys efektyviai rasti, išgauti ir vizualizuoti tinklapių medinę struktūrą. Tam yra sukurtas vizualiai tinklapių struktūros analizei atlikti skirtas instrumentas. Darbo tikslui pasiekti atlikta mokslinio potencialo analizė, parinkti efektyvūs metodai ir papildyti, atsižvelgiant į sprendžiamą darbo mokslinę ir praktinę problemą. Praktiniame darbe panaudojant class menu artumo bei urlmetodus ir taikant Prefuse vizualizaciją sukurtas instrumentas, leidžiantis ženkliai pagerinti svetainės navigaciją, struktūrą bei patobulinti svarbiausios informacijos pateikimą lankytojui. Gauti rezultatai leidžia efektyviai analizuoti, kurti, plėtoti ir vertinti tinklapių medinės struktūros navigaciją, tinklapio elementų ryšius ir vizualizacijos procesą..

Raktiniai žodžiai: tinklapio struktūros vizualizacija, tinklapio struktūros navigacija, class meniu artumo metodas, url metodas, Prefuse paketas.

Name of the project:
Semi-supervised semi-interactive methods to identify, extract and visualize navigational menus from Web sites

Author:
Gytis Sušinskas, Masters degree student at Vilnius University, Faculty of Mathematics and Informatics.

SUMMARY

This Master‘s thesis is contemporary and relevant as more efficient and convenient control of sitemap navigation and visualization is beneficial both to the user and creator of such systems. A simpler, faster sitemap, more effective visualization and more convenient structural menu are comprehensively valuable for website users. Therefore the object of this thesis is the creation of the tool which would ensure the visual analysis of site map.

This Master‘s thesis contains practical solutions how to make sitemap data extracting, filtering and rendering more effective. The goal of this thesis is to create a tool, ensuring a more efficient sitemap data extracting, filtering and visualization.

Therefore to achieve such goal, author forms the following tasks: to research and detect most efficient methods of work; to create an individual method for website sitemap extracting and filtering after applying class menu and url methods and also modifying them with classification and consolidation methods at first; successfully visualize data with the use of created tool.

The tool, which improves website map filtering, extracting and rendering is created in the practical part of thesis by applying methods and Prefuse instrument.

Obtained results allow us to upgrade and develop website navigation process, through association and visualization information even more efficiently.

This project is implemented with JAVA, Python languages. „Microsoft Office Visio 2003“ is used for representation of charts.

ĮVADAS

Temos aktualumas. Sparti interneto plėtra ir su tuo susijusių didelių informacijos srautų pateikimas tinklapiuose suponuoja informacijos kaupimo, interpretavimo bei analizavimo poreikį. Tinkamam tinklapio informacijos atvaizdavimui būtinas teisingas duomenų įvedimas ir talpinimas. Šio instrumento egzistavimas lankytojams palengvintų informacijos paiešką ir navigaciją tinklapyje. Minėti aspektai lemia tai, jog lankytojo patogumas, tinklapio navigacijos aiškumas ir funkcionalumas tampa vis labiauaktualūs kiekvienai svetainei.

Mokslinė darbo problema. Šiandien interneto erdvėje aptinkama daugybė ir įvairaus pobūdžio bei formųsvetainių, tačiau tik dalis jų yra modifikuotos ir adaptuotos patogesniam lankytojo naudojimui. Mokslinę ir praktinę darbo problemą pabrėžia dar ir tai, kad internete kuriamos naujos svetainės taip pat retai yra pritaikytos patogesniam lankytojų naudojimui. Tai lemia poreikį tobulinti esamas tinklapių medines struktūras. Norint efektyviau naudoti interneto svetainių galimybes, būtina modifikuoti svetainės navigaciją, kurią atitinka svetainės medinė struktūra. Problemai išspręsti reikalingas instrumentas, skirtas tinklapio medinės struktūros radimui, išgavimui ir vizualizacijai.

Darbo objektas yra tinklapio medinės struktūros atvaizdavimas.

Darbo tikslas – sukurti instrumentą, skirtą tinklapių medinės struktūros radimui, išgavimui ir vizualizacijai.

Darbo tikslui pasiekti iškeliami tokie **uždavinius:**

1. Atlikti tinklapių medinės struktūros radimo, išgavimo ir vizualizavimo mokslinio potencialo analizę.
2. Atlikus metodinio potencialo analizę, parinkti galimus taikyti metodus ir jų taikymo aspektus darbo tikslui pasiekti.
3. Atrinkus galimus taikyti metodus, juos patobulinti (apjungiant ir išplečiant) sukuriant individualų metodą tinklapio struktūros radimui ir išgavimui.
4. Sukurti instrumentą, skirtą svetainės medinės struktūros vizualizacijai.

Darbe naudoti metodai. Baigiamajame magistro darbe naudoti bendrieji metodai: mokslinio potencialo analizė, sintezės ir apibendrinimo, įvairių autorių nuomonių lyginimui taikytas metaanalizės metodas, duomenų apjungimui sisteminė analizė, nagrinėjant įvairius interneto tinklapių turinius naudotas turinio analizės metodas, analitinis-lyginamasis tyrimo metodas, pagrindinio instrumento sukūrimui taikytas sisteminės analizės metodas, kuris leido atlikti skirtingų autoriu nuostatų, gautų rezultatų, vertinimų ir interpretacijų tinklapio struktūros radimo, išgavimo ir vizualizavimo klausimais sintezę, kuri grindžiama logine abstrakcija. Darbas

realizuojamas JAVA bei Python kalbomis, schemų atvaizdavimui naudotas „Microsoft Office Visio 2003“ programinis paketas.

Darbo struktūra. Darbas susideda iš trijų dalių. Pirmojoje dalyje atliekama mokslinio potencialo analizė, apibendrinama esamų metodų, skirtų tinklapių medinės struktūos radimui, išgavimui ir vizualizacijai problematika. Antroje dalyje atliekamas potencialių metodų apjungimas, integruojant darbo tikslui pasiekti reikalingus savitus elementus, tokiu būdu patobulint esamus metodus. Trečiojoje dalyje kuriamas individualus instrumentas, leidžiantis rasti, išgauti ir vizualizuoti tinklapių medines strukltūras. Darbo pabaigoje pateikiamos išvados ir pasiūlymai.

Pagrindiniai literatūros šaltiniai. Darbe naudotasi mokslinėmis publikacijomis ir straipsniais pristatytais WWW, IEEE, SIGIR bei KDD konferencijose.

Autoriaus sukurto dalyko esmė ir praktinė darbo reikšmė. Sukurtas efektyvus ir patogus tinklapių medinės struktūros vizualizacijos instrumentas, skirtas analizuoti svetainės nuorodas, navigaciją, ryšius tarp puslapių tiek tinklapių kūrėjams, tiek jų savininkams. Gaunama tinklapio struktūra, efektyvi vizualizacija yra visapusiškai naudinga tinklapių analizuotojams.

Panaudotos literatūros šaltinių skaičius – 15, lentelių – 6, paveikslų – 15.

1. TINKLAPIŲ MEDINĖS STRUKTŪROS NAVIGACIJOS GAVIMO IR VIZUALIZAVIMO PROBLEMATIKA

1.1. Tinklapių medinės stuktūros radimo, išgavimo ir vizualizavimo problemos ir galimybių vertinimas

Tinklapių medinę struktūrą sudaro esantys puslapiai, kuriuose pateikiama tekstinė, vizuali ir kitokia informacija. Jie yra susieti nuorodomis, kurios veda iš kitų tinklapio puslapių. Svetainės adresas lankytoją atveda į pagrindinį puslapį, kuriame pateikiama svarbiausia, dažnai tik sutrumpinta informacija apie turinį. Pagrindiniame puslapyje naudojamas meniu arba kitos, tekstinės, vizualinės (mygtukai, paveikslėliai su nuorodomis) nuorodos, kurios leidžia pereiti į kitus puslapius. Remiantis šiuo principu galima sudaryti hierarchinę struktūrą, kuri gaunama naršant po svetainę iš pagrindinio puslapio gilyn. Tokiu atveju pagrindinis puslapis tampa aukščiausio lygio viršūne, susietos briaunomis viršūnės (puslapiai) priskiriama žemesniam lygiui, o briaunas atitinka nuorodos, vedančios iš aukštesnio lygio viršūnės. Sudarytas svetainės navigacijos medis atvaizduoja ryšius tarp puslapių bei žingsnius, kuriuos reikia žengti, norint pasiekti tam tikrą informaciją. Kiekvienam tinklalapiui yra labai svarbu pirmuosiuose žingsniuose (gyliuose) pateikti pačią svarbiausią informaciją, taip sutrumpinant lankytojui kelią jos link. Visų svetainių tikslas yra kuo geriau ir kokybiškiau pateikti turimą informaciją, todėl vienas pagrindinių uždavinių yra teisingas ir patogus navigacijos sukūrimas. Kiti uždaviniai: patrauklaus dizaino sukūrimas, tinkamas informacijos išdėstymas puslapiuose. Kadangi vienas pagrindinių darbo tikslų yra patogios navigacijos radimas, tai kiti tinklapių medinės struktūros uždaviniai darbe nėra sprendžiami.

Taigi norint išspręsti teisingos, patogios pristatomo turinio navigacijos problemą yra svarbu tinkamai parinkti puslapių išdėstymą ir susiejimą svetainėje. Todėl didžiausia užduotis yra sukurti įrankį, suteikiantį galimybę analizuoti esamos svetainės struktūrą, vizualiai pateikiant puslapių medį.

Tinklapio turinio puslapiai paprastai prieinami iš paprasto URL, vadinamo „namų“, „pradžios“ adresų. Visi svetainės URL adresai organizuoti taip, jog hierarchinėje struktūroje nuorodos leidžia skaitytojui ne tik suvokti puslapio struktūrą, bet ir yra naršymo po turinį vadovas. Dauguma svetainių turi meniu, sudarytą iš html nuorodų, nes tai palengvina paieškos sistemų rezultatus bei leidžia vartotojui lengvai rasti susistemintą informaciją. Nuorodos yra konceptualios konstrukcijos, sukurtos iš *a, area* ir *link* elementų, kurie reprezentuoja ryšį tarp

dviejų resursų, iš kurių vienas yra dabartinis dokumentas.[1] Yra dviejų rūšių nuorodos HTML. Tai nuorodos į resursus, kurie yra kituose tinklapiuose:

- *Hipersaitai*

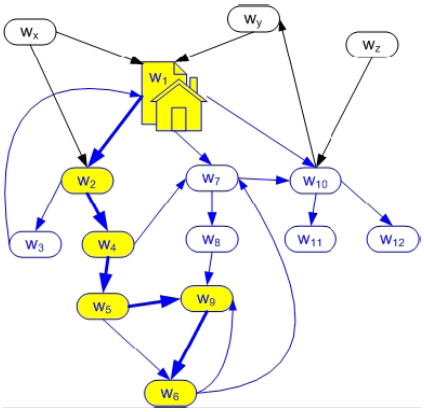
  Tai nuorodos į kitus resursus, kurie paprastai veikia „vartotojas vartotojui“ (*angl.* customer-customer), t.y. vartotojas gali naudodamasis vartotojo agentu pasiekti kitus resursus, aplankyti juos arba parsisiųsti.

- *Nuorodos į vidinius resursus*

  Tai nuorodos, kurios yra adresas į to paties domeno adreso informaciją. Paspaudęs ant jos lankytojas nepalieka svetainės.

Tačiau dažnai tinklapių medinė struktūra yra labai paini ir informacija pateikta sudėtingai. Su šiuo teiginiu sutinka autoriai P. Chandra ir G. Manjunath (2010), kurie nagrinėja svetainių navigaciją, atsirandančią dėl nuorodų painumo. Todėl didžiausias jų dėmesys yra sutelkiamas atskleisti tinklo sudėtingumą, t.y. „mąstymo ir orientavimosi sunkumus pirmą kartą apsilankius svetainėje tol, kol bus pasiektas reikiamas tikslas“[2].

.



Šaltinis: Praphul Chandra ir Geetha Manjunath. Navigational Complexity in Web Interactions.2010.

1 pav. Lankytojo kelias siekiant tam tikros informacijos

[1] http://www.whatwg.org/specs/web-apps/current-work/multipage/the-map-element.html#the-area-element (žiūrėta 2012 04 15)

[2] Praphul Chandra ir Geetha Manjunath. Navigational Complexity in Web Interactions. *WWW 2010,* April 26–30, 2010, Raleigh, North Carolina, USA.

Analizuojant lankytojo kelią ieškant tam tikros informacijos, autoriai daro prielaidą, jog žinomas pradžios ir pabaigos tikslas. Todėl analizuojant svetainės navigacijos sudėtingumą, modeliuojamas interneto per grafą (Tinklapio puslapių rinkinys) sąveikų srautas (W), kaip ir pavaizduota 1 paveiksle. Taigi, vienas svarbiausių šio darbo tikslų yra sukurti tinklapio vizualizaciją, kuri būti patogiausia vartotojui.

Vienas pagrindinių veiksnių, lemiančių patogumą lankytojui yra svetainės meniu. Tinkamai sukurtas svetainės meniu padeda nepasiklysti lankytojams, pasiekti svarbiausią suskirstytą turinį. Tinklapio meniu yra nuorodų (meniu punktų) rinkinys, naudojamas svetainės navigacijai ir pirmiausia rodomas kaip hierarchinis nuorodų sąrašas arba tam tikri nuorodų blokai, vedantys į vidinius puslapius[3]. Taigi remiantis literatūros šaltiniais galima išskirti Meniu navigacijos teikiamus privalumus:

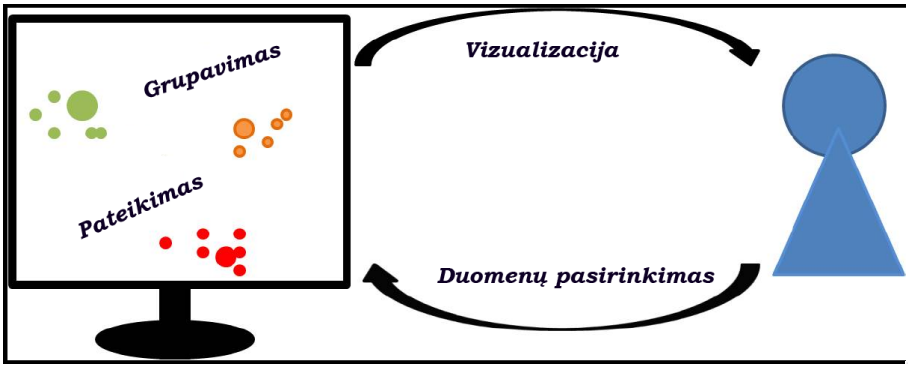
- Sukuriamas meniu ir puslapio žemėlapis, kad greičiau padėtų vartotojams rasti ieškomą informaciją.
- Sudaroma galimybė pateikti skaitytojams svarbesnius puslapius. Tai leistų užtikrinti, jog jie nepasimes dideliame informacijos kiekyje.
- Tikslinga pateikti trumpas ir aiškias pavadinimų nuorodas, kad vartotojai žinotų, kokį turinį pasieks paspaudę ant nuorodos.

Nėra abejonės, jog navigacija užima svarbų vaidmenį ir lemia vartotojų sėkmę svetainėje. Jeigu žmonės negali lengvai naršyti po svetainės turinį, nusivylę tiesiog paliks ją. Geros navigacijos sistemos turėjimas reiškia, jog galima išlaikyti daugiau vartotojų bei sulaukti kito jų apsilankymo. Žinant, jog meniu yra vienas pagrindinių tinklapio patogumo veiksnių ir naudojamas visose svetainėse yra tikslinga sudaryti meniu medį ir atvaizduoti kaip svetainės struktūrą. Dėl šios priežasties yra kuriamas „class“ radimo metodas, kuris bus analizuojamas tolimesniuose skyriuose.

Kaip minėta, ne tik meniu veiksnys yra svarbus tinklapių lankytojų patogumui, bet didelė reikšmė tenka ir svetainės vizualizacijai. Efektyvus tinklapių vizualinių objektų valdymas lemia sėkmę tinklapio struktūros patogumui. Taigi vizuali analizė turėtų būti įtraukiama tarp kitų tinklapio informacijos apdorojimo metodų (informacijos atradimo, duomenų analizės, duomenų valdymo). Tokių tinklapio vizualinių tyrimų srities tikslas yra rasti įžvalgą tarp dinaminių, masinių ir dažnai konfliktuojančių duomenų[4]. Todėl autorių D.Keim, A. Mansmann ir kt. nuomone, vizualios ir interaktyvios analizės duomenų išskyrimas yra pagrindinė vizualios analitikos tema.

[3] Jock D. Mackinlay, Stuart K. Card, Ben Shneiderman (eds.) (1999). Readings in information visualization: using vision to think. Morgan Kaufmann Publishers.
[4] Keim, D. A., Mansmann, F., Schneidewind J., and Ziegler, H. 2006. Challenges in visual data analysis. In Proceedings of IEEE Interntional Conference on Information Visualization.

Taigi vykdant tyrimus tinklapių medinės struktūros srityje duomenų išgavimo užduotims atlikti, pavyzdžiui, duomenų aprašymui, tyrimo duomenų analizei ir apibendrinimui, vizualizacijos gali suteikti didelę pagalbą. Vizualus dokumentų tyrinėjimas ir analizė leidžia vartotojui gauti apibendrintus duomenis, jų visų neskaitant ir į juos nesigilinant. Tokia informacijos peržiūra suteikia tiesioginį duomenų atvaizdavimą, pateikiant pagrindinius faktus ir išryškinant juos vartotojui aktualiausiu būdu. Be to, kaip teigia J. Leskovec, M. Grobelnik ir kt. autoriai, šis semantinis atvaizdavimas yra ne tik naudingas vizualizuojant dokumentą, bet taip pat turi didelę reikšmę bendros dokumento santraukos sudarymui[5]. Tai atliekama klasifikuojant sakinius iš pirminio teksto, kur savybės yra išgaunamos iš dokumento ir jo semantinio grafo. Tokiam darbui atlikti yra keletas įrankių, skirtų gauti dokumentų vizualizacijas: kai kurie yra koncentruoti tik tam tikriems duomenims, kaip naujienų kolekcijoms, kiti sukurti paremti vien tik tekstų dokumentuose analizei:



Šaltinis: Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. Learning Sub-structures of Document Semantic Graphs for Document Summarization

2 pav. Vizualizacijos pateikimas vartotojui

Nors nėra griežtai priimta naudoti vizualizacijos įrankius duomenų atvaizdavimui, tačiau jie yra labai naudingi atvaizduoti dideliems duomenų rinkiniams. Kaip rodo praktika, vis dažniau tenka apdoroti daugiau ir daugiau duomenų, todėl tikėtina, kad kai kurie atvaizdavimo metodai turės būti priimti. Vizualizacijos paketai suteikia daug patogumo ir universalumo įvairiems uždaviniams spręsti, taip pat vizualiai supažindinti vartotoją su duomenų sandara bei leisti atlikti kokybišką ir greitą vizualią analizę. Keletas vizualizacijai naudojam paketų:

- JgraphT – Java programavimo kalba paremtas vizualizavimo paketas, skirtas atlikti paprastoms ir lengvai plečiamoms užduotims.

[5] Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. Learning Sub-structures of Document Semantic Graphs for Document Summarization. Workshop on Link Analysis and Group Detection (LinkKDD) at KDD 2004 (Seattle, USA, August 22 – 25, 2004).

- Prefuse – Java programavimo kalba pagrįstas paketas, skirtas interaktyviam informacijos vizualizavimui.
- VisAD - Java komponentų biblioteka interaktyvioms ir bendradarbiavimo vizualizacijoms.
- The Visualization Toolkit - C++ programavimo kalba, pritaikytos daugumai platformų vizualizacijos rinkinys.

Taigi yra daug skirtingų platformų vizualizacijos paketų, pritaikytų įvairiems vizualizavimo variantams, parašytiems skirtingomis programavimo kalbomis ir turinčių savo privalumų bei trūkumų. (pvz.: Graphviz, Geomantics, JGraphT, Protovis ir kt.). Tačiau autoriaus nuomone, vizualizacijos sistemoms kurti geriausiai tinkamas Prefuse vizualizavimo paketas. Šis paketas puikiai atitinka keliamus reikalavimus: lengvai koreguojamas ir pritaikomas pagal nestandartinius grafų atvaizdavimus, pateikia patrauklią ir patogiai naršomą vizualinę tinklapio medinę struktūrą.

Taigi tinklapio medinė struktūra pirmiausiai turi turėti patogų, nesudėtingai valdomą meniu bei patrauklią, informatyvią vizualinę erdvę, kurios sukūrimas naudojant kompleksinius metodus ir yra pagrindinis autoriaus darbo tikslas.

1.2. Tinklapių medinės struktūros radimo, išgavimo ir atvaizdavimo taikymo būdų analizė

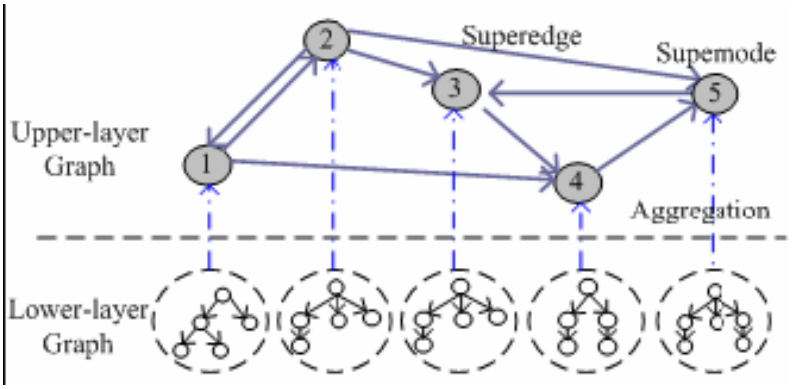
Kaip minėta ankstesniame skyriuje, tinklapių medinės struktūros kūrime svarbiausia yra meniu ir vizualinės sistemos pateikimas. Tačiau kyla ir nemažai problemų, kuriant tiek navigacijos sistemą, tiek galvojant apie vizualizacijos projektus. Nors yra pateikiamas ne vienas būdas, generuoti navigacijos medį naudojant programinę įrangą, kaip All WebMenus Pro 5.0.710, Sothink DHTML Menu 8.2, Flash Menu Factory 1.0., tačiau šie būdai yra netinkami dėl to, kad medis veikia tik kuriant svetainę, o vėlesniuose etapuose jo nebelieka.

Kita problema yra efektyvi nuorodų analizė. Kadangi esami nuorodų analizės metodai iš esmės sukurti remiantis faktais, jog visa nuorodų struktūra paremta kaip vieši tinklo dokumentai. Tačiau nuorodų informacija grafuose, tokia kaip žmonių ekonominiai santykiai, retai randami visuomenėje[6]. Tinklo dokumentų nuorodų struktūros yra surenkamos skenuojant tinklo informaciją. Tai atlieka nuskaitymo agentai, kurie keliauja po viešus dokumentus. Tinkle intensyviai naudojami nuorodų analizės algoritmai, kurie reikalingi renkant informaciją. Tačiau

[6] Jun Sakuma ir Shigenobu Kobayashi. Link Analysis for Private Weighted Graphs. SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

dabartiniai nuorodų analizės algoritmai iš esmės veikia kaip tiesioginių nuorodų grafas ignoruojant hierarchinę tinklo struktūrą. Dėl šios priežasties dažniausiai susiduriama su tokiomis problemomis: tinklo padrikumu ir naujai atsirandančių puslapių šališku rangavimu. Gui-Rong ir kt. siūlo neįprastą rangavimo algoritmą, vadinamą hierarchiniu reitingu, kaip šios problemos sprendimą. Tai leidžia panaikinti tiek hierarchinės, tiek tinklo nuorodų struktūros sunkumus. Jų algoritme tinklo puslapiai visų pirma apjungiami į hierarchinę struktūrą pagal direktoriją, domeno arba host lygius ir nuorodų analizė atliekama sudarytame grafe. Tuomet kiekvienos viršūnės atitinka atskirą puslapį ir priklauso hierarchinei struktūrai [7]. Remiantis šių autorių nuomone ir sprendžiant nuorodų analizės problemą, buvo sudaromas url metodas, kuris bus aprašytas vėliau.

Taigi X. Gui-Rong ir kt.[7] siūlo naudoti susietą hierarchinę struktūrą tinklui pagal URL. Pavyzdžiui, http://www.cs.vilnius.edu/Research/Projects/, būtų galima tikėtis rasti su projektu susijusią informaciją apie išradimus, atliktus Vilniaus kompiuterinio mokslo departamente. Tam pritaria ir H. A. Simon'as[8] kuris teigia, „jog visos sistemos turėtų būti organizuotos hierarchinės struktūros". Visas žiniatinklis yra puikus hierarchinės organizacijos pavyzdys. Žvelgiant iš tinklo pusės, tinklo puslapiai yra hierarchine struktūra organizuoti, kur hierarchinė informaciją atstoja vietą struktūroje. Žvelgiant visuotiniu mastu, žiniatinklis taip pat yra organizuotas hierarchine struktūra, kurioje pirmąjį lygį atitinka aukščiausio lygio domenai (kaip stanford.edu). Kitus lygius atitinka virtualūs host, virtualūs aplankai ir tinklo puslapiai. X. Gui-Rong ir kt.[7] pateikia skirtingai nei įprastai tradicinių nuorodų grafe variantą. Grafo nuorodas sudaro du sluoksniai, t.y viršutinysis sluoksnis ir apatinysis (3 paveikslas).



Šaltinis: Gui-Rong Xue, Qiang Yang, Hua-Jun Zeng, Yong Yu, Zheng Chen .Exploiting the Hierarchical Structure for Link Analysis. SIGIR'05, August 15–19, 2005, Salvador, Brazil.

[7] Gui-Rong Xue, Qiang Yang, Hua-Jun Zeng, Yong Yu, Zheng Chen .Exploiting the Hierarchical Structure for Link Analysis. SIGIR'05, August 15–19, 2005, Salvador, Brazil.
[8] H. A. Simon. The Sciences of the Artificial. MIT Press, Cambridge, MA, 3rd edition, 1981.

3 pav. Hierarchinė tinklo grafo struktūra

Apatinio sluoksnio grafas yra hierarchinė medžio struktūra, kurioje kiekviena viršūnė yra atskiras tinklo puslapis, o briaunos atitinka hierarchines nuorodas tarp puslapių. Vartotojas pradeda ieškoti informacijos nuo viršutiniojo sluoksnio ir gali arba pereiti į bet kurią to paties sluoksnio viršūnę, arba keliauti hierarchiškai nuoroda žemyn į apatinįjį sluoksnį. Paremtą šiuo atsitiktinio kelio medeliu, pateikiamas Hierarchinio rango algoritmas, skirtas tinklo puslapių svarbos apskaičiavimui. 1 lentelėje peikiama metodo terminologijos lentelė su URL pavyzdžiu.

1 lentelė. Terminologija

Terminas	**Pavyzdys: http://www.vu.lt/lt/tyrimai/index.html**
Adresas	**http://www.vu.lt**
Domenas	**www.vu.lt**
Direktorija	**http://www.vu.lt/lt/tyrimai/**
Puslapis	**http://www.vu.lt/lt/tyrimai/index.html**

Viena iš svarbiausių tinklapių medinės struktūros kūrimo problemų yra hierarchinės sistemos sukūrimas. Šiai problemai analizuoti yra daugybė mokslinių - tiriamųjų darbų. Nadav ir kt.[9] teigė, kad būtų naudinga hipersaitus eksploatuoti "lokaliai", kad būtų galima URL struktūrą susieti hierarchiškai ir, kad didžioji dalis organizacijų informacijos ypatybių internete yra nuspėjamos turint hierarchinės struktūros žinių. L. Laura ir kt. pasiūlytame hierarchiniame modelyje[10] teigė, jog „kiekvienas puslapis patekęs į grafą yra priskiriamas pastoviai reikšmei, regionui, kuriam jis priklauso, ir nuoroda gali susieti tik tame pačiame regione esančias viršūnes".[10]

G. Scanniello ir kt[11] pristato būdą, paremtą informacijos paieška ir klasterizavimo technikomis, skirtą automatiškai didinti tinklapių navigacijos struktūrą. Pagrindinis autoriaus siūlomo modelio tikslas yra optimizuoti navigacinių nuorodų skaičių, sukuriant semantinį navigacijos žemėlapį. Tokios nuorodos tarp viršūnių yra sudarytos taip, kad būtų galima suformuoti prieigos struktūras (pvz, navigacijos modelius, vedlius, indeksus ir kt.), paremtas specifiniais vartotojo informacijos priėmimo tikslais (pvz. greita prieiga prie svarbiausio ir dažniausiai naudojamo turinio) arba specifiniais aplikacijos reikalavimais (pvz. Sukurti vadovaujamą turą per turinį, specifine tema). Visi autoriaus siūlomi metodai ir įrankiai, skirti

[9] D. Cai, X. F. He, J. R. Wen ir W.Y. Ma. Block-level Link Analysis. The 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'2004), July 2004.
[10] L. Laura, S. Leonardi, G. Caldarelli, ir P. D. L. Rios. A Multi-Layer Model for the Web Graph. In 2nd International Workshop on Web Dynamics, Honolulu, 2002
[11] Giuseppe Scanniello, Damiano Distante, Michele Risi. An approach and an Eclipse-based environment for enhancing the navigation structure ofWeb sites. Journal International Journal on Software Tools for Technology Transfer (STTT), Volume 11 Issue 6, November 2009

tinklapių analizei ir evoliucijai paremti keletu pagrindinių technikų. Jas sudaro grupavimo algoritmai, restruktūrizacija, informacijos paieška ir klonų aptikimo metodai ir kt.

Vartotojai, kurie tinklapyje apsilanko pirmą kartą pasirenka nuorodas pagal pavadinimus ir aprašymus, esančius puslapyje. Tačiau iškyla problema, jeigu šie aprašymai yra neteisingi, o vartotojas naršydamas atlieka klaidingus ėjimus. Tokiems sunkumams išspręsti, Hollink ir kt.[12] pristato algoritmą, skirtą automatiškai rasti nuorodas su problemiškais aprašymais hierarchiniuose meniu. Algoritmas ieško registro failų navigacijos modeliuose, kur parodoma, kiek vartotojų padarė naršymo klaidų. Pasikartojančios klaidos byloja, jog vartotojas nesuprato nuorodos aprašymo. Algoritmo atradimai gali padėti tinklapio kūrėjams pagal vartotojų elgseną patobulinti svetainės nuorodų pavadinimus[13] . Šis algoritmas ir jo panaudojimo privalumai yra svarbus ir teksto autoriui. V. Hollink siūlomo algoritmo teigiamos pusės yra naudojami kuriant tinklapio vizualizacijos sistemos projektą. Navigacijos tobulinimui teisingas nuorodų pavadinimų priskyrimas ir susiejimas su adresais gali palengvinti vartotojui suvokti kokia tiksliai informacija bus pateikiama tam tikrame puslapyje.

Taigi tinklapių medinės struktūros radimo, išgavimo ir vizualizacijos kūrimas kelia nemažai problemų, todėl kitame skyriuje autorius pateiks efektyviausius metodus joms spręsti.

[12] Vera Hollink, Maarten van Someren ir Bob J. Wielinga. A semi-automatic usage-based method for improving hyperlink descriptions in menus. Journal International Journal of Human-Computer Studies, Volume 67 Issue 4, April, 2009

[13] Vera Hollink, Maarten van Someren ir Bob J. Wielinga. A semi-automatic usage-based method for improving hyperlink descriptions in menus. Journal International Journal of Human-Computer Studies, Volume 67 Issue 4, April, 2009

2. TINKLAPIŲ MEDINĖS STRUKTŪROS RADIMUI, IŠGAVIMUI IR VIZUALIZAVIMUI SKIRTŲ METODŲ MOKSLINIO POTENCIALO ANALIZĖ

Atlikus literatūros šaltinių analizę ir išskyrus pagrindines problemas, kurios kyla kuriant tinklapių struktūrą, matyti, jog tai lemia svetainėje esančios nuorodos. Todėl tik teisingas jų ryšių nustatymas leidžia sukurti hierarchinę struktūrą ir sudaryti svetainės struktūros medinį atvaizdavimą. Tokiems uždaviniams atlikti, autorius pasirenka šiuos metodus:

- **Class meniu artumo:**

Žinant, jog svetainės meniu yra vienas pagrindinių navigavimo svetainėje būdų, bei geriausiai atspindi svetainės pagrindinę sandarą, kuriamas metodas skirtas rasti svetainės meniu komponentus, išgauti teisingą jų struktūrą bei sukurti vizualizacijai skirtą duomenų išsaugojimą. Šis metodas remiasi „class“ ir „id“ atributų reikšmėmis programiniame kode, kuris apibūdintų svetainės meniu.

- **URL metodas:**

Visi svetainėje esantys url adresai, skirstant juos pagal skirtingus lygius per skirtuką „/“, gali būti susiejami tarpusavyje ir sudaryti medinę struktūrą. Šiuo principu sukuriamas metodas, kuris išgauna esančias nuorodas svetainėje, suskirsto į skirtingus lygius, sudaro medinę struktūrą ir pateikia duomenis vizualizacijai.

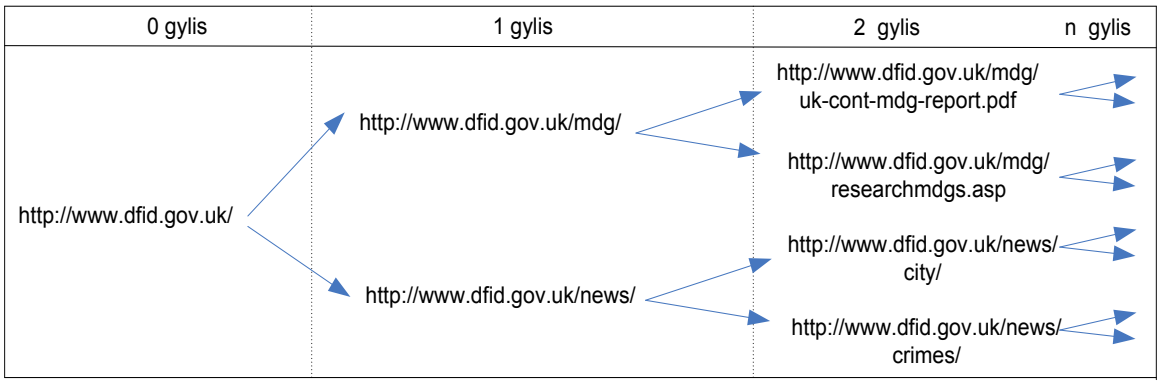
Taigi teisingai pritaikant šiuos metodus, nustatant tikslų informacijos apdrojimo kelią, duomenys yra perduodami vizualizacijos procesui. Autoriaus pasirinkimu, vizualizavimas atliekamas naudojant Prefuse paketą. Tam yra sudaromos viršūnių ir ryšių lentelės. Būtent šiose lentelėse saugoma informacija apie kiekvieną viršūnę. Duomenis sudaro nuoroda, jos pavadinimas, metodas, kuriuo buvo rasta viršūnė bei ryšys pagal viršūnės ID. Ryšys apibūdinamas parteikiant ID iš „source“ nuorodos į „target“ nuorodą. Taip yra sukuriamos briaunos, siejančios visas nuorodas. Taigi metodų, kuriuos autorius naudoja tinklapių struktūros radimui, išgavimui ir vizualizavimui analizė pateikiama tolimesniuose skyriuose.

2.1. Tinklapių medinės struktūros radimui, išgavimui ir vizualizavimui taikomų class meniu artumo ir url metodų efektyvumo analizė

Nors yra nemažai metodų, tinkamų sukurti sveitainės struktūrai, tačiau efektyviausi, autoriaus nuomone, yra class meniu artumo ir url metodai. Nors ir skirtingi šie metodai, tačiau jie yra efektyvūs norint rasti svetainės struktūrą, ją išgauti ir sudaryti hierarchinį medį

Taigi išanalizavus įvairius metodus, autorius daro išvadą, kad pagrindinis įrankis, kuriant tinklapio struktūrą ir vizualizaciją, yra class meniu artumo metodas. Jo veikimas pagrįstas svetainės meniu medžio radimu. Šis metodas naudingas tuo, kad sumažina bendrą nuorodų skaičių, gaunant pagrindinę navigaciją iš tam tikro puslapio. Duomenims rasti naudojamas tiesioginis svetainių turinio (nuorodų) skenavimas. Tačiau svetainė turi daugybę nuorodų, todėl aktualu medžio struktūrą ir skenavimus atlikti interaktyviai t.y. išgauti nuorodas iš tam tikro puslapio skirtinguose gyliuose, taip neapkraunant vizualizacijos ir susikoncentruojant į aktualiausią duomenų šaką. Taigi vykdant programą class meniu artumo metodu, iškviečiamas python skriptas, kurio užduotis gauti meniu sandarą ir gražinti suskirstytus duomenis lentelėse. Šio metodo rezultatas yra visų vidinių navigacinių nuorodų esančių svetainės meniu radimas.

Pasirinktas Python skriptas todėl, kad šia programavimo kalba yra paprasta prisijungti prie svetainės, išgauti ir apdirbti joje esančias nuorodas. Skripte esančio kodo užduotis – ne tik gauti tinklalapio nuorodas, bet ir rasti meniu struktūrą. Jai atrinkti pasirenkamas tam tikras skenavimo gylis. Skenavimo gylis - tai meniu medžio mazgų nutolusių nuo pagrindinio mazgo atstumas.



Šaltinis: sukurta autoriaus

4 pav. Meniu elementų paieškos į gylį schema (sukurta autoriaus)

Autorius savo darbui naudoja class meniu artumo metodą, tačiau atvaizdavimą vykdo interaktyviai: kaskart gauna artimiausio gylio viršūnes. Metodo veikimas paremtas klasių (class) paieška pagal pavadinimą. Kadangi programuojant meniu komponentai yra suskirstomi aprašant juos klasėse, todėl class artumo metodas yra pats tinkamiausias juos iššlifuoti. Išgavus visas nuorodas iš tinklalapio, atliekama paieška pagal id ir class atributus, kuriuose būtų “menu”, „meniu“ arba “nav” pavadinimu elementai. Kitas žingsnis - klasės viduje rasti nuorodą esančią <a> ribose, kuri yra priskirta už *href* atributo. Pagal gautas nuorodas naudojant BeautifulSoup modulį gražinamas tinklapio nuorodų medis. Šis medis yra perduodamas vizualizavimui. Kaskart pasirinkus viršūnę yra iškviečiamas python skriptas ir gaunamos visos jo „vaikinės“ viršūnės.

Class meniu artumo metodo privalumai yra:

- Gaunamas svarbiausias tinklapio skirstymas pagal meniu komponentus
- Teisingai gaunama žemesnio lygio viršūnės skirtos interaktyviai atvaizduoti kitą lygį.
- Aprinkamas daugumoje svetainių pagal naudojamą išgavimo būdą.

Metodo trūkumai:

- Dėl neteisingo class arba id atributų naudojimo arba nepriskyrimo gali neveikti.
- Pateikia tik svarbiausią navigaciją svetainėje.

Kadangi class meniu artumo modelis yra nepakankamas autoriaus darbui atlikti, todėl savo praktiniame darbe renkasi naudoti ir url metodą.

Svetainės url adresas ir puslapių kelias dažnai suskirstomas hierarchine tvarka dėl patogumo ir aiškumo tiek paieškos sistemoms, tiek lankytojams. Url adresas dažniausiai atitinka svetainės navigacijos kelią.

Prasta URL struktūra:

- http://www.domain.com/2012/01/10/archyvas/p=?2176

Efektyvi URL struktūra:

- http://www.domain.com/pavadinimas/sritis/

Norint gauti svetainės struktūrą pagal url adresą ir atlikti nuorodų adresų, navigacijos ir pavadinimų analizę, autorius url metodą panaudojo taip, kad būtų pateikiama vizualizacija ir kartu įgyvendinama paskirta užduotis. Pagrindinis principas yra išgauti visas nuorodas iš svetainės ir pagal „/“ skirtukus suskirstyti url adresus į skirtingus lygius. Pirmoji užduotis skirta gauti esančias nuorodas svetainėje. Tai atliekama prisijungus prie svetainės pagrindinio adreso surenkant visas esančias http nuorodas bei nuorodų pavadinimus (link text). Šis nuorodų sąrašas yra filtruojamas paliekant tik vidines svetainės nuorodas, vedančias į vidinius resursus. Kitas žingsnis atliekamas suskirstant gautus adresus pagal skirtingus gylius. Tuomet kiekviena viršūnė pagal pradžios adresą yra priskiriama aukštesniojo lygio viršūnei. (pvz.: www.vu.lt/studijos/ priskiriamas www.vu.lt/ viršūnei). Visa ši struktūra yra patalpinama į duomenų lenteles ir pateikiama vizualizavimui.

Metodo privalumai:

- Dažniausiai naudojamas hierarchinis, meniu struktūrą atitinkantis puslapių rikiavimas pagal direktoriją.

- Gali būti pritaikomas beveik visuose tinklapiuose

Metodo trūkumai:

- Puslapiai ir kelias iki jų gali būti išmėtyti padrikai.
- Gali būti rasti visiškai nesuprantamai pavadinti arba jokios prasmės neduodantys tarpiniai adresai.
- Dažnai būna didelis kiekis url adresų naudojamų pvz paveikslėliams, darbiniams failams laikyti.
- Flash technologijas naudojantys puslapiai dažniausiai nekeičia savo url adreso.

Taigi naudojant ir tobulinant šiuos metodus, autorius galės sukurti modernią programą skirtą, tinklapių medinės struktūros radimo, išgavimo ir vizualizavimo procesams efektyvinti.

2.2. Tinklapių medinės struktūros vizualizavimui skirto klasifikatoriaus taikymo būdai

Kadangi ne visada galima taikyti abu metodus (class meniu artumo ar url metodas) dėl tinklalapių skirtingos medinės struktūros bei vizualizacijos, todėl tikslinga atrinkti konkrečiam tinklapiui labiausiai tinkantį metodą efektyviausios tinklapio medinės struktūros gavimui. Tam autorius pasirenka klasifikavimo būdą. Naudojant ID3 algoritmą yra atliekamas metodų įvertinimas pagal apmokymui sudarytą klasifikavimo medį. Klasifikavimo pasirinkimų medžiui sudaryti pasitelkiama įrankio vartotojo pagalba. Jis priskiria teisingus puslapius, kurie bus naudojami kaip duomenys apmokymui. Vartotojui yra pateikiamas visais metodais sudarytas tam tikro gylio medis. Išsirinkęs matomus meniu ir kitas tinklapyje esančias navigacines viršūnes (pagal pavadinimą arba pagal url adresą) patvirtina, jog jos yra teisingos. Iš šių duomenų sukuriamas pasirinkimų medis klasifikavimui. Patekus į naują lygį yra atliekamas klasifikavimas ir naujosios viršūnės yra pažymimos kaip pagalba vartotojui, pagal kurį buvo geriausiai įvertintas metodas.

Rinkinys yra sudarytas iš teigiamų ir neigiamų tikslinės koncepcijos pavyzdžių.

Autoriaus atveju rinkinys atitinka koncepciją:

Ar nuoroda atitinka svetainės medinę struktūrą? {Taip, Ne}.

Taip pat yra priskiriami atributai naudojami kiekvienai nuorodai vertinti:

1 Metodas (Class meniu artumo) = {Taip, Ne}

2 Metodas (URL) = {Taip, Ne}

Duomenis klasifikuoti remtasi Weka sistema, naudotasi J48 (ID3) klasifikatoriumi. Problemoms spręsti pasitelkiamas sprendimų medžio apsimokinimo algoritmas (angl. Desision tree learning), sistemą apmokant tikrais testiniais duomenimis. Sprendimų medžio apsimokinimas naudojamas kaip nuspėjimų modelis, kuris pagal turimus duomenis palygina ir

pateikia išvadas naujai suteiktai reikšmei. Pagrindinis tikslas yra sukurti modelį, kuris gavęs kintamąjį nuspėja jos reikšmę arba priskiria tam tikrai grupei, remdamasis jau įvestų kintamųjų informacija[14]. Sprendžiant geriausio metodo radimo uždavinį naudojamas klasifikacijos medžio modelis, kuomet nuspėjamasis kintamas priskiriamas klasei, kuriai pagal savo duomenis labiausiai atitinka.

Šiuo principu atliekamas klasifikavimas bei pateikiami geriausio metodo atrinkimo rezultatai. Klasifikatoriaus pritaikymas autoriaus praktiniame darbe, kuriame konstruojamas įrankis vizualiai tinklapių struktūros analizei.

2.3. Tinklapių medinės struktūros radimui, išgavimui ir vizualizavimui skirtų metodų apjungimas ir pritaikymas

Nors naudojant klasifikatorių galima surasti vieną labiausiai tinkamą metodą vizualiai tinklapių struktūros analizei, tačiau sujungus jų geriausias savybes galima pasiekti efektyvesnių rezultatų. Šiam tikslui pasiekti, reikia atlikti sisteminius veiksmus, kurie suvienodina viršūnių ir briaunų duomenų lentelių formatus, bei sukurti strategiją kaip realizuoti ir pateikti vartotojui patogų įrankį. Metodų apjungimo tikslas yra gauti interaktyvią vizualizaciją, kurioje būtų abiejų metodų gaunamos svetainės struktūros. Todėl pirmajame lygyje (pagrindiniame tinklapio adrese) yra pateikiamos viršūnės tokiu veikimo principu:

1. Gaunama class artumo metodo svetainės struktūra
2. Atliekamas prijungimas viršūnių, gautų url metodu tokiu principu: Tikrinama ar jau yra url adresu viršūnė vizualizacijoje. Jeigu viršūnė yra, papildomas elementas užrašu („url“) papildomajame stulpelyje. Jeigu viršūnės nėra, ji yra pridedama ir sudaromas ryšys su pagrindine.

Antrajame lygyje, paspaudus bet kurią viršūnę, yra:

1. Atliekamas naujų žemesnio lygio viršūnių gavimas pagal perduotąją class artumo metodu.
2. Randamos jau gautame medyje pagal url perduodamosios viršūnės kaimynės. Prijungiamos tuo pačiu principu kaip ir ankstesniame lygyje.
3. Visos atvaizduotos viršūnės yra naudojamos klasifikavimo medžiui sudaryti. Taip pat pažymėjus vartotojui tinkamas sukuriamas apmokomųjų duomenų sąrašas.
4. Pagal gautus rezultatus yra paryškinamos tinkamiausio metodo viršūnės, taip palengvinant vartotojui rasti aktualiausias.

[14] Ian H. Witten,Eibe Frank,Mark A. Hall Data Mining. *Practical Machine Learning Tools and Techniques* 539-605. 2011

Kad vartotojas galėtų žymėti teisingas viršūnes medžio apmokymui, šalia atliekamos vizualizacijos yra užkraunamas naršomas tinklapis. Šiuo principu yra sukuriamas įrankis, suteikiantis galimybę analizuoti svetainės struktūrą, pagal abu metodus, kadangi jie vienas kitą papildo, taip pagerindami įrankio veikimo rezultatus.

Praktikoje įrankis gali būti pritaikomas tinklapių navigacijos, informacijos pateikimo, nuorodų pavadinimų vizualiai analizei. Kadangi paeiliui galima matyti visas tam tikro puslapio nuorodas, pavadinimus, galima optimizuoti svetainės lankymo žingsnius. Taip pat galima peržvelgti esančias nuorodas, optimizuoti jas pagal pavadinimus paieškos sistemoms.

Taigi toks metodų apjungimas yra kuriamas autoriaus praktiniame darbe, kurio detalus aprašymas pateikiamas 3 dalyje.

3. TINKLAPIO MEDINĖS STRUKTŪROS RADIMAS, IŠGAVIMAS IR VIZUALIZAVIMAS TAIKANT CLASS MENIU ARTUMO IR URL METODUS

3.1. Tinklapio medinės struktūros radimo, išgavimo ir vizualizacijos aprašymas

Praktinis darbas „Tinklapių medinės struktūros radimas, išgavimas ir vizualizacija“ atliktas remiantis mokslinės literatūros šaltiniais, pritaikant daugiafunkcinius metodus. Class meniu artumo ir url metodų pagalba sukurtas įrankis naudojamas svetainės struktūros medžio atvaizdavimui, t.y. sumodeliuojamas vizualus svetainės struktūros vaizdas, kuriuo būtų galima vizualiai gerinti svetainės navigaciją. Taigi autoriaus darbo esmė - surasti html puslapiuose medinę struktūrą, patį medį pateikti navigacijos puslapyje ir sukurti įrankį, skirtą analizuoti tinklapio sandarai. Atvaizdavimo procesui naudojamas Prefuse vizualizacijos paketas, kuris suteikia galimybę realizuoti meniu atvaizdavimo reikalavimus.

Norint pabrėžti navigacijos patogumą, pagrindinis uždavinys yra didinti svarbiausių navigacinių nuorodų skaičių skirtinguose medžio lygiuose sukuriant navigacijos žemėlapį. Dabartiniai nuorodų analizės algoritmai iš esmės veikia kaip tiesioginių nuorodų grafas ignoruodamas hierarchinę tinklo struktūrą. Dėl šios priežasties dažniausiai susiduriama su tokiomis problemomis: tinklapio padrikumu ir naujai atsirandančių puslapių neteisingu rangavimu bei nuorodų į juos priskyrimu. Kad tai neįvyktų, reikia gauti svetainės struktūrą sudarančias nuorodas bei teisingai jas priskirti, susieti ryšiais. Visą procesą galima skaidyti į keletą etapų. Visų pirma reikia prisijungti prie svetainės, gauti joje esančias vidines nuorodas naudojant skirtingus metodus (url, class radimo). Iš visų nuorodų sąrašo pagal skirtingus atributus ir taisykles atrenkamos tik tinkamos nuorodos, jų pavadinimai. Antroji užduotis yra esamų nuorodų suskirstymas į skirtingus lygius, bei ryšių tarp jų sudarymas. Tam naudojama url adresų analizė bei skirstymo taisyklių sukūrimas ir, išgaunant nuorodas, teisingas išsaugojimas duomenų lentelėse. Sekanti užduotis yra gautų nuorodų (viršūnių) ir ryšių (briaunų) tarp jų tinkamas perdavimas vizualizavimo procesui. Visos kitos užduotys informacijos pristatymui ir pateikimui vartotojui yra atliekamos vykdant vizualizacijos procesą.

Autoriaus sukurtas įrankis, tinklapių medinės struktūros radimo, išgavimo ir vizualizacijos procesams analizuoti, apjungia skirtingus svetainių struktūros išgavimo metodus į vieną labiausiai tinkantį. Šiai užduočiai spręsti naudojamas klasifikavimas. Pagal vartotojo patvirtintą nuorodų sąrašą sudaromas apmokomųjų duomenų klasifikavimo medis bei atliekamas klasifikatoriaus apsimokymo procesas. Sudaromas modelis, kurio duomenys yra vizualizacijoje atvaizduotos nuorodų url ir class meniu artumo metodų reikšmės (Taip, Ne) bei pagrindinis atributas, atitinkantis vartotojo patikrintos teisingų navigacinių nuorodų reikšmes. Atlikus klasifikavimą, randamas efektyviausias metodas nagrinėjamos svetainės medinei struktūrai rasti.

Interaktyviai skleidžiant svetainės medį, kitame lygyje vartotojui yra pateikiamos rekomendacijos, pagal tinkamiausią metodą, išryškinant to metodo rastas viršūnes.

Taigi šalia kitų informacijos apdorojimo metodų rezultatų pateikimui naudojama ir vizuali analizė. Dėl šios priežasties autoriaus konstruojamas įrankis sukuria svetainės struktūros medžio vizualizaciją. Duomenų analizės užduotims vizualus rezultatų nagrinėjimas suteikia daug patogumo ir aiškumo. Vizualizacijos paketai yra universalūs ir turi daug galimybių įvairiems tinklapių medinės struktūros radimo, išgavimo ir vizualizacijos uždaviniams spręsti. Be to, leidžia vizualiai supažindinti vartotoją su duomenų sandara bei atlikti kokybišką ir greitą duomenų analizę.

3.2. Konkretaus tinklapio medinės struktūros vizualizacijai skirto Prefuse paketo taikymas

Praktiniam darbui atlikti autorius pasirenka Prefuse paketą. Prefuse - programinės įrangos įrankių rinkinys, skirtas kurti duomenų vizualizacijoms. „Prefuse flare“ rinkinys suteikia vizualizacijos ir animacijos priemones panaudojant ActionScript ir Adobe Flash Player. Šis paketas palaiko didelį kiekį rinkinių, skirtų duomenų modeliavimui, vizualizacijai ir sąveikai. Tai leidžia optimizuoti lentelių, grafikų ir medžių duomenų struktūras, išdėstymo ir vaizdo kodavimo būdus, suteikia galimybę naudoti animaciją, dinamiškas užklausas, integruotas paieškas ir naudojimąsi duomenų bazėmis. Prefuse parašyta naudojant Java 2D grafinę biblioteką, yra lengvai integruojama į Java Swing ir interneto programas. Paketas yra platinamas pagal BSD licenziją ir gali būti laisvai naudojamas komerciniais ir nekomerciniais tikslais.

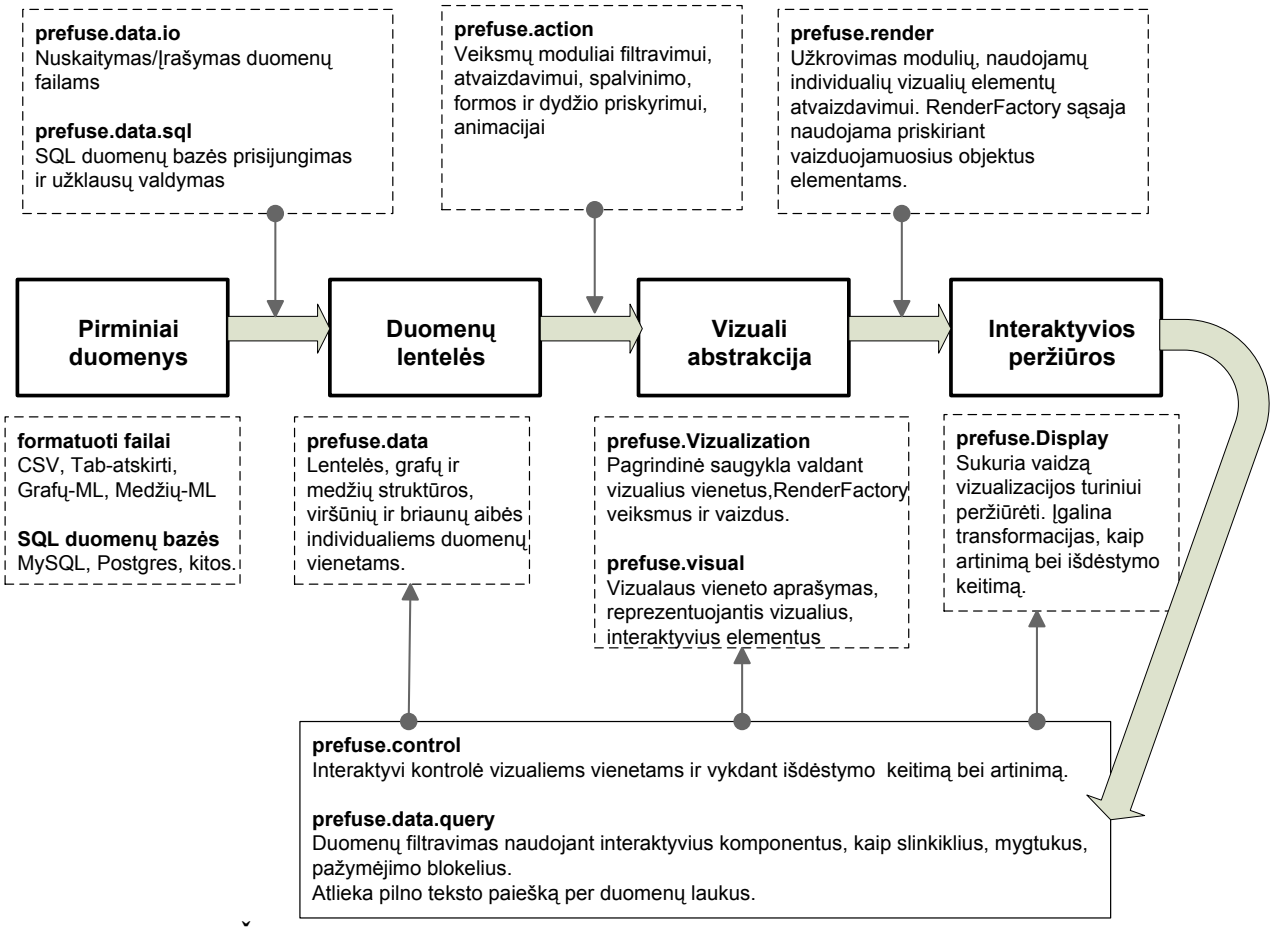
Prefuse praplečia programinės įrangos sistemą, padeda programinės įrangos kūrėjams modeliuoti interaktyvias informacijos vizualizavimo programas. Jis gali būti naudojamas kuriant atskiras programas, vaizdo komponentus, naudojamus didesnių programų ar tinklo programų. Prefuse ketina iš esmės supaprastinti procesus, skirtus reprezentuojant ir efektyviai perduodant duomenis, juos apdorojant ir vizualizuojant (pvz., matmenys, forma, spalva ir t.t.).

Autorius: Jeffrey Heer / Berkeley universitetas
projekto tinklapis: http://prefuse.org
Paskutinė versija: beta, 2007spalio 21
Reikalavimai: Java 1.4
Pagrindinės duomenų struktūros: Lentelė, Grafas, Medis
Integruotos vizualizavimo technikos: Fisheye Menu, Radial Graph, Treemap, Scatterplot, zipdecode, DOITree, Graph View, Data Mountain

Galimi failų formatai: Grafo failas (GraphML (XML)), Medžio failas (TreeML (XML)), tekstas, CSV

Architektūra:

Prefuse įrankių rinkinys sudarytas remiantis informacijos vizualizavimo pavyzdiniu rinkiniu. Tai programinės įrangos kūrimo būdas, skirstantis vizualizavimo procesą į smulkesnius etapus. Pagrindiniai jų yra pirminių duomenų pasirinkimas, duomenų lentelių sukūrimas, modeliavimas ir vizualaus kodavimas (abstrakcija) bei interaktyvus atvaizdavimas. (4 pav.)



Šaltinis: http://www.infovis-wiki.net/index.php/Prefuse, versta autoriaus

5 pav. Prefuse paketo architektūra

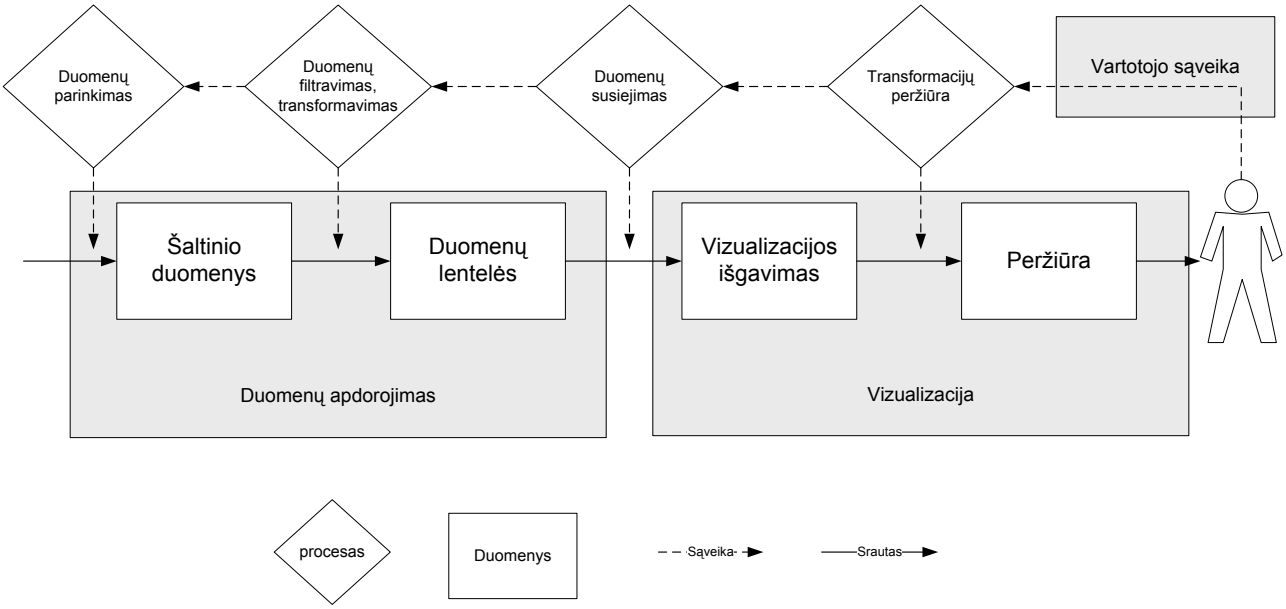
Patį Prefuse paketo veikimą būtų galima suskirstyti į duomenų, vizualizacijos ir užduoties etapus. Tai pagrindiniai trys etapai, kurie Prefuse yra dar smulkiau suskirstyti į paketus. Duomenų skiltyje duomenys yra surenkami arba užkraunami iš tam tikro resurso, paskui suskirstomi į duomenų lenteles. Norint sudaryti lenteles atliekamas duomenų transformavimas ir filtravimas. Kitas etapas yra susiejimas vizualių duomenų su vizualizacijos išgavimo moduliu. Paskutinysis etapas yra medžio pateikimas vartotojui. Be to, sudaroma galimybė vartotojui

pateikti tam tikrus parametrų pakeitimus ir gauti nuo jų priklausančius rezultatus (pvz. grafo dydį, gylį, priartinimą ir kt.)

Paketus sudaro iš anksto apibrėžti komponentai. Vieni pagrindinių elementų - tai kraštinės ir mazgai, kurie atlieka standartines funkcijas atvaizduojant elementus. Žemiau pateikiami kiti pagrindiniai vizualizavimo komponentai:

- Susiejimas (Renderers): EdgeRenderer naudojamas norint susieti ir priskirti briaunų atvaizdavimo parametrus,. LabelRenderer – susieti teksto elementus, o PolygoneRenderer susieti geometrinius elementus.
- Išdėstymas (Layouts): RandomLayout, GridLayout, ForceDirectedLayout, TreeLayout, RadialTreeLayout nurodo medžių ir grafikų išdėstymo maketus.
- Kontrolė: DragControl, ZoomControl, PanControl, ToolTipControl ir keletas kitų naudojami sukurti vartotojo sąsajai.

Prefuse vizualizacijos etapai



Šaltinis: Sukurta autoriaus

6 pav. Prefuse vizualizacijos etapai

- Duomenų parinkimas: parenkami duomenis iš įvairių išteklių. (GraphML (XML), TreeML (XML), Text, CSV.)

  Šaltinio (pirminiai) duomenys: parenkami neapdoroti duomenys. Jie dar negali būti naudojami tiesiogiai, kad vyktų vizualizacija.

- Duomenų filtravimas, transformavimas. Atrenkami tik vizualizavimui skirti duomenys ir transformuojami taip, kad būtų galimi naudoti priklausomai nuo vizualizavimo tipo (Grafas, medis ir kt.).
- Duomenų lentelės. Visi transformuoti duomenys surašomi į duomenų lenteles, kurios naudojamos vizualizacijai gauti.
- Duomenų susiejimas. Susiejami duomenys taip, kad būtų galimas jų atvaizdavimas, nurodomi ryšiai.
- Vizualizacijos išgavimas. Nustatomos tam tikri atvaizdavimo požymiai. (Pvz.: spalva, dydis, vieta).
- Transformacijų peržiūra. Vizualizacijos atvaizdavimas pakeitus tam tikrus parametrus. (Pvz. Peržiūros gylį, priartinimą).
- Peržiūra. Tai galutinis vizualizacijos pateikimas vartotojui.

Prefuse paketo apibendrinimas

Prefuse yra labai efektyvus paketas, turintis didelį kiekį komponentų ir metodų informacijos vizualizavimui. Paketas suteikia galimybes atvaizduoti net ir didžiausius kriterijus reikalaujančias vizualizacijas. Kadangi Prefuse sukurtas naudojant išskirstytą vizualizavimo dizainą, todėl jis yra labai lankstus ir gali realizuoti bet kokius reikalingus funkcionalumus bei reikalavimus.

Taigi Prefuse skirtas atvaizduoti tarpusavyje susijusius duomenis, kurie gali būti pateikti medžio arba grafo struktūros, bet taip pat gali naudoti duomenų lenteles saugoti nesusietiems duomenims. Esant atskirtiems susiejimo ir duomenų procesams atsiranda piešimo nepriklausomumas. Tuomet spalvinimo, priartinimo, centravimo, koordinačių parinkimo metodų naudojimas tampa atskirtas nuo loginių procesų.

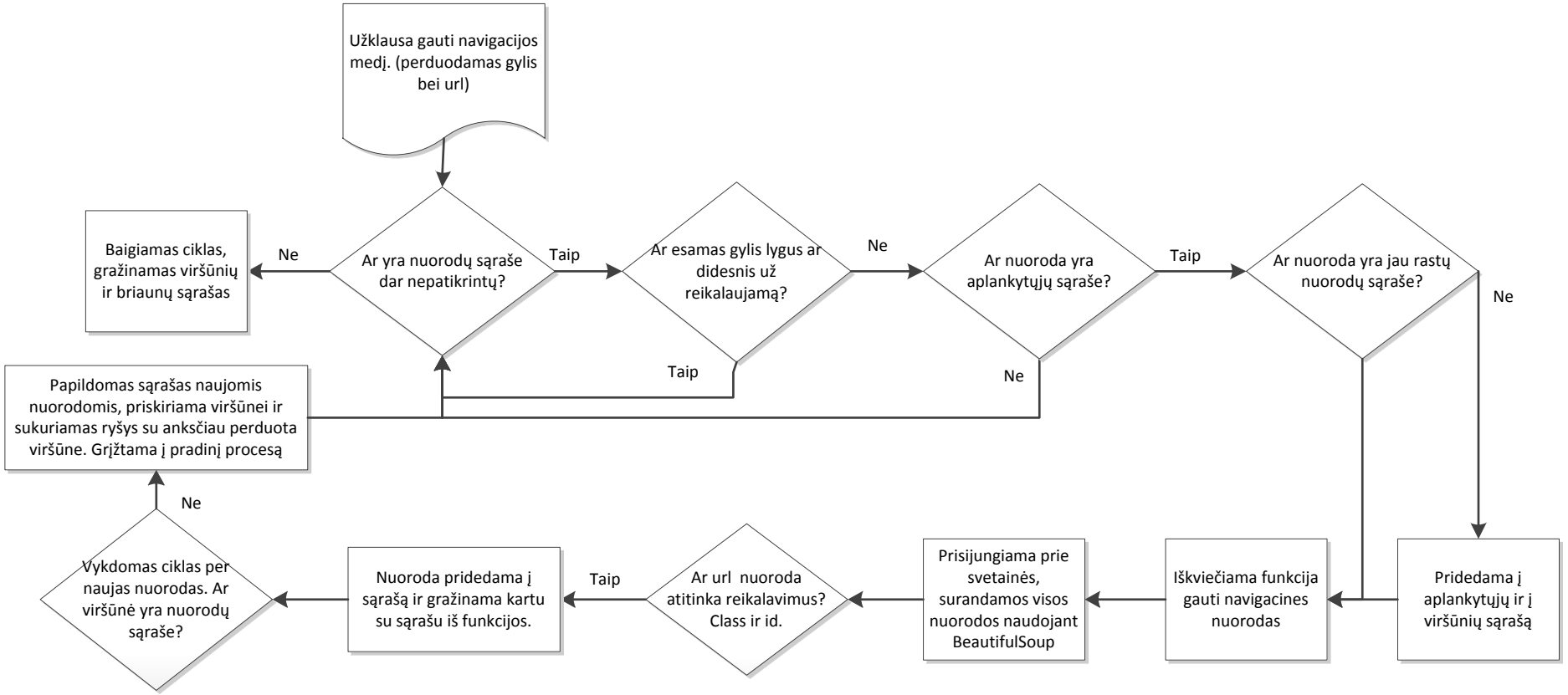
3.3. Tinklapio medinės struktūros radimas, išgavimas ir vizualizavimas taikant pasirinktus metodus

3.3.1. Class meniu artumo metodo taikymas

Vienas iš išnagrinėtų ir pasirinktų metodų yra class meniu artumo. Jis padės realizuoti tinklapio medinės struktūros radimo, išgavimo bei vizualizavimo uždavinį. Class meniu artumo metodas yra skirtas surasti ir išgauti svetainės meniu esančių nuorodų medį.

Metodo algoritmą sudaro šios dalys:

- Python skripto naudojimas rasti ir gauti svetainės medį.
- Funkcionalumų realizavimas gauti kokybišką vizualizaciją.
- Tolimesnis vizualizacijos vystymas.



Šaltinis: Sukurta autoriaus

7 pav. Python skripto veikimo schema

Realizuojant class meniu artumo metodą pirmiausia iškviečiamas python skriptas, kuriam yra perduodamas pagrindinis svetainės adresas. Atliekamas patikrinamas ar nuoroda pabaigiama „/“ ženklu. Jeigu ne, ji yra papildoma. Visų pirma yra sudaromas jau aplankytų puslapių sąrašas, pridedant pagrindinį tinklapio adresą, kuris perduodamas navigacinių nuorodų gavimo funkcijai. Šioje funkcijoje yra prisijungiama prie svetainės ir gaunami visi navigaciniu tipu priskirti adresai. Yra remiamasi semantine puslapio informacija. (t.y. class ir id atributais). Perduotasis adresas yra patikrinamas, ar jame nėra pdf, jpg, jpeg, png, gif, bmp, wav, midi, mp3, ps, doc, ppt, tif, tiff formato failų. Tuo atveju jeigu nuorodą sudaro šie atributai, tolimesni veiksmai neatliekami. Prisijungti prie svetainės ir gauti duomenis iš URL naudojamas urllib2 modulis. Šis modulis suteikia vieningą kliento sąsają su HTTP, FTP ir gopher. Jis automatiškai išrenka tinkamą protokolą priklausomai nuo URL perduoto bibliotekai „urllib“ modulis turi keletą įdomių savybių kaip:

- Atidaryti URL ir parinkti automatiškai jam tinkamą sąsają.
- Turi funkciją URL apdorojimo, pavyzdžiui prisijungus prie HTML puslapio, jo parametrų tvarkymą.
- Proxy ir HTTP bazinio autentifikavimo galimybės.

urlopen() atlieka HTTP GET (http gavimo) užklausą ir gauna pateikto puslapio turinį. Papildomi parametrai taip pat gali būti perduodami ir gaunami kartu su URL pateikimu užklausoje. Šiuo atveju naudojame urllib2.urlopen(page_url) puslapio duomenų gavimui.

Kitas etapas yra duomenų išnagrinėjimas, navigacinių nuorodų išgavimas. Tinklo duomenų nagrinėjimui ir nuorodų gavimui rasti galima naudoti *lxml* arba *BeautifulSoup* modulius. Šie moduliai atlieka duomenų išgavimą iš tinklapių ir gali sudaryti jų medį. Dokumento medžio samprata - Analizatorius (parser) sukuria medį. *Lxml* unikalus tuo, jog apjungia išsamias XML funkcijas ir veikia labai greitai.[15] Tačiau rekomenduojama naudoti *BeautifulSoup,* nes turi geresnį ir patogesnį platformų palaikymą, kodo aptikimo suportą, be to, lengviau konfigūruojamas. Jis išnagrinėja XML (gali būti ir netvarkingas) arba HTML dokumentą į medžio formos atvaizdavimą. Jo pateikiami metodai ir Python kalbos galimybės leidžia lengvai naviguoti, atlikti paiešką ir keisti medį. Pagrindinė idėja naudojant BeautifulSoup yra ta, jog kiekvienas <tag> HTML kode yra priskiriamas medžio viršūnei. Daugybė viršūnių, kaip lentelių viršūnės, taip pat turi savo vaikines viršūnes. BeautifulSoup sudaro metodai, kurie leidžia atlikti paiešką medyje arba medžio dalyje, pradedant pasirinkta viršūne. Kai ieškoma viršūnė yra randama, duomenys iš jos gali būti gaunami arba išspausdinami. Aprašomajame

[15] http://wiki.python.org/moin/beautiful%20soup

metode naudojamas BeautifulSoup modulis gražina visų tinklapio nuorodų medį. Kitas žingsnis yra atrinkimas tinkamų nuorodų, kurios atitiktų meniu navigaciją. Atliekama paieška grąžintame nuorodų medyje pagal id ir class atributus, kuriuose būtų "menu" , „meniu" arba "nav" elementas. Tam pasitelkiamos regex funkcijos paieškai dokumente. Atrankos eiliškumas:

- Atrinkimas pagal id ir class (nav|menu|meniu)
- Atrinkimas atributų <a> ribose href nuorodų radimui.
- Javascript kode paieška nuorodų su atributu href, nuorodai rasti.
- Atmetamos nuorodos su skirtingu baziniu url, kad liktų tik svetainės vidinė struktūra.

Atlikus šiuos veiksmus funkcija grąžina jau atrinktų ir suskirstytų į medį nuorodų sąrašą. Tuomet vyksta ciklas per visas nuorodas, patikrinama, ar ji nėra jau lankytųjų sąraše ir tuomet sukuriamas ryšys tarp naujosios nuorodos ir tuo metu esančios pagrindinės nuorodos. Tokiu principu yra randamos nuorodos, jos išgaunamos ir grąžinamos jau paruoštos atvaizduoti medinei struktūrai.

Vizualizavimo pakete visų pirma gautasis viršūnių ir briaunų sąrašas yra išsaugomas jas apibūdinančiose duomenų lentelėse. Papildomai pridedamas papildomas stulpelis pavadinimams.

Viršūnių lentelės pavyzdys (pirmajame lygyje, kol neatlikta pavadinimų gavimo funkcija, pavadinimų stulpelis yra tuščias):

2 lentelė: Viršūnių duomenų lentelė

url adresas	**Nuorodos pavadinimas**
http://www.vu.lt/lt/studijos/aktualu/	Aktualu
http://www.vu.lt/lt/apiemus/struktura/	Struktūra
http://www.vu.lt/lt/apiemus/istorija/	Istorija

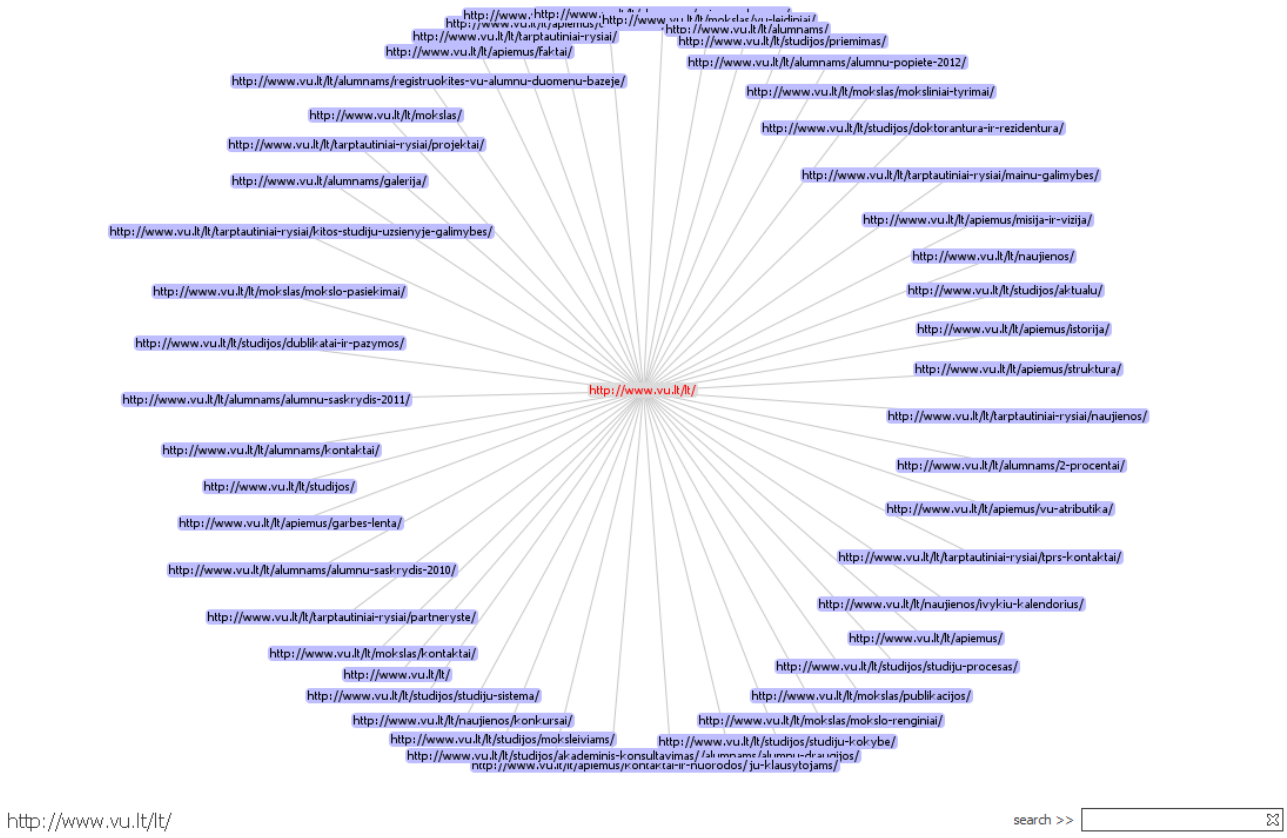
3 lentelė: Briaunų lentelės pavyzdys

Viršūnės ID iš kurios yra rodoma	**Viršūnės ID į kurią yra rodoma**
0	3
0	4
1	7

Tuomet yra priskiriami ir sukonfigūruojami pačiai vizualizacijai skirti elementai:

- Parenkama viršūnių, briaunų spalva, teksto šriftas, dydis, forma, medžio išdėstymas.

- Sukuriami veiksmai: priartinimas, pozicijų automatinis priskyrimas, spalvų, tekstų perpiešimas pagal parinktus parametrus. Įterpiama paieškos, aktyvios nuorodos teksto išvedimo funkcijos.
- Sukuriamos animacijos ir kontrolės po vartotojo atliktų veiksmų.



Šaltinis: Sukurta autoriaus, programos rezultatas

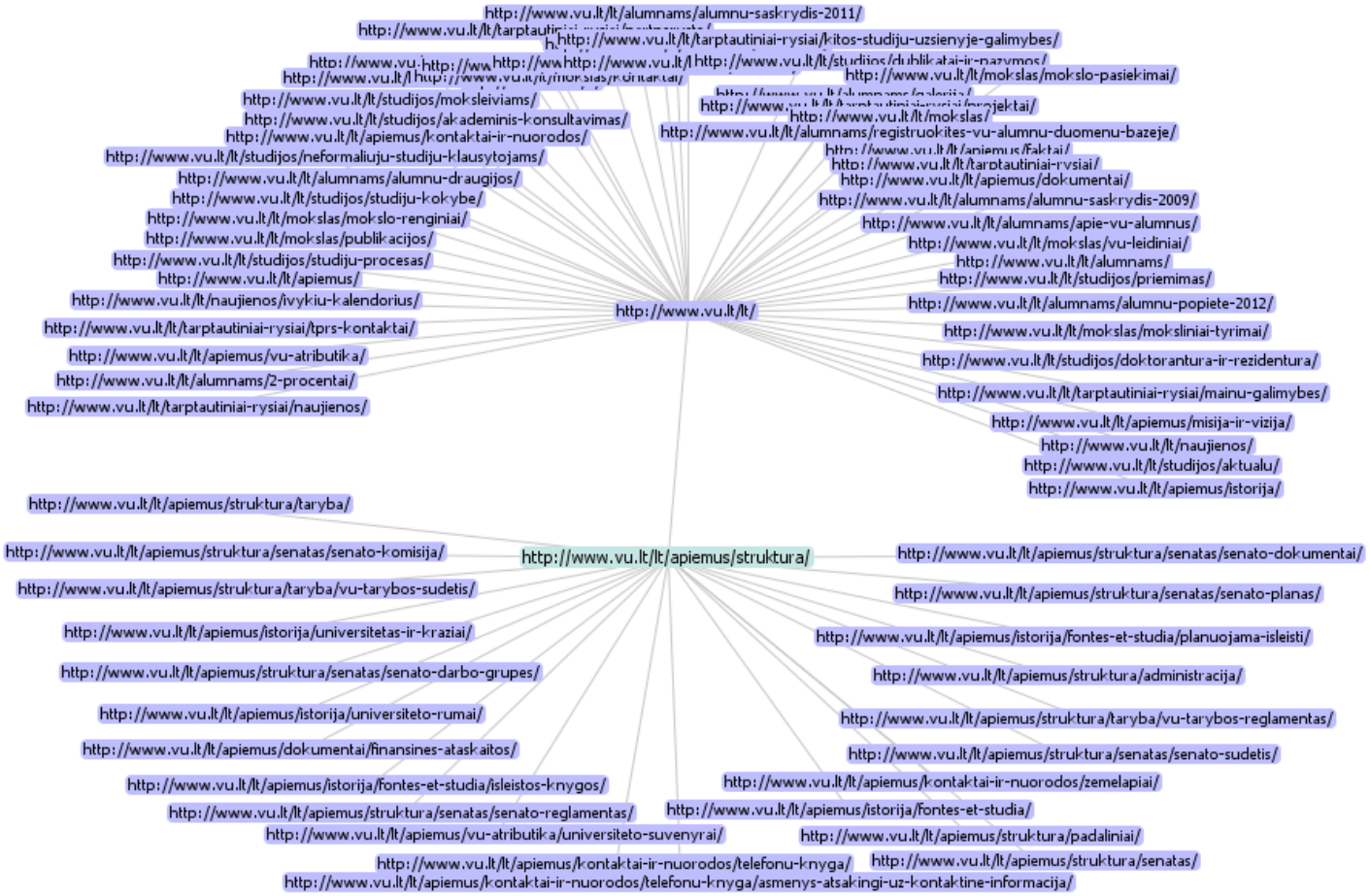
8 pav. Svetainės pirmojo lygio vizualizacija

Tolesnės vizualizacijos vystymas apima naujų funkcijų sukūrimą:

1. Dvigubo pelės paspaudimo ant bet kurios viršūnės veikimas, norint gauti tolimesnį jos lygį.
2. Pavadinimų gavimas ir atvaizdavimas paspaudus dešinį pelės mygtuką ant bet kurios viršūnės
3. Nepriklausomos viršūnės, mygtuko sukūrimas ir funkcijos sukūrimas, kad būtų atkeičiami viršūnių pavadinimai į url adresus.

Norint atlikti interaktyvų vizualizavimą, reikia sukurti veiksmą, kuris suteiktų galimybę skleisti sukurtą medį. Tam pasitelkiamas kontrolės adapteris, kuris suveikia paspaudus du kartus ant bet kurios viršūnės. Yra gaunamas url adresas viršūnės kuri yra paspausta ir perduodama į funkciją naujam grafui sudaryti. Šiam veiksmui naudojamas jau aprašytas python skriptas, kuris grąžina pirmojo lygio kaimynines viršūnes iš pateiktosios viršūnės.

Taip sukuriamas antrasis grafas. Toliau yra iškviečiama funkcija grafų apjungimui. Perduodamas atvaizduotas grafas, naujasis grafas, pagrindinio grafo viršūnių skaičius, aktyvi viršūnė ir aukštesniojo lygio viršūnės ID. Toliau atliekamas ciklas per naujųjų viršūnių sąrašą bei atliekamas ciklas per pagrindinių viršūnių sąrašą, patikrinant, ar viršūnės nėra dvejinamos. Jeigu tikrinama viršūnė nėra dvejinama, nėra perduotoji viršūnė ir nėra aukštesnio lygio viršūnė, ji yra pridedama prie pagrindinio grafo. Taip pat yra sukuriamas ryšys iš perduotosios viršūnės ID, į naujai sukurtos viršūnės ID. Tokiu būdu yra papildomas grafas su naujai atrastomis, kito lygio viršūnėmis. Toliau atliekami vizualizacijos perpiešimo ir atnaujinimo veiksmai.



Šaltinis: sukurta autoriaus, programos rezultatas

9 pav. Aukštesniojo lygio viršūnės išskleidimas

Kadangi atvaizduotos viršūnės yra pavadintos nuorodų vardais, kurie yra neaiškiai pavadinti, todėl dažnai gali būti sunku suprasti, kurį puslapį iš tiesų ta viršūnė atitinka. Dėl šios priežasties sukuriama funkcija leidžianti vartotojui gauti nuorodų pavadinimus. Yra sukuriamas valdymo

adapteris, kuris aktyvuojasi paspaudus ant bet kurios viršūnės dešinį pelės mygtuką. Tuomet visų pirma yra gaunamas paspaustosios viršūnės url pavadinimas ir perduodamas į naują funkciją „Gauti pavadinimus (paspaustas url)“.

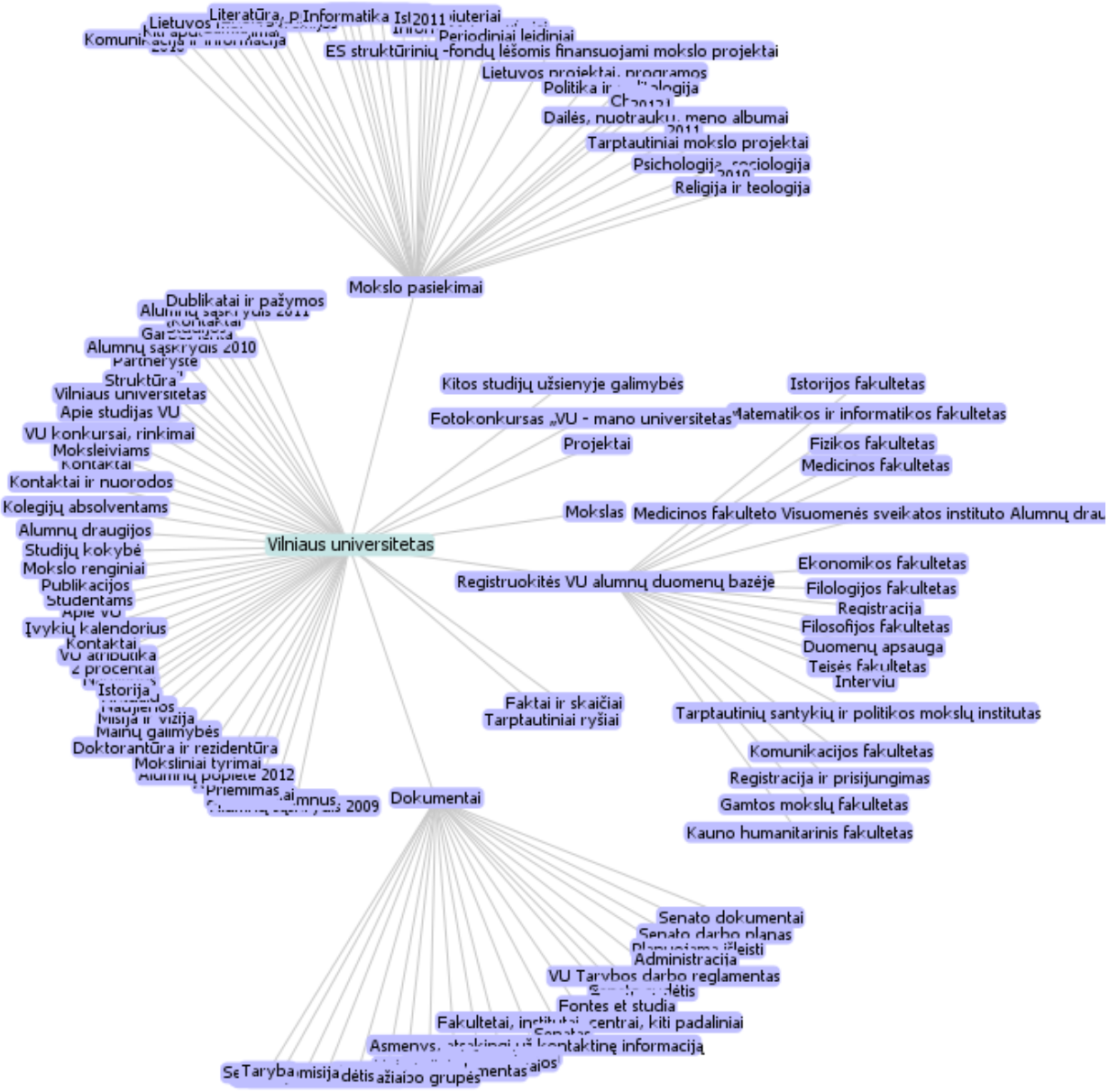
Funkcijos tikslas yra gauti sąrašą pavadinimų, susietų su url adresais. Visų pirma funkcijoje sukuriami sąrašai saugoti url adresams ir pavadinimams ir iškviečiamos funkcijos jiems gauti. Gavimo veikimas panašus, Jsoup modulio pagalba yra prisijungiama prie pateiktojo adreso ir yra gaunamas puslapio kodas. Adresų gavimo funkcija suranda visus "a[href]" adresus ir atfiltruoja turinčius nuorodas pagal "abs:href" paiešką. Tuomet yra grąžinamas sąrašas. Pavadinimų gavimas yra atliekamas ta pačia struktūra, norint, kad nuorodų ir pavadinimų kiekis nesikeistų ir būtų galimybė juos susieti. Taip pat yra prisijungiama prie svetainės, gaunamos puslapyje esančios nuorodos ir tuomet atliekamas ciklas per adresų sąrašą. Yra tikrinama ar yra *link.text()* priskirta prie gautųjų nuorodų. Jeigu yra, tuomet patalpinamas nuorodos pavadinimas į sąrašą, jeigu nėra, tuomet pridedamas esamas url adresas, taip nepaliekant visiškai tuščios informacijos pavadinimų sąraše. Grąžinti abu sąrašai yra perduodami į „dublikatai“ funkciją, kurioje panaikinami dublikatų reikšmes turintys url adresai. Prieš grąžinant rezultatą iš pagrindinės funkcijos, yra apjungiami sąrašai. Gaunama tokia struktūra:

4 lentelė : Pavadinimų ir adresų masyvas

Url adresas 1	Pavadinimas pagal url adresą 1	Url adresas 2	Pavadinimas pagal url adresą 2	Url adresas 2	Pavadinimas pagal url adresą 2

Tuomet daroma iteracija per visas atvaizduotas viršūnes ir gautąjį sąrašą, lyginami url adresai. Radus vienodus adresus pavadinimai yra priskiriami kiekvienam atskirai į viršūnių lentelės stulpelį. Atliekamas pakartotinis veiksmų įvykdymas, perpiešiant vizualizacijos elementus ir suanimuojant.

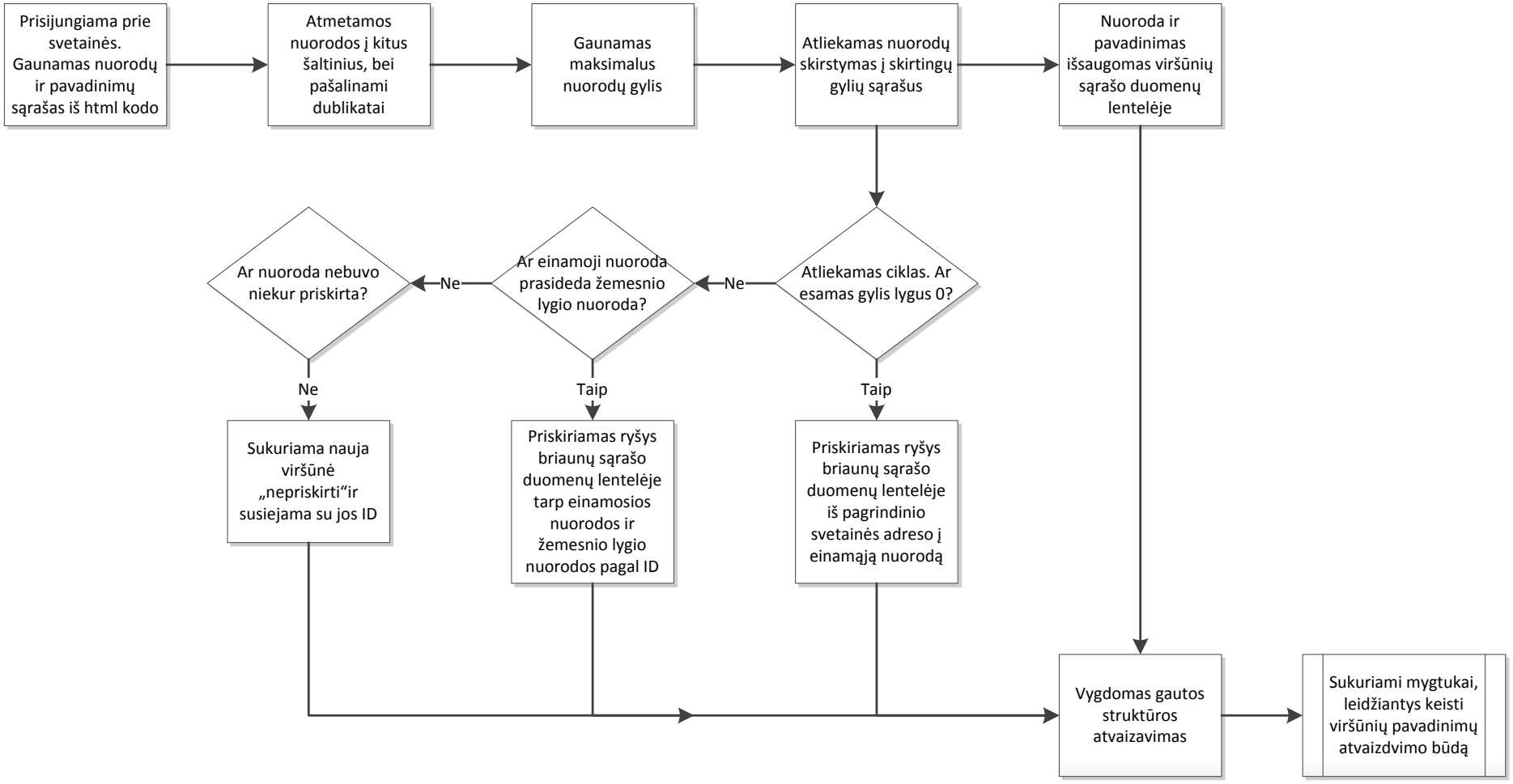
Norint pakeisti pavadinimus į url adresus yra sukuriama nepriklausoma viršūnė, ant kurios paspaudus dešinį perlės mygtuką atliekami pervadinimo veiksmai. Parenkamas „url“ lentelės stulpelis kaip pagrindinis viršūnių tekstas bei atliekamas vizualizacijos atnaujinimas.



Šaltinis: Sukurta autoriaus, programos rezultatas

10 pav. Vizualizacija, atvaizduojant nuorodų pavadinimus

3.3.2. Url metodo taikymas



Šaltinis: Sukurta autoriaus

11. pav URL metodo veikimo schema tinklapio struktūros vizualizavimui

URL metodo veikimas yra pagrįstas gaunamų nuorodų teisingo suskirstymo į medinę struktūrą atrinkimas. Visų pirma atliekamas prisijungimas prie svetainės ir visų nuorodų joje radimas. Tam naudojamas Jsoup modulis, kuris iš pateiktojo adreso gauna visą puslapio kodą ir atlieka teksto paiešką jame. Pagal *"a[href]"* atributą surandamos nuorodos esančios <a> atributo ribose su href identifikatoriumi (abs:href) ir patalpinamos sąraše. Tokiu pačiu principu gaunami nuorodų pavadinimai (link text). Papildomai yra priskiriama pavadinimams url adresas, jeigu prie nuorodos jis nėra pateikiamas link text atribute. Gautieji sąrašai yra perduodami funkcijai kuri atlieka dublikatų panaikinimą. Remiamasi url adresų lyginimu. Taip gaunami unikalių nuorodų sąrašai, kurie toliau naudojami teisingam jų suskirstymui ir priskyrimui viršūnių ir briaunų lentelėse.

Kitas žingsnis yra tinkamų nuorodų atrinkimas ir didžiausio gylio nuorodose radimas. Tam sukuriamos iteracijos per adresų ir pavadinimų sąrašą. Yra atliekamas kiekvieno adreso tikrinimas pagal pagrindinį svetainės adresą, taip atfiltruojant nuorodas į kitus šaltinius. Taip pat atmetami adresai, kurie yra tik svetainės adreso ilgio su „/“ atskyrimu bei adresai prasidedantys „~” ženklu po svetainės adreso, kadangi tai dažniausiai neaktualios nuorodos. Iš turimų adresų pašalinamas pradinis svetainės adresas, taip paliekant tik labiausiai mus dominančią adreso dalį. Naudojant *spulit(„/“).length* funkciją yra skaičiuojamas nuorodos gylis. Kaskart išsaugojant didžiausią rastą gylį iteracijoje surandamas maksimalus esamas gylis.

Tolimesnei eigai reikia atrinkti nuorodas ir suskirstyti į skirtingų gylių sąrašus. Šiai užduočiai yra sukuriamas dvimatis masyvas, kuriame bus saugoma skirtingų gylių adresai. Atliekamas ciklas maksimalaus gylio skaičiui bei iteracija per visas esančias nuorodas. Paskaičiuojamas kiekvienos nuorodos gylis ir jeigu jis yra tam tikro ciklo etapo (pvz. Gylis=1) skaičiui, ji yra patalpinama į tą masyvą bei kartu sukuriama viršūnė grafo su visais atributais. Priskiriamas pavadinimas bei ID. Kiekvienas ID taip pat išsaugojamas masyve, kad būtų galimybė vėliau jį panaudoti viršūnių susiejimo procese.

Ryšių sudarymui atliekama iteracija per jau sudarytus rastų viršūnių, suskirstytų pagal gylį masyvų sąrašus. Paeiliui yra atliekamas ciklas kiekvieno gylio sąraše juos susiejant. Atliekamas lygio viršūnių susiejimas tokiu principu:

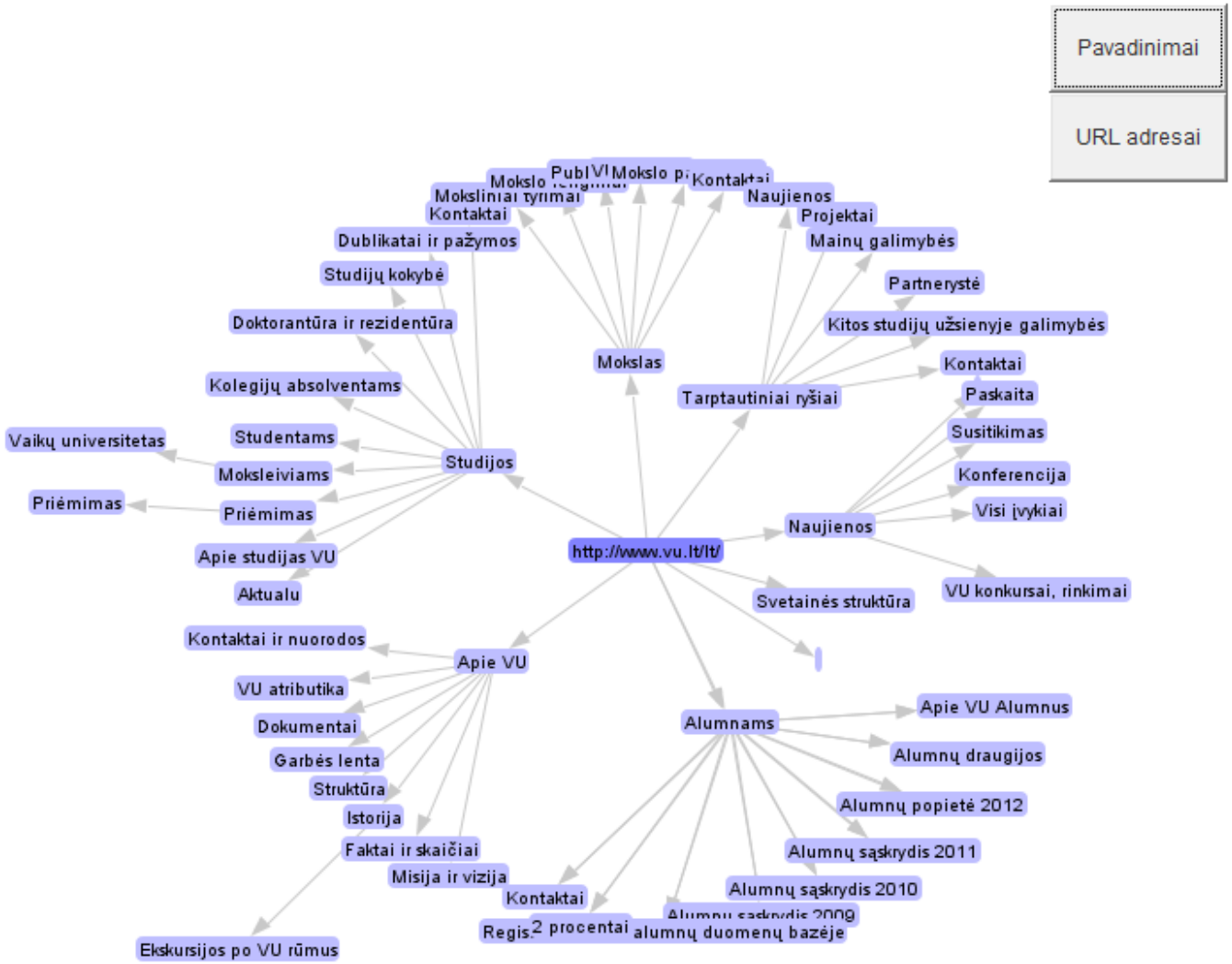
- Einamoji ciklo viršūnė (pagrindinis puslapio adresas) yra priskiriamas kaip pagrindinis
- Gaunamas aukštesnio lygio nuorodų sąrašas. Atliekama iteracija per jo elementus
- Jeigu gylis yra 0 tuomet visos sąraše esančios nuorodos yra priskiriamos pagal pagrindinės viršūnės ID
- Jeigu gylis yra aukštesnio lygio atliekamas tikrinimas ar aukštesnio lygio nuoroda prasideda einamąja nuoroda. Tuomet sukuriamas ryšys pagal jų ID.

- Yra galimybė, jog nuorodos gali neturėti savo žemesnio lygio viršūnės. (pvz. Nėra sukurta puslapio adresu www.vu.lt/lt/pastabos tačiau yra viršūnė www.vu.lt/lt/pastabos/mokymai/) Tuomet sukuriama viršūnė pavadinimu „nepriskirti“, su kuria yra susiejamos neturinčios tėvinės viršūnės nuorodos.

Šiuo metodu yra sudaromos viršūnių ir briaunų lentelės medžio vizualizacijai. Jos pateikiamos atvaizduoti. Sukuriami viršūnių ir briaunų atvaizdavimo veiksmai, parenkami nustatymai bei kontrolės. Skirtingai nei class artumo metode, šiuo atveju yra atvaizduojamas visas medis, o ne skirtingų gylių medis bei interaktyvus keliavimas per jį. Papildomai yra sukuriami mygtukai, kuriais vartotojas gali keisti vizualizacijos atvaizdavimą. Gali peržiūrėti svetainės struktūros medį pagal pavadinimus arba pagal adresus.

Šaltinis: Sukurta autoriaus, programos rezultatas

12 pav. www.vu.lt svetainės medis pagal url metodą, pateikiant nuorodų adresus



Šaltinis: Sukurta autoriaus, programos rezultatas

13 pav. Svetainės struktūros medžio atvaizdavimas pateikiant pavadinimus

3.3.3. Tinklapio medinės struktūros klasifikuota metodų vizualizacija

Metodų apjungimas suteikia galimybę praplėsti gaunamą svetainės medį ir rasti tinkamiausią metodą būtent nagrinėjamos svetainės struktūrai rasti. Visų pirma reikia suvienodinti naudojamų metodų viršūnių ir briaunų duomenų lenteles. Viršūnių lentelės sandara yra pakeičiama, siekiant pritaikyti tolimesniems procesams:

5 lentelė: Pavyzdinė viršūnių duomenų lentelė

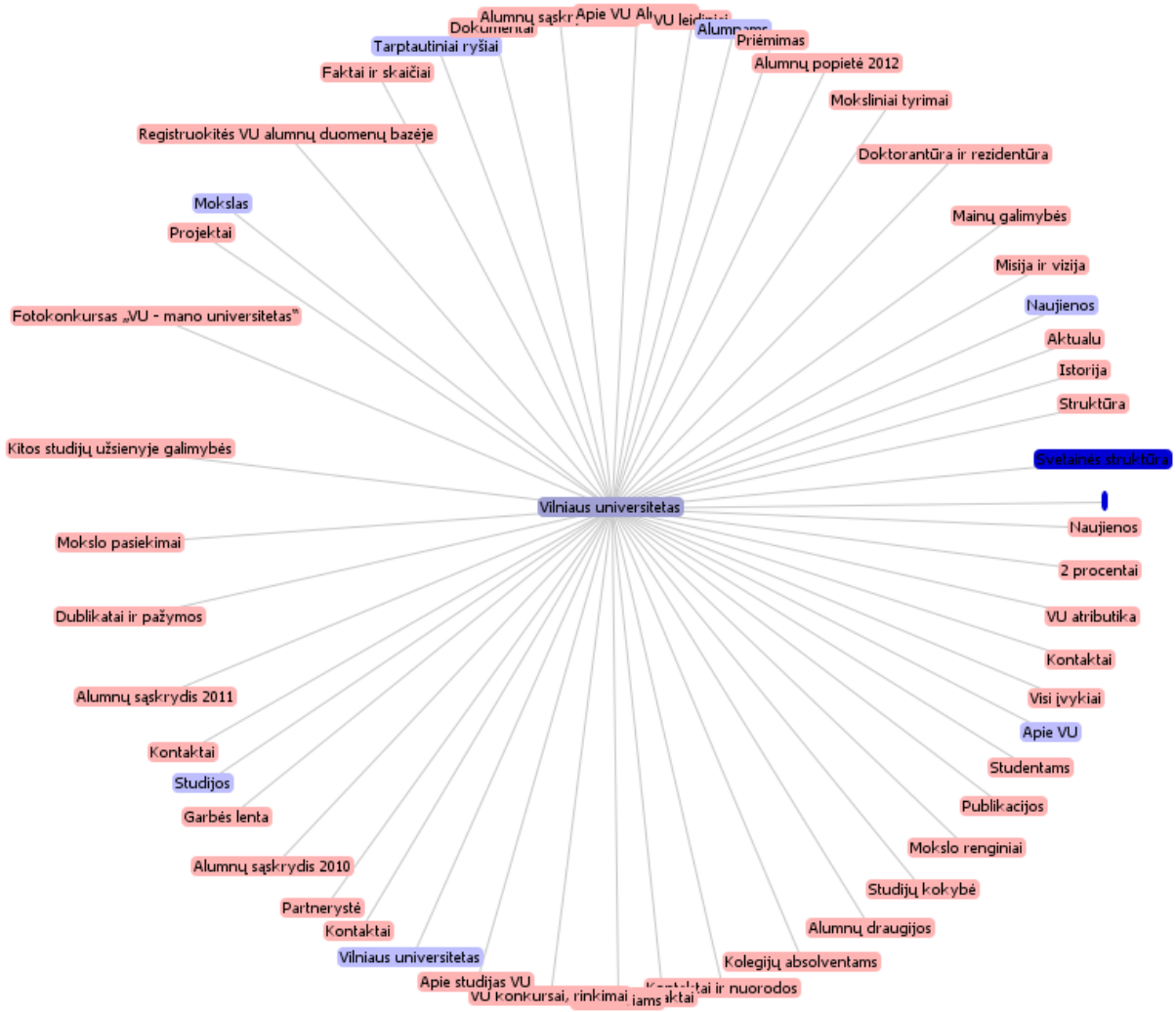
url adresas	**Nuorodos pavadinimas *"pav"***	**Metodas *"metod"***	**Tinka *"tinka"***
http://www.vu.lt/lt/studijos/aktualu/	Aktualu	url	Ne
http://www.vu.lt/lt/apiemus/struktura/	Struktūra	Class	Taip
http://www.vu.lt/lt/apiemus/istorija/	Istorija	Class, url	Ne

Papildomas stulpelis „metod“ naudojamas metodo, kuriuo jis yra gautas priskyrimui. Stulpelis „tinka“ turintis „taip“ arba „ne“ reikšmes bus naudojamas klasifikavimo uždaviniui spręsti.

Apjungimo metodas visų pirma atlieka class meniu artumo metodo funkcijas, skirtas svetainės navigacinio medžio radimui ir išgavimui. Metodo veikimo metodika yra aprašyta ankstesniame skyriuje. Šiuo būdu yra gaunamos viršūnių ir briaunų duomenų lentelės. Kadangi yra aktualu iškart turėti nuorodų pavadinimus viršūnių lentelėje, nuorodų pavadinimų išgavimas ir išsaugojimas atliekamas adekvačiai. Kitas procesas yra url metodo iškvietimas ir svetainės medžio radimas bei išgavimas. Atliekamas kreipiamasi į url metodo gavimo funkciją, kuri grąžina svetainės medžio struktūros duomenų lenteles. Kadangi class meniu artumo metodas pateikia tik pagrindinės nuorodos pirmojo lygio nuorodas, o url metodas grąžina pilną svetainės struktūrą, todėl yra reikalingas metodų apjungimo sprendimas. Šiai užduočiai įvykdyti yra sukuriama funkcija, kuriai perduodami abiejų metodų duomenų lentelės, pagrindinė viršūnės adresas bei jos ID.

Norint rasti pagrindinės nuorodos „kaimynes“, apjungimo funkcijoje yra vykdomas ciklas per gautąsias url metodo viršūnes tikrinant briaunų lentelėje ar perduotosios nuorodos ID yra lygus tikrinamos nuorodos ID („*source*“). Tenkinančiose šią sąlygą briaunose randamos viršūnės į kurias rodo pagrindinės viršūnės ID („*target*“). Pagal gautąjį ID ieškomas svetainės adresas viršūnių lentelėje. Šiuo būdu randamos visos kaimyninės nuorodos url metodo medyje iš perduotosios pagrindinės viršūnės. Toliau atliekama paieška per class radimo metodo viršūnes, lyginant adresus su esamu url metodo adresu. Radus, yra išsaugojamas jos ID ir pagal jį yra priskiriamas metodo lauko „url“ pavadinimas. Jeigu adresas nėra rastas esamos vizualizacijos viršūnėse, duomenų lentelėse yra pridedama nauja viršūnė. Jai priskiriami adreso, pavadinimo,

gautojo metodo („url“), tinkamumo („ne“) atributai su reikšmėmis bei sukuriamas ryšys su jos aukštesniojo lygio viršūne. Grąžinus medžio duomenų lenteles yra sukuriama vizualizacija pagal nustatymus.



Šaltinis: Sukurta autoriaus, programos rezultatas

14 pav. Tinklapio struktūros atvaizdavimas apjungus metodus

(Rezultatai: meslsvai – abu metodai, rausvai – class artumo metodas, mėlynai – url metodas rado viršūnes).

Sekantis žingsnis yra realizuoti metodų apjungimą vykdant tolimesnį medžio skleidimą. Šios užduoties vykdymui sukuriama kontrolė, kuri yra aktyvuojama paspaudus du kartus kairįjį pelės klavišą ant pasirinktos viršūnės. Tuomet iškviečiama naujų class meniu artumo kaimyninių viršūnių radimo ir prijungimo funkcija. Papildomai yra perduodama paspaustosios viršūnės adresas ir jos ID apjungimo funkcijai, siekiant rasti ir apjungti url metodu gautas tos pačios viršūnės artimiausias nuorodas. Skirtingam elementų atvaizdavimui sukuriama kontrolė, kuri aktyvuojama paspaudus dešinį pelės mygtuką du kartus ant bet kurio atvaizduoto elemento. Ji pakeičia visų esančių viršūnių pavadinimus į nuorodų adresus. Norint atkeisti viršūnių tekstą į

nuorodų pavadinimus, sukuriama kontrolė, kuri aktyvuojasi paspaudus dešinį pelės mygtuką ant viršūnės tris kartus. Tokiu būdu vartotojas gali keisti pavadinimų atvaizdavimo būdus.

Norint atrinkti geriausiai tinkantį metodą yra atliekamas metodų klasifikavimas. Šiai užduočiai įvykdyti naudojamas J48 klasifikavimo algoritmas, kuris realizuojamas Weka bibliotekomis. Jis yra realizuojamas skleidžiant medį pagal pasirinktą viršūnę. Visų pirma vartotojas sužymi paspausdamas ant kiekvienos atitinkančios struktūrą viršūnės pagal šalia vizualizacijos matomą svetainės vaizdą. Skleidžiant naująjį lygį yra sukuriamas apmokymui skirtas klasifikavimo medis. Norint apibūdinti informacijos išgavimą tiksliai, visų pirma reikia gauti entropiją, kuri charakterizuoja turimų kolekcijų pavyzdžių grynumą (negrynumą) ir parodo kiek duotajam atributui priklauso informacijos. Rinkinys S yra sudarytas iš teigiamų ir neigiamų tikslinės koncepcijos pavyzdžių.

Mūsų atveju rinkinys S atitinka koncepciją:

Ar nuoroda atitinka svetainės medinę struktūrą? {Taip, Ne}.

6 lentelė: Klasifikavimo duomenų rinkinio S pavyzdys

Meniu dalis	Class meniu artumo	URL	Ar nuoroda atitinka svetainės medinę struktūrą
1	Taip	Taip	Taip
2	Taip	Ne	Ne
3	Ne	Taip	Taip
4	Taip	Ne	Taip
5	Ne	Ne	Ne
6	Taip	Taip	Taip
7	Taip	Ne	Ne

Atributai:

1 Metodas (Class meniu artumo) = {Taip, Ne}

2 Metodas (URL) = {Taip, Ne}

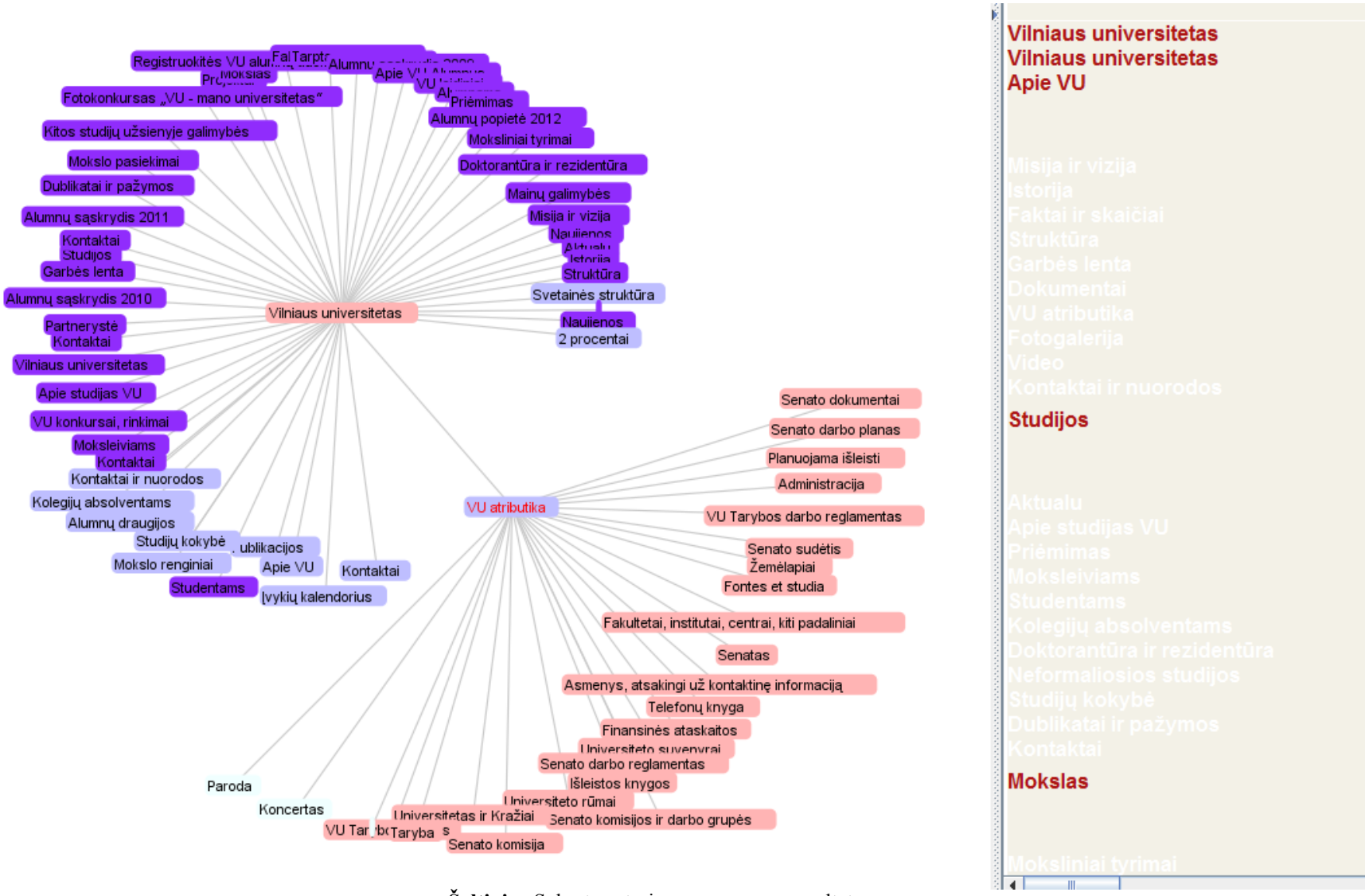
Pirmiausia paskaičiuojama Entropija duoto S rinkinio pagrindiniam vertinimui :

$$\text{Entropija}(S) = - pt \log_2 pt - pn \log_2 pn$$

Kur pt yra S rinkinio teigiamų („taip“) reikšmių ir viso reikšmių santykis, o pt yra S rinkinio neigiamų („Ne“) reikšmių ir viso reikšmių santykis.

Esant didesniam kiekiui apmokomųjų, pavyzdinių duomenų rinkinių pati tinkamiausias metodas gali kisti. Todėl atlikus gausesniam tinkamų viršūnių pažymėjimo procesui, bus tiksliau nuspėjamas tinklapio medinę struktūrą atitinkantis metodas.

Norint suteikti vartotojui patogumo instrumento naudojimui, šalia svetainės medinės struktūros vizualizacijos yra pateikiamas tinklapio naršymo langas. Tokiu būdu vartotojas gali nesunkiai rasti, bei žymėti svetainės pagrindinę struktūrą atitinkančias viršūnes.



Šaltinis: Sukurta autoriaus, programos rezultatas

15 pav. Tinklapio medinės struktūros vizualizacija pritaikius metodų klasifikavimą

3.4. Gautų rezultatų vertinimas

Atliekant tinklapių medinės struktūros radimo, išgavimo ir vizualizavimo uždavinius buvo išnagrinėti literatūros šaltiniai susijusiais darbo tema klausimais, sukurti url, class meniu artumo metodai atskiram jų naudojimui, pritaikius metodų klasifikavimą pagal ID3 klasifikatorių ir apjungus metodus, sukurtas instrumentas, skirtas svetainės medinės struktūros atvaizdavimui. Vizualizacijoje buvo pritaikytas patogus įrankio naudojimas vartotojui įgyvendinant galimybes: pateiktą medį artinti, tolinti, akcentuoti aktualias viršūnes, atlikti paiešką ir gauti nuorodų pavadinimus bei adresus, interaktyviai plėsti medį, pateikiant žemesnio lygio nuorodas. Dėl laiko stokos nepavyko sukurti ir apjungti svetainės medžio bei navigacinės eilutės metodų. Jos būtų leidę praplėsti įrankio rezultatų gavimo galimybes. Apibendrinant rezultatus galima teigti, jog darbo tikslas buvo pasiektiems.

Naudota programinė įranga:

1. Eclipse IDE for Java Developers. Java kodo rašymui, prefuse paketo redagavimui ir vizualizacijai kurti.
2. Java JDK 1.6. platforma java aplikacijoms vykdyti.
3. Python 2.5. Meniu struktūros išgavimo skripto redagavimui, testavimui ir kodo rašymui.
4. Weka 3.7.6. klasifikavimo uždaviniams spręsti taikoma programa

Rekomenduotina naudoti minėtas programinės įrangos versijas, kad nekiltų nesklandumų peržiūrint metodo veikimą.

IŠVADOS IR PASIŪLYMAI

Atlikus mokslinės literatūros analizę ir atlikus tyrimą, gautos šios išvados:

1. Atlikus metodinio potencialo analizę nustatyta, kad sprendžiant tinklapių medinės struktūros klausimus esminės mokslo ir praktikos problemos yra šios: trūksta adekvačių metodų ir/ar metodikų, skirtų pilnam ir išsamiam tinklapio struktūros radimui, išgavimi ir atvaizdavimui tiek Lietuvos, tiek pasaulio mastu, mokslinėje literatūroje pateikiami metodai nepilnai atitinka šiandieninius reikalavimus, išskiriamiems metodams būdingas jų taikymo fragmentiškumas bei metodų naudojimo nuoseklumo stoka.
2. Atlikus tinklapių medinės struktūros radimo, išgavimo ir vizualizavimo mokslinio potencialo analizę, identifikuoti šie galimi darbe taikytini metodai:
    a. *class meniu artumo* metodas, kuris leidžia atvaizduoti tinklapio meniu sandarą pagal tinklapio kode esančius atributus,
    b. *url* metodas, leidžiantis tinklapio nuorodas suskirstyti pagal jame esančius vidaus adresus.
3. Atrinkus aukščiau nurodytus metodus kaip potencialius darbo tikslui pasiekti skirtus instrumentus bei juos išanalizavus nustatyta, kad galimas class meniu artumo ir url metodų apjungimas, juos papildant klasifikavimo pagal struktūros tinkamumą požymiais, tokiu būdu sukuriant pilną ir išsamų darbo autoriaus individualų metodą, skirtą svetainės struktūros radimui, išgavimui bei atvaizdavimui.
4. Sukurtas efektyvus ir patogus tinklapių medinės struktūros vizualizacijos instrumentas, leidžiantis analizuoti tinklapio vidaus nuorodas, jų pavadinimus, navigaciją, tinklapio elementų tarpusavio ryšius. Naudojant darbo autoriaus sukurtą instrumentą išgaunama tinklapio struktūra ją pateikiant vizualiai. Sukurtas instrumentas dedikuotas tiek tinklapių kūrėjams, tiek jų savininkams, o efektyvi tinklapio struktūros vizualizacija yra visapusiškai naudinga šių struktūrų tyrėjams.

Siekiant plėtoti tinklapių struktūrų vizualizavimo klausimus, formuojami šie pasiūlymai:

- Pritaikius sukurtą instrumentą atlikti skirtingų tinklapių tipų medinės struktūros analizę, rasti bei išskirti panašumus. Panaudoti gautus duomenis tinklapių kūrimo navigacijos optimizavimo procese.
- Galimas instrumento plėtojimas sukuriant kitus rečiau taikytinus metodus: svetainės medžio, nuorodų eilutės, apjungiant į bendrą metodą kartu su url ir class meniu artumo.

- Galimas instrumento plėtojimas pritaikant įrankį svetainės lankytojams, suteikiant galimybę patiems išskirti svarbiausią turinį jų nuomone. Pagal gautus rezultatus įvertinti turinio perskirstymo galimybes.
- Pritaikyti kitokius tinklapio struktūros atvaizdavimo būdus.

LITERATŪRA

http://www.whatwg.org/specs/web-apps/current-work/multipage/the-map-element.html#the-area-element (žiūrėta 2012 04 15)

[CG10] Praphul Chandra ir Geetha Manjunath. *Navigational Complexity in Web Interactions*. *WWW 2010,* April 26–30, 2010, Raleigh, North Carolina, USA.

[Jms99] Jock, D.; Mackinlay, S. K.; Card, Ben Shneiderman (eds.). 1999. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers.

[Kms+] Keim, D. A., Mansmann, F., Schneidewind J., and Ziegler, H. 2006. *Challenges in visual data analysis*. In Proceedings of IEEE Interntional Conference on Information Visualization.

[Lgm04] Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. *Learning Sub-structures of Document Semantic Graphs for Document Summarization*. Workshop on Link Analysis and Group Detection (LinkKDD) at KDD 2004 (Seattle, USA, August 22 – 25, 2004).

[Ssk09] Sakuma ir Shigenobu Kobayashi. *Link Analysis for Private Weighted Graphs*. SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

[Xyz+] Gui-Rong Xue, Qiang Yang, Hua-Jun Zeng, Yong Yu, Zheng Chen. *Exploiting the Hierarchical Structure for Link Analysis.* SIGIR'05, August 15–19, 2005, Salvador, Brazil.

[Has81] H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 3rd edition, 1981.

[Cwm04] D. Cai, X. F. He, J. R. Wen ir W.Y. Ma. *Block-level Link Analysis*. The 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'2004), July 2004.

[Llc+] L. Laura, S. Leonardi, G. Caldarelli, ir P. D. L. Rios. *A Multi-Layer Model for the Web Graph*. In 2nd International Workshop on Web Dynamics, Honolulu, 2002

[Sdr09] Giuseppe Scanniello, Damiano Distante, Michele Risi. *An approach and an Eclipse-based environment for enhancing the navigation structure ofWeb sites*. Journal International Journal on Software Tools for Technology Transfer (STTT), Volume 11 Issue 6, November 2009

[Hsw09a] Vera Hollink, Maarten van Someren ir Bob J. Wielinga. *A semi-automatic usage-based method for improving hyperlink descriptions in menus*. Journal

International Journal of Human-Computer Studies, Volume 67 Issue 4, April, 2009

[Hsw09b] Vera Hollink, Maarten van Someren ir Bob J. Wielinga. *A semi-automatic usage-based method for improving hyperlink descriptions in menus*. Journal International Journal of Human-Computer Studies, 68(4), April, 2009

[Iem+] Ian H. Witten,Eibe Frank,Mark A. Hall Data Mining. *Practical Machine Learning Tools and Techniques* 539-605. 2011

http://wiki.python.org/moin/beautiful%20soup (žiūrėta 2011 05 15)

Priedas Nr.1

Platesnis www.vu.lt tinklapio medinės struktūros vizualizavimas