TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse Instituut
Informaatika eriala

Oliver Soop

Eesti sotsiaalmeedia RSS-voogude roomaja

Bakalaureusetöö (6 EAP)

Juhendaja: Peep Küngas
Kaasjuhendaja: Meelis Kull

Autor: …… "….." mai 2012

Juhendaja: ……. "….." mai 2012

Juhendaja: ……..... "….." mai 2012

Lubada kaitsmisele

Professor: ……..... "….." mai 2012

TARTU 2012

Sisukord

Sissejuhatus

Sotsiaalmeedia abil edastatakse maailmas suurel hulgal informatsiooni. Käesoleval ajal on informatsiooni omamisel ja selle sihipärasel kasutamisel suur väärtus. Sotsiaalmeedias edastatud informatsioon on killustunud ning sellest kasuliku teadmuse eraldamine keeruline ja samuti ka kulukas tegevus. Teadmuse kogumise teevad keeruliseks eelkõige erinevad viisid informatsiooni edastamiseks: blogid, mikroblogid, sotsiaalveebirakendused, uudisportaalid jne. Käesoleva bakalaureusetöö eesmärgiks on luua rakendus, mis võimaldaks koguda Eesti sotsiaalmeedias edastatavat informatsiooni. Selle eesmärgi saavutamiseks loodi töö raames roomajad, mis kasutavad informatsiooni kogumisel ära RSS formaadi eeliseid. Ülevaade antakse rakenduse algoritmidest ning ülesehitusest, samuti analüüsitakse ka rakenduse efektiivsust töö raames toimunud eksperimendi tulemuste põhjal.

Sotsiaalmeedia võimaldab erinevaid tehnoloogiaid kasutades informatsioon edastada nii üksikisiku kui ka kogukondade ja ettevõtete poolt ning RSS(*Really Simple Syndication*) pakub võimalust edastada sisu struktureeritud formaadis. Mõlemad mõisted on detailsemalt kirjeldatud juba järgnevates peatükkides.

Sama valdkonna uurimusi on tehtud eelnevaltki, küll ei ole keskendutud Eesti sotsiaalmeedia kaardistamisele ja roomamisele. Nii on näiteks artiklis „Mapping the Blogosphere — Towards a Universal and Scalable Blog-Crawler“[1] eesmärgiks seatud universaalse blogosfääri roomaja loomine. Kõige suurem erinevus käesoleva töö raames valminud rakendusega tuleneb mahtudest, blogisid on artiklis viidatud andmete põhjal vähemalt 133 miljonit, Eesti sotsiaalmeedia kohta täpseid andmeid teadmata võib eeldada, et mahud on väiksemad. Sellest tulenevalt on universaalse blogiroomaja rakendamisel kasutatud efektiivsuse saavutamiseks heuristikuid – kogemusepõhine teadmus efektiivsuse optimeerimisest[2]. Võrreldes käesoleva töö raames valminud RSS-voogude leidja ja nende sisu roomajaga, kus on eelistatud põhjalikkust, heuristikuid pole rakenduse töös märkimisväärselt kasutatud. Kuid leidub ka sarnasusi kahe töö vahel – eelkõige on need seotud roomamise algoritmilise lahenduse põhimõtete sarnasusega. Mõlemas rakenduses kasutatakse RSS-voogude leidmisel samalaadset lahendust - HTML DOM (Dokumendi objekti mudel) struktuurist vastavate märgendite leidmist ja nende kõrvutamist otsitava RSS-voogudele eripäraste väärtustega. Eelnevalt nimetatud töö oli mõningal määral aluseks ka

RSS-voogude roomaja algoritmi efektiivsemaks tegemisel – bakalaureusetöö rakenduses implementeeriti üks töös kirjeldatud võimalus RSS-voogude eristamiseks.

Teine töö, mis kuulub osaliselt samasse valdkonda on „Searching the Blogosphere“[3], mille üheks osaks on samuti RSS-voogude roomaja, mis nagu eelnevas artiklis kirjeldatud roomaja, keskendub blogidele. Antud töös püütakse leida seost erinevate võtmesõnade vahel, mis kirjeldavad ühte sündmust või objekti. Analüüsi teostamiseks roomatakse blogisid iga 12 tunni järel ja roomatavate blogide nimekiri saadakse sisendist. Võrreldes antud tööd käesolevaga, on eelkõige tegemist kahe erineva eesmärgi saavutamisega – selle töö puhul efektiivse roomaja loomine ning artiklis kirjeldatu puhul võtmesõnade vaheliste relatsioonide analüüsimine. Samuti on artiklis kirjeldatud rakendusele lisatud rämpsblogide eemaldamine, millega antud töös pole arvestatud. Ühise joonena saab välja tuua mõlema töö puhul olulise komponendina efektiivselt RSS-voogudest sisu allalaadimise ning selle vastaval kujul andmebaasi salvestamise.

Töö koosneb neljast põhilisest osast: esmalt antakse ülevaade ning tutvustatakse töö eesmärgi saavutamisega seotud mõisteid – roomajad, RSS, sotsiaalmeedia; teises peatükis tutvustatakse tehnoloogia valikuid; kolmandas peatükis kirjeldatakse valminud rakenduse üht komponenti RSS-voogude roomajat – probleemid, mis lahendamist vajasid, funktsionaalsuse ja algoritmide kirjeldust; neljas peatükk käsitleb rakenduse teist komponenti RSS-voogude sisu roomajat – samuti probleemide ja funktsionaalsuse kirjeldust; neljandas peatüki aluseks on eksperiment ja analüüs, mis rakendusega läbi viidi.

Rakenduse lähtekood ja kasutusjuhend on kaasa pandud CD plaadil lisas 5, täpsemad failide kirjeldused asuvad samuti lisas 5.

1. RSS, sotsiaalmeedia ja roomajate ülevaade

1.1 RSS

RSS (inglise keeles *Rich Site Summary* või *Really Simple Syndication*) on vorming, milles edastatakse sagedasti uueneva veebi sisu. Enim rakendust on RSS leidnud online meedia ettevõtete juures ning samuti ka blogide sisu edastamisel.

RSS-i üldisem suunitlus seisneb veebis kuvatava sisu standardiseeritud ja struktureeritud lühikokkuvõtte edastamises infost huvitatud lugejatele. Väljendit lugeja võib RSS-ide puhul mõista kaheti – lugeja võib olla nii isik kui ka vastav tarkvara, mis RSS-voogusid kuvab. Kasutajal, kes eelnevalt pidi huvipakkuva informatsiooni leidmiseks külastama lugematu arv erinevaid veebisaite, on nüüd võimalus leida meelepärane info ühest rakendusest selgelt struktureeritud infokuvalt, kuhu on kasutaja poolt seadistatud huvipakkuvad RSS-vood. Taolist RSS-voogude kogumist ja kuvamist pakuvad ka paljud meilikliendid - näiteks Microsoft Outlook. Kuivõrd RSS-voos on metaandmete hulgas ka infoallikas, siis on võimalus tutvuda ka esialgse sisuga. Kõige levinumad RSS-voo edastajad on kõikvõimalikud blogid ja meediaportaalid, näiteks on Tartu Ülikooli blogil www.blog.ut.ee oma RSS-voog www.blog.ut.ee/feed/. Samuti kasutavad ka Eesti suuremad meediaportaalid oma uudisvoo edastamiseks RSS-voogusid, veebisaidil www.postimees.ee on olemas RSS-voog http://www.postimees.ee/rss/.

Enne RSS-i loomist eksisteeris mitmeid sama eesmärgiga formaate veebi sisu restruktureerimiseks ja standardiseeritud kujul kuvamiseks, näiteks Pointcast(beeta versioon aastal 1996)[4]. Samas ei leidnud need vormingud laia kasutust mitmete puudujääkide tõttu, kuid peamiselt põhjusel, et need olid kirjeldatud ühe info edastaja tarvis ega olnud lihtsasti laiendatavad. Järgmine tähelepanuväärsem samm veebi sisu standardiseeritud kuvamisel oli RDF (*Resource Description Framework*) loomine 1999. aastal, mis seisneb metaandmete andmemudeli kirjeldamises. Samal aastal lasti Netscape-i poolt välja *RDF Site Summary*, mis oli esimeseks RSS versiooniks ja kandis versiooni numbrit 0.9[5]. Selle versiooni loojateks olid Dan Libby ja Ramanathan V. Guha. Sama aasta juuli kuus järgnes veel versioon 0.91[6], milles olid RDF elemendid juba eemaldatud. 2001. aastal Netscape-i poolne RSS-i arendus ja toetus lõpetati, mis tähendas, et uuel väljatöötatud formaadil puudus omanik. Tekkis kaks osapoolt, kes töötasid edasi RSS-i arendamise nimel, üks neist oli RSS-DEV arendusgrupp, kuhu kuulus ka esimese versiooni looja Guha. Nemad lõid RSS versiooni 1.0[7], mis oli suur

samm edasi võrreldes eelnevatega ning lubas kasutada XML-i nimeruume - nimeruumide vaba kasutamine võimaldab defineerida märgendeid, mida algses RSS formaadis ei eksisteeri ning formaatida erinevat tüüpi infoobjekte. Teine osapool, kuhu kuulus Dave Winer avaldas 2002. aastal RSS versioon 2.0, mis on viimane loodud versioon ja leiab praegusel hetkel kõige enam kasutust erinevate RSS-voogude edastamisel[8].

RSS-voogude lihtne ülesehitus on tingitud märgendite selgest nimest ning asjaolust, et RSS märgendite arv ei ole väga suur, kuigi seda on võimalik laiendada lõpmatult erinevaid XML-i nimeruume kasutusele võttes. Järgnevalt on näitena toodud Tartu Ülikooli blogi (www.blog.ut.ee) RSS-voog 26. aprillil 2012, kus kasutatakse RSS-i versiooni 2.0.

```
1.  <rss  xmlns:content="http://purl.org/rss/1.0/modules/content/"
2.          ...
3.  xmlns:dc="http://purl.org/dc/elements/1.1/"
4.  xmlns:atom="http://www.w3.org/2005/Atom" version="2.0">
5.  <channel>
6.          <title>UT Blog</title>
7.          <atom:link href="http://blog.ut.ee/feed/" rel="self" type="application/rss+xml"/>
8.          <link>http://blog.ut.ee</link>
9.          <description>University of Tartu News, Views, Ways</description>
10.         <lastBuildDate>Thu, 26 Apr 2012 10:20:40 +0000</lastBuildDate>
11.         <language>en</language>
12.         <sy:updatePeriod>hourly</sy:updatePeriod>
13.         <sy:updateFrequency>1</sy:updateFrequency>
14.             ...
15.         <item>
16.                 <title>Tartu by You</title>
17.                 <link>http://blog.ut.ee/tartu-by-you/</link>
18.                 <comments>http://blog.ut.ee/tartu-by-you/#comments</comments>
19.                 <pubDate>Thu, 26 Apr 2012 07:48:26 +0000</pubDate>
20.                 <dc:creator>Inga Külmoja</dc:creator>
21.                 <category><![CDATA[ Tartu ]]></category>
22.                     ...
23.                 <guid isPermaLink="false">http://blog.ut.ee/?p=1881</guid>
24.                 <description>
25.                 <![CDATA[This is a collection of interesting content recently created either about or
26.                 in Tartu by nice people amongst and around us. <a href="http://blog.ut.ee/tartu-by-
27.                  you/">Continue reading <span class="meta-nav">&#8594;</span></a>
28.                    ...
29.                 </description>
30.                 <content:encoded>
31.                 <![CDATA[<p><em>This is a collection of interesting content recently created
32.                 about or in Tartu by nice people amongst and around us.</em></p> <p><span
33.                 id="more-1881"></span></p> <p><script
34.                 src="http://storify.com/TartuUniversity/tartu-by
35.                 you.js?header=false&#038;sharing=false&#038;border=false">
36.                 </script><noscript>[<a href="http://storify.com/TartuUniversity/tartu-by-you"
37.                 target="_blank">
38.                   ...
39.                 </content:encoded>
40.                 <wfw:commentRss>http://blog.ut.ee/tartu-by-you/feed/</wfw:commentRss>
41.                 <slash:comments>0</slash:comments>
```

```
42.          </item>
43.          …
```

Joonis 1. www.blog.ut.ee/feed RSS-voo näide(Osa voost asendatud punktidega ning lisatud ridade nummerdus)

Struktureeritud ja selge formaat võimaldab ka tavainimestel lugeda ja lihtsasti mõista RSS-voo sisu. Iga RSS-i puhul on esmatähtis RSS märgendi olemasolu, millel on atribuudiks versioon ning uuemate versioonide puhul ka XML nimeruumid, eelnevalt toodud näites on versiooniks 2.0 (Joonis 1, rida 4) ja kasutusel on ka laialt levinud nimeruum Dublin Core (Joonis 1, rida 3). Seejärel kirjeldatakse RSS-voo edastaja andmed: pealkiri (Joonis 1, rida 6, märgend <title>), lühitutvustus (Joonis 1, rida 9, märgend <description>), keel (Joonis 1, rida 11, märgend <language>) jne. Samuti võib olla kirjeldatud ka logod või ka muu info, mis on RSS formaadiga lubatud. Sellele järgneb RSS-voo sisuosa, kus iga objekt on eraldatud märgendiga *<item>*, joonis 1 esimene objekt algab real 11 ja lõppeb real 42. Vastavalt erinevatele versioonidele on objektil oma kohustuslikud elemendid. Näitena väljatoodud UT blogi RSS-voos (Joonis 1) on kasutusel ka teiste XML nimeruumide märgendid, nii on näiteks kirjeldatud märgendiga <sy:updatePeriod> ajaline määratlus, mille järel voo sisu uueneb. Bakalaureusetöö raames valminud RSS-voogude sisu roomaja eesmärgiks on osata töödelda kõiki märgendeid vajalikul kujul, et salvestada andmed korrektselt edasiseks töötluseks andmebaasi.

1.2 Roomajad

Roomaja (inglise keeles *crawler*) on programm, mille eesmärgiks on külastada internetis kuvatavaid veebilehti, veebirakendusi või teisi erinevaid teenuseid süstemaatiliselt ning eesmärgiga, et leitav info leiab mingil viisil rakendust. Kõige rohkem on roomajaid rakendanud info kogumiseks erinevad otsingumootorid, kelle eesmärgiks on omada kõige päevakohasemat infot.

Esimeseks roomajaks peetakse Matthew Gray poolt väljatöötatud *World Wide Web Wanderer*-i 1993. aastal ning eesmärgiks oli seatud veebi kasvu ja suuruse analüüsimine.

Roomajad jagunevad küll erinevatesse alamkategooriatesse, järgnevalt on kirjeldatud veebiroomaja üldisem töökäik[9]:

1. Sisendiks antakse URL(inglise keeles *Uniform Resource Locator*), mille sisu laetakse alla
2. Seejärel leitakse sisust kõik URL-id, mis lisatakse külastatavate URL-ide nimekirja, mida nimetatakse URL-ide frondiks (inglise keeles *frontier*)
3. Olenevalt eesmärgist võib roomaja ka sisu töödelda ning eraldada ja salvestada huvipakkuvat infot.
4. Peale URL-i edukat töötlemist jätkatakse URL-ide frondis olevate kirjete töötlemist.
5. Roomaja töö võib olla piiratud lõpetamistingimusega või jätkuda lõpmatult kuni sekkumiseni.

Roomajate puhul üks tähtsamaid komponente on niinimetatud URL-ide front, mis üldjuhul salvestatakse puhvermällu või mahukamate roomamiste puhul juba kõvakettale. Tüüpiliselt realiseeritakse front FIFO tüüpi magasini põhimõttel, kuid olenevalt roomaja spetsiifikast võib kasutusel olla ka mõni heuristik, mille alusel kirjeid rajast eemaldatakse või ümber järjestatakse. Teemaspetsiifiliste roomajate puhul on näiteks heuristikuna kasutusel järgmine meetod: näite URL-ide põhjal luuakse Bayes'i klassifikaatorid ning enne iga URL-i lisamist leitakse relevantsuse näitajad vastavalt eelpool kirjeldatud klassifikaatoritele ja seejärel URL lisatakse rajasse sobivale kohale. Teine oluline aspekt roomajate juures seisneb korduvate kirjete rajasse mittelisamises, selle ülesande jaoks kasutatakse erinevaid meetodeid: hoitakse eraldi külastatud URL-ide nimekirja või nende räsi.

Roomajate üldise käitumise kirjeldavad neli erinevat strateegiat[10]:

1. Roomatavate lehekülgede valimise strateegia – milliseid lehekülgi roomaja külastab
2. Taaskülastamise strateegia – ajaline määratlus, kas ja millal roomaja peaks külastatud veebilehekülgi uuesti alla laadima
3. Viisakusstrateegia – kuidas vältida külastatavate veebilehtede ülekoormust ja mitte sattuda musta nimekirja
4. Paralleelsusstrateegia – kuidas toimub paralleelne roomamine ja selle töö koordineerimine

1.3 Sotsiaalmeedia

Selle bakalaureusetöö vaatluse all on Eesti sotsiaalmeedia, seetõttu järgneb ülevaade ja detailsem kirjeldus sotsiaalmeedia olemusest. Sotsiaalmeediat kuvatakse mitmes erinevas vormis: ajakirjade ja ajalehtede online veebilehtedena, foorumitena, blogidena, vikidena jne. Üks tähelepanuväärsemaid fakte viimase 10 aasta jooksul seisneb asjaolus, et kui interneti algusperioodil olid peamised sisutootjad professionaalid(nt. ajakirjanikud), siis nüüdseks loovad suurema osa sisust lõppkasutajad. 2009. aastal oli Technorati.com andmetel aktiivseid blogisid 200 miljonit[11] ning 2010. aastal avaldas mikroblogijate lipulaev Twitter andmed, et neil on 75 miljonit kasutajat, kellest küll ainult 15 miljonit on aktiivsed[12].

Praeguseks ajaks võib eeldada, et vastavad numbrid on mõnevõrra kasvanud. Sotsiaalmeedia osakaal on tõusnud ning kindlasti jätkab oma tõusu lähiaastatel. See tähendab, et osa infot ja teadmust on muutunud rohkem laialivalgunuks ning sellise info käitlemine omajagu keerulisemaks.

Sotsiaalmeedia puhul nii nagu iga loodud väärtuse puhul on tähtis ka asjaolu, kes on sotsiaalmeedias loodud sisu omanik. On kaks osapoolt, keda tuleb arvestada - lõppkasutajad ja teenusepakkujad. Teenusepakkujad justkui pole sisu loonud, kuid nemad pakuvad võimalust sellist sisu edastada. Samuti on mõnede sotsiaalmeedia teenusepakkujate kasutamistingimustes lisatud punkt, kus märgitakse sisu omanikuks just teenusepakkuja.

2. Valminud rakenduste tehnoloogia valikud

Selleks, et lahendada eespool kirjeldatud probleemi, mis seisneb erinevate RSS-voogude leidmises, tuli teha valikud, millised tehnoloogiad ja vahendid oleksid kohased ning kõige optimaalsemad. Kuigi bakalaureusetöös kirjeldatakse kahe suurema probleemi lahendamist eraldi on tehnoloogiavalikud ja põhjendused kirjeldatud sõltumata probleemi olemusest ühe peatüki all. Mõlema lahenduse puhul olid kasutusel samad tehnoloogilised vahendid, kuigi iga probleemi lahendamisel oli vaatluse all erinev kontekst tehnoloogia kasutamisel.

Veebisaitide leidmiseks ja sellele järgnevaks töötlemiseks tuli otsustada, millisele programmeerimiskeelele peaks suurema osa rakendusest üles ehitama. Valikul tuli arvestada mitmeid asjaolusid: tegemist oleks kergekaalulise keelega, seetõttu et rakenduslik pool ei nõuaks väga suuri arendusressursse; võimalike tuleviku arenduste tarvis oleks tegemist laialt levinud keelega. Seetõttu osutus valituks PHP (inglise keeles Hypertext Preprocessor), mis ei olegi niivõrd programmeerimiskeel kui serveripoolne skriptide kirjutamiseks mõeldud keel[13]. Üks peamisi põhjusi, miks rakendus valmis PHP-s, seisnes sellest, et PHP on juba suur osa vajalikust funktsionaalsusest realiseeritud põhifunktsioonidena. Nii näiteks oli HTTP GET ja POST päringute esitamine põhifunktsioonide abil tehtud lihtsaks, samuti oli abi ka põhifunktsionaalsusest dokumendiobjektide mudeli (inglise keeles Document Object Model) koostamisel. Võrdluses teiste programmeerimiskeeltega ei ole võimalik konkreetseid eeliseid peale PHP põhifunktsioonide välja tuua, osaliselt oli valik kindlasti suunatud autori eelnevatest kogemustest antud programmeerimiskeele kasutamisega. Samas leidus üks puudujääk, nimelt ei toeta PHP algsel kujul mitmelõimelisust ning seega oli paralleelse töötluse toe loomine vähesel määral keerukam.

Teine valik, mis tuli teha oli seotud kogutud andmete salvestamisega ning siinkohal langes valik MySQL andmebaasi kasuks[14]. MySQL on kõige enam levinud vabavaraline relatsioonilise andmebaasi juhtimise tarkvara. Teise valikuna oleks olnud võimalik kasutada ka PostgreSQL andmebaasi tarkvara[15], mis on samuti vabavaraline, kuid otsus MySQL-i kasuks langetati eelkõige tulenevalt põhjusest, et andmete edasiseks töötlemiseks loodud rakendus oli kirjutatud arvestades MySQL-i ning komplikatsioonide vältimiseks kasutatakse seda ka roomaja rakenduses.

3. RSS voogude roomaja

3.1 Probleemid

Sotsiaalmeedias kajastatakse suurel hulgal teadmust ja informatsiooni, mis ei ole alati esitatud süstematiseeritult, see tähendab, et kaks allikat võivad rääkida ühel teemal ja pidada dialoogi, kuid konkreetne teineteisele viitav seos nende vahel puudub. Lisaks sellele pole sotsiaalmeedia allikate leidmine lihtne, esiteks teeb situatsiooni keeruliseks asjaolu, et neid on väga palju, näiteks mikroblogi teenusepakkujal Twitteril oli 2012. aasta veebruariks üle 500 miljoni registreeritud kasutaja ning see number jätkab aktiivset kasvu[16]. Nende 500 miljoni hulgas on vähemalt 100 miljonit aktiivset kasutajatat, kes sisenevad oma kasutajakontole korra kuus – vastav arv avaldati küll juba 2011. aasta septembris ning võib praeguseks hetkeks olla märkimisväärselt kasvanud[17]. Keerukaks kujunes ka RSS-voo keele identifitseerimine voos kirjeldatud andmete põhjal, kus ettenähtud märgendite kirjeldamisel enamus RSS-voogude sisu edastajaid ei pööranud keele märgendile palju tähelepanu. Seetõttu on keeruline piiritleda, kas mõni allikas kuulub vaadeldavasse konteksti või mitte. Samuti ei leidu vastava suunitlusega sotsiaalmeedia indekseerijat, kuhu oleks haaratud kogu Eesti sotsiaalmeedia kontekst täies mahus. Üks mahukam blogide nimekiri leidub blogs.station.ee-s, kuid kindlasti pole see täielik, sest lisamine sinna toimub lõppkasutaja algatusel.

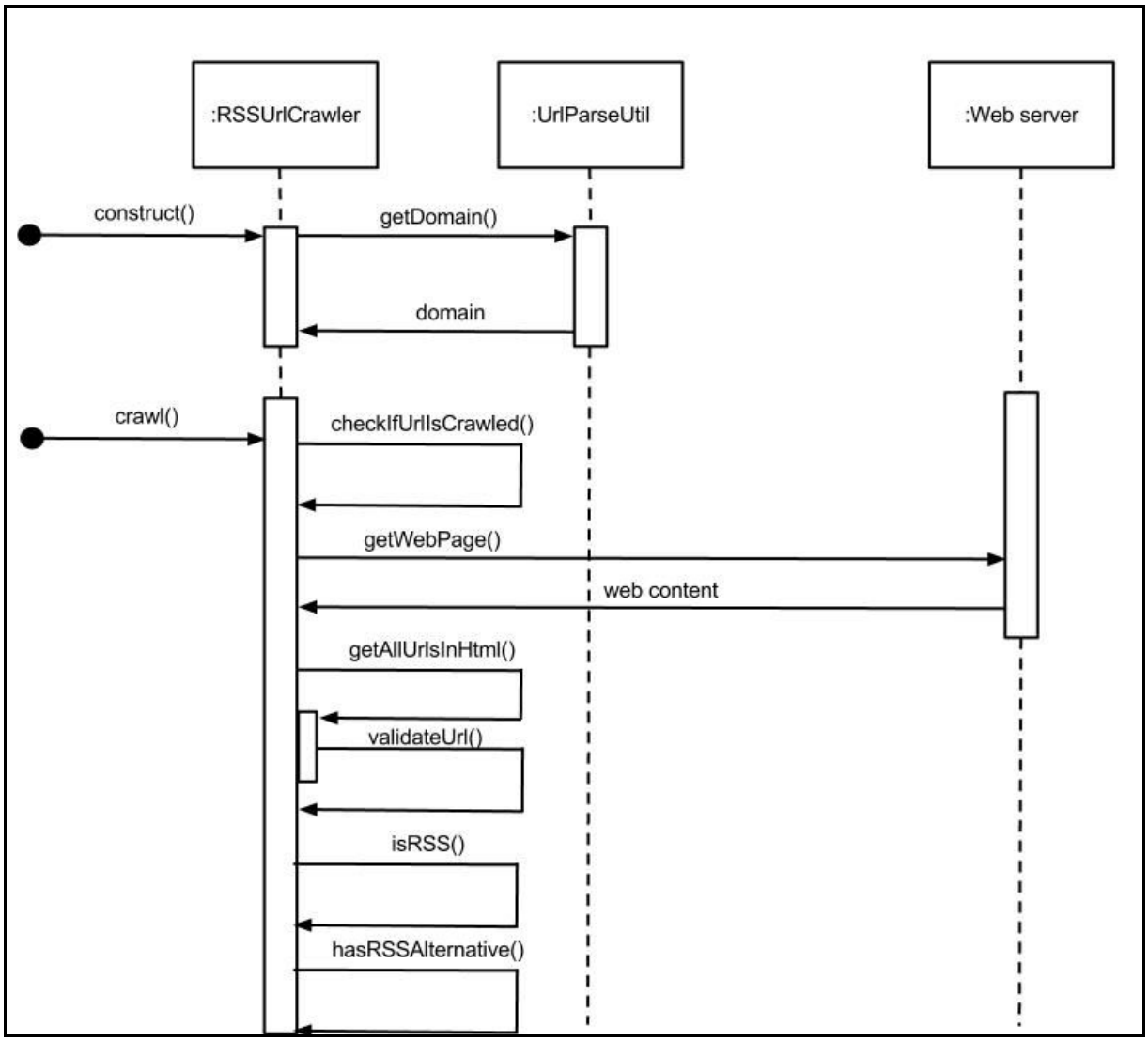
Erinevad sotsiaalmeedia teenusepakkujad võimaldavad kasutajatel edastada ja jagada oma sisu erinevates formaatides. See tähendab, et edastatav sisu on HTML (Hyper Text Markup Language) struktuuri poolest varieeruv erinevate allikate puhul[18]. Selleks, et huvipakkuv sisu oleks võimalik HTML koodis kirjeldatud metaandmete põhjal üles leida ning eraldada muust, oleks tarvis kirjeldada igale teenusepakkujale eraldi vastavad algoritmid ning käsitleda ka teenusepakkuja sisu kuvamise varieeruvusi, mis oleks liigselt ajakulukas. Seetõttu kasutatakse selle bakalaureusetöö raames loodud sotsiaalmeedia allikatena ainult RSS-voogusid. RSS formaat on selge struktuuriga ja võimaldab vähemate raskustega eraldada huvipakkuv informatsioon mittevajalikust. Kõik tuntumad sotsiaalmeedia teenusepakkujad võimaldavad kasutajatel sisu edastada RSS või Atom[19] formaadis ning vastavalt Technorati läbiviidud küsitlusele[20] edastab 74% kõikidest blogijatest kogu sisu RSS voogudes, vastav number firmade seas on küll väiksem - 55%, kuid siiski on RSS voogudes edastatava sisu maht märkimisväärne. Kui eelnevalt on kirjeldatud blogide sisu võimalikku roomamist, siis edastavad RSS-voogusid ka teised sotsiaalmeedia tüübid nagu foorumid ja wikid. Vastavalt RSS formaatidele läbi ajaloo on võimalik sisu endale sobivale kujule teisendada ning

salvestada edasiseks töötlemiseks. Kuna vaatluse all on ka Eestis leiduvad mittestandardsed veebisaidid ning need pole piiratud eespool nimetatud sotsiaalmeedia väljunditega, siis on neist RSS voogude leidmine mõnevõrra keerukam ja eeldab põhjalikumat veebisaitide töötlemist. Konkreetselt on uurimise all ja sisendiks selle bakalaureusetöö raames Eesti firmade veebisaidid ja nende RSS voogude leidmine.

Eesti Interneti Sihtasutuse andmetel oli 4. aprilli seisuga registreeritud 65529 Eesti domeeni (st. lõpuga ee)[21]. Kuigi sellega võiks piirata kogu huvipakkuva konteksti, siis on enamus sotsiaalmeediast edastatud välismaiste domeenidega URL-idel ning samuti leidub suur hulk Eesti veebisaite, kes kasutavad teisi domeene näiteks .eu ja .com. Nii on RSS-voogude ülesleidmiseks valmis seatud lai tööpõld ja nende avastamine on küllaltki ajakulukas.

3.2 Valminud rakendus

Käesolevas peatükis antakse ülevaade valminud RSS-voogude roomaja rakenduse ülesehitusest ning algoritmidest. Tegemist on ühe komponendiga kahest, mis antud bakalaureusetöö raames valminud rakendus sisaldab. Rakenduse tööd võib kujutleda graafi läbimisena, kus iga URL viitab ühele tipule ja iga tippu läbitakse üks kord. Roomaja ei kasuta URL-ide läbimisestrateegias heuristikuid, seega keskendutakse põhjalikkusele. Rakenduse arhitektuuri kirjeldamiseks on kasutatud 4 + 1 arhitektuurivaate mudelit[22]. Definitsiooni järgi on 4 + 1 arhitektuurivaate mudel kasutamiseks tarkvarapõhiste rakenduste kirjeldamiseks, mille raames kirjeldatakse rakenduse loogiline, arendus, protsessi ja füüsiline vaade ning kasutajalood.



Joonis 2. RSS-voogude leidja järgnevusskeem

Loogiline vaade - järgnevusskeem (inglise keeles sequence diagram)

Järgnevusskeem (Joonis 2) annab madala taseme ülevaate RSS-voogude leidja tööst – programmikoodi taseme ülevaade ei kätke endas kogu funktsionaalsust ning samuti kuvab skeem ainult roomaja põhitööd. Järgnevalt kirjeldatakse kõigi skeemidel väljatoodud funktsioonide eesmärgid ning tagastatavad väärtused, samuti ülevaade objektidest.

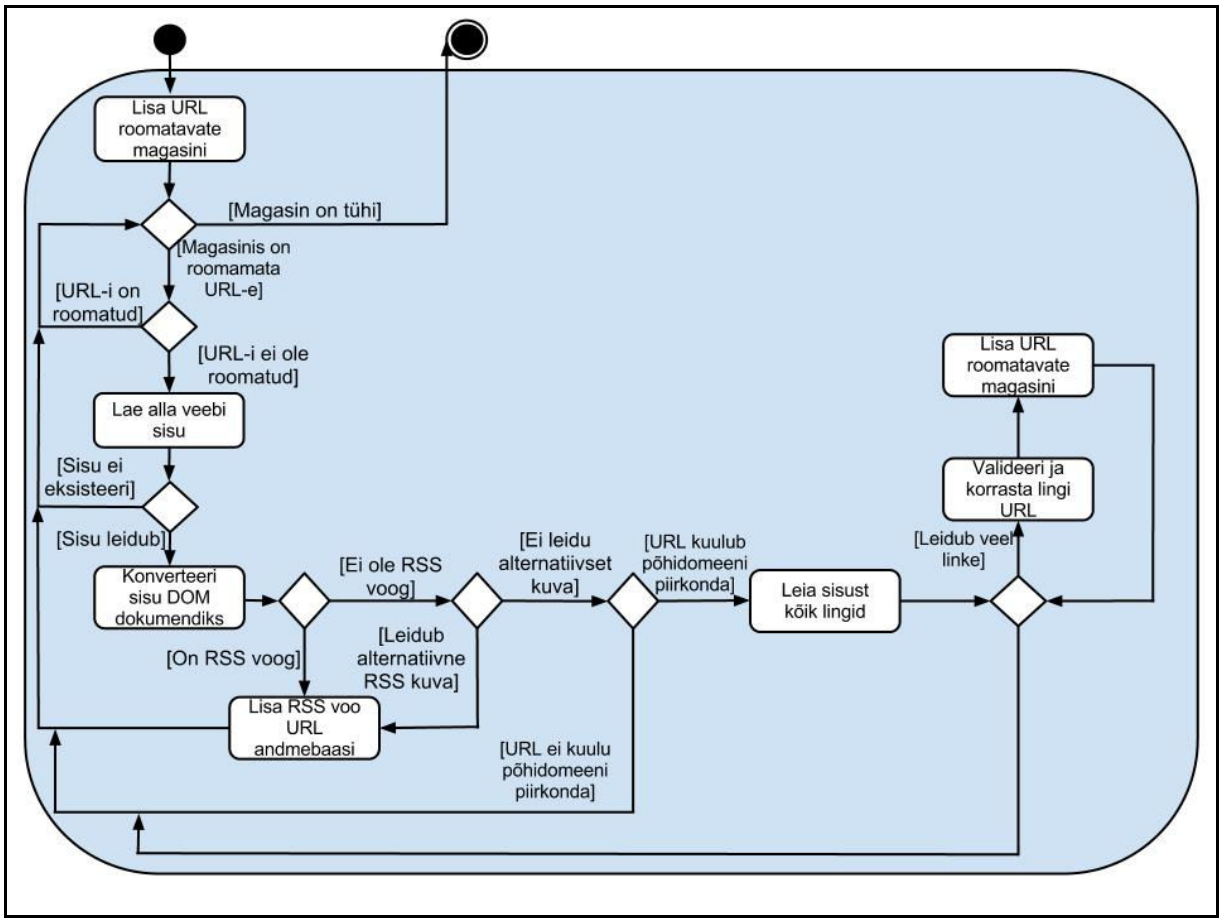
Objektid

1. RSSUrlCrawler – klass, kus toimub peamine töötlus RSS-voogude roomamisel
2. UrlParseUtil – klass, mis sisaldab URL-ide töötlemiseks staatilisi meetodeid nagu näiteks kahe URL-i põhidomeeni võrdlemiseks

3. Web server – antud skeemil kujutab igasugu veebiserverit, kuhu esitatakse HTTP päring ning tagastatakse veebilehe sisu

Meetodid ja tagastusväärtused

1. getDomain() – meetodile antakse parameetrina kaasa roomajale sisendiks olnud URL ning selle põhjal leitakse põhidomeen, mis salvestatakse ja edasisel töötlusel kontrollitakse, et järgnevad roomatavad URL-id asuksid domeeni piires. Tagastatakse domeen – domain.
2. checkIfUrlIsCrawled() – selle meetodiga kontrollitakse andmebaasist, kas sisendiks olevat URL-i on eelnevalt juba roomatud. Kui selline kirje eksisteerib andmebaasis, siis selle URL-i roomamist ei jätkata.
3. getWebPage() – kasutades PHP cURL moodulit laetakse URL-i viitava veebisaidi sisu alla, kui URL viitab mitteeksisteerivale lehele katkestatakse edasine roomamine sisendiks olnud URL-iga.
4. getAllUrlsInHtml() – seejärel leitakse allalaetud sisust kõik URL-id ning need omakorda valideeritakse meetodis validateURL(). See tähendab URL-id korrastatakse kui need on relatiivsed või on tegemist nii-öelda tühjade URL-idega, näiteks on URL-is on kasutatud trelle (#), mis puhul see edaspidisest töötlusest eemaldatakse.
5. isRSS() – tegemist on meetodiga, mis kontrollib, kas allalaetud sisu puhul on tegemist RSS-iga - kontrollitakse <rss> märgendi olemasolu.
6. hasRSSAlternative() – igal veebilehel on võimalik kirjeldada metaandmetes alternatiivsed võimalused veebilehe kuvamiseks, antud meetod otsib taolise kirjelduse olemasolu. Täpsemalt on meetodi töö kirjeldatud järgnevas - protsessi vaates.



Joonis 3. RSS-voogude leidja tegevusskeem

Protsessivaade - tegevusskeem (inglise keeles activity diagram)

Loogilise vaate (Joonis 2) kirjeldamisel anti rakenduse kirjeldus kooditasemel ning seetõttu ei haaranud kogu rakenduse käitumist. Tegevusskeemiga (Joonis 3) antakse koos järgneva diagrammi seletusega detailsem ülevaade RSS-voogude leidja tööst. Samuti kirjeldatakse rakenduse funktsionaalsust eelnevaltki näiteks olnud www.blog.ut.ee varal.

Tegemist on rakenduse põhitöö skeemiga ja sellele eelnevad eeltegevused ja järeltegevused.

Eeltegevused

1. Roomaja konstrueerimisel antakse sisendina kaasa ainsa parameetrina URL, kus alustatakse roomamist. See URL lisatakse FIFO tüüpi roomatavate URL-ide magasini - fronti. FIFO magasin käitub järjekorra põhimõttel, objektid magasinis töödeldakse samas järjekorras, kuidas need sinna sisestati. Sisend URL-i põhjal leitakse põhidomeen – näiteks kui sisendiks on Tartu Ülikooli blogi kande URL http://www.blog.ut.ee/tartu-

student-fashion-through-time/, siis on leitud põhidomeeniks blog.ut.ee ja see on ka domeeni piiriks.

2. Tulenevalt PHP seadistusest võib tekkida vajadus määrata maksimaalne skripti käivitusperiood, mis vaikeväärtusena on 30 sekundit. Kuna tõenäoliselt on skripti tegelik käivitusperiood pikem võib selle seadistamine aidata hoiduda enneaegsest skripti töö lõpetamisest.
3. Kolmanda piiranguna on võimalik määrata maksimaalne RSS-voogude arv, mida ühe sisendiks olnud URL-i käivitamisel leitakse.
4. Veel on võimalik määrata maksimaalne ühe sisend URL-i töötlemise aeg. Selle piirangu seadmine on tingitud veebisaitidest, mis genereerivad pidevalt erinevaid räsisi sisaldavaid URL-e linkideks ning seega võib roomaja jääda tsüklisse.

Järgnevalt käivitatakse roomaja põhitöö, mille kohta on ka eelnev tegevusskeem.

Põhitöö

1. URL-ide töötlemine käib tsükliliselt ning tsüklist väljumiseks on kaks juhtu: URL-ide front saab tühjaks või kui eelnevalt seadistatud piirang maksimaalse arvu RSS-voogude leidmiseks on kätte jõudnud.
2. URL-ide frondist eemaldatakse järgmine URL ja kontrollitakse andmebaasist, kas seda URL-i on juba roomatud. Kui on roomatud jäetakse URL vahele ning siirdutakse järgmise URL-i töötlemise juurde.
3. Järgneb URL-i viidatud veebi sisu allalaadimine – kui roomatav URL ei viita eksisteerivale veebisaidile lõpetatakse selle URL-i edasine töötlemine.
4. Allalaetud veebi sisu konverteeritakse edaspidise töötluse lihtsustamiseks DOM dokumendiks.
5. Nüüd on võimalik kontrollida, kas tegemist on RSS vooga, selle kontrollimiseks otsitakse DOM dokumendipuust <rss> märgendit. Näitena toodud www.blog.ut.ee/feed/ (Joonis 1) puhul asub <rss> märgend real 1.
6. Kui veebisaidi sisus kuvati RSS-voogu, siis salvestatakse antud URL andmebaasi teise RSS-voogude sisu roomajale sisendiks.
7. Kui sisus ei leidunud <rss> märgendit ning seega ei edastanud veebisait ka RSS-voogu, kontrollitakse kas eksisteerib alternatiivne sisu kuvamise võimalus. Selleks leitakse DOM dokumendi puust kõik <link> märgendid ning kontrollitakse selle atribuudi *rel* väärtust, kui väärtuseks on *alternate*,

kontrollitakse ka atribuudi *type* väärtust ning kui selleks on *application/rss+xml*, leidub veebisaidil alternatiivne kuva RSS voo näol ja märgendi <link> atribuudiks olev *href* väärtus salvestatakse andmebaasi. Näiteks www.blog.ut.ee puhul – mis on esileheks antud veebisaidil, on välja toodud joonisel 4.

```
1.  <!DOCTYPE html>
2.  <html dir="ltr" lang="en-US">
3.  <head>
4.          <meta property="og:image" content="http://blog.ut.ee/wp-
5.          content/uploads/2012/02/share_logo.jpg" />
6.          <meta charset="UTF-8" />
7.          <title>UT Blog | University of Tartu News, Views, Ways</title>
8.          <link rel="profile" href="http://gmpg.org/xfn/11" />
9.          <link rel="stylesheet" type="text/css" media="all"
10.         href="http://blog.ut.ee/wp-content/themes/twentyten/style.css" />
11.         <link rel="pingback" href="http://blog.ut.ee/xmlrpc.php" />
12.         <link rel="alternate" type="application/rss+xml" title="UT Blog
13.         &raquo; Feed" href="http://blog.ut.ee/feed/" />
14.         <link rel="alternate" type="application/rss+xml" title="UT Blog
15.         &raquo; Comments Feed" href="http://blog.ut.ee/comments/feed/"
16.         />
17.         ...
```

Joonis 4. www.blog.ut.ee HTML koodi väljavõte

Joonisel 4, read 12-16 kirjeldavadki alternatiivset viisi sisu edastamiseks ning roomaja oma töö käigus need leiab ja salvestab vastavad URL-id andmebaasi. Antud juhul on kaks kuva esitatavad alternatiivselt RSS-i näol - uudised ja kommentaarid.

8. Järgmine kontroll, kus kontrollitakse, kas roomatav URL asub põhidomeeni piirides tagab selle, et roomaja ei jääks lõputult roomama. Põhidomeenide võrdluse saab kõige paremini esitada järgneva näitega, kus põhidomeeniks on www.blog.ut.ee ja roomaja on vaatluse alla võtnud domeeni http://blog.ut.ee/category/career/, siis antud domeen kuulub põhidomeeni piiridesse, aga näiteks domeen www.neti.ee ei kuulu enam põhidomeeni

piiridesse. Domeeni piirkonna kontrollimine on sisse seatud sellesse töötlemise faasi seetõttu, et eelnevalt oleks võimalik kontrollida, kas näiteks www.neti.ee puhul on tegemist RSS vooga ja siis alles edasine töötlus katkestada. Siinkohal on eesmärgiks RSS-voogude registrite roomamise võimaldamine, kus register asub ühes domeeni piirkonnas ja registris olevad RSS-voogude URL-id teises.

9. Kui roomatava URL-i puhul oli tegemist põhidomeeniga samasse piirkonda kuuluva domeeniga, siis järgnevalt otsitakse üles kõik <a> märgendid ning selle märgendi atribuudiks olev *href* väärtused.
10. Leitud väärtused on URL-id, mis valideeritakse ja korrastatakse. Valideerimine antud juhul tähendab seda, et iseendale viitavad URL-id, mida kasutatakse Javascripti tarvis ning URL-i väärtuseks on vaid trellid (#) eemaldatakse edasisest töötlemisest. Korrastamise puhul kõik relatiivsed URL-id muudetakse ümber täielikeks URL-ideks. Eelnevaltki olnud näite www.blog.ut.ee puhul:

    <a href="#content" title="Skip to content">

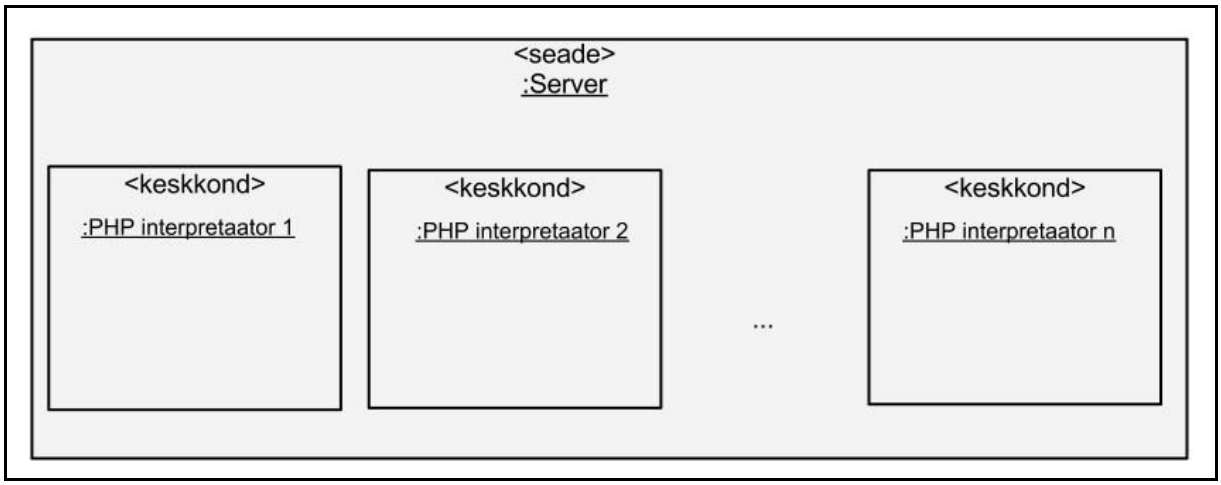
    Antud linki ei töödelda, kuna tegemist on Javascripti lingiga.

    <a href="/contact/">

    Siin korrastatakse URL nii, et tulemusena näeb see välja järgnev

    http://blog.ut.ee/contact/
11. Järgmise sammuna lisatakse kõik leitud ja valideeritud ning korrastatud URL-id URL-ide fronti, mida edaspidi on võimalik roomata.
12. Sellega saabki üks tsükli käik ja URL-i roomamise põhitöö ring läbi.

RSS-voogude leidja puhul on tegemist roomajaga, mis peaks garanteerima, et roomamine katab Eesti sotsiaalmeedia võrgustiku sobiva sisendi korral võimalikult suures ulatuses. Erinevalt paljudest teistest roomajatest pole käesoleva roomaja puhul kasutatud heuristikuid, mis suunaks roomaja tööd muutes URL-ide järjekorda frondis. Heuristiku puudumine küll suurendab ajalist kulu roomamise töö läbi viimisel. Arvestades asjaolu, et taoline roomamine viiakse iga domeeni piires läbi vaid üks kord, mitte pidevalt ja korduvalt, seetõttu on eelistatud põhjalikkust kiirusele.



Joonis 5. RSS-voogude leidja rakendusdiagramm

Füüsiline vaade – rakendusdiagramm (inglise keeles deployment diagram)
Füüsilise vaadet kirjeldav rakendusdiagramm (Joonis 5) ei sisalda palju keerukat infot, kuna rakenduse käivitamisel ei ole tarvis kasutada mitmeid rakendusservereid. Peamine eesmärk selle vaate kirjeldamisel on anda ülevaade, kuidas on lahendatud paralleeltöötlemise probleem. See on tingitud asjaolust, et PHP programmeerimiskeel ei toeta mitmelõimelisust.

Selleks, et oleks võimalik paralleelselt käivitada mitu roomajat suurema efektiivsuse tagamiseks, loodi rakendusele lisaks skript, mida on võimalik UNIX[23] keskkondades käivitada. Skripti käivitamisel on võimalik parameetrina kaasa anda käivitatavate PHP interpretaatorite arv. Seejärel käivitab vastav skript juba PHP päringu, mis tagastab andmebaasis leiduvate URL-ide arvu. Nii jagatakse võrdses osas andmebaasis leiduvad URL-id PHP interpretaatorite vahel ära. Nüüd on võimalik käivitada PHP skriptid, mis omakorda etteantud vahemikus URL-ide roomamise käivitavad.

4. RSS voogude sisu roomaja

4.1 Probleemid

RSS-voogude sisu roomaja eesmärgiks on RSS-voogudes kuvatava sisu allalaadimine, edasiseks töötluseks vastavasse formaati teisendamine ning andmebaasi salvestamine. Nii nagu eelnevas peatükis mainitud, edastavad paljud sotsiaalmeedia veebisaidid ning ka teised veebisaidid enda infovoogu RSS formaadis. Läbi ajaloo on loodud vähemasti 6 rohkem kasutatavat RSS versiooni nende hulgas on hetkel kõige enam kasutatavam versioon 2.0[24]. Alates versioonist 1.0 tekkis võimalus kasutada teisi XML-i nimeruume, mis tähendab, et RSS-voog ei pruugi alati sisaldada ainult RSS nimeruumi märgendeid. Taoline suur varieeruvus tekitab RSS-voogude sisu alla laadimisel ja edasisel töötlemisel omajagu raskusi. Bakalaureusetöö raames valminud rakenduse raames ei leitud antud probleemile üldisemat lahendit, mis oskaks iseseisvalt iga sisendi korral leida õige meetodi sisu töötlemiseks. Iga nimeruumi korral on tarvis kirjeldada kindla struktuuriga algoritm huvipakkuva märgendi töötlemiseks.

Näiteks on üks kasutatumaid XML nimeruume Dublin Core-i nimeruum ja paljud Eesti uudiste kui ka teised RSS voogude sisu edastajad, kellel on tähtis märkida iga sisu elemendi juurde selle autor, seda ka kasutavad. Joonisel 1 asub see real 20. Kuigi RSS-i versioonis 2.0 on kasutusel ka <author> märgend, kasutatakse ka eelnevat Dublin Core-i märgendit <dc:creator>. Seega on RSS justkui selgelt struktureeritud ja kindla formaadiga sisu edastamise vorming, samas on antud RSS-voogude sisu edastajatele suur vabadus formaadi laiendamiseks.

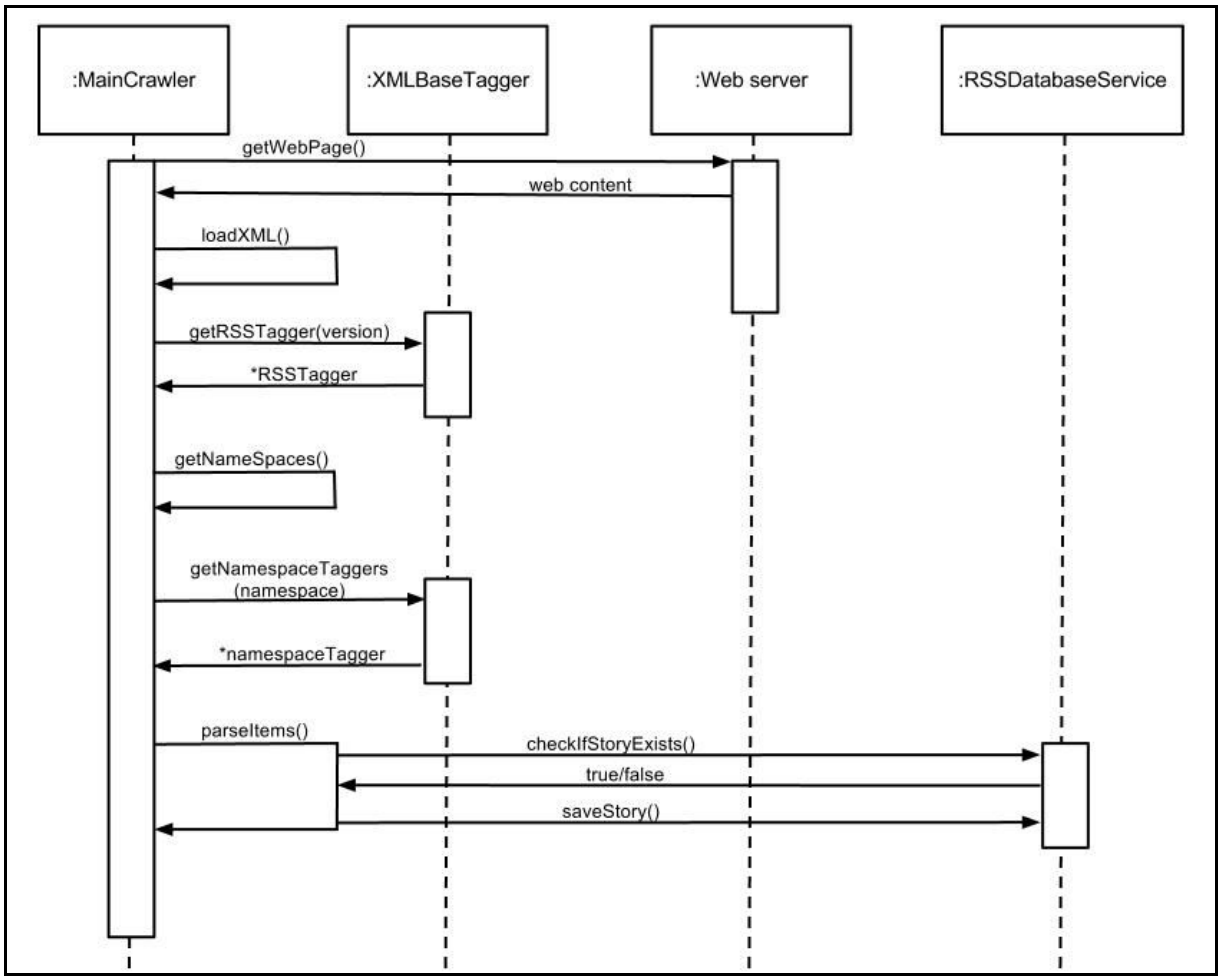
4.2 Valminud rakendus

RSS-voogude sisu roomaja töötleb kuvatavaid infovoogusid sobivale kujule, et need esmalt salvestada andmebaasi ning seejärel juba vastavaid infoobjekte edasi töödelda või analüüsida. Rakenduse töö põhineb suuresti DOM-i (dokumendi objekti mudel, inglise keeles Document Object Model) võimaluste rakendamises. Eelkõige kasutatakse selles PHP põhifunktsionaalsusi, mida on kasutatud vastavalt eesmärgile leida dokumendi sisupuust huvipakkuvad ja töödeldavad DOM objektid. Rakendus jaguneb üldises plaanis kolme erinevasse osasse:

1. RSS-voo sisu alla laadimine
2. RSS-voost uudisobjektide leidmine ja väljasõelumine

3. Uudisobjektide töötlemine ja andmebaasi salvestamine

Järgnevalt on RSS-voogude sisu roomaja arhitektuuri kohta ülevaade, mis põhineb 4 + 1 arhitektuurivaate mudelil[1]. 4 + 1 arhitektuurivaate mudelit on kirjeldatud eelnevas peatükis ning selle rakenduse juures on samuti kirjeldatud 3 vaadet: füüsiline, protsessi ja loogiline vaade.



Joonis 6. RSS-voogude sisu roomaja järgnevusskeem

Loogiline vaade - järgnevusskeem (inglise keeles sequence diagram)

Skeem (Joonis 6) annab rakenduse loogilise ülevaate – näidatud on roomaja üldisem töökäik vastavate meetodite nimedega nii nagu need on rakenduse PHP koodis implementeeritud. Kuigi iseseisvaid olemeid on rakenduses veelgi, siis põhivoo kirjeldamises ei ole neid kasutatud ning neid kirjeldatakse teistel skeemidel ja töövoo põhjalikumas ülevaates. Skeemil kuvatud objektid ning vastavad objektimeetod on kirjeldatud järgnevalt.

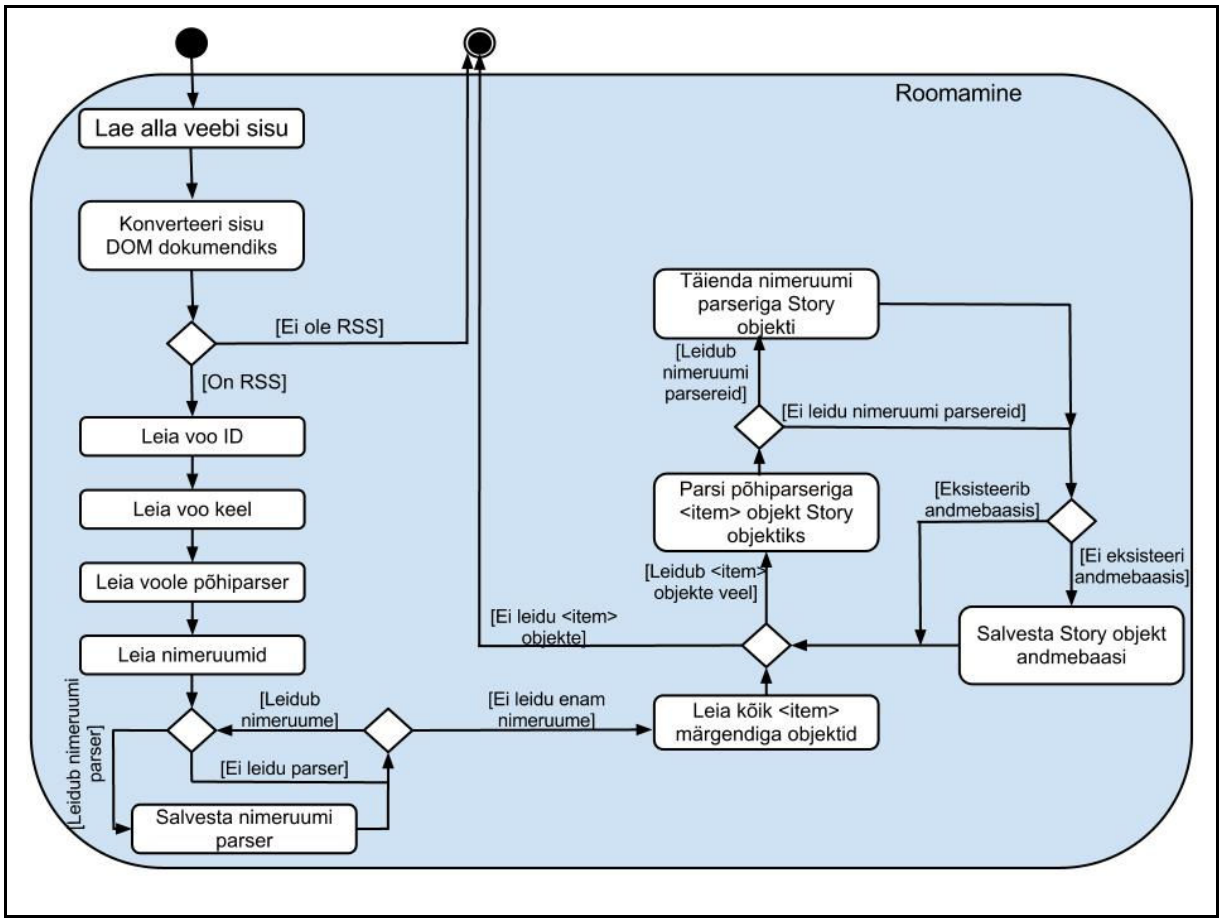
Objektid

1. MainCrawler – on RSS-voo sisu roomaja põhiklass, kus toimub suurem osa töövoost
2. XMLBaseTagger – tegemist on klassiga, mis on erinevate RSS versioonide ja nimeruumi parserite register
3. Web server – antud skeemil kujutab igasugu veebiserverit, kuhu esitatakse HTTP päring ning tagastatakse veebilehe sisu
4. RSSDatabaseService – tegemist on objektiga, mis kujutab endast vahekihti andmebaasiga suhtluses

Meetodid ja tagastusväärtused

1. getWebPage() – meetod, mille käigus veebiserverile esitatakse HTTP päring ning tagastatakse veebi sisu(antud kontekstis RSS-voo sisu) – web content
2. loadXML() – meetod, kus eelnevalt alla laetud veebi sisu laetakse DOM dokumenti edasiseks töötlemiseks
3. getRSSTagger(version) – meetod, täpsemalt päring XMLBaseTaggerile, kus sisendiks antakse RSS versioon ning registrist tagastatakse vastav RSSTagger, mida kasutatakse esmase ja peamise sisu töötlejana
4. getNameSpaces() – meetod, mis leiab RSS-voo pealkirjast kõik antud voos kasutatavad nimeruumid
5. getNameSpaceTaggers() – antud meetod esitab taaskord päringu XMLBasetaggerile ning annab parameetrina kaasa ühe eelnevalt leitud nimeruumidest, tulemusena tagastab XMLBaseTagger vastava nimeruumi Tagger objekti, kui see leidub registris, ning seda kasutatakse samuti sisu edaspidisel töötlemisel
6. parseNewsItems() – RSS-voogude sisus edastatakse objekte <item> märgendite vahel ning antud meetod tegeleb nende märgendite vahele jääva sisu parsimisega. Selleks kasutatakse eelnevalt leitud Tagger klasse, kus on vastav parsimisloogika kirjeldatud.
    a. checkIfStoryExists() – kontrollitakse, kas leitud objekt juba eksisteerib andmebaasis – tagastatakse tõene, kui eksisteerib ja vale kui ei eksisteeri

b. saveStory() – vastavalt eelnevale vastusele objekt, kas salvestatakse või mitte



Joonis 7. RSS-voogude sisu roomaja tegevusskeem

Protsessivaade - tegevusskeem (inglise keeles activity diagram)

Kui järgnevusskeem (Joonis 6) kirjeldas rakenduse tööd väga madalal tasemel – programmikoodi tasemel, siis tegevusskeem (Joonis 7) on veidike abstraktsem ning annab selgema ülevaate rakenduse käitumisest. Selleks, et anda spetsiifilisem ja parem ülevaade kogu RSS-voo sisu roomaja tegevusest järgneb iga tegevusskeemil kirjeldatud tegevuse seletus ning detailsem kirjeldus ning samuti ka erinevate otsustuspunktides aluseks olevate väärtuste kirjeldamine.

Tegemist on rakenduse põhitöö skeemiga ja sellele eelnevad eeltegevused ja järeltegevused.

Eeltegevused

1. Roomaja konstrueerimisel antakse ette kaks sisendit: RSS-voo URL ja kasutajapoolne RSS-voo ID
2. Võimalik on ümber muuta PHP-l vaikimisi seadistatud skripti käivitusperioodi pikkust. See võib vajalikuks osutuda väga mahukate RSS-voogude sisu roomamisel.

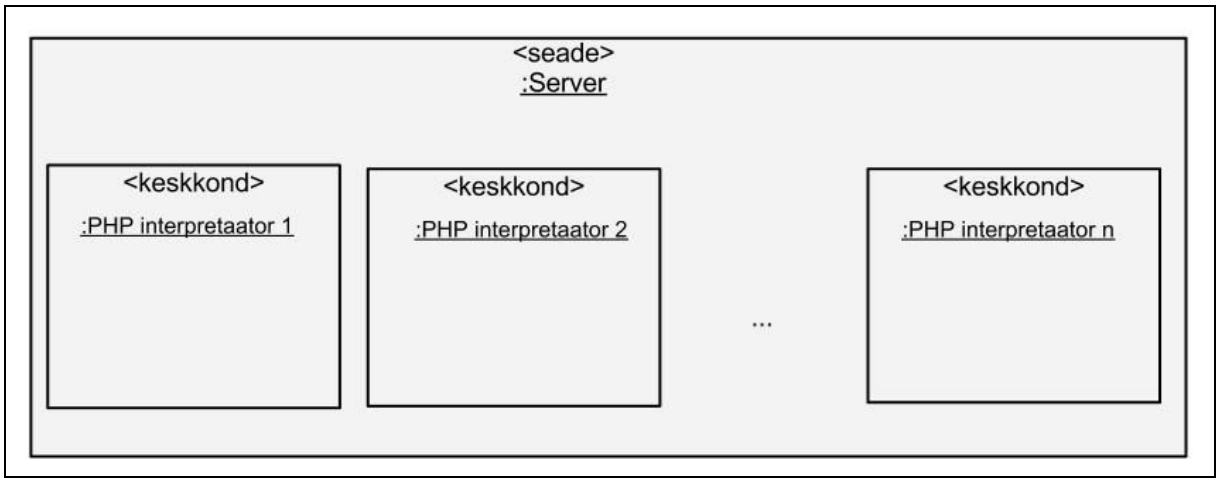
Edasi järgneb põhitöö käivitamine, mida on kirjeldatud ka eelneval tegevusskeemil (Joonis 7).

Põhitöö

1. Esmalt laetakse sisend URL-ilt alla veebi sisu, selleks kasutatakse PHP cURL moodulit.
2. Järgnevalt konverteeritakse allalaetud veebi sisu DOM dokumendiks, mis võimaldab edaspidi lihtsamat töötlemist.
3. Toimub kontroll veendumaks, et sisendiks antud URL viitab RSS-voole, selleks kontrollitakse, kas allalaetud sisus leidub <rss> märgend. Eespool näitena toodud www.blog.ut.ee/feed/ (Joonis 1) puhul asub märgend real 1. Kui antud märgend puudub pole tegemist RSS vooga ja edasine töötlus katkestatakse.
4. RSS-voo ID ehk tunnuse leidmiseks otsitakse märgendi <channel> piirkonda kuuluvat <title> märgendit. Samuti on www.blog.ut.ee/feed/ (Joonis 1) puhul asub see märgend real 6. Seega on blog.ut.ee/feed/ voo identifikaatoriks ehk pealkirjaks UT Blog.
5. Edasisel töötlemisel võib huvi pakkuda ka allalaetud RSS-voo sisu keeleline aspekt, seetõttu tuvastatakse roomamisel ka keel, mis on kanali infos kirjeldatud. Selleks leitakse <channel> märgendi piirkonda kuuluv <language> märgend. Näitena toodud blog.ut.ee/feed/ (Joonis 1) puhul asub see real 11. Kuigi tegelikult on blog.ut.ee/feed/ sisu eestikeelne, siis on seal kirjeldatud see väärtusega en, mis vastab inglise keelele.
6. Järgnevalt üritatakse leida põhiparseri klass, mis on edaspidi peamiseks aluseks sisu töötlemisel. Selleks leitakse, mis RSS versiooni formaati voo edastamisel kasutatakse, kui enamus versioonide puhul leidub <rss> märgendil versiooni kirjeldus, siis RSS versioon 1.0 eristamiseks on aluseks märgendi <rdf:RDF> olemasolu. Lõpptulemusena tagastatakse klass, mis vastutab edaspidise põhilise <item> märgendi piirkonda kuuluva sisu töötlemise eest.

www.blog.ut.ee/feed/ (Joonis 1) näites on kasutusel versioon 2.0, kirjeldatud real 4.

7. RSS-voo kuvamisel kasutatud nimeruumid on kirjeldatud samuti märgendi <rss> omadustena ning need eraldatakse edasiseks töötlemiseks. www.blog.ut.ee/feed/ puhul on kasutatud suur hulk erinevaid nimeruume, joonisel 1 on neist kuvatud 3 - rida 1, rida 3 ja rida 4.
8. Kui RSS voo sisu kuvamisel on kasutatud nimeruume leitakse, kas nende nimeruumidele vastavaid parsereid on rakenduses kirjeldatud või mitte. Hetkel on kirjeldatud kaks nimeruumi parserit: Yahoo Media nimeruumi parser ja Dublin Core nimeruumi parser. Seega võetaks praeguses rakenduses www.blog.ut.ee/feed/ sisu edasisel töötlemisel samuti kasutusele Dublin Core ja Yahoo Media parserid.
9. Järgneb sisu parsimine, esmalt leitakse kõik <item> märgendiga objektid, mis sisaldavadki kogu huvipakkuva informatsiooni.
10. Iga <item> objekt parsitakse eraldi. Parsimine antud kontekstis tähendab <item> objekti väärtuste vastavusele viimist andmebaasi objektiga Story. Esmalt kasutatakse parsimisel eelnevalt leitud põhiparserit ning seejärel rakendatakse põhiparseri poolt tagastatud Story objektil nimeruumi parsereid, mis täpsustavad või parandavad puuduvaid väärtusi. www.blog.ut.ee/feed/ näide joonisel 1, kus põhiparser eraldab järgnevad märgendid: <title>, <link>, <pubDate>,<description> ja <content> (vastavalt read: 16, 17, 19, 24-29 ja 30-39). Nimeruumi parser, mis leiab kasutust, on Dublin Core-i parser ja see eraldab ainult <dc:creator> märgendi (Joonis 1, rida 20). Ülejäänud märgendeid ignoreeritakse, tulevikus võib nende tarvis kirjeldada veel nimeruumi parsereid.
11. Peale <item> objekti töötlust ja Story objektiga vastavusse viimist kontrollitakse andmebaasis ega selline objekt juba andmebaasis ei eksisteeri. Otsustamise aluseks on <title> ehk pealkirja väärtus. Kui andmebaasis objekti ei eksisteeri, siis see salvestatakse.



Joonis 8. RSS-voogude sisu roomaja rakendusdiagramm

Füüsiline vaade – rakendusdiagramm (inglise keeles deployment diagram)

Ka RSS-voogude sisu roomaja puhul on kasutusel sama paralleeltöötlemise ülesehitus, mida kasutati RSS-voogude roomaja puhul – erinevusi implementatsioonis ei esine.

5. Eksperiment

Mõlema roomaja töö efektiivsuse mõõtmiseks ning saamaks informatsiooni ka Eesti sotsiaalmeedia aktiivsuse kohta rakendati valminud roomajaid lühikese perioodi jooksul. Töö tulemuste analüüs järgneb käesolevas peatükis. Eksperiment toimus lühikese ajavahemiku jooksul ning seetõttu ei ole võimalik Eesti sotsiaalmeedia kohta põhjalikke hinnanguid lähtuvalt RSS-voogudest võimalik anda. Kuigi leidub mõningaid indikeerivaid tulemusi sotsiaalmeedia kohta, ning millele on ka tähelepanu juhitud. Tulemusena on võimalik anda hinnang roomajate efektiivsusele ja samuti on väljatoodud ka kitsaskohad, mida võiks rakenduses edaspidi parandada.

Samuti tuleb juhtida tähelepanu ka ühele roomajate valdkonnas tähtsale strateegiale – viisakusstrateegia. Viisakusstrateegia peaks välistama olukorra, kus roomaja satub nii-öelda musta nimekirja – ehk tema tööd püütakse tõrjuda. Antud roomajate puhul oleneb viisakusstrateegia suuresti sellest, kuidas neid käivitatakse, millise perioodi tagant. RSS-voogude leidja puhul on viisakusstrateegia tähtsus väiksem, sest roomatakse ühte URL-i ainult üks kord. Sisu roomaja puhul, aga tuleb roomaja strateegiat hoolikalt valida. Kuigi sisu roomaja ei tekita märkimisväärset võrguliiklust, vaid laeb alla ainult huvipakkuva veebisaidi (RSS-voo) sisu.

5.1 Eksperimendi tingimused

Eksperiment viidi läbi ajavahemikul 26. aprill 2012, kell 22:00 kuni 02. mai 2012 kell 22:00. Kahe roomaja töö, RSS-voogude roomaja ja sisu roomaja, käivitati erinevatel aegadel. Esmalt käivitati RSS-voogude roomaja töö, antud tulemused läksid sisu roomaja sisendiks. Nii käivitati sisu roomaja 24 tundi hiljem ning mõlema roomaja töö katkestati 02. mail 2012. Antud perioodi jooksul toimusid ka mõnetunnised katkestused roomajate töös põhjustatuna kitsaskohtadest roomajate ehituses kui ka väliste mõjutajate tõttu.

Eksperimendi sisendiks oli nimekiri Eesti ettevõtete veebisaitide URL-idest, mis sisaldas 36177 URL-i. Nimekirjas leidus ka URL-e, mis enam ei ole aktiivsed, kuid nende osakaal on väike ning samuti ei mõjutanud nende töötlemine tulemusi, kuna ajaline kulu selliste leidmiseks ja kõrvaldamiseks roomaja poolt oli minimaalne.

5.2 RSS-voogude leidja ja sisu roomaja tulemused

RSS voogude roomaja juures salvestati ning analüüsiti järgnevaid andmeid:

1. Roomatud URL-ide arv
2. Roomatud domeenide arv
3. Leitud RSS-voogude arv
4. RSS-voogude sisu postituste arv nädalapäevade lõikes
5. RSS-voogude sisu postituste arv perioodil 1. jaanuar 2012 kuni 02. mai 2012
6. RSS-voogude sisu postituste arv nädalapäevade lõikes, samuti ka kellaaja lõikes

Siin kohal anname ülevaate üldisematest tulemustest ja arvudest, mis vaatluse all oleval perioodil roomajatega saavutati.

RSS-voogude roomaja

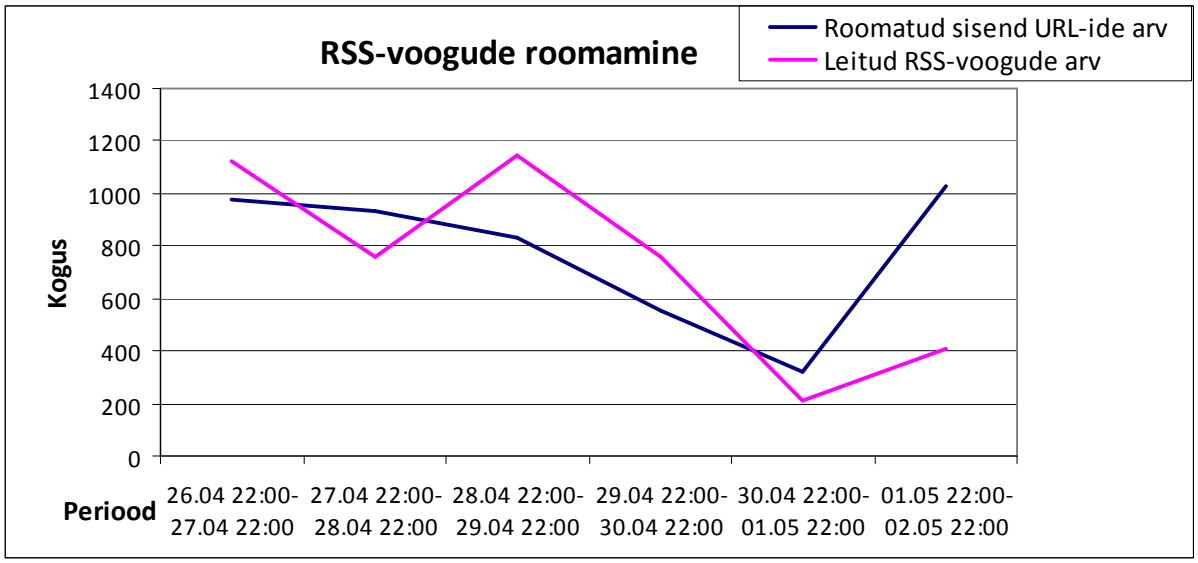
- 6 ööpäeva jooksul roomati sisendiks olnud 36177 URL-ist 4641 URLi, mis moodustab 12.83% sisendist
- 6 ööpäeva jooksul külastati 1016284 URL-i ning töödeldi vastavate URL-ide viitavate veebilehtede sisu
- 6 ööpäeva jooksul leiti 4402 RSS-voogu

RSS-voogude sisu roomaja

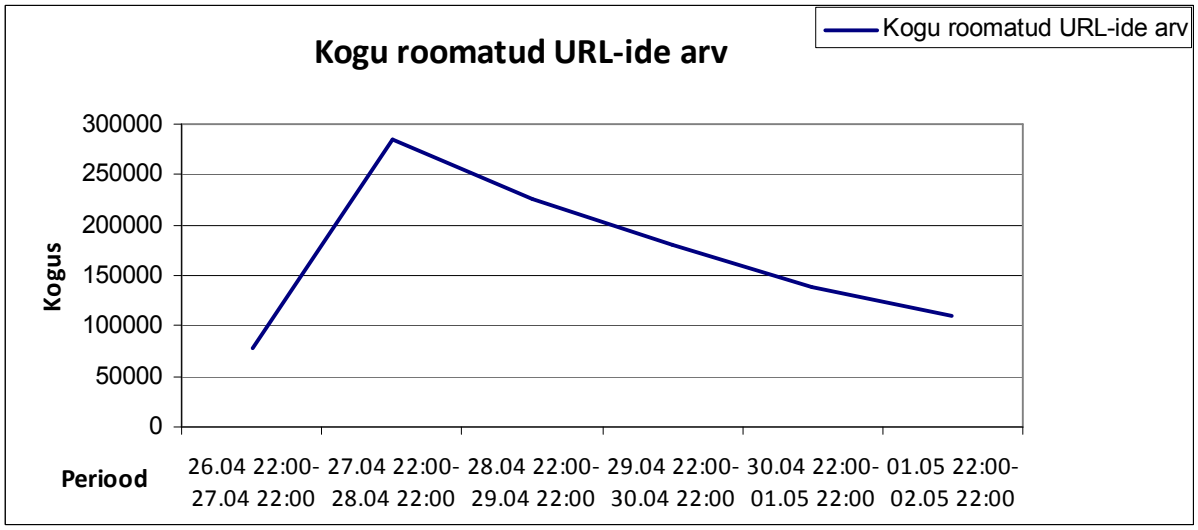
- 5 ööpäeva jooksul, alates 27. aprill 2012 kuni 02. mai 2012 leiti, töödeldi ja salvestati 38409 <item> märgendiga objekti.
- 38409-st on märgitud eesti keelseks 10.2% (3906), inglise keelseks 17% (6498) ja ülejäänud 72.8% (28005) pole võimalik keelt eristada või on tegemist mõne teise keelega, mis ei kuulu vaatluse alla – ei ole eesti ega inglise keel. Paraku pole see statistika täpne sisu kohalt, kuna paljude voogude juures kasutatakse inglise keelt justkui vaikimisi väärtusena, samas kui sisu on näiteks eesti keelne.

5.3 Tulemuste analüüs

Parem ülevaade roomamise efektiivsusest on võimalik anda diagrammide abil, käesolevas alapeatükis on väljatoodud statistika, mida võiks tõlgendada ja võtta aluseks hinnangu andmiseks roomajate efektiivsusele. Peatüki teises pooles analüüsitakse ja püütakse leida omapärasid seoses uudiste ja info publitseerimisega leitud voogudes. Diagrammidel kujutatud andmed on kaasa pandud lisades.



Joonis 9. RSS-voogude leidja roomamise statistika



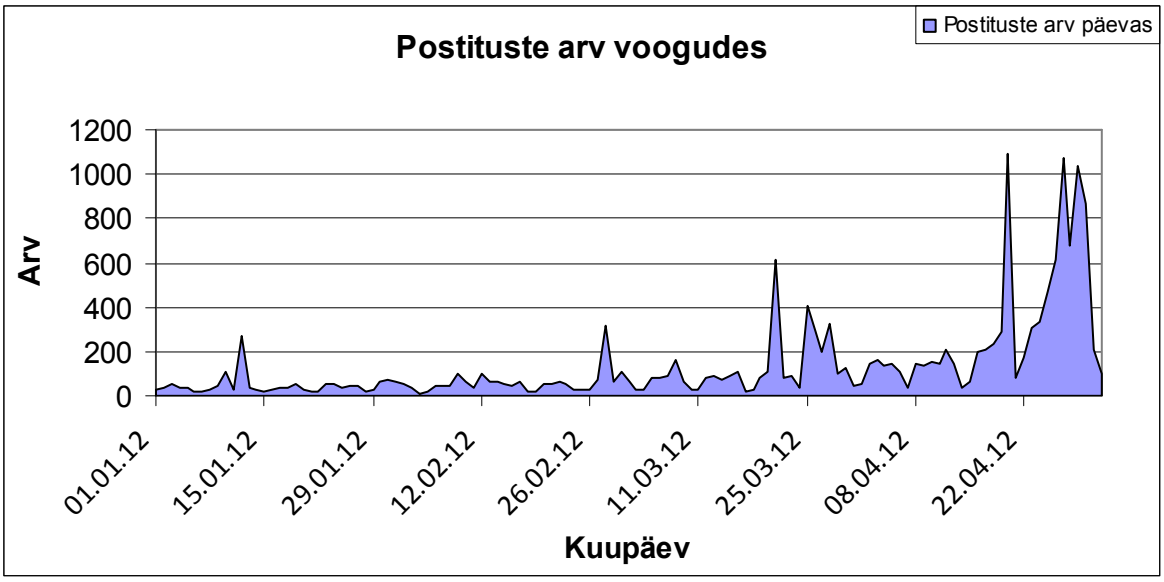
Joonis 10. RSS-voogude leidja kogu roomatud URL-ide arv

Diagrammid 1.1 ja 1.2 kirjeldavad RSS-voogude roomamise statistikat 24 tunni lõikes, konkreetsed arvud on välja toodud lisas 1. Esimesel diagrammil (Joonis 9) on näha mitu sisendiks olnud URL-i roomati ning võrdluseks leitud RSS-voogude arv. Teisel diagrammil (Joonis 10) on samuti võrdlemiseks toodud välja kogu URL-ide arv, mida roomati 24 tunni

jooksul.

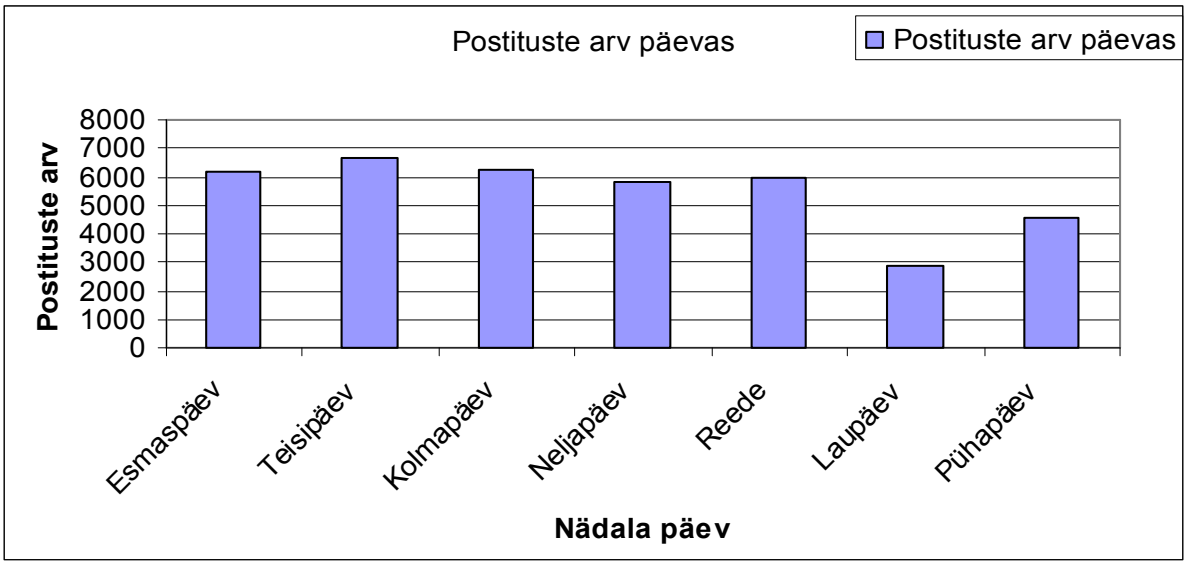
Esimeselt diagrammilt võib välja lugeda, et kõige edukamad päevad RSS-voogude leidmisel olid 1. ja 3. päev. Samas kui roomatud sisendiks olnud URL-ide ja leitud RSS-voogude vahel võib märgata skeemil mõningast korrelatsiooni – välja arvutatud korrelatsiooniväärtuseks on 0,49, mis kinnitab, et tegemist on huvipakkuva korrelatsiooniga. Samas muutus siinkohal teisiti kogu roomatud URL-ide arv. Kui esimesel päeval roomati 977 sisend URL-i ja leiti 1120 RSS-voogu, siis oli kogu roomatud URL-ide arv 76906. Võrdluseks teisel päeval roomati peaaegu, et sama palju sisend URL-e - 934. RSS-vooge leiti, aga juba märkimisväärselt vähem, 757 – küllaltki ootamatu oli siinkohal tervelt 285043 URL-i läbimine. Taolise korrelatsiooni puudumine või kõrvalekalde tekkimine võib olla tingitud mitmest asjaolust: esiteks polnud sisendiks olevad URL-id järjestatud kuidagi, võis leiduda palju mitteaktiivseid URL-e esimeses osas; teiseks võis põhjuseks olla sisendiks olnud URL-ide viitamine sisutihedatele (sisaldavad palju URL-e) veebilehtedele, kus ei leidunud ühtegi RSS-voogu.

Keskmiselt roomati päevas 773.5 sisendiks olnud URL-i, mis moodustab 2% kogu sisendist. Siinkohal toon välja paar asjaolu, mis selgusid eksperimendi käigus ja takistasid parema efektiivsuseni jõudmise. Eksperiment sooritati ühe andmebaasiga, seega polnud andmebaasiga ühenduse loomised ja päringud jagatud kahe andmebaasiga nii nagu rakendus, seda tegelikult võimaldab. Taolise muudatuse tegemine võimaldaks efektiivsust mingil määral kindlasti paremaks muuta. Teise asjaoluna selgusid mõned vead rakenduses, mis põhjustasid mõnel korral roomamise peatumise mõneks tunniks, põhjustatuna esmakordsest suure mahuga testimisest. Vigadeks olid näiteks andmebaaside ühendumise vead, samuti ka tsüklisse sattumine, mis oli tingitud URL-idest, mis sisaldavad räsisid. RSS-voogude roomaja seisukohalt on rakendusse viidud siis vastavad parandused ning usun, et võimekuse poole pealt võiks oodata efektiivsuse kahekordistumist – seda eelkõige kahe andmebaasi kasutamisele võtuga. Selle hüpoteesi kontrollimiseks toimus 6 tunnine eksperiment, mille tulemusena roomati 382 sisendiks olnud URL-i. Tegemist oli sama sisendiga, mis oli ka pikemalt kestnud eksperimendi aluseks. See viitab 100% efektiivsuse kasvule, samas hüpoteesi kinnitamiseks on vajalik kauem kestva eksperimendi läbiviimine.

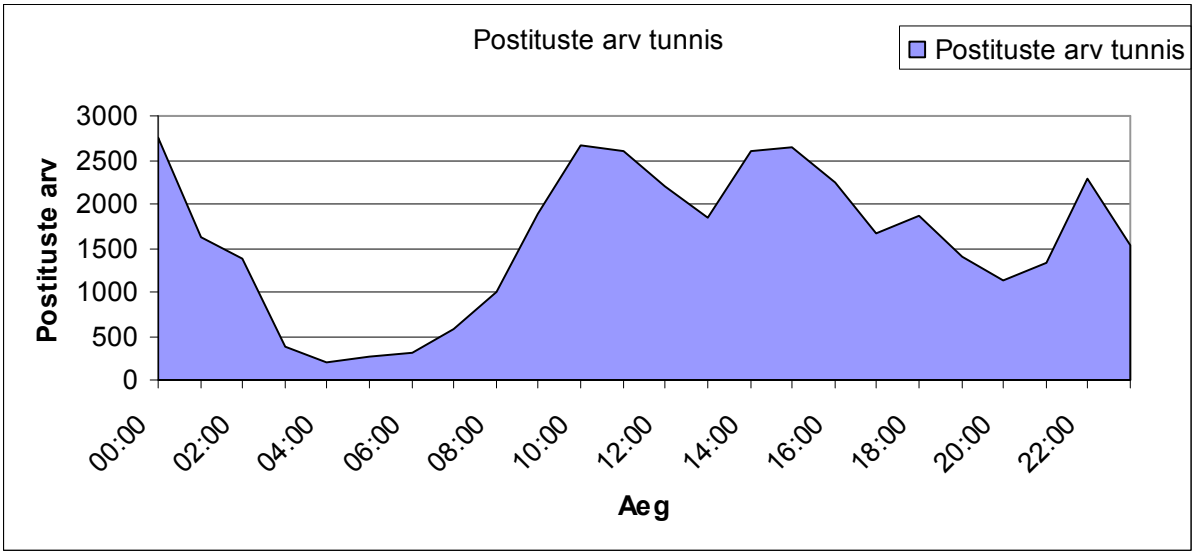


Joonis 11. Postituste arv RSS-voogudes 01.01.2012 – 02.05.2012 (Andmed lisas 2)

Jooniselt 11 on näha, et enamus RSS-vooge edastab ja salvestab ainult viimaseid postitusi, sellest selgub, et ka suurem osa töödeldud postitusi on edastatud viimase kuu aja jooksul. Samas ilmneb aktiivsuses ka kohatisi hüppeid, millel leidub erinevaid põhjuseid. Lähemalt uurides oli 19. märtsil toimunud tipnemine tingitud ühe firma saidilt avastatud Picasa pildialbumi kommentaaride RSS-voost, kus iga kommentaar oli üheks postituseks ning antud album avaldati firma saidil antud kuupäeval. Seega ei ole antud andmete põhjal võimalik tuua toimunud postituste arvu äkilise kasvu ja näiteks maailma sündmuste vahel paralleele. Kindlat mustrit omavad ka postituse arvude madalad päevad, mis toimuvad samuti 7 päevase vahedega. Sellele annab kinnitust ka järgmine diagramm 1.4, kus laupäevane postituste arv moodustab 47% keskmiselt tööpäevadel tehtud postitustest.
Antud diagrammi põhjal võib edasiseks RSS-voogude sisu roomamiseks luua strateegiaid – näiteks võib praeguste andmete põhjal eeldada, et postitused osades RSS-voogudes uuenevad umbkaudu nädala ajaga. See aeg on graafikul viimase postituste publitseerimise kasvamisperioodi algusest 24. aprillil kuni postituste arvu hõrenemiseni peale 1. maid.



Joonis 12. Kogu roomatud postituste arv päevade lõikes (Andmed lisa 3)



Joonis 13. Kogu roomatud postituste arv tundide lõikes (Andmed lisa 4)

Ootuspäraselt võib jooniselt 13 välja lugeda, et enamus postitusi tehakse perioodil kella 10:00 kuni 16:00. Samas on ka öistel aegadel äkilised postituste edastamise arvu kasvud – nii näiteks perioodil 00:00 kuni 00:59 edastatakse sama palju postitusi kui keskpäeval. Selle põhjuseks võib ühelt poolt olla asjaolu, et postituste edastamisel määratakse küll kuupäev, kuid aeg jäetakse määramata.

Kombineerides antud diagrammil esitatut nädalapäevade lõikes oleva statistikaga on võimalik edaspidiseks RSS-voogude sisu roomamiseks luua efektiivsem strateegia. Ei ole otstarbekas käivitada voogude sisu roomamist sama tihedalt perioodil 04:00 kuni 08:00, sel ajal oleks palju kasulikum käivitada RSS-voogude roomaja.

Kokkuvõte

Bakalaureusetöö käigus loodud kaks RSS-voogude roomajat, nii sisu roomaja kui ka voogude leidja, suudavad efektiivselt leida ja töödelda RSS-voogude sisu. Optimaalsema roomamise strateegia loomiseks töös eksperimendi käigus väljatoodud statistika põhjal, peab edasisel roomajate kasutamisel jälgima keskkonnast tulenevaid erisusi – see võib olla geograafilisest asukohast tulenev ajavöönd. Lõpptulemusena saavutatud küllaltki laiaulatuslikud roomamise rakendused võimaldavad kasutada seda edaspidi Eesti meedia indekseerimise ja analüüsimise projektis Tartu Ülikoolis.

Mõlema rakenduse puhul leidus keerulisi nüansse, millele selle töö käigus leiti lahendus, kuid efektiivsuse parandamiseks võib neid lahendusi ümber konstrueerida või veel arendada. RSS-voogude roomaja puhul oli kõige suuremaks väljakutseks tsüklilisest roomamisest hoidumine, mis oli eelkõige tingitud räsidega URL-ide kasutamisest. Sisu roomaja korral oli kõige keerukam ja ka huvitavam parima töötlemisviisi leidmine – erinevate nimeruumide jaoks töötlemisviisi kirjeldamine võiks olla kõige efektiivsem. Samas nõuab erinevate nimeruumide suur arv, et edaspidisel kasutamisel peaks igaühe jaoks soovi korral töötlemisviisid rakenduses juurde kirjeldama.

Roomajate efektiivsuse parendamiseks võiks tuleviku arendustena ette näha parema meetodi leidmise räsidega URL-ide töötlemisel. See tagaks tsüklitest varasema väljumise ja roomamise mahu suurenemise. Teine valdkond, kus võiks edasine arendus vajalikuks kujuneda seisneb RSS-voogude sisus kasutatava keele väljaselgitamises voo sisu tekstide põhjal. Nii nagu töös on välja toodud, siis RSS-voogude edastamisel ei pöörata erilist tähelepanu RSS-voogude formaadis kasutatavale kanali keele märgendi õigele väärtustamisele. Paljud eesti keelset sisu omavad vood kirjeldavad RSS-voo andmetes keelena inglise keelt ja ka vastupidi. Seega kui bakalaureusetöö eesmärk oli piiritleda töötlemist Eesti sotsiaalmeediaga, õnnestus selline piiritlemine ainult ühes faasis sisendi andmisel ning voogude roomamise faasis enam sisus kasutavale keelele tähelepanu ei pööratud. Sisupõhine keele tuvastamine võimaldaks ehk arendada ka efektiivsust, kuigi lisatöötluse kasutamine roomamisel võib mõjuda ka vastupidiselt.

Eksperimendi tulemuste põhjal saab luua edasise roomamise strateegiaid. Sarnaste eksperimentide kordamisel pikema perioodi jooksul võib tulevikus näiteks uurida postituste arvu, samuti ka sisu seost maailma sündmustega. Samuti võib huvi pakkuda erinevate sotsiaalmeedia kanalite vaheline sõltuvus, sellist sõltuvust on uuritud mikroblogi Twitter ja tavablogide vahel artiklis „Comparing Information Diffusion Structure in Weblogs and Microblogs“[25]. Näiteks võiks uurida ka erinevate sündmuste või info levimist Eesti sotsiaalmeedia kontekstis – selline teadmine aitaks arendada veebiturundust.

Crawler for Estonian Social Media RSS Feeds

Bachelor Thesis

Oliver Soop

Abstract

The aim of the thesis is to develop two RSS (RSS stands for Rich *Site Summary* and *Really Simple Syndication)* feed crawlers that would help to gather the information published in Estonian social media. The crawlers will be used by a work group of Tartu University for indexing and analysing Estonian media. The first crawler developed was a web crawler which crawled the web for RSS feeds given and input of URL's. The second crawler concentrated on processing the content of the RSS feeds and saving the content in desired format to database for further analysis. The thesis is made up of four chapters.

The first chapter of the thesis gives a background for the remainder of the work. Gives an overview of the nature of the social media and its connection to the work. A thorough insight is given of the evolution of the RSS format and its main concept on providing a format for publishing web content. Web crawlers are another topic covered in the first chapter.

Second chapter focuses on the first of the two crawlers built for this bachelor thesis – the RSS feed crawler. Finding RSS feeds from the internet given a certain amount of URL's as an input. The problems that arose and how they were dealt with – for example how to decide on one site being an RSS feed. The chapter also covers the application from the coding and algorithmic stand point.

Third chapter gives a view of the RSS feed content crawler and parser. This crawler usually crawls the feeds at certain intervals found by the previous crawler. There were also specific difficulties that had to be overcome – most complex was processing feeds that also used external namespaces. Low level aspect of the application is included as well.

The last chapter concludes the application overview with and experiment done using the crawlers with an input of Estonian companies' web sites. Statistics and analysis of the results of the experiment are given with reasoning. This chapter mainly focuses on the effectiveness of the first crawler but also implies on the current state of Estonian social media. Some idea of

how to use these results on future development and usage of the crawlers is given as well.

The bachelor thesis results indicate that the crawlers developed are broad-based and suitable for the upcoming crawling of social media for the Tartu University work group indexing and analysing Estonian media.

Viited

[1] Philipp Berger, Patrick Hennig, Justus Bross, Christoph Meinel; *Mapping the Blogosphere -Towards a Universal and Scalable Blog-Crawler*; Konverentsil *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom),* leheküljed 672-677, 9-11 okt. 2011; http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6113195

[2] Wikipedia, Heuristics, http://en.wikipedia.org/wiki/Heuristic_(computer_science), (viimati vaadatud 07.05.2012)

[3] Nilesh Bansal, Nick Koudas; *Searching the Blogosphere*; Konverentsil *Tenth International Workshop on the Web and Databases (WebDB 2007)* Beijing, China , oktoober 2007; http://leo.saclay.inria.fr/events/WebDB2007/Papers/p37.pdf

[4] Wikipedia, PointCast(dotcom), http://en.wikipedia.org/wiki/PointCast_(dotcom), (viimati vaadatud 08.05.2012)

[5] RSSBoard, http://www.rssboard.org/rss-0-9-0, (viimati vaadatud 08.05.2012)

[6] RSSBoard, http://www.rssboard.org/rss-0-9-1-netscape, (viimati vaadatud 08.05.2012)

[7] RDF Site Summary, http://web.resource.org/rss/1.0/spec, (viimati vaadatud 08.05.2012)

[8] RSS Usage worldwide by version, 2007, http://www.peachpit.com/articles/article.aspx?p=674690, (viimati vaadatud 08.05.2012)

[9] Wikipedia, Web crawler, http://en.wikipedia.org/wiki/Web_crawler, (viimati vaadatud 08.05.2012)

[10] Carlo Castillo *Effective Web Crawling* 2004, University of Chile, http://www.chato.cl/papers/crawling_thesis/effective_web_crawling.pdf

[11] State of blogosphere, http://technorati.com/social-media/article/state-of-the-blogosphere-introduction/, (viimati vaadatud 09.05.2012)

[12] 75M Twitter Users But Growth Slowing, http://www.twitterrati.com/2010/01/26/75m-twitter-users-but-growth-slowing/, (viimati vaadatud 09.05.2012)

[13] PHP, http://www.php.net/, (viimati vaadatud 09.05.2012)

[14] MySQL, http://www.mysql.com/

[15] PostgreSQL, http://www.postgresql.org/

[16] An estimate of active Twitter accounts per ultimo February 2012, http://www.twopblog.com/2012/03/estimate-of-active-twitter-accounts-per.html, (Viimati vaadatud 09.05.2012)

[17] One hundred million voices, http://blog.twitter.com/2011/09/one-hundred-million-voices.html, (viimati vaadatud 09.05.2012)
[18] Introduction to HTML, http://www.w3schools.com/html/html_intro.asp, (Viimati vaadatud 09.05.2012)
[19] Wikipedia, Atom, http://en.wikipedia.org/wiki/Atom_(standard), (Viimati vaadatud 09.05.2012)
[20] Technology, Traffic, Revenue; http://technorati.com/social-media/article/how-technology-traffic-and-revenue-day/, (Viimati vaadatud 09.05.2012)
[21] Eesti Interneti Sihtasutus, http://www.internet.ee/, (Viimati vaadatud 10.05.2012)
[22] Philippe Kruchten; *Architectural Blueprints—The "4+1" View Model of Software Architecture*; Avaldatud ajakirjas IEEE Software Volume 12, number 1, leheküljed 42-50 jaanuar 1995; http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf
[23] Wikipedia, Unix, http://en.wikipedia.org/wiki/Unix, (Viimati vaadatud 10.05.2012)
[24] Distribution of RSS Versions, http://www.syndic8.com/stats.php?Section=rss#tabtable, (Viimati vaadatud 10.05.2012)
[25] Jiang Yang, Scott Counts; *Comparing Information Diffusion Structure in Weblogs and Microblogs*; Konverentsil *4. ICWSM 2010: Washington, DC, USA*; 23-26. mai 2010 http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1467/1897

Lisad

Lisa 1. RSS-voogude leidja tulemused

Tabel 1. Roomatud URL-ide ja leitud voogude arv

	26.04 22:00-27.04 22:00	27.04 22:00-28.04 22:00	28.04 22:00-29.04 22:00	29.04 22:00-30.04 22:00	30.04 22:00-01.05 22:00	01.05 22:00-02.05 22:00
Roomatud sisend URL-ide arv	977	934	830	552	321	1027
Kogu roomatud URL-ide arv	76906	285043	225084	179852	139389	110010
Leitud RSS-voogude arv	1120	757	1146	758	210	411

Lisa 2. RSS-voogude postituste arv 01.01.2012 – 02.05.2012

Tabel 2. Postituste arv päevas

Kuupäev	Kogus	Kuupäev	Kogus	Kuupäev	Kogus	Kuupäev	Kogus
01.01.12	30	07.02.12	47	15.03.12	88	21.04.12	78
02.01.12	38	08.02.12	49	16.03.12	104	22.04.12	167
03.01.12	51	09.02.12	97	17.03.12	21	23.04.12	304
04.01.12	37	10.02.12	63	18.03.12	29	24.04.12	336
05.01.12	34	11.02.12	32	19.03.12	82	25.04.12	467
06.01.12	20	12.02.12	98	20.03.12	106	26.04.12	611
07.01.12	14	13.02.12	61	21.03.12	613	27.04.12	1075
08.01.12	25	14.02.12	63	22.03.12	84	28.04.12	677
09.01.12	48	15.02.12	53	23.03.12	87	29.04.12	1034
10.01.12	112	16.02.12	46	24.03.12	36	30.04.12	867
11.01.12	31	17.02.12	62	25.03.12	403	01.05.12	207
12.01.12	275	18.02.12	17	26.03.12	290	02.05.12	96
13.01.12	35	19.02.12	20	27.03.12	199		
14.01.12	28	20.02.12	54	28.03.12	326		
15.01.12	20	21.02.12	55	29.03.12	95		
16.01.12	29	22.02.12	65	30.03.12	126		
17.01.12	38	23.02.12	51	31.03.12	47		
18.01.12	40	24.02.12	26	01.04.12	58		
19.01.12	52	25.02.12	28	02.04.12	145		
20.01.12	28	26.02.12	27	03.04.12	166		
21.01.12	17	27.02.12	71	04.04.12	131		
22.01.12	22	28.02.12	316	05.04.12	146		
23.01.12	52	29.02.12	61	06.04.12	110		
24.01.12	57	01.03.12	104	07.04.12	40		
25.01.12	38	02.03.12	64	08.04.12	148		
26.01.12	47	03.03.12	25	09.04.12	134		
27.01.12	44	04.03.12	27	10.04.12	156		
28.01.12	14	05.03.12	80	11.04.12	140		
29.01.12	31	06.03.12	82	12.04.12	207		
30.01.12	61	07.03.12	93	13.04.12	148		
31.01.12	71	08.03.12	166	14.04.12	35		

01.02.12	66	09.03.12	61	15.04.12	66		
02.02.12	50	10.03.12	26	16.04.12	196		
03.02.12	37	11.03.12	25	17.04.12	206		
04.02.12	13	12.03.12	83	18.04.12	234		
05.02.12	18	13.03.12	88	19.04.12	288		
06.02.12	43	14.03.12	74	20.04.12	1088		

Lisa 3. Roomatud RSS-voogude postituste arv päevade lõikes

Tabel 3. Postitused päevade lõikes

Nädalapäev	Esmaspäev	Teisipäev	Kolmapäev	Neljapäev	Reede	Laupäev	Pühapäev
Postituste arv päevas	6191	6679	6257	5811	5997	2906	4568

Lisa 4. Roomatud RSS-voogude postituste arv tundide lõikes

Tabel 4. Postitused tundide lõikes

Tund	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00
Postituste arv	2764	1616	1381	384	203	260	305	582
Tund	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Postituste arv	998	1882	2659	2596	2204	1854	2599	2647
Tund	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Postituste arv	2241	1661	1863	1396	1138	1343	2296	1537

Lisa 5. CD rakendusega, mis sisaldab kahte roomajat

Tabel 5. Rakendusel asuvad failid ja nende kirjeldus

Fail	Faili kirjeldus
crawler/BaseCrawler.php	Roomajate põhiklass
crawler/MainCrawler.php	RSS-voogude sisu roomaja põhiklass
crawler/RSSUrlCrawler.php	RSS-voogude leidja põhiklass
database/crawler.sql	Baasimuudatuse skript, millega esialgset andmebaasi korrigeeritakse roomajate tarvis ning lisatakse uued tabelid
database/Database.php	Klass RSS-voogude sisu roomaja andmebaasiga ühenduse loomiseks
database/ekktt.sql	Esialgne andmebaasi koostamise skript (*Ei valminud töö raames, oli eelnevalt olemas sisendina)
database/RSSDatabaseService.php	RSS-voogude sisu roomaja teenuskiht andmebaasiga suhtlemiseks
database/URLDatabase.php	Klass RSS-voogude leidja andmebaasiga ühenduse loomiseks
database/URLDatabaseService.php	RSS-voogude leidja teenuskiht andmebaasiga suhtlemiseks
model/DCTagger.php	Dublin Core märgendite töötlemise klass

model/DownloadedXML.php	XML dokumendi klass
model/RSSBaseTagger.php	RSS-voo versioon 2.0 <item> objekti põhitöötluse klass
model/RSSV10Tagger.php	RSS-voo versioon 1.0 <item> objekti põhitöötluse klass
model/XMLBaseTagger.php	Põhitöötluse ja märgendite töötluse klasside vanemklass
model/YahooMediaTagger.php	Yahoo Media märgendite töötlemise klass
scripts/CrawlDatabaseUrls.php	Skript millega käivitada URL-ide roomamine, et leida RSS-vooge
scripts/CrawlRSSFeeds.php	Skript, millega käivitada RSS-voogude roomamine
scripts/getAmountOfRSSFeeds.php	Skript, millega pärida roomamata RSS-voogude arv andmebaasist
scripts/getAmountOfUrls.php	Skript, millega pärida roomamata sisend URL-ide arv
scripts/startRSSCrawling.sh	Shell skript käivitamaks mitme PHP interpretaatoriga CrawlRSSFeeds.php skripte
scripts/startUrlCrawling.sh	Shell skript käivitamaks mitme PHP interpretaatoriga CrawlDatabaseUrls.php skripte
ui/UserInterface.php	Lihtne kasutajaliides, mis võimaldab mõningast töö juhtimist
ui/UserInterfaceBackend.php	Kasutajaliidese kontroller ehk juhtklass
utils/CrawlDatabaseFeeds.php	Skript andmebaasis olevate RSS-voogude roomamise käivitamiseks
utils/CSVFileImporter.php	Klass, mis impordib etteantud csv failist kõik väärtused
utils/csvImport.php	Skript, mis käivitab CSVFileImporter.php töö etteantud faili asukohaga
utils/StoryEnum.php	Enum, kus on kirjeldatud kõik Story objekti väärtused
utils/UrlParseUtil.php	Utiliit URL-ide töötlemiseks
Story.php	Andmeobjek, milleks RSS-voo <item> objektid töödeldakse ja millisel kujul andmebaasi salvestatakse. (*Ei valminud töö raames, oli eelnevalt olemas sisendina)
kasutusjuhend.txt	Rakenduse kasutusjuhend