



Universiteti i Prishtinës “Hasan Prishtina”

FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

Departamenti i Kompjuterikes

---

# PUNIM DIPLOME

## Master

Rangimi semantik i ueb faqeve - Wikipedia  
si një rast studimi

**Mentori:**

**Prof. Asoc. Dr. Lule Ahmedi**

**Kandidati:**

**Bsc. Gëzim Sejdiu**

Korrik 2014, Prishtinë

## Abstrakti

Përdorimi i internetit në dekadën e fundit është rritur me një hov shumë të madh dhe ka gjetur pothuajse zbatim në çdo aspekt të jetës dhe të planifikimit ditor. Peshën më të lartë në këtë mega përdorim e kanë patjetër makinat kërkuese (siç janë Google, Bing, Yahoo, etj), përmes së cilave bëhet edhe përmbledhja e informative të nevojshme në internet.

Në mënyrë që shfrytëzuesi t'i ketë në dispozicion të gjitha informatat e nevojshëm, jo vetëm në aspektin sasior por edhe cilësor, është punuar që këto informata të jenë të klasifikuara apo të renditura sipas rëndësisë/peshës së tyre. Për këtë arsye është propozuar metoda e ashtuquajtur PageRank. Përmes kësaj metode bëhet edhe kalkulimi i peshës së faqeve apo informatave të cilat kërkohen përmes shfrytëzuesit të internetit.

Lëmia shkencore e cila merret me semantikën e uebit (ang. Semantic Web) merret pikërisht me semantikën e ueb faqeve të cilat listohen gjatë kërkimit të informatave. Përdorimi i semantikës në dhënien e peshës së faqeve, gjegjësisht kalkulimi semantik i PageRank-ut është një qasje tjetër e gjetjes së lidhshmërisë ndërmjet faqeve duke kalkuluar rangun përmes semantikës të shprehur në ontologji. Kohëve të fundit kombinimi i tij me rajtingun e autorëve është duke u konsideruar si peshë e madhe në impaktin e shfaqjeve të rezultateve.

Përmes këtij punimi është paraqitur në mënyrë shumë të detajuar rangimi semantik i faqeve, parametrat që marrin pjesë në kalkulimin e tij, si dhe mënyra e kalkulimit, konkretisht në një rast studimi, duke e marrë si shembull Wikipedia-n.

Pastaj janë përshkruar në detaje hapat të cilat duhet të merren, duke filluar nga mapimi i faqeve të dhëna nga libreria e Wikipedia-s në formatin standard të ontologjisë në ueb (owl), importimi i instancave, klasave dhe vetive tek ontologjia ekzistuese për rangim të ueb faqeve *SemanticSearchOnto*. Shtimi i attributeve të reja për vlerësimin e faqeve/artikujve, aplikimi i rregullave SWRL në kalkulimin e rangut semantik, e deri tek shfaqja e rezultateve duke përdorur ueb aplikacionin i cili është zhvilluar me qëllim shfaqjen dhe prezantimin e faqeve tek shfrytëzuesi.

Pastaj ne kemi shpjeguar konceptet bazë të uebit semantik, metodën dhe mënyrën e llogaritjes së rangimit semantik, modelimin dhe dizajnimin e ontologjisë, mënyrën e mapimit të të dhënave nga ato relacionare në atë RDF të ontologjive dhe në fund metodologjinë e evaluimit të rezultateve dhe prezantimit të tyre të aplikuar në Wikipedia përmes uebit semantik.

# Përmbajtja

|         |  |    |
|---------|--|----|
| 1       | Hyrje.....   | 8  |
| 1.1     | Motivimi.....  | 9  |
| 1.2     | Definimi i problemit.....  | 9  |
| 1.3     | Qëllimi i punimit.....   | 9  |
| 1.4     | Struktura e punimit të tezës.....  | 10 |
| 2       | Konceptet themelore .....  | 11 |
| 2.1     | Ueb semantik .....   | 11 |
| 2.1.1   | Hiperlinqet URI.....   | 12 |
| 2.1.2   | Gjuha markuese XML.....  | 13 |
| 2.1.3   | RDF .....  | 13 |
| 2.1.4   | RDF Skema .....  | 14 |
| 2.1.5   | Ontologjitë dhe OWL.....   | 16 |
| 2.1.6   | Pyetësorët SPARQL .....  | 17 |
| 2.1.7   | Rregullat SWRL .....   | 18 |
| 2.2     | Rangimi i faqeve .....   | 18 |
| 2.2.1   | Struktura e faqeve.....  | 18 |
| 2.2.2   | Algoritmi PageRank .....   | 19 |
| 2.2.2.1 | Definimi i PageRank-ut.....  | 19 |
| 2.2.2.2 | Faktori <i>Damping</i> .....   | 20 |
| 2.2.2.3 | Procedura Iterative e kalkulimit të PageRank-ut.....                                     | 21 |
| 3       | Puna paraprake.....  | 22 |
| 4       | Ontologjia dhe rregullat SWRL të rangimit – Wikipedia si një rast studimi.....           | 23 |
| 4.1     | Ontologjia <i>SemanticSearchOnto</i> .....   | 24 |
| 4.2     | Rregullat SWRL për kalkulimin e PageRank-ut semantik .....                               | 28 |
| 4.3     | Mapimi i të dhënave të Wikipedia në ontologji.....                                       | 31 |
| 4.3.1   | Wikipedia e cunguar (ang. <i>dump</i> ).....   | 31 |
| 4.3.2   | D2R Mapimi – migrimi i të dhënave relacionare .....                                      | 32 |
| 4.3.3   | Importimi i të dhënave RDF.....  | 36 |
| 5       | Implementimi i rangimit semantik - Wikipedia si një rast studimi .....                   | 39 |
| 5.1     | Kalkulimi i rangut semantik në mënyrë iterative.....                                     | 39 |
| 5.2     | Aplikacioni <i>SemanticSearchWikiApp</i> .....   | 43 |
| 5.2.1   | Pyetësorët SPARQL në aplikacionin <i>SemanticSearchWikiApp</i> .....                     | 43 |
| 5.2.2   | Ndërfaqja e <i>SemanticSearchWikiApp</i> .....   | 46 |
| 6       | Rezultatet dhe eksperimentet .....   | 50 |
| 6.1     | Evaluimi kundrejt një bashkësie eksperimentale .....                                     | 50 |
| 6.2     | Evaluimi i rezultateve të rangimit kundrejt shënimeve të “ <i>Wikipedia dump</i> ” ..... | 54 |
| 7       | Përfundimi dhe puna e ardhshme.....  | 57 |
| 8       | Referencat.....  | 58 |
| 9       | Shtojcat .....   | 60 |
| 9.1     | Veglat e dobishme .....  | 60 |
| 9.2     | Kodi burimor.....  | 62 |

|       |   |    |
|-------|---|----|
| 9.2.1 | Pjesë kodi nga kalkulimi iterativ i PageRank-ut ..... | 62 |
| 9.2.2 | Pjesë kodi nga ueb aplikacioni .....                  | 65 |

---

## Lista e figurave

|   |    |
|---|----|
| Figura 1 Ndërlidhja e faqeve.....   | 8  |
| Figura 2 Shtresat e Uebit Semantik.....   | 12 |
| Figura 3 Tresha RDF.....  | 13 |
| Figura 4 Paraqitja e tresheve për shembullin nga Listimi 2.....   | 14 |
| Figura 5 Definimi i klasës FOAF Person si pjese e vokabularit FOAF [28].....                                    | 15 |
| Figura 6 Ekzekutimi i SPARQL pyetësorit.....  | 17 |
| Figura 7 Lidhshmëria ndërmjet faqeve. ....  | 19 |
| Figura 8 Numri i linqeve për faqe.....  | 20 |
| Figura 9 Pamje nga ontologjia co-AuthorOnto [9] - relacioni ndërmjet foaf, portal dhe support ontologjive. .... | 22 |
| Figura 10 Paraqitja e përmbajtjes në Wikipedia.....   | 23 |
| Figura 11 Ontologjia SemanticSearchOnto – hierarkia e klasave.....  | 24 |
| Figura 12 Një pamje e hierarkisë së klasës “portal:Web-Site”.....   | 25 |
| Figura 13 Klasat e mapuara në ontologji. ....   | 25 |
| Figura 14 Disa nga vetitë e klasës portal:Web-Site .....  | 26 |
| Figura 15 Vetitë e klasës wiki:Web-SiteLinks.....   | 26 |
| Figura 16 Paraqitja e lidhshmërisë së vetisë hasOutLink.....  | 27 |
| Figura 17 Vetitë e klasës wiki:Categories.....  | 27 |
| Figura 18 Vetitë e klasës Constants.....  | 27 |
| Figura 19 Paraqitja e një instance të klasës Web-Site, dhe relacioni i saj. ....                                | 28 |
| Figura 20 SWRL rregullat për kalkulimin e rangut semantik. ....   | 29 |
| Figura 21 Database Diagrami – “Wikipedia dump” .....  | 31 |
| Figura 22 Procedura e mappimit të të dhënave relacionare në ontologji.....                                      | 32 |
| Figura 23 Prezantimi i OutLinks.....  | 36 |
| Figura 24 Lidhjet e jashtme për faqen specifike.....  | 36 |
| Figura 25 Konvertimi i llojit të vetisë përmes editorit Protege.....  | 37 |
| Figura 26 Importimi i ontologjisë SemanticWikiOnto tek SemanticSearchOnto. ....                                 | 38 |
| Figura 27 Pamje nga importimi i të dhënave.....   | 38 |
| Figura 28 Metoda iterative e kalkulimit të rangimit semantik.....   | 39 |
| Figura 29 Ndërfaqja e kalkulimit iterativ të rangut semantik [11]. ....   | 42 |
| Figura 30 Arkitektura e SemanticSearchWikiApp. ....   | 43 |
| Figura 31 Ekzekutimi dhe rezultatet nga SPARQL pyetësori. ....  | 44 |
| Figura 32 Procesi i ekzekutimit të pyetësorit SPARQL. ....  | 45 |
| Figura 33 Diagrami i sekuencës për procesin e kërkimit në SemanticSearchWikiApp. ....                           | 47 |
| Figura 34 Pamje nga faqja kryesore e aplikacionit SemanticSearchWikiApp.....                                    | 48 |
| Figura 35 Pamje nga katalogu i faqeve në SemanticSearchWikiApp.....   | 48 |
| Figura 36 Relacionet ndërmjet faqeve (rasti me tri instanca). ....  | 50 |
| Figura 37 PageRank evaluimi (shembulli me 3 instanca). ....   | 53 |
| Figura 38 Rangu semantik inicial i dhe ai pas 5 iteracioneve nga shembulli ne Figura 36. ...                    | 53 |
| Figura 39 Paraqitja grafike e 10 faqeve që prijnë sipas fjalës kyçe "d". ....                                   | 55 |
| Figura 40 D2R- Serveri.....   | 60 |

## Lista e tabelave

|   |    |
|---|----|
| Tabela 1 Raporti i lidhjeve të faqeve nga shembulli i dhënë në fig. 36..... | 51 |
| Tabela 2 Procedura e kalkulimit të PageRank-ut në mënyrë të detajuar.....   | 51 |
| Tabela 3 Formulatat për kalkulimin e PageRank-ut. ....                      | 52 |
| Tabela 4 Rezultati i detajuar i PageRank-ut për 5 iteracione. ....          | 52 |
| Tabela 5 Rezultati i përgjithësuar i PageRank-ut pas 5 iteracioneve .....   | 53 |

---

|  |    |
|--|----|
| Tabela 6 10 Web-Site në krye të gjeneruara nga SPARQL për fjalën kyçe "d"..... | 55 |
| Tabela 7 Krahasimi i rezultateve përmes PageRank <-> WikiRank. ....            | 56 |

## Lista e ekuacioneve

|            |    |
|------------|----|
| ( 1 )..... | 19 |
| ( 2 )..... | 20 |
| ( 3 )..... | 20 |
| ( 4 )..... | 21 |
| ( 5 )..... | 21 |
| ( 6 )..... | 21 |
| ( 7 )..... | 50 |
| ( 8 )..... | 53 |

---

për nusën time të dashur, Mimosën dhe  
për djalin tim Jonin

# 1 Hyrje

Në të gjitha aspektet e jetës ndikimi i uebit është evident dhe mjaft i madh, ku rolin më të madh në këtë konsum e zënë makinat kërkuese. Duke e parë kërkesën e madhe të aplikimit të makinave kërkuese është ardhur deri tek nevoja e specializimit të këtyre kërkueseve, në mundësimin e gjetjes së rezultateve më të përafërta.

Dimë që makinat kërkuese ruajnë një numër të madh të ueb faqeve të cilat në vetvete përmbajnë një numër të madh të linqeve. Në këtë mënyrë është paraqitur nevoja e një algoritmi i cili jep faqet më relevante të kërkua përmes pyetësorit.

PageRank është një algoritm i bazuar në analizën e linqeve, i propozuar dhe përdorur në makinën më të madhe kërkuese Google [1], që në mënyrë empirike përcakton një vlerë numerike për secilin element të dokumenteve të lidhura me linqe. Kjo lloj peshe numerike që i përcaktohet secilit element  $E$  quhet PageRank i  $E$  dhe shënohet me  $PR(E)$ .

Ky proces është i lidhur ngushtë me linqet të cilat faqet i ndajnë ndërmjet veti. Kur një faqe  $P1$  është e lidhur me një faqe tjetër  $P2$ , atëherë në mënyrë efektive faqja  $P1$  shkakton një votë për faqen  $P2$ . Sa më shumë vota që ka një faqe, aq më e rëndësishëm është ndikimi i saj në kërkim. Dhe gjithashtu faqja e cila jep një votë për faqen tjetër gjithashtu ka rëndësi edhe pesha e saj (pra kush ka dhënë votë) në përcaktimin e vlerës numerike të asaj vote.

Në Figura 1 është ilustruar procesi i ndërlidhjes së faqeve:

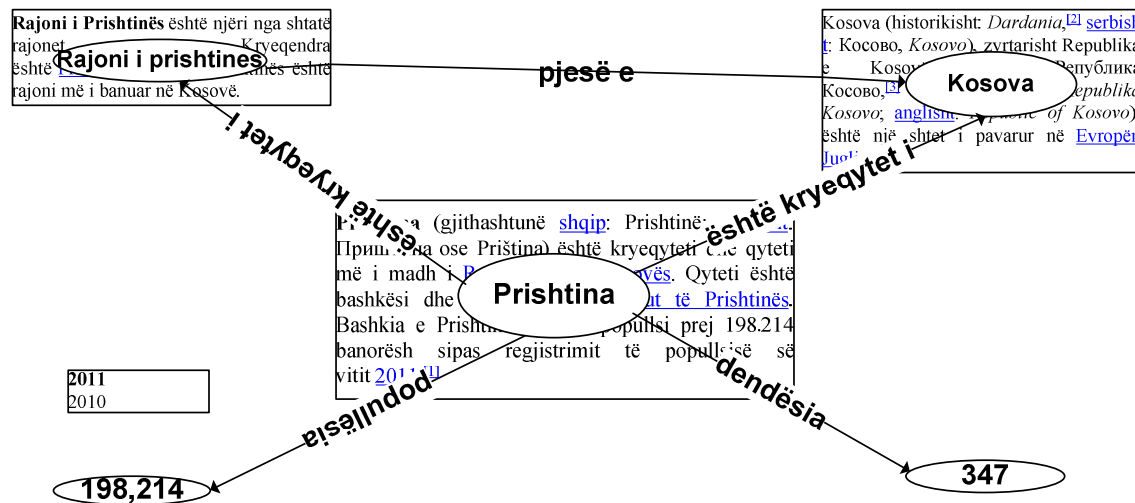


Figura 1 Ndërlidhja e faqeve.

Në mënyrë që të përcaktohet edhe kjo peshë për faqet e caktuara, makinat kërkuese përdorin algoritmin PageRank duke e vendosur në një kalkulator i cili edhe bën matjet e këtyre vlerave numerike.

Një kalkulim i tillë si parametra kryesor ka këto (gjatë kërkimit):

- Gjej të gjitha faqet që i përshtaten fjalës kyçe të kërkuar.
- Rangoj duke përdorur PageRank Algoritmin.
- Dhe shfaq këto rezultate.

Përpos aplikimit të tij në makinat kërkuese, PageRank në mënyrë të suksesshme është përdoret për të zgjidhur probleme shumë të rëndësishme siç është detektimi i spamave [25], gjetja e impaktit të citimeve [26], dhe shumë aplikime të tjera.



---

## 1.1 Motivimi

Dimë që në shumicën e faqeve tek të cilat ne qasemi ato përmbajnë një sasi të madhe të informatave siç mund të jenë informata mbi syllabusët e lëndëve, publikimet shkencore, kontakte personash, operatorë ekonomik apo shumë të tjera. Këto të dhëna në aspektin vizual duken se janë mirë të organizuara por për makinat të cilat i përpunojnë këto të dhëna ato nuk janë asgjë përpos tekst i thjeshtë. Për këtë arsye duhet që këto të dhëna të merren dhe të organizohen në mënyrë që të mund të mapohen në ontologjinë për përpunim semantik të mëtutjeshëm.

Sfida në konvertimin e bazave të shënimeve relacionare në ato semantike ka qenë njëri ndër motivimet e këtij punimi, duke mos e anashkaluar semantikën në rangimin e këtyre të dhënave, duke krijuar rregulla për kalkulim të rangut semantik si dhe shfaqjen/krahasimin e rangimit klasik me atë semantik.

## 1.2 Definimi i problemit

Implementimi i rregullave semantike në realizimin e rangimit të faqeve të Wikipedia-së ka qenë edhe ideja fillestare e kësaj teme, në mënyrë që paraqitja e faqeve të Wikipedia-së (bazës që ne e posedojmë) të bëhet përmes këtij rangimi. Mirëpo deri tek arritja e këtij qëllimi ka sjellë edhe çështje të tjera për të cilat jemi munduar që ti zgjidhim në mënyrë që të arrihet edhe qëllimi i këtij punimi.

Duke u nisur nga ky fakt pikat më të rëndësishme të cilat janë shtjellua në këtë punim janë:

- Sigurimi i bazës së shënimeve të Wikipedia-së,
- Adaptimi dhe mappimi i databasës relacionare në formatin RDF,
- Importimi dhe krijimi i ontologjisë nga formati i fituar RDF,
- Shtimi i vetive të reja në ontologji për kalkulimin e rangimit semantik,
- Krijimi i rregullave SWRL për kalkulimin e numrit të lidhjeve të jashtme të faqeve në ontologjinë e krijuar,
- Kalkulimi i rangut semantik përmes SQWRL rregullave të thirrura nga aplikacioni i cili do të krijohet duke u mbështetur në strukturën e kësaj ontologjie,
- Krijimin e një ueb aplikacioni për shfaqjen e rezultateve të cilët do të thirren duke i shfrytëzuar pyetësorët SPARQL.

Posedimi i faqeve të Wikipedia-së me një rangim të ri në mënyrë semantike, do të jetë i mundur përmes ontologjisë së krijuar dhe shfaqjes së tyre përmes aplikacionit të zhvilluar.

## 1.3 Qëllimi i punimit

Përmes këtij punimi të diplomës, ne synojmë që të nxjerrim rezultate në mënyrë semantike brenda të dhënave të artikujve të bazës së të dhënave të publikimeve të Wikipedia-së. Aplikimi i rangimit në publikimet në këtë databazë të kalkuluar përmes rregullave SWRL [10, 11] është edhe qëllimi kryesor gjatë këtij punimi. Zgjedhja e kësaj database (Wikipedia) është bërë për shkak se të dhënat në këtë datotekë ofrojnë një gamë të gjerë të artikujve/faqeve, dhe ne jemi munduar që të klasifikojmë aplikimin e semantikës në lëmin e ngushtë të uebit semantik.

Përmes rregullave të cilat do të aplikohen do të mundësohet edhe kalkulimi i rankut të faqeve, duke mundësuar rankim të faqeve sipas peshës së tyre. Rëndësia e peshës së kalkuluar ndikon direkt në paraqitjen e saj gjatë kërkimit (duke implementuar shfaqjen e rezultateve përmes një ueb aplikacioni), në këtë mënyrë edhe nxjerrja e top faqeve që kanë rankim më të lartë, në mënyrë që të shihet edhe pesha e faqeve.

---

## 1.4 Struktura e punimit të tezës

Punimi është i organizuar në atë mënyrë që të përfshijë të gjitha pjesët të cilat shtjellohen në punim rreth realizimit të rangimit semantik të faqeve.

Në seksionin e dytë do të jetë një hyrje mbi konceptet themelore të uebit semantik si dhe shpjegimi në mënyrë matematikore dhe kronologjike të disa definicioneve të kalkulimit klasik të rangut të faqeve.

Pastaj në seksionin e tretë është paraqitur puna e bërë deri më tani, ndërsa në seksionin e katërt është përshkruar në detaje mënyra e modelimit të ontologjisë, duke përfshi procesin e mapimit të ontologjisë, importimin e të dhënave, implementimin e semantikës në kalkulimin e rangut të faqeve të Wikipedia-së duke paraqitur SWRL rregullat për kalkulim të këtij rangui.

Në seksionin e pestë është paraqitur metoda e kalkulimit iterativ të rangut semantik të faqeve si dhe zhvillimin e një ueb aplikacioni për kërkim të të dhënave nga ontologjia përmes pyetësorëve SPARQL.

Seksioni gjashtë paraqet në mënyrë vizuale disa nga rezultatet e fituara dhe eksperimentet. Në fund, seksioni shtatë përmbledhë të gjithë temën.

---

## 2 Konceptet themelore

Në këtë seksion janë përfshirë bazat e Uebit Semantik [6] si hyrje, ku paraqitet si një shtesë mbi uebin normal duke e ngjyrosur me semantik në dokumentet/përmbajtjen të cilën e ka. Gjatë këtij kapitulli do të përshkruhen në detaje e gjithë struktura e uebit semantik, komponentët, mënyra se si gjenerohet semantika e të dhënave dhe se si aplikacionet ndërveprojnë me të. Pastaj gjatë këtij seksioni është paraqitur edhe kuptimi dhe formulimi matematik i PageRank-ut [1] klasik, duke përshkruar të gjithë akterët të cilët marrin pjesë në një kalkulim të tillë, në rastin e përgjithshëm.

### 2.1 Uebi semantik

Duke u nisur nga fakti se shumica e të dhënave në ditët e sodit janë të organizuara në atë mënyrë që duken mjaft mirë të lexuara nga pikëpamja e shfrytëzuesit por jo edhe të makinës. Nuk po e them këtë se kompjuteri nuk mundet ti lexoi këto përmbajtje (shembull mund të jenë të formatit HTML, linqe apo dokumente të cilat edhe i lexon), por makina nuk është në gjendje të kuptoi semantikën mbrapa tyre.

Në mënyrë që të kuptohet semantika duhet që të dihet se çka përmban një faqe. Një faqe nuk është asgjë tjetër veçse një bashkësi e dokumenteve, të ndërlidhur ndërmjet veti përmes linqeve. Ku këto dokumente përmbajnë informata të cilat shfrytëzuesit e këtyre faqeve i shohin dhe i lexojnë. Kjo do të thotë që makinat përpunuese ose kompjuterët nga gjithë ai volum i të dhënave nuk mund të përpunojnë apo të kuptojnë semantikën e tyre.

Në këtë nivel vjen në konsideratë intervenimi i ofruesit të informacioneve, të cilët mund ti ngjyrosin këto informacione me semantik, në këtë mënyrë kompjuterët mund të bëjnë më shumë me këto të dhëna, dhe të ofrojnë rezultate sipas nevojave të shfrytëzueseve.

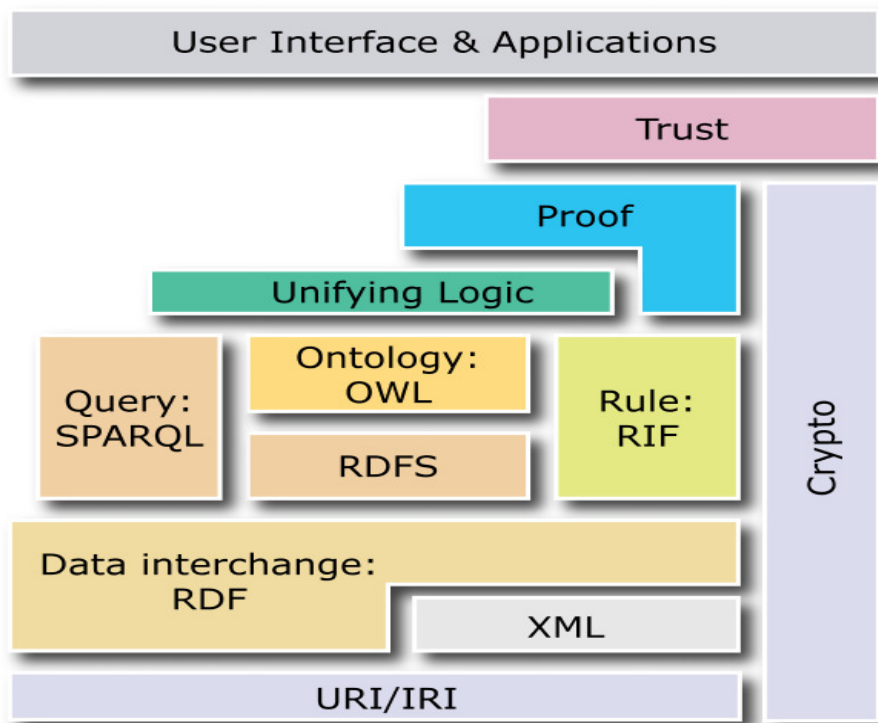
Lëmia shkencore e cila merret me semantikën e uebit (ang. Semantic Web) [6] paraqet një grup metodash apo teknologjisë të përdorura të cilat ia mundësojnë makinës që të kuptoi domethënien/semantikën e përmbajtjes apo informatave në ueb.

Mundësimi i kësaj semantike kundrejt makinës nuk paraqet domosdoshmërisht bërjen inteligjent të makinës, por përmbajtja e cila përpunohet nga makina është e paraqitur në atë mënyrë që të mundësohet implementimi i rregullave formale në atë përmbajtje. Marrim si shembull një pyetësor i cili bën gjetjen e pikave më të rëndësishme turistike të një lokacioni të cilat ofrojnë një mori vetish, siç mund të jenë internet, kafe, artefakte, etj. Në mënyrë klasike do të ishte shume e vështirë të gjinden këto pika, e më lere të përpunohen dhe të gjinden vetit përcjellëse të tyre në mënyrë që të përcaktohet edhe renditja e tyre.

Këto edhe janë problemet themelore që sjellin në pah aplikimin e uebit semantik. Një nga vetitë e uebit semantik është implikimi i të dhënave në ueb – një ueb i përbërë nga lidhshmëria ndërmjet dokumenteve [26].

Në mënyrë që ky nocion i ri i paraqitjes së të dhënave në ueb të aplikohet, është propozuar [7] një arkitekturë e kësaj teknologjie që edhe është paraqitur në Figura 2.

Por duhet cekur që uebi semantik nuk paraqitet si një ueb i ndarë [8] nga uebi i zakonshëm, por është një zgjerim në domethënien e uebit të zakonshëm.



**Figura 2 Shtresat e Uebit Semantik.**

Siç shihet në Figura 2, arkitektura e paraqitur përbëhet nga:

- RDF - Resource Description Framework – një shumëllojshmërie ndërveprimeve ndërmjet formateve, siç mund të jenë: RDF/XML,
- RDFS - RDF Schema,
- OWL – Web Ontology Language,
- Rregulla,
- Pyetësorë

Duke u mbështetur në skemën, shihet qartë se të gjitha këto komponentë përbëjnë një përshkrim formal të termave, koncepteve dhe lidhjeve ndërmjet tyre për ndonjë domen të njohur.

Për secilën shtresë më në detaje do ti përshkruajmë në seksionet në vijim.

### 2.1.1 Hiperlinqet URI

Uniform Resource Identifiers (URI) janë të definuara në [15] si bashkësi karakteresh të kompletuara të cilat përdoren për identifikimin e një resursi abstrakt ose fizik.

Këto lloj identifikues përdoren zakonisht për qasje në ndonjë ueb (<http://fiek.uni-pr.edu/>), por është shumë e rëndësishme që të kuptohet se roli kryesor i URI-së është që të identifikoi këto resurse, e jo të ndërveproi ndërmjet tyre. Duke u nisur nga ky fakt konstatojmë se çdo gjë e cila mund të identifikohet mund të shprehet përmes URI-ve, siç mund të jenë ueb faqet, librat, njerëzit apo edhe numrat. Por specifika kryesore e URI-ve është që të specifikoi saktë resurset e qasshme nga makina, siç mund të jetë ueb faqet, gjë të cilën e paraqet më saktë nën bashkësi të URI-së, URL (ang. Uniform Resource Locators).

Duke shfrytëzuar URL-të bëhet përshkrimi, apo përmban më shumë informata lidhur me objektin i cili po adresohet. Nga ky këndvështrim, një URL përmban që të dyja: identifikimin e objektit dhe ri-drejtimin tek përshkrimi i objektit.

Si shembull të URL do të konsideronim : [http://en.wikipedia.org/wiki/University\\_Prishtines](http://en.wikipedia.org/wiki/University_Prishtines), që na referon tek të dyja:

- “Universiteti i Prishtinës” si një institucion, dhe
- Artikullin apo faqen në Wikipedia.

### 2.1.2 Gjuha markuese XML

eXtensible Markup Language (XML) [16, 17] është një meta-gjuhë për definimin e tagjeve në një format standard, në mënyrë që përmes tyre të bëhet strukturimi i dokumenteve dhe të dhënave në ueb. Kontributi kryesor i XML-së është që ofron një sintaksë të përbashkët për të gjithë dokumentet në ueb.

| Listimi.1 | XML reprezentimi i faqës së Universitetit të Prishtinës në Wikipedia.     |
|-----------|---|
| 1         |   |
| 2         | <mediawiki xmlns="http://www.mediawiki.org/xml/export-0.8/"               |
| 3         | xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"                     |
| 4         | xsi:schemaLocation="http://www.mediawiki.org/xml/export-0.8/              |
| 5         | http://www.mediawiki.org/xml/export-0.8.xsd" version="0.8" xml:lang="en"> |
| 6         | <siteinfo>  |
| 7         | <sitename>Wikipedia</sitename>  |
| 8         | <base>http://en.wikipedia.org/wiki/Main_Page</base>                       |
| 9         | <generator>MediaWiki 1.22wmf16</generator>                                |
| 10        | <case>first-letter</case>   |
| 11        | <namespaces>...</namespaces>  |
| 12        | </siteinfo>   |
| 13        | <page>  |
| 14        | <title>Universiteti i Prishtinës</title>                                  |
| 15        | <ns>0</ns>  |
| 16        | <id>6823710</id>  |
| 17        | <revision>  |
| 18        | <id>571061203</id>  |
| 19        | ....  |
| 20        | <text xml:space="preserve" bytes="16353">                                 |
| 21        | {{Two other uses the university located in Pristina -----                 |
| 22        | </text>   |
| 23        | <sha1>chbyzbt3cmo0vz9pxvvlwx5lo3i4e7f</sha1>                              |
| 24        | <model>wikitext</model>   |
| 25        | <format>text/x-wiki</format>  |
| 26        | </revision>   |
| 27        | </page>   |
| 28        | </mediawiki>  |
| 29        |   |

### 2.1.3 RDF

Resource Description Framework (RDF) [18] është një modelim i të dhënave në një format të cilët reprezentojnë meta-data në uebin semantik, që edhe paraqet gjuhën kryesore në uebin semantik. Pra të dhënat në Uebin Semantik nuk janë thjesht të dhëna të publikuara në formatin e tabelave apo listave në ndonjë dokument HTML, por si “*treshe*” duke iu referuar RDF-it.

RDF paraqitet si një bashkëdyzim ndërmjet grafit dhe modelit të të dhënave, i përbërë nga treshet (*Subjekti*, *Predikati* dhe *Objekti*) dhe nga formati RDF/XML që shfrytëzohet për serializimin e grafit me treshet.

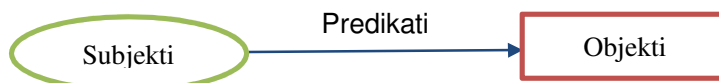


Figura 3 Tresha RDF.

Në Figura 3 janë paraqitur përbërësit e treshes së RDF-it: *subjekti*, *predikati*, dhe *objekti* dhe definohen siç vijon:

- **Subjekti:** Subjekti paraqet një resurs për të cilin dëshirojmë të krijojmë përmbajtje. Një subjekt zakonisht është i përbërë nga një URI ose një nyje të zbrazët.
- **Predikati:** Predikati e definon saktë se çfarë lloji të informatave dëshirojmë të shprehim përmes subjektit. Predikati përmban, apo duhet të përmbaj një URI.
- **Objekti:** Objekti definon vlerën e shprehur nga predikati. Objekti mund të jetë ose një URI, një label (tekst i thjesht) ose një nyje e zbrazët.

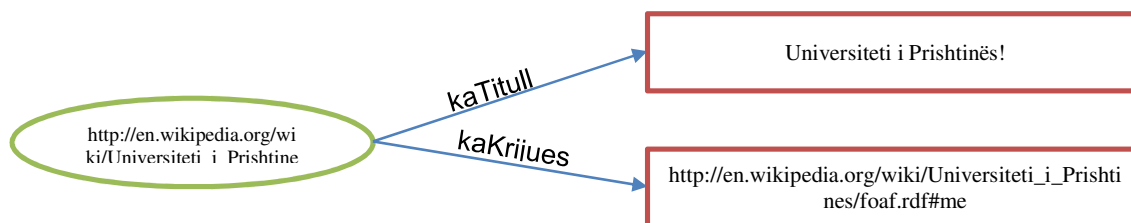
Më poshtë paraqitet reprezentimi i një shembulli RDF.

| Listimi.2 | RDF reprezentimi.   |
|-----------|---|
| 1         |   |
| 2         | <rdf:RDF  |
| 3         | xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"                             |
| 4         | xmlns:dc="http://purl.org/dc/elements/1.1/">  |
| 5         | <rdf:Description  |
| 6         | rdf:about="http://en.wikipedia.org/wiki/Universiteti_i_Prishtines">                 |
| 7         | <dc:title>Universiteti i Prishtinës!</dc:title>                                     |
| 8         | <dc:creator   |
| 9         | rdf:resource="http://en.wikipedia.org/wiki/Universiteti_i_Prishtines/foaf.rdf#me"/> |
| 10        | />  |
| 11        | />  |
| 12        | </rdf:RDF>  |
| 13        |   |

Në Listimi 2 është paraqitur një shembull RDF, i përbërë prej dy pjesëve:

- Pjesa 1: Paraqet gjendjen e një faqeje, të dhënë me URI (shih. rreshtin 6) : [http://en.wikipedia.org/wiki/Universiteti\\_i\\_Prishtines](http://en.wikipedia.org/wiki/Universiteti_i_Prishtines), si një resurse identifikues për atë faqe, dhe se përshkrimi/titulli i saj është “Universiteti i Prishtinës!” (shih. rreshtin 7), derisa
- Pjesa 2: Shpjegohet se për të njëjtën faqe është një identifikues që drejton tek resursi lokal (shih. rreshtin 9) : [http://en.wikipedia.org/wiki/Universiteti\\_i\\_Prishtines/foaf.rdf#me](http://en.wikipedia.org/wiki/Universiteti_i_Prishtines/foaf.rdf#me), që të dërgon tek krijuesi i këtij resursi (i cili pastaj është një entiten në vete).

Në Figura 4 është paraqitur shembulli nga Listimi 2 edhe përmes RDF- grafit.



**Figura 4 Paraqitja e treshëve për shembullin nga Listimi 2.**

RDF përdor këtë lloj prezantimi në formatin e treshëve për transferim të të dhënave, duke shfrytëzuar formatin XML.

#### 2.1.4 RDF Skema

RDF Schema (RDFS) [19] është një ontologji e thjeshtë e cila lejon që shfrytëzuesit e thjeshtë të definojnë shprehje të nevojshme për përshkrimin e resurseve në domen të caktuar përmes meta – të dhënave. Në mënyrë që kjo ontologji të definohet, RDFS përdor formatin e tresheve RDF. Tek RDFS shfrytëzuesit mund të definojnë klasat, vetitë, dhe relacionet për modelim të koncepteve në domen të caktuar.

Në vazhdim do të japim një hyrje për disa nga modelet bazike që përdoren për definimin e ontologjisë në RDFS. Termet të cilat janë të definuara në RDFS përmbajnë prefiksën “rdfs”, ndërsa ato të definuara në RDF përmbajnë prefiksën “rdf”.

Ky modelim përmban klasa dhe veti. Më poshtë do të paraqesim disa nga klasat e përdorura në modelimin e ontologjisë:

- `rdfs:Class` Kjo është klasa e resurseve të cilat janë RDF klasa.
- `rdfs:Literal` Kjo është klasa e vlerave të lateraleve siç janë lista e karaktereve dhe numrave.
- `rdfs:Property` Kjo është klasa e RDF vetive.

Ndërsa vetitë janë të përdorura për të definuar relacionet në ontologji. Disa nga vetitë do ti listojmë më poshtë:

- `rdf:type` Lidhë në mënyrë të drejtpërdrejte një resurs tek klasa e saj. Me fjalë të tjera, kjo veti përdoret për të deklaruar një resurs që të jetë një instancë e klasës së dhënë.
- `rdfs:subClassOf` Lidhë një klasë të dhënë me mbi-klasën e saj. Të gjitha instancat e klasës janë instanca edhe të mbi-klasës. Një klasë mund të jetë nën-klasë e më shumë se një klase.
- `rdfs:domain` Specifikon domenin e vetisë, që është klasa e resurseve që mund të jenë si subjekt në treshet që përmbajnë këtë predikat. Nëse ky domen nuk është i specifikuar atëherë çdo resurs është i paradefinuar si subjekt.
- `rdfs:range` Specifikon rangun e një vetie, që është klasa e këtyre resurseve që mund të jenë si objekte në një treshë që përmbajnë këtë predikat.

Përmes skemës RDF para-përcaktohen edhe konceptet në RDF, kjo është një zgjerim i RDF-it. Ndër konceptet që i përdor RDF skema mund të jenë : `rdfs:Class`, `rdfs:Property`, etj.

Në Figura 5 është paraqitur një skemë RDFS e klasës `Person` nga vokabulari FOAF [28].

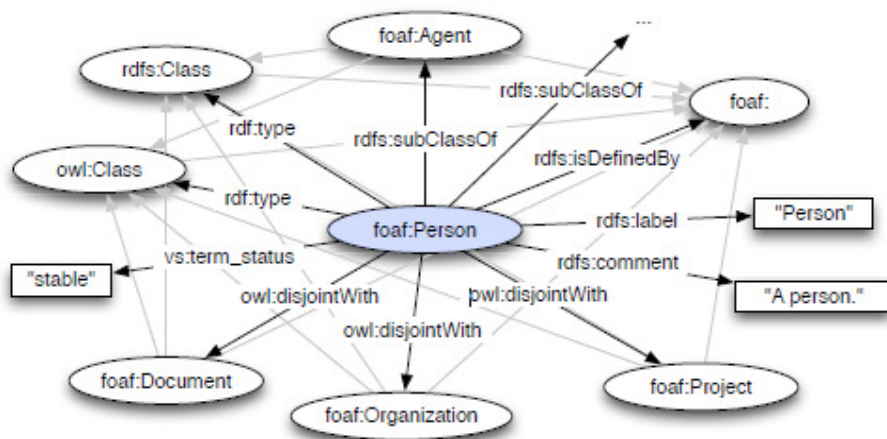


Figura 5 Definimi i klasës FOAF Person si pjese e vokabularit FOAF [28].

---

## 2.1.5 Ontologjitë dhe OWL

Web Ontology Language (OWL) [20] është gjuhë reprezentuese e njohurive përmes RDF formatit, duke mundësuar paraqitjen e të dhënave në ueb dhe aplikimin e kontrollueseve në të. Paraqitet si një gjuhë vetë-shpjeguese dhe dinamike, duke mundësuar që pala e tretë të krijojë detyra/kërkesa të pavarura.

Është e bazuar në formatin RDF dhe RDF skema, por përdor edhe pjesë të tjera në aspektin shprehës duke i aplikuar ato në paraqitjen e klasave dhe lidhjeve ndërmjet klasave apo instancave.

Siç e përshkruam edhe më parë, RDF përdor standardin RDF/XML, rrjedhimisht OWL është e bazuar në këtë format, pra RDF/XML.

Më poshtë është paraqitur një format i sintaksës së prezantimit në OWL formatin.

| Listimi.3 | OWL reprezentimi.   |
|-----------|---|
| 1         |   |
| 2         | <code>&lt;owl:Class rdf:ID="Publication"&gt;</code>                                       |
| 3         | <code>  &lt;rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Document"/&gt;</code> |
| 4         | <code>  &lt;owl:equivalentClass</code>  |
| 5         | <code>    &lt;rdf:Description</code>  |
| 6         | <code>      rdf:about="http://www.aktors.org/ontology/portal#Publication"&gt;</code>      |
| 7         | <code>        &lt;owl:equivalentClass rdf:resource="#Publication"/&gt;</code>             |
| 8         | <code>      &lt;/rdf:Description&gt;</code>   |
| 9         | <code>    &lt;/owl:equivalentClass&gt;</code>   |
| 10        | <code>  &lt;/owl:Class&gt;</code>   |
| 11        |   |

Siç shihet nga Listimi 3, sintaksa përdorë vetit dhe klasat nga skema RDFS, si dhe prefikset tjera si RDF, OWL. Prefikset e tilla thirren zakonisht në krye të dokumentit, si psh.:

| Listimi.4 | Inicializimi i prefiksve.  |
|-----------|--|
| 1         |  |
| 2         | <code>xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"</code>      |
| 3         | <code>xmlns:owl="http://www.w3.org/2002/07/owl#"</code>              |
| 4         | <code>xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"</code> |
| 5         |  |

Në këtë shembull të OWL-së (Listimi 3) thuhet se ekziston një klasë me ID: `rdf:ID="Publication"` (shih.rresh.2), e cila thuhet se është nënklasë e klasës `Document` (shih.rresh.3), nga ontologjia `foaf`, dhe është ekuivalente me klasën `Publication` (shih.rresh.4), nga ontologjia `portal`.

Varësisht nga kërkesa që një aplikacion kërkon në aspektin shprehës, dallojmë tri lloje të OWL-së [20]:

- OWL Lite
- OWL DL (Description Logic)
- OWL Full

**OWL Lite** : Është më pak komplekse në aspektin formal. Ajo mundëson vetëm disa veti që të shtohen tek ontologjia, përkatësisht tek RDF Skema. Siç mund të jetë rasti kur mundëson shtimin e kushteve:

- (ang. equality, inequality) për klasat dhe individët ose
- (ang. cardinality) për vetitë (por vetëm me vlerat 0 dhe 1).

**OWL DL** :Është dizajnuar në atë mënyrë që të pranoi maksimalisht shprehje dhe gjithashtu kontrolluesit (ang. reasoners) nuk do të paraqiteshin problem në aplikim.



---

**OWL Full** :Shihet nga emri që lejon operim të plotë në të, pra përmban të gjitha mundësit që i ofron një ontologji. Si shembull që mund të marrim, atë që lejon ky lloj varianti:

- Lejimin që të krijohen instanca të klasave nga klasat tjera në të njëjtën kohë (kjo nuk lejohet në OWL DL).

Kjo lejon mundësi maksimale të të shprehurit, por nuk garanton stabilitet, për shkak të kontrollueseve.

### 2.1.6 Pyetësorët SPARQL

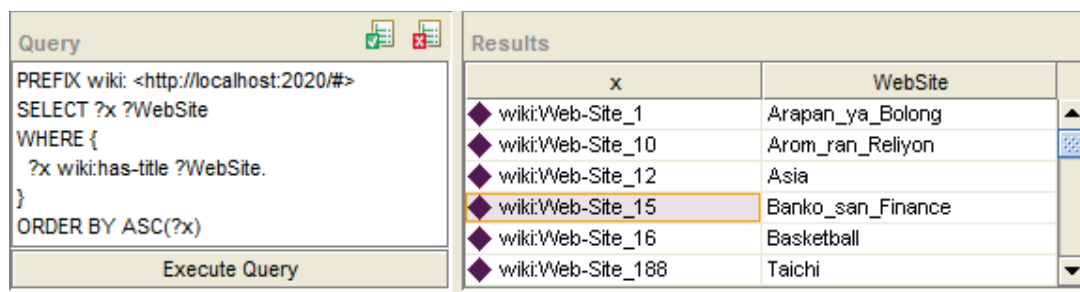
Pyetësorët SPARQL (ang. The SPARQL Query Language) përdoren në RDF në mënyrë që përmes tyre të nxirren të dhëna të çfarëdoshëm [21].

Në shikim të parë duket e përafërt me gjuhën pyetësore SQL për baza të të dhënave, por ajo përmban më shumë veti sesa SQL.

Në listimin e mëposhtëm është paraqitur një shembull i thjeshtë i pyetësorit SPARQL.

| Listimi.5 | Shembull i një pyetësori SPARQL.       |
|-----------|--|
| 1         |  |
| 2         | PREFIX wiki: <http://localhost:2020/#> |
| 3         | SELECT ?x ?WebSite                     |
| 4         | WHERE {                                |
| 5         | ?x wiki:has-title ?WebSite.            |
| 6         | }                                      |
| 7         | ORDER BY ASC(?x)                       |
| 8         |  |

Ndërsa në Figura 6 është prezantuar pyetësori dhe ekzekutimi i tij duke i paraqitur rezultatet e prekura nga pyetësori.



The screenshot shows a query editor on the left and a results table on the right. The query is: PREFIX wiki: <http://localhost:2020/#> SELECT ?x ?WebSite WHERE { ?x wiki:has-title ?WebSite. } ORDER BY ASC(?x). The results table has two columns: 'x' and 'WebSite'. The results are: wiki:Web-Site\_1 (Arapan\_ya\_Bolong), wiki:Web-Site\_10 (Arom\_ran\_Reliyon), wiki:Web-Site\_12 (Asia), wiki:Web-Site\_15 (Banko\_san\_Finance), wiki:Web-Site\_16 (Basketball), and wiki:Web-Site\_188 (Taichi). The row for 'Banko\_san\_Finance' is highlighted.

| x                 | WebSite           |
|-------------------|-------------------|
| wiki:Web-Site_1   | Arapan_ya_Bolong  |
| wiki:Web-Site_10  | Arom_ran_Reliyon  |
| wiki:Web-Site_12  | Asia              |
| wiki:Web-Site_15  | Banko_san_Finance |
| wiki:Web-Site_16  | Basketball        |
| wiki:Web-Site_188 | Taichi            |

**Figura 6 Ekzekutimi i SPARQL pyetësorit.**

Përdorimi i prefikseve përmes komandës PREFIX bën të mundur realizmin e thirrjeve të resurseve nga po ai prefiks (duke përdorur si shkurtesë për resursin e thirrur). Në shembullin e dhënë (Listimi 5), prefiksi definon shkurtesën për ontologjinë wiki (shkurtesa e URI-së http://localhost:2020/#) (shih.rresh.2). Këto prefikse bëhen me qëllim që më vonë gjatë përdorimit të tyre në pyetësorë të mund të bëhet thirrja e resurseve përmes shkurtesave, që edhe e bëjnë më të lexueshëm pyetësorin, duke mos e shkruar të gjithë linkun URI të resursit. Në shembullin e prezantuar thirrja wiki:has-title do të mundësohej edhe përmes thirrjes http://localhost:2020/#/has-title.

Shembulli i prezantuar (Listimi 5) merr të gjitha titujt e faqeve dhe vet instancën e faqeve (shih.rresh.3). Rezultati i tyre është prezantimi i tresheve të përbërë nga ndryshorja has-title dhe vlera e tyre. Kështu që rezultati është shfaqja e dy ndryshoreve (vetë instancës së faqes dhe emërtimit të saj, ?x ?WebSite) (të paraqitur në Figura 6).

## 2.1.7 Rregullat SWRL

Rregullat SWRL (ang. Semantic Web Rule Language) janë rregulla të bazuara në kombinimin e gjuhëve përshkruese OWL DL dhe OWL Lite të shprehura përmes veprimeve Unare apo Binare në markapin DataLog RuleML [22].

SWRL është e bazuar në OWL: të gjitha rregullat janë të shprehura si terme e koncepteve të OWL-së (si klasat, vetitë, individët, literalet, etj.), dhe paraqitja e tyre bëhet si pjesë përbërëse e *atomeve*.

Si shembull i SWRL rregullës është paraqitur në Listimi 6, ku bëhet listimi i të gjitha faqeve që gjinden në ontologjinë përkatëse. Ku kjo qasje mundësohet përmes përdorimit të OWL klasës *Web-Site* dhe paraqitja e saj bëhet përmes përdorimit të komandës `sqwrl:select` siç është paraqitur në Listimi 7.

| Listimi.6 | Shembull i një rregulle SWRL.                     |
|-----------|---|
| 1         |   |
| 2         | <code>wiki:Web-Site(?w) → sqwrl:select(?w)</code> |
| 3         |   |

SWRL rregullat përpos që përdoren për listim apo nxjerrje të rezultateve ato më së shpeshti përdoren për ekzekutim komandash duke shkruar vlera në veti përkatëse mbi instancat.

**Listimi.7** Ekzekutimi i rregullës SWRL dhe rezultati përmes SWRLJes Tab-it.

The image shows two parts of the SWRLJesTab interface. Part (a) shows the SWRL Rule editor with the rule `wiki:Web-Site(?w) → sqwrl:select(?w)`. Part (b) shows the execution results in the 'Imported Individuals' tab, listing various instances of `wiki:Web-Site` with their IDs, such as `wiki:Web-Site(wiki:Web-Site_8700)` through `wiki:Web-Site(wiki:Web-Site_295)`. The instance `wiki:Web-Site(wiki:Web-Site_8701)` is highlighted in blue.

a) Formulimi i rregullës SWRL

b) Rezultatet pas ekzekutimit me SWRLJesTab

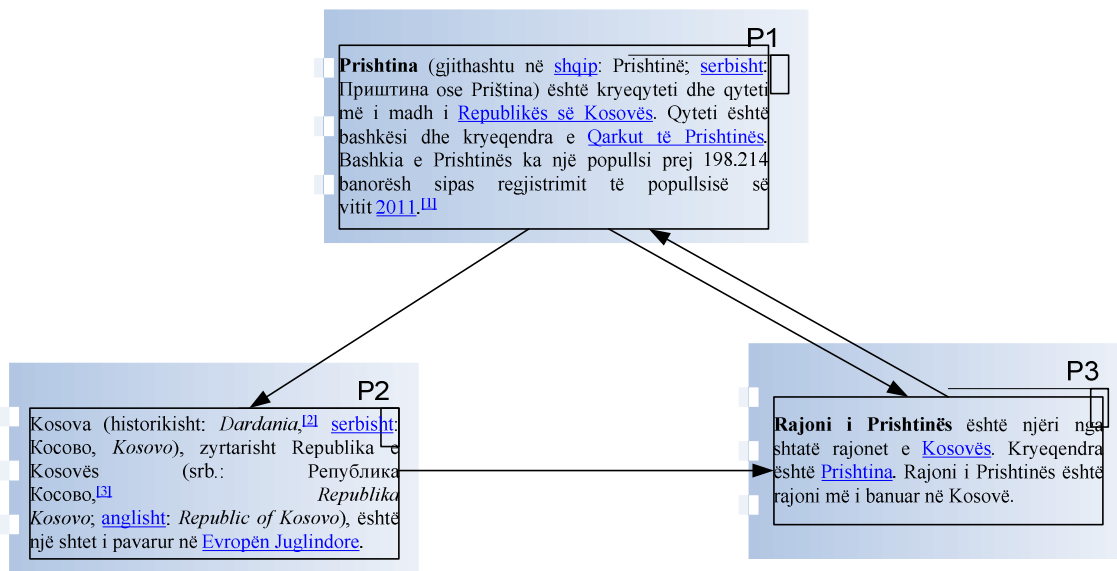
Ndërsa ekzekutimi dhe krijimi i rregullave mundësohet përmes SWRLTab-it [34] nga Protege.

## 2.2 Rangimi i faqeve

Në mënyrë që të përcaktohet rëndësia e një faqe gjatë momentit të kërkimit të saj, duhet që në një mënyrë makina kërkuese të bëjë sortimin e tyre. Për këtë arsye është propozuar [1] metoda PageRank, një metodë e cila kalkulon rangimin e çdo faqe duke u bazuar në lidhshmërinë e faqeve.

### 2.2.1 Struktura e faqeve

Pasi që uebi përmban një numër të madh të faqeve, në mënyrë që ato të jenë më të qasshme gjatë kërkimit aplikohet rregulli i ashtuquajtur –grupimi apo kategorizimi i faqeve. Faqet janë të përbëra nga një numër i caktuar liqesh, të cilët mund të jenë linqe në të (faqe) apo nga ajo (faqe) (shih. Figura 7).



**Figura 7 Lidhshmëria ndërmjet faqeve.**

Në Figura 7 është paraqitur lidhshmëria e faqeve ku shihet që faqet *P1* dhe *P3* janë lidhje në (faqen *P2*).

Sa më shumë lidhje që ka një faqe aq më e rëndësishme është në procesin e rangimit, por jo gjithmonë ka rëndësi numri i lidhjeve që ka ajo faqe, por sa janë të rëndësishëm ata linqe.

## 2.2.2 Algoritmi PageRank

Siç e përshkruam edhe në seksionin paraprak (Seksioni 2.2.1), një faqe është më e rëndësishme nëse shumta e rangimit të linqeve që janë në të (faqe) është më e madhe.

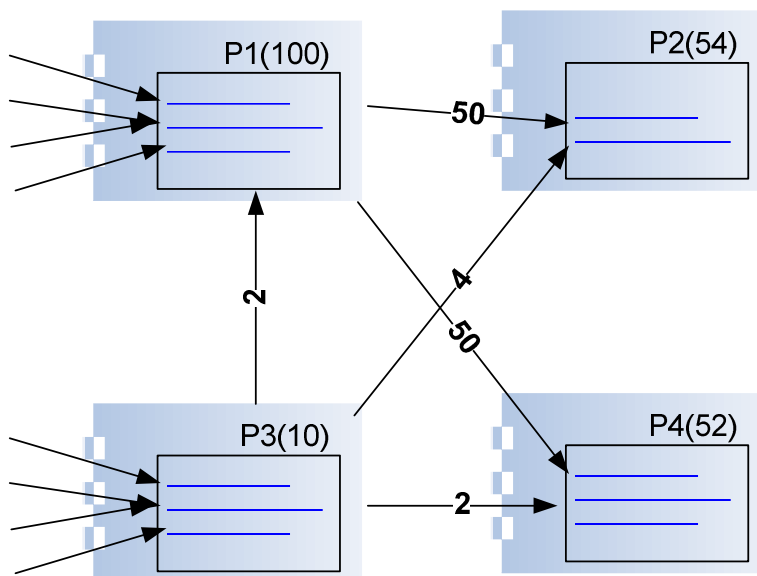
Më poshtë është paraqitur definicioni i rangimit të faqeve.

### 2.2.2.1 Definimi i PageRank-ut

Le të jetë  $p$  një ueb faqe. Pastaj le të jetë  $N_p$  një bashkësi faqesh  $p$  të cilat referojnë në të, dhe  $B_p$  le të jetë një bashkësi faqesh të cilat referojnë në  $p$ . Atëherë PageRank i faqes  $p$  paraqitet përmes shprehjes  $PR(p)$ , që paraqet shumën e të gjithave PageRank-ëve të gjithave faqeve të cilat referojnë në faqen  $p$  [2,3,4].

$$PR(p) = \sum_{v \in B_p} \frac{PR(v)}{N_v} \quad (1)$$

Në mënyrë që të kuptohet saktë ekuacioni (1) atëherë do ta thjeshtojmë paraqitjen e tij përmes katër faqeve (shih Figura 8).



**Figura 8** Numri i linqeve për faqe.

Figura 8 paraqet peshën e faqeve të cilat kanë linqe referencë tek faqja përkatëse.

### 2.2.2.2 Faktori *Damping*

Faktori *Damping* është një faktor ndikues në kalkulimin e ueb faqeve i cili ndodh në mënyrë të rastësishme nga klikimet e ueb shfrytëzuesit. Nëse një klikim i tillë ndodh në mënyrë të rastësishme dhe përsëritet disa herë atëherë ai faktor merr pjesë në peshën e faqes sepse në një farë mënyre faqja e cila arrihet përmes këtij modaliteti është, apo ka ndikim përmes këtyre faqe të klikuara (por jo të lidhura drejtpërdrejt me faqen në fjalë).

Në mënyrë që të kalkulohet ky lloj faktori është propozuar [1] një konstant  $d$ , e cila paraqet gjasën që do të ndihmoj këtë lloj ndikimi të rastësishëm që të vazhdoi në çfarëdo pike të caktuar që të ndodh edhe për linqet të cilat nuk janë drejtpërdrejt të referuara në ueb faqen. Sipas [1] kjo konstantë është përafëruar në vlerën **0.85**.

Faktori ndikues është paraqitur përmes shprehjes:

$$damp. = \frac{1 - d}{N} \quad (2)$$

Ku

$$d = 0.85$$

$N$  – numri i faqeve

Pra shprehja (2) paraqet raportin ndërmjet zbritjes së faktorit nga 1 dhe numrit total të faqeve.

Në mënyrë që faktori të merr pjesë në kalkulimin e PageRank-ut është zgjeruar ekuacioni (1) duke u paraqitur në këtë mënyrë:

$$PR(p) = \frac{1 - d}{N} + d \sum_{v \in B_p} \frac{PR(v)}{N_v} \quad (3)$$

Ku

$p$  – numri i faqeve të cilat merren në konsiderim,

---

$B_p$  – numri i faqeve të cilat lidhen në  $p$ ,  
 $N_v$  – numri linqeve të jashtëm tek  $p$ , dhe  
 $N$  – numri total i faqeve

Vlera e  $d$  aplikohet tek të gjitha faqet në mënyrë që edhe nëse faqet nuk kanë asnjë lidhje të jashtme të kenë vlerë në rangim.

### 2.2.2.3 Procedura Iterative e kalkulimit të PageRank-ut

Një nga metodat më të përdorura në kalkulimin e rangimit është Metoda Iterative [3,4], proces i cili do të përsëritet derisa sa vlerat e rangimit eventualisht do të konvergjojnë tek disa vlera finale që do të jenë të qëndrueshme.

Le të jetë  $k$  numri i iteracioneve dhe supozojmë se në fillim të gjitha faqet kanë rangimin e njëjtë, pra

$$PR_0(p) = 1 \quad (4)$$

Atëherë funksioni për kalkulimin e rangimit merr këtë formë.

$$PR_{k+1}(p) = \frac{1-d}{N} + d \sum_{v \in B_p} \frac{PR_k(v)}{N_v} \quad (5)$$

Ku

$PR_k(p)$  – pagerank-u i faqes  $p$  në iteracionin  $k$ ,  
 $d$  – dumping faktori,  
 $B_p$  – numri i faqeve të cilat lidhen në  $p$ ,  
 $N_v$  – numri linqeve të jashtëm tek  $p$ , dhe  
 $N$  – numri total i faqeve

Por në shumicën e rasteve gjatë kalkulimit të PageRanku-t nuk merret parasysh parametri i numrit total të faqeve, por vetëm diferenca ndërmjet numrit të plotë 1 dhe faktorit  $d$ . Pra në këtë rast funksioni (5) merr formën:

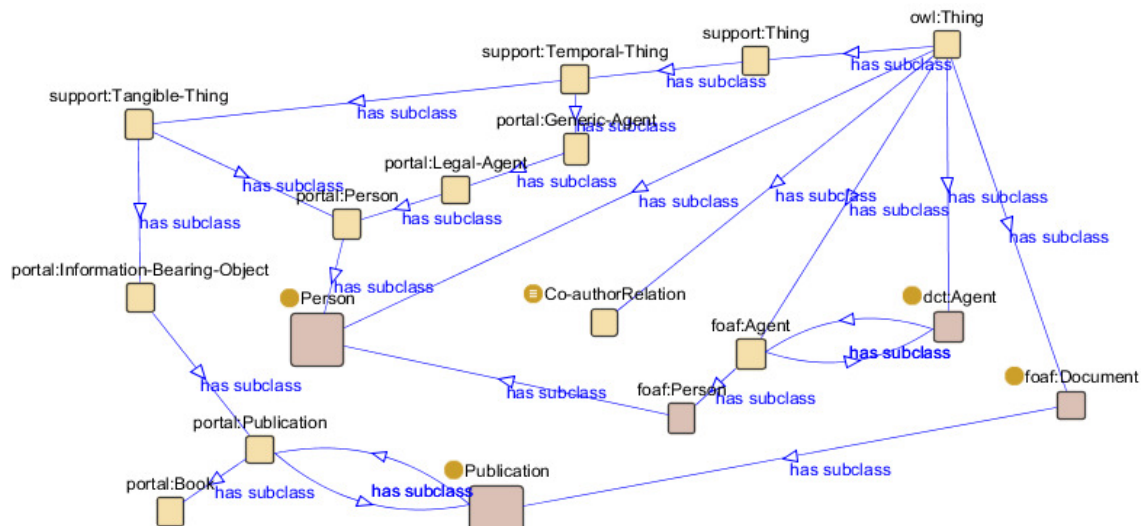
$$PR_{k+1}(p) = 1 - d + d \sum_{v \in B_p} \frac{PR_k(v)}{N_v} \quad (6)$$

Për më shumë rreth procedurës së kalkulimit të PageRank-ut shikoni në [1].

### 3 Puna paraprake

Për tu bërë e mundur kalkulimi i rangimit semantik në domenin e Wikipedia-së më parë duhet të përshkruhet puna e bërë për përgatitjen e terrenit për një kalkulim të tillë.

Dhe në këtë kontekst konsiderojmë punën e cila është bërë në zgjerimin e vokabularit FOAF në ontologjinë e ashtu quajtur *co-AuthorOnto* (Figura 9) [9] në mënyrë që të modelohen relacionet ndërmjet autorëve si dhe entiteteve të cilat janë shtuar tek ontologjitë referente të AKT-së (ontologji të cilat janë integruar në këtë ontologji të zgjeruar, konkretisht ontologjia referente *portal* dhe *support*).



**Figura 9** Pamje nga ontologjia *co-AuthorOnto* [9] - relacioni ndërmjet *foaf*, *portal* dhe *support* ontologjive.

Pastaj është punuar në zgjerimin e vetive tek klasa *Person*, veti të cilat përcaktojnë edhe peshën e rangimit të një personi (vetia *hasPR* dhe *hasAR*), si dhe krijimin e një klase ndihmëse (klasa *Constants*) e cila konsiderohet si klasë e cila ruan ndryshoret globale, të cilat përdoren gjatë kalkulimeve të ndryshme në ontologji [10].

Pasi që ka qenë e pa pamundur të bëhet kalkulimi iterativ i vlerave në Protege, është punuar në shtimin e një veti ndihmëse *hasPRTemp* e cila ruan rezultatin gjatë kalkulimit iterativ të vlerës. Dhe për këtë është bërë realizimi i një metode ndihmëse e cila bën të mundur përdorimin e kësaj vetie duke përdorur SWRL rregullat dhe ekzekutimin e tyre përmes Jena API-të [11].

Duke u mbështetur nga puna paraprake që është bërë në aspektin e kalkulimit semantik të rangimit atëherë pjesa e punës që është bërë në këtë punim ka qenë përshtatja, zgjerimi, realizimi i kalkulimit, importimi dhe mapimi i të dhënave nga domeni i Wikipedia-së në ontologjinë e zgjeruar.

## 4 Ontologjia dhe rregullat SWRL të rangimit – Wikipedia si një rast studimi

Wikipedia është enciklopedia më e madhe e editueshme në rrjet. Ajo është lansuar në fillit të vitit 2001 nga Jimbo Wales dhe Larry Sanger, por ideja e parë e një përmbajtje si wiki është paraqitur nga Ward Cunningham [23] dhe që nga atëherë përmban miliona artikuj në mijëra gjuhë të botes [24].

Wikipedia është thjesht një sistem për menaxhimin e përmbajtjes, që edhe lexuesit të thjesht ia jep mundësin që ta ndryshoi përmbajtjen ekzistuese apo të shtoi ndonjë të re.

Një përmbajtje e tillë është e kategorizuar dhe e ndarë në artikuj, në mënyrë që të ofroi një kërkim të plotë të përmbajtjes.

Në Figura 11 është paraqitur një shembull i rezultatit/redaktimit të një përmbajtje (Fjala kyçe “Prishtina”, dhe gjuha “shqipe”).



The image shows a screenshot of the Wikipedia article for "Prishtina" in Albanian. On the left is the Wikipedia logo and navigation menu. The main content area shows the article title "Prishtina" and a warning message: "Ky artikull ose seksion duhet të përkrahë ndihmën e ndihmësive". Below this, there is a line of text: "Ky artikull merret me Komunën e Prishtinës. Për artikullin Qarku i Prishtinës". The article text is mostly obscured by a large redacted area.

Figura 10 Paraqitja e përmbajtjes në Wikipedia.

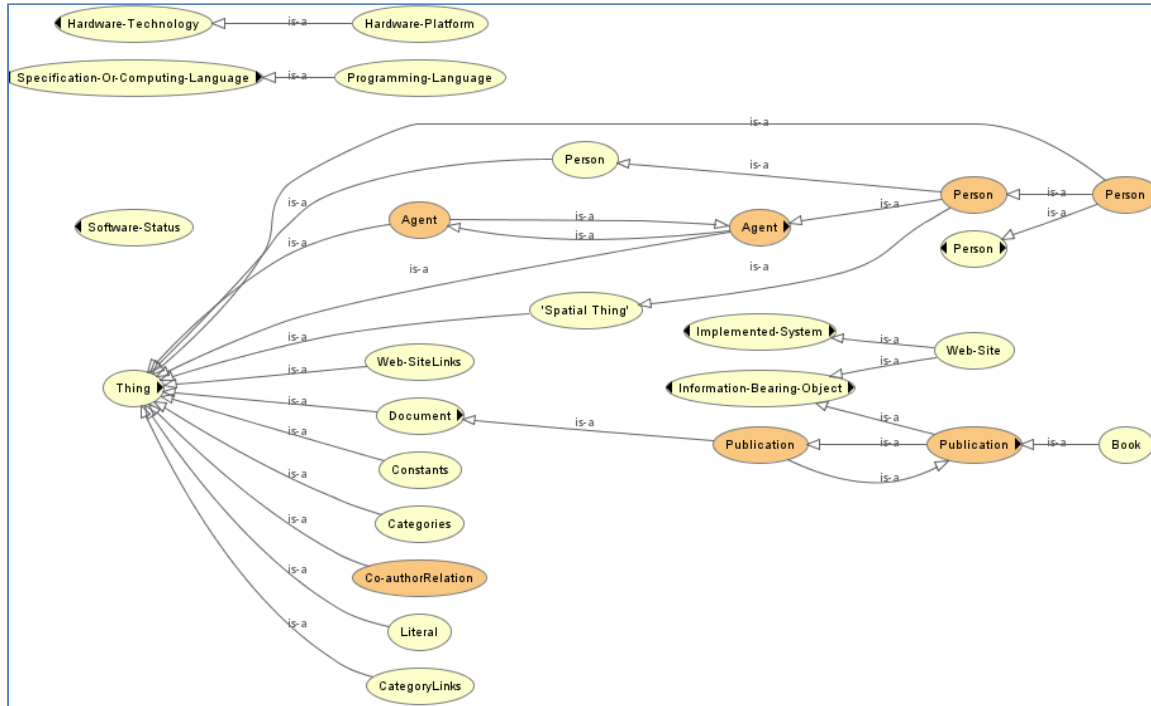
Struktura e Wikipedia-së është e përbërë nga këto karakteristika [5]:

- *kategoritë*, që kategorizon artikujt bazuar në përmbajtjen e tyre.
- *tipet e linqeve*, që klasifikon linqet ndërmjet artikujve duke u bazuar në domethënien e tyre, dhe
- *atributet*, që specifikon vetitë e thjeshta bazuar në përmbajtjen e artikullit.

Duke u bazuar në karakteristikat që i posedon Wikipedia, atëherë tek ky domen është e mundur të bëhet implementimi i rangimit të faqeve (artikujve) duke u mbështetur tek lidhjet të cilat janë të paraqitura, por gjithashtu është një set i mirë për ta mundësuar rangimi semantik[11] të faqeve të Wikipedia-së.

## 4.1 Ontologjia SemanticSearchOnto

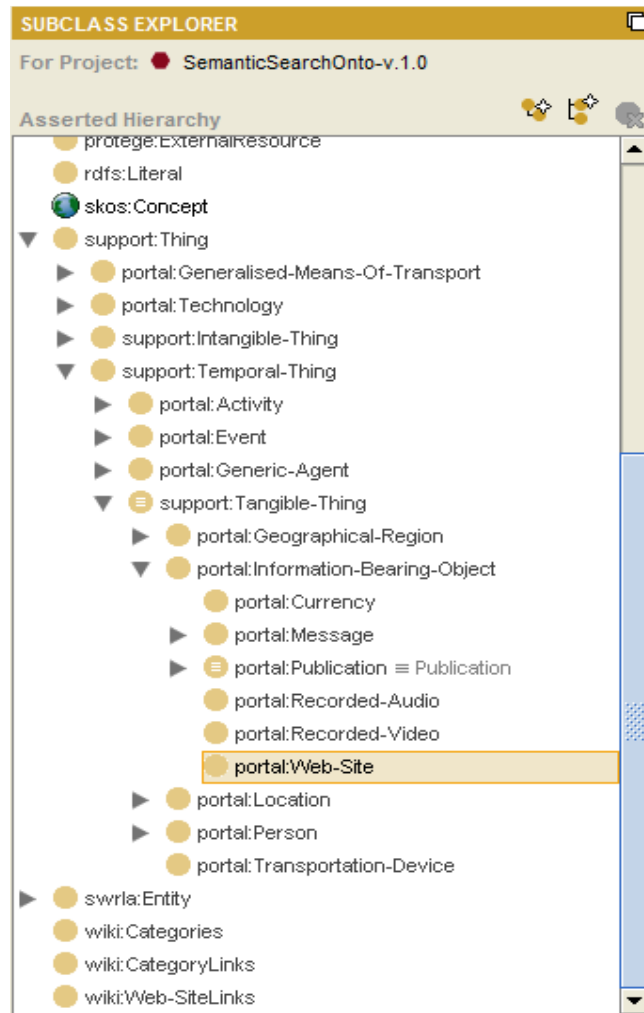
Duke shfrytëzuar ontologjinë që më parë është modeluar [10] dhe importimin e ontologjisë së krijuar/gjeneruar nga mapimi i të dhënave nga databaza e Wikipedia-së (të paraqitur tek Seksioni 4.3), kjo ontologji është zgjeruar dhe ka këtë strukturë.



**Figura 11 Ontologjia SemanticSearchOnto – hierarkia e klasave.**

Siç shihet nga Figura 11 nuk ka qenë e nevojshme të krijohet një klasë e veçantë për të modeluar rangimin e ueb faqeve përderisa ekziston klasa "portal:Web-Site", që është një nën-klasë në hierarkinë e klasës "portal#Information-Bearing-Object", por vetëm është mapuar nga baza e shënimeve sipas prefiksit portal, e paraqitur në Figura 12.





**Figura 12** Një pamje e hierarkisë së klasës “portal:Web-Site”.

Në mënyrë që të bëhet përcaktimi konkret i lidhjeve ndërmjet faqeve janë krijuar këto klasa në ontologji të mapuara më parë nga baza e shënimeve të Wikipedia-së:



**Figura 13** Klasat e mapuara në ontologji.

të cilat përmbajnë veti dhe instanca.

- portal:Web-Site është klasa e cila përmban faqet e importuara nga databasa e Wikipedia-së.

Figura 14 paraqet disa nga vetitë kryesore të kësaj klase.

CLASS EDITOR for portal:Web-Site (instance of owl:Class, rdfs:Class)

For Class: <http://www.aktors.org/ontology/portal#Web-Site>  Inferred View

rdfs:subClassOf

- portal:Implemented-System
- portal:Information-Bearing-Object

Properties

| Property               | Cardinality | Type  |
|------------------------|-------------|-------|
| hasAlly                | Multiple    | float |
| hasAR                  | Multiple    | float |
| hasInLink ↔ hasOut...  | Multiple    |       |
| hasOutLink ↔ hasInL... | Multiple    |       |
| hasPR                  | Multiple    | float |
| hasTotalNrOutLinks     | Multiple    | int   |
| hasWebAlly             | Multiple    | float |
| hasWebPR               | Multiple    | float |
| portal:has-title       | Multiple    |       |
| portal:has-URL         | Multiple    |       |
| wiki:has-page_random   | Multiple    |       |
| wiki:hasCategories     | Multiple    |       |

Figura 14 Disa nga vetitë e klasës portal:Web-Site .

Shikoni në Figura 12 se si klasa Web-Site është e renditur si klasë nga tipi portal e ontologjisë AKT [27], e treguar në krye të figurës.

Duke iu referuar ontologjisë [10], tek e cila janë përdorur vetitë për përcaktimin/kalkulimin e rangut për faqe përkatësisht autorë, prandaj edhe në këtë projekt janë përdorur të njëjtat veti, vetëm se janë adaptuar për ontologjinë SemanticSearchOnto.

Më poshtë janë përshkruar vetitë të cilat janë gjeneruar përmes mapimit nga baza e shënimeve si dhe vetitë e tjera, po ashtu edhe klasat e shfrytëzuara nga ontologjia [10] ekzistuese .

- o portal:has-title përcakton emrin unik të faqes nga baza e shënimeve, e cila identifikon faqen, duke u mapuar me tipin e njejtë nga klasa portal.
  - o wiki:has-page\_random paraqet një rangim të cilin Wikipedia e ka bërë për faqen përkatëse.
  - o hasOutLink kjo veti është e tipit ObjectType dhe përmban të gjitha lidhjet e jashtme të cilat përfaqëson faqja përkatëse. Domeni dhe rangi i kësaj vetie është “Web-Site” dhe “Web-SiteLinks” respektivisht.
  - o wiki:hasCategories kjo veti është e tipit ObjectType dhe përmban listën e kategorive të cilave i takon faqja (kategoritë janë faqe në vetvete). Domeni i kësaj vetie është “Web-Site” ndërsa rangi i saj është “CategoryLinks”.
  - o hasWebPR vetia e shfrytëzuar nga ontologjia [10] vetëm duke u riemëruar nga hasPR, është edhe vetia përcaktuese për rangimin e faqes. Domeni i kësaj vetive është vendosur “Web-Site”, ndërsa rangi i saj është “float”.
  - o hasWebPRTemp është përdorur si veti ndihmëse për ruajtjen e vlerave gjatë kalkulimit në mënyrë iterative të rangut të faqeve. Domeni i kësaj vetive është “Web-Site”, ndërsa rangi i saj është “float”.
- wiki:Web-SiteLinks është klasa e cila përmban lidhjet ndërmjet faqeve të importuara nga databaza e Wikipedia-së.

CLASS EDITOR for wiki:Web-SiteLinks (instance of rdfs:Class)

For Class: <http://localhost:2020/#Web-SiteLinks>  Inferred View

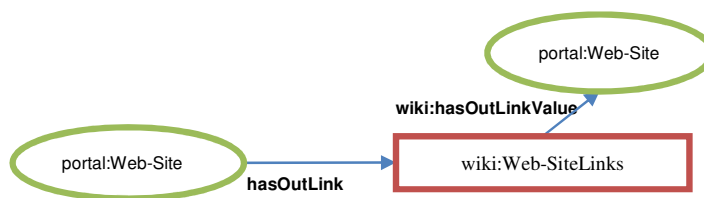
rdfs:subClassOf

Properties

| Property             | Cardinality | Type |
|----------------------|-------------|------|
| wiki:hasOutLinkValue | Multiple    |      |

Figura 15 Vetitë e klasës wiki:Web-SiteLinks.

- o wiki:hasOutLinkValue kjo veti është e tipit ObjectType dhe përmban instancën përkatëse të një lidhje të jashtme nga vetia hasOutLink. Domeni dhe rangu i kësaj vetie është “Web-SiteLink” dhe “Web-Site” respektivisht.



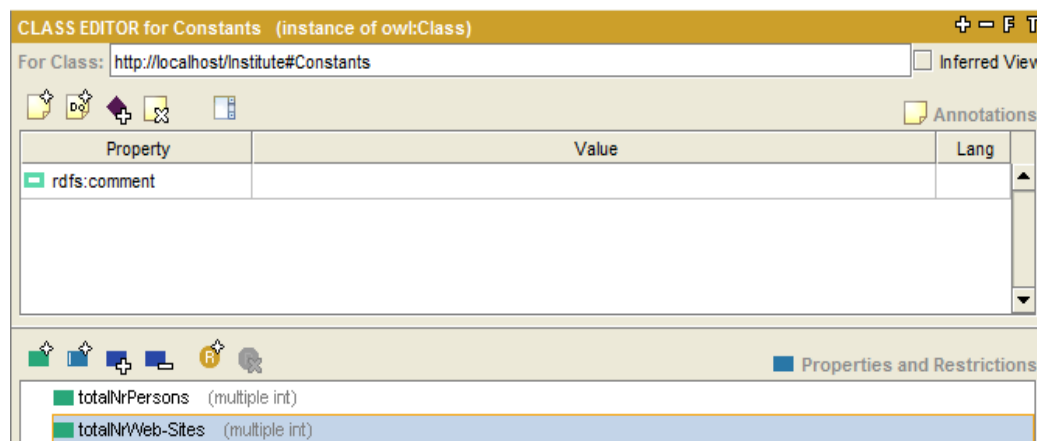
**Figura 16 Paraqitja e lidhshmërisë së vetisë hasOutLink.**

- wiki:Categories është klasa e cila përmban kategoritë e faqeve të importuara nga databasa e wikipedia-së.



**Figura 17 Vetitë e klasës wiki:Categories.**

- o wiki:has-Cat\_title përcakton emrin unik të kategorisë nga baza e shënimeve.
- o wiki:hasCategoryLinks përcakton lidhjen ndërmjet kategorive. Domeni i kësaj vetie është “Categories” ndërsa rangi i saj është “CategoryLinks”.
- wiki:CategoryLinks është klasa e cila përmban lidhjet ndërmjet kategorive të importuara nga databasa e wikipedia-së.
- Constants është klasa e cila përdoret për ruajtjen e numrit total të faqeve që përmban ontologjia dhe atë duke e ruajtur vlerën në vetinë "totalNrWeb-Sites", veti e re e shtuar tek kjo klasë, nga ontologjia [10].

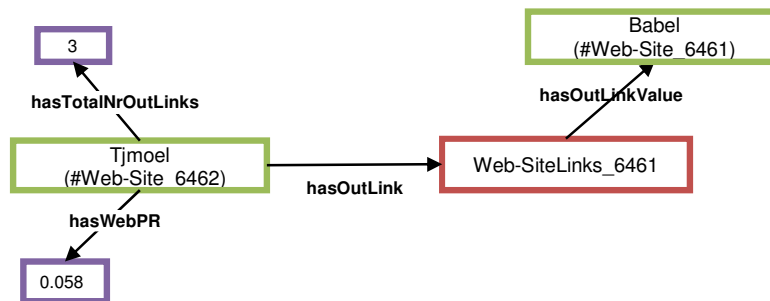


**Figura 18 Vetitë e klasës Constants.**

- o `totalNrWeb-Sites` ruan vlerën totale të faqeve në ontologji. Ku domeni është "Constants" dhe rangu "int".  
Kjo veti konsiderohet si një variabël globale për ruajtjen e rezultatit mbi numrin total të instancave nga klasa `Web-Site`, duke shtuar një instancë të vetme (`Constant1`) e cila mban vlerën e numrit total të faqeve në ontologji.

Disa nga vetitë e paraqitura në këtë seksion do të plotësohen me vlera duke përdorur rregullat SWRL të aplikuara në ontologji, që do të paraqiten në seksionin e ardhshëm.

Një model i paraqitjes së instancës duke i përdorur vetitë e prezantuara më sipër është paraqitur në Figura 19.



**Figura 19 Paraqitja e një instance të klasës `Web-Site`, dhe relacioni i saj.**

Figura 19 bën ilustrimin e një faqe (Tjmoel) dhe vlerës së saj në raporti e relacionit ndaj lidhjeve të jashtme. Në këtë nivel përdorimi i vetisë `hasOutLink` për të prezantuar lidhjet si dhe vetisë `hasOutLinkValue` për përcaktim të këtij relacioni, duke simuluar në këtë rast që si lidhje ndërmjet faqes "Tjmoel" dhe "Babel" përdoret relacioni `Web-SiteLinks_6461`. Si dhe duke i prezantuar edhe vetitë e rëndësishme të klasës `Web-Site`, siç është numri total i lidhjeve të jashtëm (`hasTotalNrOutLinks`) që është "3", si dhe vlera e rangut semantik (`hasWebPR`) të kësaj faqe "0.058".

Prandaj, pasi që janë përshkruar klasat dhe vetitë të cilat do të përdoren gjatë manipulimit me ontologji, atëherë vjen momenti i paraqitjes së rregullave të realizuara për kalkulimin e rangut semantik të ueb faqeve.

## 4.2 Rregullat SWRL për kalkulimin e PageRank-ut semantik

Në mënyrë që të manipulohet me të dhënat në dispozicion të ontologjisë, dhe ato të dhëna të kenë domethënie semantike janë përdorur rregullat SWRL për kalkulim dhe paraqitje të instancave apo vlerave të reja në ontologji.

Rregullat bazike për kalkulimin semantik të PageRank-ut janë definuar në [10], por në mënyrë që këto rregulla të jenë të aplikueshme tek domeni ynë, janë zgjeruar dhe adaptuar sipas strukturës së ontologjisë `SemanticSearchOnto`.

Në pajtim me nevojën për kalkulim të rangut semantik të ueb faqeve të Wikipedia-së janë krijuar këto rregulla:

| Enabled                             | Name   | Expression   |
|-------------------------------------|--------|--|
| <input type="checkbox"/>            | Rule-1 | $\rightarrow \text{portal:Web-Site(?w)} \circ \text{sqwrl:makeSet(?s, ?w)} \circ \text{sqwrl:size(?n, ?s)} \rightarrow \text{totalNrWeb-Sites(Constant1, ?n)}$   |
| <input type="checkbox"/>            | Rule-2 | $\rightarrow \text{portal:Web-Site(?p)} \wedge \text{hasOutLink(?p, ?a)} \circ \text{sqwrl:makeSet(?s, ?a)} \wedge \text{sqwrl:groupBy(?s, ?p)} \circ \text{sqwrl:size(?size, ?s)} \rightarrow \text{hasTotalNrOutLinks(?p, ?size)}$   |
| <input checked="" type="checkbox"/> | Rule-3 | $\rightarrow \text{hasOutLink(?wj, ?wrel)} \wedge \text{wiki:hasOutLinkValue(?wrel, ?wi)} \wedge \text{hasTotalNrOutLinks(?wj, ?cj)} \wedge \text{totalNrWeb-Sites(Constant1, ?n)} \wedge \text{swrlm:eval(?yj, "1/cj", ?cj)} \rightarrow \text{hasWebPR(?wi, ?pri)}$                                |
| <input type="checkbox"/>            | Rule-4 | $\rightarrow \text{hasPR(?wj, ?pri)} \wedge \text{hasOutLink(?wj, ?wrel)} \wedge \text{wiki:hasOutLinkValue(?wrel, ?wi)} \wedge \text{hasTotalNrOutLinks(?wj, ?cj)} \wedge \text{totalNrWeb-Sites(Constant1, ?n)} \wedge \text{swrlm:eval(?yj, "1/cj", ?cj)} \rightarrow \text{hasWebPR(?wi, ?pri)}$ |

Figura 20 SWRL rregullat për kalkulimin e rangut semantik.

Ndërsa më poshtë do të paraqitet edhe rregullat në detaje dhe përshkrimin e tyre.

| Listimi.8 | Rule-1 (Numri total i Web-Site).  |
|-----------|---|
| 1         |   |
| 2         | $\text{portal:Web-Site(?w)} \circ \text{sqwrl:makeSet(?s, ?w)} \circ \text{sqwrl:size(?n, ?s)} \rightarrow$ |
| 3         | $\text{totalNrWeb-Sites(Constant1, ?n)}$  |
| 4         |   |

Rregulla **Rule-1** gjen numrin total të instancave të Web-Site në ontologji, duke marruar të gjitha faqet ?w dhe duke i vendosur ato në një bashkësi ?s (funksioni makeSet), pastaj ruajtja e numrit total të elementëve, ?n, në një bashkësi (funksioni size). Pastaj ky numër i llogaritur, ?n, vendoset tek vetia globale totalNrWeb-Sites e instancës Constant1, në klasën Constants të ontologjisë.

| Listimi.9 | Rule-2 (Numri total i lidhjeve të jashtëm për Web-Site-in përkatës).                                    |
|-----------|---|
| 1         |   |
| 2         | $\text{portal:Web-Site(?p)} \wedge \text{hasOutLink(?p, ?a)} \circ \text{sqwrl:makeSet(?s, ?a)} \wedge$ |
| 3         | $\text{sqwrl:groupBy(?s, ?p)} \circ \text{sqwrl:size(?size, ?s)} \rightarrow$                           |
| 4         | $\text{hasTotalNrOutLinks(?p, ?size)}$  |
| 5         |   |

Kjo rregull **Rule-2** kalkulon numrin e lidhjeve të jashtëm për çdo Web-Site të dhënë. Njëjtë sikurse tek Rule-1 ashtu edhe këtu bëhet nxjerrja e të gjitha faqeve ?p, pastaj numri i lidhjeve të jashtëm ?a duke i vendosur në një bashkësi ?s (funksioni makeSet) dhe grupimi është aplikuar (groupBy) që të bëjë grupimin sipas faqeve, dhe në fund kalkulimin e numrit të elementeve të afektuar (funksioni size). Pastaj ky numër i llogaritur, ?size, vendoset tek vetia hasTotalNrOutLinks e instancës ?p, në klasën portal:Web-Site të ontologjisë.

| Listimi.10 | Rule-3 (Kalkulimi inicial i rangut për Web-Site).  |
|------------|--|
| 1          |  |
| 2          | $\text{hasOutLink(?wj, ?wrel)} \wedge \text{wiki:hasOutLinkValue(?wrel, ?wi)} \wedge$                  |
| 3          | $\text{hasTotalNrOutLinks(?wj, ?cj)} \wedge \text{totalNrWeb-Sites(Constant1, ?n)} \wedge$             |
| 4          | $\text{swrlm:eval(?yj, "1/cj", ?cj)} \circ \text{sqwrl:makeBag(?s, ?yj)} \wedge$                       |
| 5          | $\text{sqwrl:groupBy(?s, ?wi)} \circ \text{sqwrl:sum(?ally, ?s)} \wedge \text{swrlm:eval(?pri, "(0.15$ |
| 6          | $/ n) + 0.85 * ally", ?n, ?ally)} \rightarrow$   |
| 7          | $\text{hasWebPR(?wi, ?pri)}$   |
| 8          |  |

Përmes rregullës **Rule-3** bëhet kalkulimi fillestar i rangut të Web-Site, duke mundësuar iniciimin e kalkulimit iterativ të rangut.

Kalkulimi i tillë merr për bazë formulën e paraqitur në ekuacionin (6) për kalkulimin e PageRank-ut.

Kjo rregull është e organizuar në këtë mënyrë gjatë ekzekutimit të saj:

- Së pari bën procedimin e të gjitha lidhjeve të jashtëm që ka faqja ?wj, duke marr në konsideratë rregullën e relacionit, ku ?wrel prezanton lidhjen ndërmjet faqes ?wj dhe faqes ?wi .
- Llogaritë parametra të nevojshëm për kalkulimin e më vonshëm në rregull si numrin total të lidhjeve të jashtëm ?cj, raportin e këtyre lidhjeve ?yj, pastaj numrin total të faqeve në ontologji ?n, etj.
- Aplikon shumën (funksioni sum) e të gjitha lidhjeve të jashtëm ?ally që marrin pjesë në relacion të jashtëm tek faqja ?wi, duke e llogaritur më parë këtë vlerë (funksioni eval ) ku ky raport është llogaritur përmes shprehjes ?yj=1/?cj, që edhe paraqet peshën e faqes ?wj, ku ?cj është numri total i lidhjeve të jashtëm që i posedon faqja ?wj.
- Dhe në fund bëhet edhe llogaritja e rangut duke marrë në konsideratë po ashtu edhe faktorin “dumping” [1] që është i parapërcaktuar me vlerën 0.85 si dhe vlerën totale të faqeve ?n në ontologji dhe duke e vendosur atë vlerë ?pri tek vetia hasWebPR e instancës ?wi, në klasën portal:Web-Site të ontologjisë.

|                   |   |
|-------------------|---|
| <b>Listimi.11</b> | <b>Rule-4 (Kalkulimi i rangut për Web-Site.).</b> |
|-------------------|---|

|   |   |
|---|---|
| 1 | hasWebPR(?wj, ?prj) $\wedge$ hasOutLink(?wj, ?wrel) $\wedge$                    |
| 2 | wiki:hasOutLinkValue(?wrel, ?wi) $\wedge$ hasTotalNrOutLinks(?wj, ?cj) $\wedge$ |
| 3 | totalNrWeb-Sites(Constant1, ?n) $\wedge$ swrlm:eval(?yj, "prj/cj", ?prj, ?cj)   |
| 4 | ° sqwrl:makeBag(?s, ?yj) $\wedge$ sqwrl:groupBy(?s, ?wi) °                      |
| 5 | sqwrl:sum(?ally, ?s) $\wedge$ swrlm:eval(?pri, "0.15 / n + 0.85 *               |
| 6 | ally", ?n, ?ally) $\rightarrow$   |
| 7 | hasWebPR(?wi, ?pri)   |
| 8 |   |
| 9 |   |

Derisa sa është përcaktuar dhe është llogaritur rangimi fillestar i faqeve (Rule-3) atëherë është e mundur të implementohet rangimi iterativ për çdo instancë të Web-Site, duke marruar për bazë rangimin fillestar të instancës.

Rregulla është e organizuar në këto pjesë gjatë ekzekutimit:

- Së pari bën procedimin e të gjitha lidhjeve të jashtëm që ka faqja ?wj, duke marr në konsideratë rregullën e relacionit, ku ?wrel prezanton lidhjen ndërmjet faqes ?wj dhe faqes ?wi , si dhe çka është ma kryesorja merr edhe vlerën e para-llogaritur të rangut të faqes (rangimi fillestar) ?prj (nga vetia hasWebPR).
- Llogaritë parametra të nevojshëm për kalkulimin e më vonshëm në rregull si numrin total të lidhjeve të jashtëm ?cj, raportin e këtyre lidhjeve ?yj pastaj numrin total të faqeve në ontologji ?n etj.
- Aplikon shumën (funksioni sum) e të gjitha lidhjeve të jashtëm ?ally që marrin pjesë në relacion të jashtëm tek faqja ?wi, duke e llogaritur më parë këtë vlerë (funksioni eval ) ku ky raport është llogaritur përmes shprehjes ?yj= ?prj/?cj, (siç shihet këtu merr pjesë rankimi aktual i llogaritur i faqes ?wj ) që edhe paraqet peshën e faqes ?wj, ku ?cj është numri total i lidhjeve të jashtëm që i posedon faqja ?wj.
- Dhe në fund bëhet edhe llogaritja e rangut duke marrë në konsideratë po ashtu edhe faktorin “dumping” [1] që është i parapërcaktuar me vlerën 0.85, vlerën e rangut paraprak të llogaritur nga rregulla initiale (Rule-3) si dhe vlerën totale të faqeve ?n në ontologji dhe duke e vendosur atë vlerë ?pri tek vetia hasWebPR e instancës ?wi, në klasën portal:Web-Site të ontologjisë.

Kjo rregull ekzekutohet tek të gjitha instancat e faqeve (si instanca primare dhe ato nga relacionet duke shfrytëzuar vlerën e lidhjes ?wrel nga vetia e relacionit wiki:hasOutLinkValue).

---

Rregulla Rule-4 paraqet edhe rregullën kryesore për llogaritjen e rangut semantik, që edhe përcakton rangun e një faqe.

### 4.3 Mapimi i të dhënave të Wikipedia në ontologji

Gjatë këtij seksioni do të përshkruajmë në detaje procesin e mapimit të të dhënave për rastin tonë të studimit.

#### 4.3.1 Wikipedia e cunguar (ang. *dump*)

Në mënyrë që rangimi semantik të aplikohet tek faqet e Wikipedia-së, ne e kemi në dispozicion bazën e cunguar të të dhënave të ofruara nga Wikipedia [35].

Të dhënat të cilat ne i kemi në dispozicion janë të përbëra nga tri nivele të organizimit:

1. Lidhshmëritë ndërmjet faqeve,
2. Lidhshmëritë ndërmjet kategorive dhe
3. Lidhshmëritë ndërmjet faqeve dhe kategorive.

Në Figura 21 është prezantuar diagrami i kësaj baze të cunguar të Wikipedia-së.

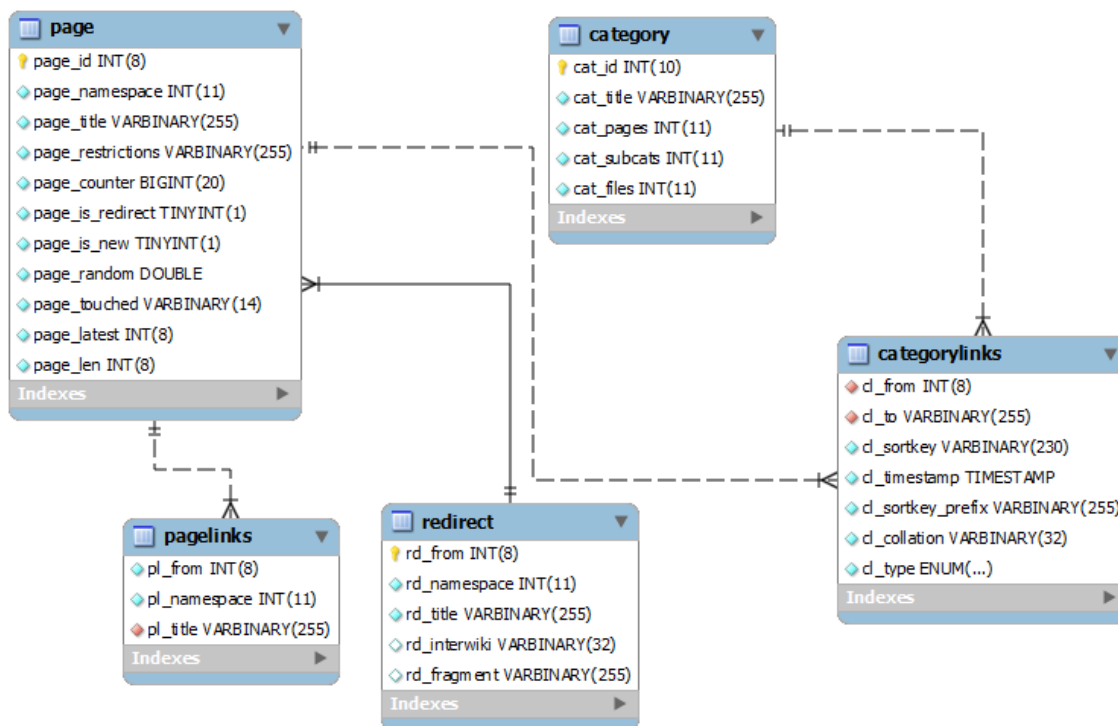


Figura 21 Database Diagrami – “Wikipedia dump”

Për shkak se tabelat e ofruara nga Wikipedia përmbajnë një numër të madh të fushave dhe detajeve, ndërsa ne në këtë projekt do ti përdorim ato të cilat janë të nevojshme për rastin tonë.

Pasi që të dhënat që ofron Wikipedia janë të formatit të databasave relacionare, për shkak se migrimi i tyre në mënyrë direkte në formatin OWL/RDF nuk është e mundur ne më poshtë do të paraqesim edhe modelin e mappimit të të dhënave nga ky format në atë OWL/RDF.

### 4.3.2 D2R Mapimi – migrimi i të dhënave relacionare

Të dhënat të cilat i kemi në dispozicion janë të ruajtura në formatin SQL, konkretisht në MySQL. Procesi i mappimit fillon duke bërë konvertimin e këtyre të dhënave relacionare në ato të formatit RDF sipas një skeme të dëshiruar.

Në mënyrë që këto treshe RDF të krijohet duhet ndjekur procedurën e shkrimit të fajllit mappues të formatit .ttf në D2R Serverin [14], tek i cili fajll përshkruhen në detaje relacionet ndërmjet databazës relacionare dhe semantikës (OWL/RDF fajllit që do të gjenerohet).

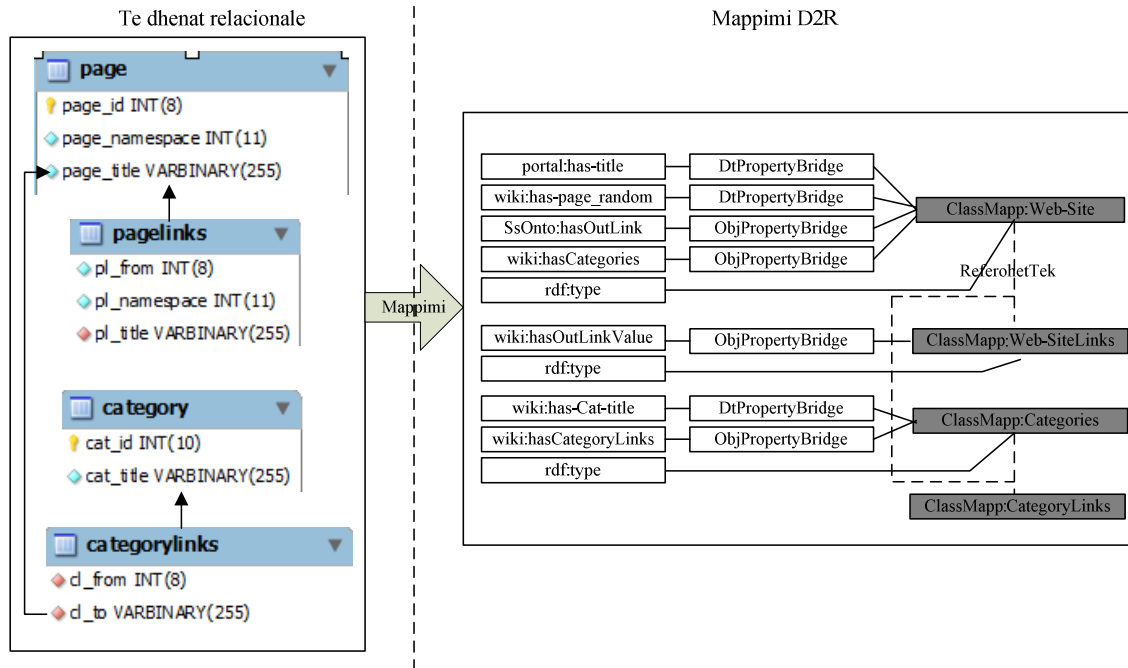


Figura 22 Procedura e mappimit të të dhënave relacionare në ontologji.

Në Figura 22 paraqitet procedura e mappimit të bazës së shënimeve në ontologjinë e dëshiruar, dhe kjo arrihet përmes fajllit mapues i cili krijohet sipas kërkesës së shfrytëzuesit dhe ekzekutohet në server.

Në Listimi 12 është paraqitur fajlli mapues, i ekzekutuar në D2R serverin.

| Listimi.12 | Fajlli mapues.  |
|------------|---|
| 1          |   |
| 2          | <code>#-- namespace configuration</code>  |
| 3          | <code>#-- path to the mapping file</code>   |
| 4          | <code>@prefix map: &lt;file:/C:/Users/gezim.sejdiu/Desktop/Wiki/D2RQ/d2rq-0.8.1/mapping-</code> |
| 5          | <code>wikipedia.n3#&gt; .</code>  |
| 6          | <code>@prefix vocab: &lt;vocab/&gt; .</code>  |
| 7          | <code>@prefix rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; .</code>                 |
| 8          | <code>@prefix rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; .</code>                      |
| 9          | <code>@prefix owl: &lt;http://www.w3.org/2002/07/owl#&gt; .</code>                              |
| 10         | <code>@prefix xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; .</code>                           |
| 11         | <code>@prefix d2rq: &lt;http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#&gt; .</code>        |
| 12         | <code>@prefix dc: &lt;http://purl.org/dc/elements/1.1/&gt; .</code>                             |
| 13         | <code>@prefix dcterms: &lt;http://purl.org/dc/terms/&gt; .</code>                               |
| 14         | <code>@prefix foaf: &lt;http://xmlns.com/foaf/0.1/&gt; .</code>                                 |
| 15         | <code>@prefix skos: &lt;http://www.w3.org/2004/02/skos/core#&gt; .</code>                       |
| 16         | <code>@prefix db: &lt;&gt; .</code>   |
| 17         | <code>@prefix wiki: &lt;#&gt; .</code>  |
| 18         | <code>@prefix SsOnto: &lt;http://localhost/Institute#&gt; .</code>                              |
| 19         | <code>@prefix portal: &lt;http://www.aktors.org/ontology/portal#&gt; .</code>                   |



```

20 @prefix jdbc: <http://d2rq.org/terms/jdbc/> .
21 #-- namespace configuration
22
23 #-- database connection parameters
24 map:database a d2rq:Database;
25     d2rq:jdbcDriver "com.mysql.jdbc.Driver";
26     d2rq:jdbcDSN "jdbc:mysql://localhost/wikipedia?autoReconnect=true";
27     d2rq:username "root";
28     d2rq:password "root";
29     jdbc:keepAlive "3600";# sends noop-query every 3600 seconds
30 #     jdbc:keepAliveQuery "SELECT 1";# optional custom noop-query
31 .
32 #-- database connection parameters
33
34     # Table page
35 map:Web-Site a d2rq:ClassMap;
36     d2rq:dataStorage map:database;
37     d2rq:uriPattern "http://localhost/Institute#Web-Site_@@page.page_id@";
38     d2rq:class portal:Web-Site;
39 .
40 map:page_title a d2rq:PropertyBridge;
41     d2rq:belongsToClassMap map:Web-Site;
42     d2rq:property portal:has-title;
43     d2rq:column "page.page_title";
44 .
45 map:page_random a d2rq:PropertyBridge;
46     d2rq:belongsToClassMap map:Web-Site;
47     d2rq:property wiki:has-page_random;
48     d2rq:column "page.page_random";
49 .
50 map:hasOutLink a d2rq:PropertyBridge;
51     d2rq:belongsToClassMap map:Web-Site;
52     d2rq:property SsOnto:hasOutLink;
53     d2rq:refersToClassMap map:Web-SiteLinks;
54     d2rq:condition "pagelinks.pl_from != 0";
55     d2rq:join "pagelinks.pl_title_ = page.page_title_";
56 .
57 map:hasOutLinkValue a d2rq:PropertyBridge;
58     d2rq:belongsToClassMap map:Web-SiteLinks;
59     d2rq:property wiki:hasOutLinkValue;
60     d2rq:refersToClassMap map:Web-Site;
61     d2rq:join "pagelinks.pl_title_ = page.page_title_";
62     d2rq:join "pagelinks.pl_from = page.page_id";
63 .
64 map:hasCategories a d2rq:PropertyBridge;
65     d2rq:belongsToClassMap map:Web-Site;
66     d2rq:property wiki:hasCategories;
67     d2rq:refersToClassMap map:CategoryLinks;
68     d2rq:join "categorylinks.cl_to_ = page.page_title_";
69 .
70     # Table pagelinks
71 map:Web-SiteLinks a d2rq:ClassMap;
72     d2rq:dataStorage map:database;
73     d2rq:uriPattern "http://localhost/Institute#Web-
74 SiteLinks_@@pagelinks.pl_from@";
75     d2rq:class wiki:Web-SiteLinks;
76 .
77     # Table category
78 map:Categories a d2rq:ClassMap;
79     d2rq:dataStorage map:database;
80     d2rq:uriPattern "http://localhost/Institute#Category_@@category.cat_id@";
81     d2rq:class wiki:Categories;

```

```

82      .
83      map:cat_title a d2rq:PropertyBridge;
84          d2rq:belongsToClassMap map:Categories;
85          d2rq:property wiki:has-Cat_title;
86          d2rq:column "category.cat_title";
87      .
88      map:hasCategoryLinks a d2rq:PropertyBridge;
89          d2rq:belongsToClassMap map:Categories;
90          d2rq:property wiki:hasCategoryLinks;
91          d2rq:refersToClassMap map:CategoryLinks;
92          d2rq:condition "categorylinks.cl_from != 0";
93          d2rq:join "categorylinks.cl_to=category.cat_title";
94      .
95      # Table categorylinks
96      map:CategoryLinks a d2rq:ClassMap;
97          d2rq:dataStorage map:database;
98          d2rq:uriPattern
99      "http://localhost/Institute#CategoryLinks_@@categorylinks.cl_from@";
100         d2rq:class wiki:CategoryLinks;
101

```

Siç mund të shihet nga Listimi 12 fajlli mapues është i ndarë në këto kategori:

- Kategoria e definimit të hapësirave, ku secila hapësirë punuese fillon me prefiks `@prefix` e pasuar nga shkurtesa e prefiksit dhe linku i hapësirës punuese, si p.sh:

| Listimi.13 | Inicializimi i prefiksave.   |
|------------|--|
| 1          |  |
| 2          | <code>@prefix d2rq: &lt;http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#&gt;</code> |
| 3          |  |

- Pastaj vjen kategoria e konfigurimit të serverit tek i cili referohet baza e shënimeve, duke përcaktuar se me cilin lloj drajveri është lidhur, adresën e DSN serverit, kredencialet e bazës së shënimeve dhe veti të tjera të paraqitura sipas nevojës.

| Listimi.14 | Konfigurimi i koneksionit  |
|------------|--|
| 1          |  |
| 2          | <code>map:database a d2rq:Database;</code>   |
| 3          | <code>    d2rq:jdbcDriver "com.mysql.jdbc.Driver";</code>                            |
| 4          | <code>    d2rq:jdbcDSN "jdbc:mysql://localhost/wikipedia?autoReconnect=true";</code> |
| 5          | <code>    d2rq:username "root";</code>   |
| 6          | <code>    d2rq:password "root";</code>   |
| 7          |  |

- Dhe pjesa e fundit përfshinë pjesën ku bëhet definimi i mapimit konkret të tabelave dhe kolonave nga baza e shënimeve në ontologjinë përkatëse. Më poshtë janë paraqitur disa nga mapimet të cilat përmbajnë veti të ndryshëm nga njëra tjetra.

| Listimi.15 | Mappimi i tabelave/kolonave.  |
|------------|---|
| 1          |   |
| 2          | <code># Mappimi dhe gjenerimi i klases Web-Site.</code>                             |
| 3          | <code>map:Web-Site a d2rq:ClassMap;</code>  |
| 4          | <code>    d2rq:dataStorage map:database;</code>                                     |
| 5          | <code>    d2rq:uriPattern "http://localhost/Institute#Web-</code>                   |
| 6          | <code>Site_@@page.page_id@";</code>   |
| 7          | <code>    d2rq:class portal:Web-Site;</code>  |
| 8          |   |
| 9          | <code># Mappimi i vetise hasOutLink, qe paraqet lidhjet e jashme te Web-Site</code> |

```

10 individit.
11 map:hasOutLink a d2rq:PropertyBridge;
12     d2rq:belongsToClassMap map:Web-Site;
13     d2rq:property SsOnto:hasOutLink;
14     d2rq:refersToClassMap map:Web-SiteLinks;
15     d2rq:condition "pagelinks.pl_from != 0";
16     d2rq:join "pagelinks.pl_title_ = page.page_title_";
17
18 # Mappimi i vetise hasOutLinkValue, qe paraqet nderlidhjen ndermjet
19 lidhjes se jashtme te Web-Site individit dhe Web-Site (si percaktues i
20 asaj lidhje).
21 map:hasOutLinkValue a d2rq:PropertyBridge;
22     d2rq:belongsToClassMap map:Web-SiteLinks;
23     d2rq:property wiki:hasOutLinkValue;
24     d2rq:refersToClassMap map:Web-Site;
25     d2rq:join "pagelinks.pl_title_ = page.page_title_";
26     d2rq:join "pagelinks.pl_from = page.page_id";
27

```

Siç mund të shihet nga Listimi 15 në pjesën e parë është paraqitur pjesa e krijimit të klasës Web-Site përmes mapimit të tabelës page që gjatë këtij procesi si uri link paraqitet id e fushës "[http://localhost/Institute#Web-Site\\_@@page.page\\_id](http://localhost/Institute#Web-Site_@@page.page_id)". Në këtë proces krijimi i klasës bëhet duke thirrur vetinë d2rq:ClassMap, ndërsa si burim mapues e përdor lidhjen e dhënë map:database.

Në të njëjtën mënyrë është bërë edhe krijimi i klasave:

- [wiki:Web-SiteLinks](#),
- [wiki:Categories](#),
- [wiki:CategoryLinks](#)

dhe i vetive të tyre.

Por ajo që duhet të veçohet gjatë këtij procesi është krijimi i vetive të cilat rrjedhin nga lidhshmëria e tabelave. D2R serveri e përkrahë krijimin e vetive nga lidhjet që ato kanë ndërmjet veti, siç mund të jetë përdorimi i d2rq:join, që edhe përcakton kushtet e relacionit.

Në Listimi 15 është paraqitur vetia SsOnto:hasOutLink, e cila përdor metodën d2rq:refersToClassMap, e cila përcakton tabelën tek e cila duhet të referohemi për të krijuar lidhjen. Ndërsa përcaktimi i kushtit bëhet përmes komandës "[pagelinks.pl\\_title\\_ = page.page\\_title\\_](#)". Përmes kësaj komande bëhet edhe gjenerimi i tresheve të cilat kanë lidhje të jashtme kundrejt faqeve.

Por gjithashtu gjatë këtij procesi është gjeneruar edhe një veti e cila përcakton lidhjen në nivelin tre të lidhshmërisë: duke mundësuar edhe identifikimin e lidhjes së jashtme, duke përdorur kushtet d2rq:join "[pagelinks.pl\\_title\\_ = page.page\\_title\\_](#)" dhe d2rq:join "[pagelinks.pl\\_from = page.page\\_id](#)", që edhe përcakton saktë se cila është lidhja e jashtme (e tipit Web-Site).

Pastaj pas ekzekutimit të fajllit mapues në D2R Server, serveri bën shfaqjen e ontologjisë duke mundësuar shfletimin e tresheve të mapuara.

Në Figura 23 është paraqitur realizimi i mapimit të klasës Web-Site dhe vetive të saja, por më në veqanti të vetisë hasOutLink, e cila paraqet lidhjet e jashtme që një faqe ka. Në këtë rast shihet që faqja me ID 16 (Basketball) ka dy lidhje të jashtme (faqen 1 dhe faqen 7938).

| Property            | Value   |
|---------------------|---|
| db:#has-page_random | 0.779231979962                                  |
| db:#has-title       | Basketball                                      |
| db:#hasOutLink      | <http://localhost:2020/resource/pagelinks/1>    |
| db:#hasOutLink      | <http://localhost:2020/resource/pagelinks/7938> |
| rdf.type            | wiki:page                                       |

Figura 23 Prezantimi i OutLinks.

Ndërsa nëse klikojmë mbi faqen 7938 ka këto lidhje (duke përfshi edhe faqen 16 që e zgjedhëm më parë), siç shihet në Figura 24.

| Property             | Value                                      |
|----------------------|--|
| is db:#hasOutLink of | <http://localhost:2020/resource/page/16>   |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/33>   |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/4536> |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/6861> |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/6875> |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/7315> |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/7553> |
| is db:#hasOutLink of | <http://localhost:2020/resource/page/7600> |
| rdf.type             | wiki:pagelinks                             |

Figura 24 Lidhjet e jashtme për faqen specifike.

#### 4.3.3 Importimi i të dhënave RDF

Derisa D2R serveri ka mundësinë e gjenerimit të tresheve në formate të caktuara, konkretisht RDF/XML, por edhe në .owl, prandaj gjenerimi i ontologjisë është bërë përmes përdorimit të D2R serverit.

Importimi është bërë përmes veglës Protege, por gjatë përdorimit të tresheve për manipulim me rregulla është specifikuar se gjenerimi i vetive nga ana e D2R serverit paraqet problem për shkak se nuk e përcakton se cilit lloj i takon vetia.

Gjenerimi i tresheve nga ana e D2R serverit është bërë në këtë mënyrë:

| Listimi.16 | Treshet e gjeneruara nga D2R serveri për dy veti.                             |
|------------|---|
| 1          |   |
| 2          | # Vetia e tipit Property.   |
| 3          | <rdf:Property rdf:about="http://localhost/Institute#hasOutLink">              |
| 4          | < rdfs:domain rdf:resource="http://www.aktors.org/ontology/portal#Web-Site"/> |
| 5          | </rdf:Property >  |
| 6          |   |
| 7          | # Vetia e tipit Property.   |
| 8          | <rdf:Property rdf:about="http://www.aktors.org/ontology/portal#has-title">    |
| 9          | < rdfs:domain rdf:resource="http://www.aktors.org/ontology/portal#Web-Site"/> |
| 10         | </rdf:Property >  |

Prandaj për evitim të këtij problemi është edituar përmes editorit Protege në .owl fijllin duke bërë konvertimin e vetive në llojin e dëshiruar nga `rdf:Property` në `owl:ObjectProperty` përkatësisht në `owl:DatatypeProperty`, siç shihet në Listimi 17

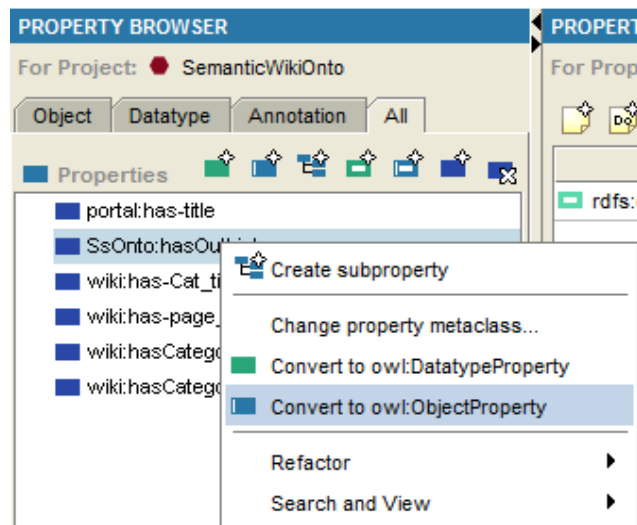
**Listimi.17** | **Korrigjimi i ontologjisë së gjeneruar për këto veti.**

```

1
2 # Vetia e tipit Object.
3 <owl:ObjectProperty rdf:about="http://localhost/Institute#hasOutLink">
4   <rdfs:domain rdf:resource="http://www.aktors.org/ontology/portal#Web-Site"/>
5 </owl:ObjectProperty>
6
7 # Vetia e tipit Datatype.
8 <owl:DatatypeProperty rdf:about="http://www.aktors.org/ontology/portal#has-
9 title">
10   <rdfs:domain rdf:resource="http://www.aktors.org/ontology/portal#Web-Site"/>
11 </owl:DatatypeProperty>
12

```

dhe e prezantuar përmes editorit në Figura 25.



**Figura 25** Konvertimi i llojit të vetisë përmes editorit Protege.

Pas këtyre korrigjimeve të nevojshme në ontologjinë e krijuar përmes mapimit të të dhënave, tani më është përgatitur terreni që kjo ontologji (së bashku me të gjitha të dhënat) të inkorporohet në ontologjinë `SemanticSearchOnto-v.1.0` të përdorur në [10,11].

Importimi tek kjo ontologji është bërë duke përdorur editorin Protege, duke shfrytëzuar mundësinë e importimit të ontologjive përmes veglës së paraqitur në Figura 26.

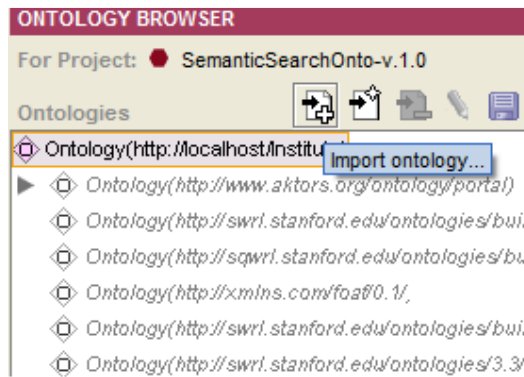


Figura 26 Importimi i ontologjisë SemanticWikiOnto tek SemanticSearchOnto.

Dhe pasi që të bëhet përcaktimi i fajllit .owl të cilin dëshirojmë ta importojmë, atëherë bëhet sinkronizimi i klasave, hierarkisë, instancave etj në mënyrë automatike nga vetë editorin (pasi që të jetë parapërcaktuar vendosja/lokacioni në hierarki tek fajlli mapues).

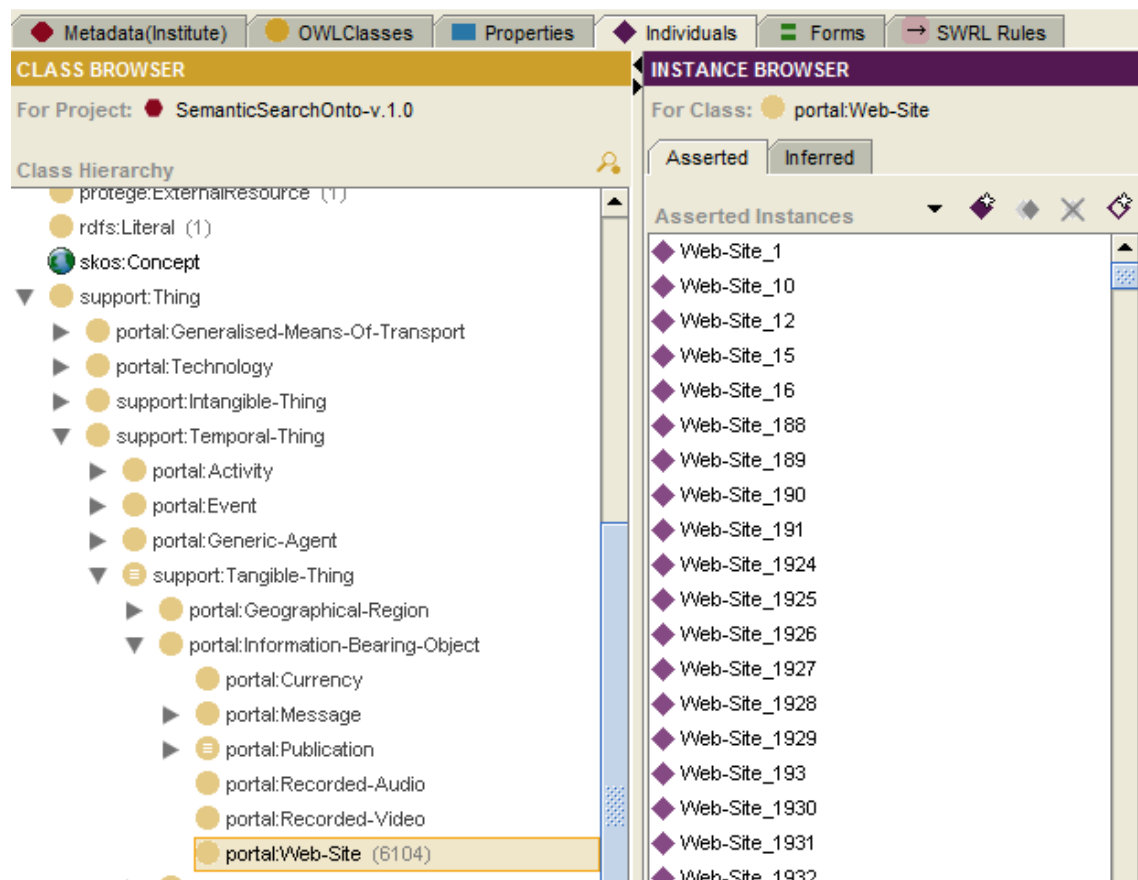


Figura 27 Pamje nga importimi i të dhënave.

Sic shihet nga Figura 27, klasa e krijuar përmes mapimit me D2RQ nuk është një klasë e re në këtë ontologji, por është sinkronizuar konkretisht nga tipi që e kemi përcaktuar, përkatësisht e tipit `portal:Web-Site`, dhe me këtë të gjitha instancat janë vendosur në atë klasë në mënyrë automatike.

## 5 Implementimi i rangimit semantik - Wikipedia si një rast studimi

Në këtë seksion do të paraqitet në detaje mënyra e kalkulimit të rangut semantik në mënyrë iterative si dhe prezantimi i rezultateve përmes një ueb aplikacioni.

### 5.1 Kalkulimi i rangut semantik në mënyrë iterative

Në punën paraprake [11] të prezentuar në Kapitulli 2 është arritur që të bëhet kalkulimi iterativ i rangut semantik, ku si sfidë ishte pamundësia që vlerat e prezantuara në Protege, të mbishkruhet apo të merren vlerat nga ato paraprake. Prandaj është punuar në krijimin e një metode ndihmëse e cila lexon nga fajlli i ontologjisë, përkatësisht nga formati RDF, duke bërë të mundur edhe zëvendësimin e vlerave të vetisë, të cilat do të përdoren për kalkulim të rangut semantik.

Kjo metodë e prezentuar në [11], bën të mundur që për çdo iteracion hap/lexon dhe mbyllë/ruan ontologjinë, në mënyrë që t'i mbishkruaj vlerat për përdorim në iteracionin e ardhshëm. Një proces i tillë është paraqitur në mënyrë vizuale në Figura 28.

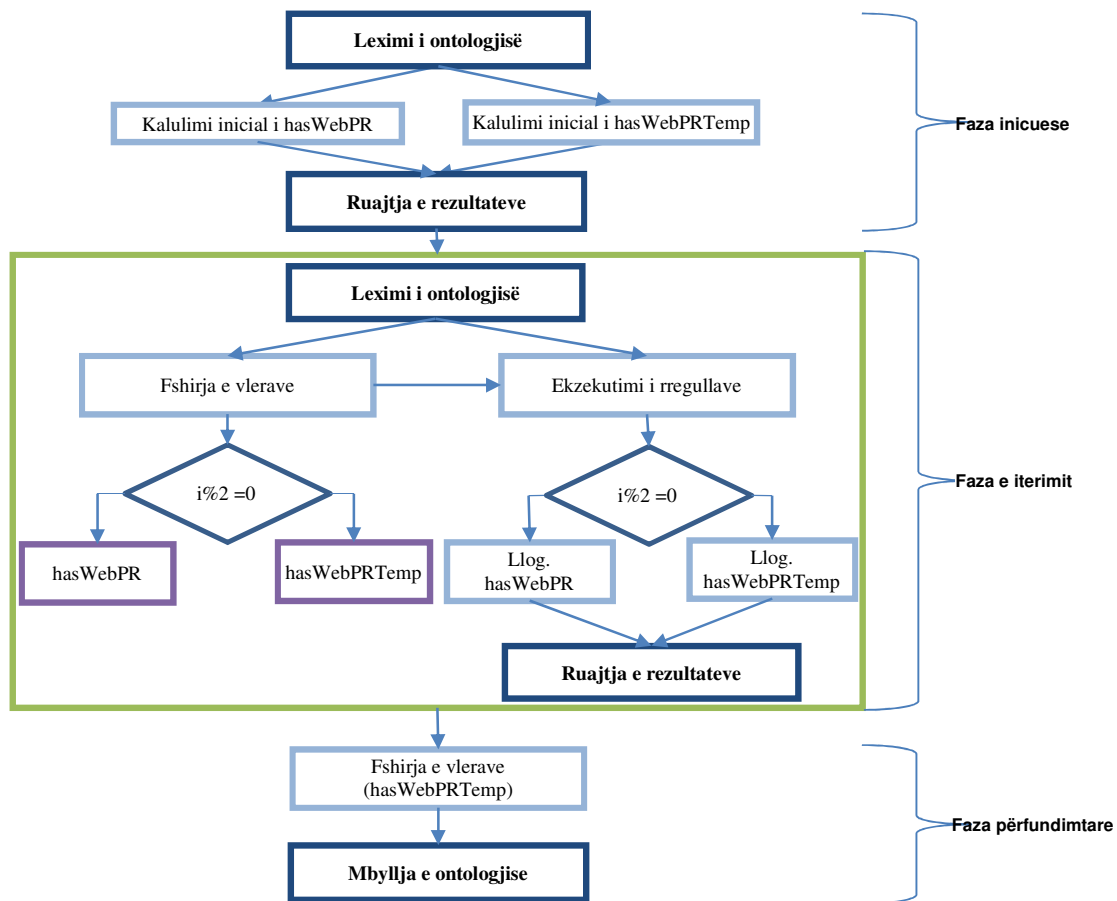


Figura 28 Metoda iterative e kalkulimit të rangimit semantik.

Në Figura 28 është paraqitur procesi i llogaritjes së rangimit semantik përmes metodës iterative. Kjo metodë në mënyrë iterative bën leximin dhe ruajtjen e vlerave në ontologji duke bërë procesin e hapjes dhe mbylljes së ontologjisë përkatësisht.

Procesi i manipulimit me ontologji është mundësuar përmes Jena API [31], ku edhe është mundësuar që përmes shfrytëzimit të klasave nga libraritë e saj të realizohet edhe kalkulimi i rangimit.

Klasat e përdorura nga kjo librari janë:

- `JenaOWLModel` përdoret për krijimin e modelit që përmban ontologjinë, varësisht nga adresa e ontologjisë.

**Listimi.18 | Krijimi i OWL modelit përmes klasës `JenaOWLModel`.**

```
1
2 JenaOWLModel owlModel =
3 ProtegeOWL.createJenaOWLModelFromURI(outputURI.toString());
4
```

- `SWRLFactory` dhe `SWRLRuleEngine` paraqesin klasat të cilat përdoren për implementimin e rregullave SWRL.

**Listimi.19 | Implementimi dhe ekzekutimi i SWRL- `SWRLFactory` dhe `SWRLRuleEngine`.**

```
1
2 SWRLFactory factory = new SWRLFactory(owlModel);
3 SWRLRuleEngine ruleEngine =
4 SWRLRuleEngineFactory.create(owlModel);
5
```

Procesi i kalkulimit të rangimit semantik (i paraqitur në Figura 28) është i ndarë në tri faza:

1. Faza iniciale – gjatë kësaj faze bëhet kalkulimi fillestar i vlerave `hasWebPR` dhe `hasWebPRTemp` duke shfrytëzuar rregullat SWRL nga Listimi 10, 11 dhe metodat nga Jena për realizimin e tyre.

**Listimi.20 | Ekzekutimi i rregullave për inicializim të `hasWebPR` dhe `hasWebPRTemp`.**

```
1
2 SWRLImp mainRule;
3 mainRule = factory.createImp("hasOutLink(?wj, ?wrel) ^
4 wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj) ^
5 totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"1/cj\", ?cj) ^
6 sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally, ?s) ^
7 swrlm:eval(?pri, \"(0.15 / n) + 0.85 * ally\", ?n, ?ally) -> hasWebPR(?wi,
8 ?pri)");
9 ruleEngine.infer();// execute rule and save values
10
11 factory.deleteImps();// delete all rules, created before.
12 mainRule = factory.createImp("hasWebPR(?wj, ?prj) ^ hasOutLink(?wj,
13 ?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj)
14 ^ totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj,
15 ?cj) ^ sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally,
16 ?s) ^ swrlm:eval(?pri, \"0.15 / n + 0.85 * ally\", ?n, ?ally) ->
17 hasWebPRTemp(?wi, ?pri)");
18 ruleEngine.infer();// execute rule and save values
19
```

Siç shihet në Listimi 20 janë deklaruar dy rregulla të cilat bëjnë llogaritjen e vlerave fillestare të `hasWebPR` dhe `hasWebPRTemp` përkatësisht (shih. rreshtat 3 dhe 12).



2. Faza e iterimit - gjatë kësaj faze bëhet kalkulimi i rankut duke shfrytëzuar vlerat e iteracionit paraprak, pra bëhet kalkulimi iterativ, si dhe bëhet përditësimi i vlerave në mënyrë që të shfrytëzohet në iteracionin e ardhshëm.

| Listimi.21 | Ekzekutimi i rregullave për kalkulimin e hasWebPR dhe hasWebPRTemp. |
|------------|---|
|------------|---|

```

1
2 while (i < iterations) {
3   if (i % 2 == 0) {
4     Remove("hasWebPR");
5   } else {
6     Remove("hasWebPRTemp");
7   }
8
9   if (i % 2 != 0) {
10    mainRule = factory.createImp("hasWebPR(?wj, ?prj) ^ hasOutLink(?wj,
11    ?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj)
12    ^ totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj,
13    ?cj) ^ sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^
14    sqwrl:sum(?ally, ?s) ^ swrlm:eval(?pri, \"0.15 / n + 0.85 * ally\", ?n,
15    ?ally) → hasWebPRTemp(?wi, ?pri)");
16  } else {
17    mainRule = factory.createImp("hasWebPRTemp(?wj, ?prj) ^ hasOutLink(?wj,
18    ?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj)
19    ^ totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj,
20    ?cj) ^ sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^
21    sqwrl:sum(?ally, ?s) ^ swrlm:eval(?pri, \"0.15 / n + 0.85 * ally\", ?n,
22    ?ally) → hasWebPR(?wi, ?pri)");
23  }
24  ruleEngine.infer();// execute rule and save values
25  }
26

```

Në Listimi 21 shihet që pas çdo iteracioni bëhet edhe përditësimi i vetive hasWebPR apo hasWebPRTemp, varësisht se në cilin mod të iterimit gjendet kalkulimi (pra cilën vlerë dëshiron që ta përdorë si vlerë pasuese), dhe këtu përdorë metodën ndihmëse Remove (shih Listimi 23), si dhe janë deklaruar dy rregulla të cilat përdorin vlerën e rangut nga iteracioni paraprak ?prj.

3. Faza përfundimtare – Në këtë fazë bëhet përmbyllja e procesit, pra përfundimit të iteracioneve, fshirjes së vlerës nga vetia hasWebPRTemp dhe ruajtjes së ontologjisë nga modeli (shih Listimi 22).

| Listimi.22 | Fshirja e vlerës hasWebPRTemp dhe ruajtja e ontologjisë. |
|------------|--|
|------------|--|

```

1
2 Remove("hasWebPRTemp");
3
4 owlModel.save(new File("SemanticSearchOnto-v.1.0.owl").toURI(),
5 FileUtils.langXMLAbbrev, errors);
6

```

Pasi që përmes rregullave SWRL të krijuara nuk është e mundur të përditësohen vlerat në ontologji, është paraqitur nevoja që të shfrytëzohet një metodë ndihmëse e cila mundëson zëvendësimin e vlerave tek vetia hasWebPR duke shfrytëzuar një veti ndihmëse hasWebPRTemp e cila ruan vlerën e iteracionit paraprak (shih Listimi 23).

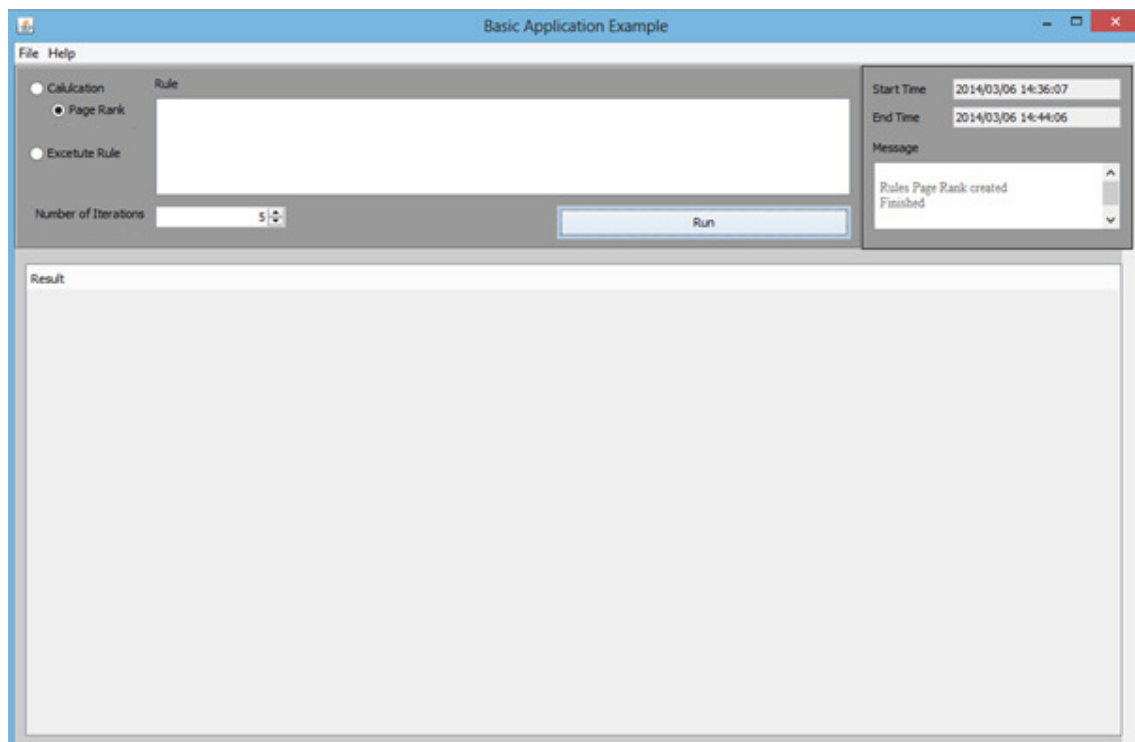
| Listimi.23 | Pjesë nga metoda për përditësim të vetive hasWebPRTemp dhe hasWebPR. |
|------------|--|
|------------|--|

```

1
2  while (true) {
3      textinLine = br.readLine();
4      if (textinLine == null) {
5          break;
6      } else if (textinLine.contains("<" + toReplace)) {
7          replacing = true;
8      } else if (replacing) {
9          if (textinLine.contains("</" + toReplace + ">")) {
10             replacing = false;
11             if (textinLine.contains("</rdf:Description >")) {
12                 sb.append("\n </rdf:Description>");
13             }
14         }
15     } else {
16         sb.append(textinLine + "\n");
17     }
18 }
19

```

Siç shihet në kodin e paraqitur tek Listimi 23, rreshti i lexuar nga ontologjia në formatin RDF bëhet mbyllja e tagut të tij duke mundësuar edhe fshirjen e vlerës së asaj vetie (shih.rresh.6 – 9), dhe në fund nëse është gjetur ajo veti bëhet edhe mbyllja e klasës `portal:Web-Site`, pasi që RDF sintaksa e saj është e taguar përmes `rdf:Description` (shih.rresh.11, 12).



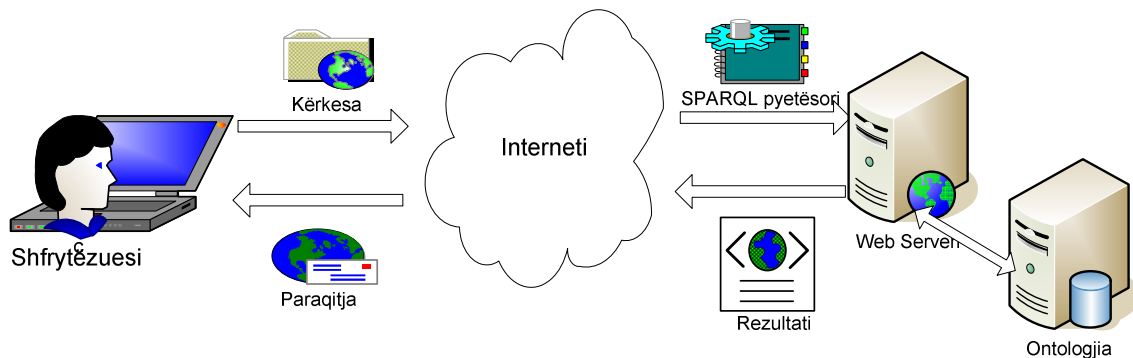
**Figura 29** Ndërfaqja e kalkulimit iterativ të rangut semantik [11].

Tek Figura 29 shihet ndërfaqja për iniciimin e kalkulimit të rangut semantik nga shfrytëzuesi, duke përzgjedhur paraprakisht metodën për kalkulim (në rastin tonë “Page Rank”), pastaj numrin e iteracioneve për kalkulim dhe në fund duke shtypur mbi butonin “Run” bëhet e mundur iniciimi i procesit të kalkulimit të rangut semantik, konkretisht procesit të paraqitur në Figura 28.

## 5.2 Aplikacioni SemanticSearchWikiApp

Në mënyrë që e gjithë puna e realizuar, duke filluar nga mapimi i bazës së shënimeve në ontologji, pastaj imkorporimi i asaj ontologjie tek ontologjia *SemanticSearchOnto*, si dhe procesi i kalkulimit të vlerave për rangim semantik të faqeve të jetë e disponueshme për shfrytëzuesit, gjatë punës së tezës është zhvilluar edhe një ueb aplikacion i quajtur *SemanticSearchWikiApp*.

Realizimi i kërkesave nga ana e shfrytëzuesit bëhet përmes përdorimit të pyetësorëve SPARQL duke përdorur teknologjinë e ashtuquajtur ARQ [32] që është një librari e vendosur tek Jena duke mundësuar ekzekutimin e pyetësorëve nga Jena që përmes Web Serverit bëhet bartja e kërkesave nga shfrytëzuesi tek ontologjia dhe pastaj përmes tij bëhet edhe përcjellja e rezultateve. Arkitektura e tillë është e paraqitur në Figura 30.



**Figura 30 Arkitektura e SemanticSearchWikiApp.**

Siç shihet nga Figura 30 shfrytëzuesi bën kërkesën përmes çfarëdo shfletuesi në ueb, dhe një kërkesë e tillë konvertohet në SPARQL pyetësor , që pastaj përmes serverit bëhet edhe qasja në ontologji duke ekzekutuar pyetësorin. Pastaj përsëri përmes serverit bëhet edhe gjenerimi i rezultateve, të cilat tek shfrytëzuesi shfaqet në formatin e lexueshëm për të.

### 5.2.1 Pyetësorët SPARQL në aplikacionin SemanticSearchWikiApp

Kërkesat e realizuara nga ana e shfrytëzuesit janë procesuar në formatin e pyetësorëve SPARQL duke përdorur teknologjinë ARQ e cila pastaj atë pyetësor e ekzekuton në server për nxjerrje të rezultateve.

Në listimin e mëposhtëm është paraqitur një shembull i pyetësorit SPARQL i përshtatshëm për editorin protege.

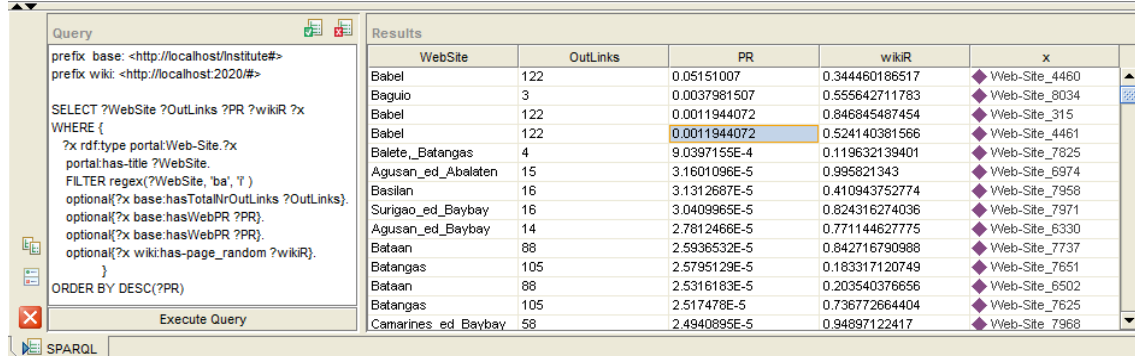
| Listimi.24 | Shembull i një pyetësoi SPARQL.                 |
|------------|---|
| 1          |   |
| 2          | prefix base: <http://localhost/Institute#>      |
| 3          | prefix wiki: <http://localhost:2020/#>          |
| 4          |   |
| 5          | SELECT ?WebSite ?OutLinks ?PR ?wikiR ?x         |
| 6          | WHERE {   |
| 7          | ?x rdf:type portal:Web-Site.                    |
| 8          | ?x portal:has-title ?WebSite.                   |
| 9          | FILTER regex(?WebSite, 'ba', 'i' )              |
| 10         | optional{?x base:hasTotalNrOutLinks ?OutLinks}. |
| 11         | optional{?x base:hasWebPR ?PR}.                 |
| 12         | optional{?x wiki:has-page_random ?wikiR}.       |

```

13     }
14 ORDER BY DESC(?PR)
15

```

Ndërsa tek Figura 31 është prezantuar pyetëtori dhe ekzekutimi i tij duke i paraqitur rezultatet e kthyer nga pyetëtori.



| WebSite             | OutLinks | PR           | wikiR          | x             |
|---------------------|----------|--------------|----------------|---------------|
| Babel               | 122      | 0.05151007   | 0.344460186517 | Web-Site_4460 |
| Baguio              | 3        | 0.0037981507 | 0.555642711783 | Web-Site_8034 |
| Babel               | 122      | 0.0011944072 | 0.846845487454 | Web-Site_315  |
| Babel               | 122      | 0.0011944072 | 0.524140381566 | Web-Site_4461 |
| Balete_Batangas     | 4        | 9.0397155E-4 | 0.119632139401 | Web-Site_7825 |
| Agusan_ed_Abajlaten | 15       | 3.1601096E-5 | 0.995821343    | Web-Site_6974 |
| Basilan             | 16       | 3.1312687E-5 | 0.410943752774 | Web-Site_7958 |
| Surigao_ed_Baybay   | 16       | 3.0409965E-5 | 0.824316274036 | Web-Site_7971 |
| Agusan_ed_Baybay    | 14       | 2.7812466E-5 | 0.771144627775 | Web-Site_6330 |
| Bataan              | 88       | 2.5936532E-5 | 0.842716790988 | Web-Site_7737 |
| Batangas            | 105      | 2.5795129E-5 | 0.183317120749 | Web-Site_7651 |
| Bataan              | 88       | 2.5316183E-5 | 0.203540376656 | Web-Site_6502 |
| Batangas            | 105      | 2.517478E-5  | 0.736772664404 | Web-Site_7625 |
| Camarines_ed_Baybay | 58       | 2.4940895E-5 | 0.94897122417  | Web-Site_7968 |

Figura 31 Ekzekutimi dhe rezultatet nga SPARQL pyetëtori.

Pyetëtori i përdorur gjatë kërkimit nga ana e shfrytëzuesit në aplikacion është paraqitur në Listimi 25.

**Listimi.25** | Pyetëtori SPARQL që përdoret gjatë kërkimit në aplikacion.

```

1
2 String dataprfx = "prefix base: <http://localhost/Institute#>";
3 String rdfprfx = "prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ";
4 String rdfsprfx = "prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> ";
5 String portalprfx = "prefix portal: <http://www.aktors.org/ontology/portal#> ";
6 String wikiprfx = "prefix wiki: <http://localhost:2020/#> ";
7
8 String QuerySearch = "";
9     if (request.getParameter("Search") != null) {
10         QuerySearch = request.getParameter("Search");
11     }
12
13 String inputQuery = dataprfx + rdfprfx + portalprfx + wikiprfx + ""
14 + "SELECT ?WebSite ?OutLinks ?PR ?wikiR ?x "
15 + "WHERE { "
16 + "?x rdf:type portal:Web-Site."
17 + "?x portal:has-title ?WebSite."
18 + "FILTER regex(?WebSite, '" + QuerySearch + "', 'i' ) "
19 + " optional{?x base:hasTotalNrOutLinks ?OutLinks}."
20 + " optional{?x base:hasWebPR ?PR}."
21 + " optional{?x wiki:has-page_random ?wikiR}."
22 + "}"
23 + "ORDER BY DESC(?PR) ";
24

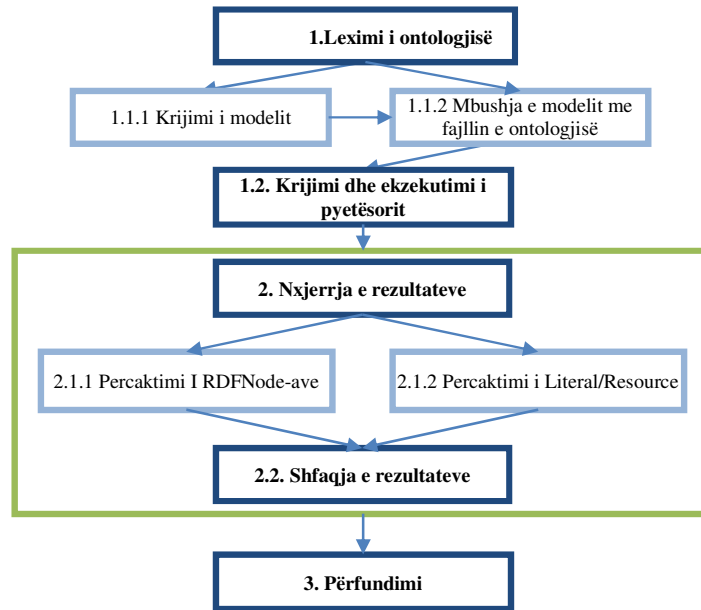
```

Siç shihet në Listimi 25 pyetëtori merr në konsideratë parametrin QuerySearch i cili paraqet fjalën kërkuese të cilën shfrytëzuesi e jep gjatë procesit të kërkimit, parametër i cili nga ana e serverit merret përmes qasjes request.getParameter("Search") (shih.rresh.10).

Pastaj ky parametër procedohet përmes pyetësorit duke bërë kërkimin e të gjitha faqeve ?WebSite që paraqesin titullin e faqes portal:has-title e cila i takon tipit rdf:type portal:Web-Site dhe që kanë përmbajtje nga ky parametër, që realizohet përmes funksionit FILTER regex(?WebSite, "'" + QuerySearch + "', 'i' ) (shih.rresh.18), ku parametri 'i' bën të mundur që fjala kërkuese të mos jetë e varur nga gjendja e shkronjave (të mëdha apo të vogla) (ang. case-sensitive) si dhe gjatë këtij pyetëtori merren edhe vlerat e lidhjeve të jashtme ?OutLinks,

rangun e një faqe `?PR` dhe numrin rangues që Wikipedia ia jep një faqe `?wikiR`. Të gjitha këto vlera janë opsionale nëse ekzistojnë merren, në të kundërtën nuk shfaqen, kjo përmes predikatit `optional`. Pastaj të gjithë rezultatit e kthyer nga parametrat e përshkruar më lartë renditet sipas rangimit të faqes në mënyrë rënëse, përmes `ORDER BY DESC (?PR)`.

Ekzekutimi i pyetësorit SPARQL është bërë përmes Jena, dhe një proces i tillë është paraqitur në Figura 32 në mënyrë të detajuar.



**Figura 32** Procesi i ekzekutimit të pyetësorit SPARQL.

Duke u mbështetur nga Figura 32 shihet që procesi i ekzekutimit të pyetësorit SPAQRL në Jena është i ndarë në 3 pjesë:

1. Modelimi i ontologjisë dhe krijimi i pyetësorit – gjatë kësaj pjese bëhet leximi i ontologjisë nga fajlli përmes `InputStream` (shih.rresh.4,9) dhe duke e vendosur atë në modelin `OntModel` të cilin e krijojmë gjatë këtij procesi (shih.rresh.2), pastaj në atë model bëhet krijimi i pyetësorit `inputQuery`, Jena pyetësorit `Query` dhe ekzekutuesit të atij pyetësori `QueryExecution` (shih.rresh.17,22) dhe vendosja e tyre në një bashkësi rezultatesh `ResultSet` (shih.rresh.25) në mënyrë që të manipulohet me elementet e vendosura aty, si dhe vendosjen e ndryshoreve në një listë `resultVars` (shih.rresh.26), siç shihet në Listimi 26.

**Listimi.26** | Modelimi i ontologjisë dhe SPARQL pyetësori.

```

1
2  OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM,
3  null);
4  InputStream in = FileManager.get().open(ONTOLOGY_URL);
5  if (in == null) {
6      throw new IllegalArgumentException("File: " + ONTOLOGY_URL + " not
7  found");
8  }
9  InputStream in_wiki = FileManager.get().open(ONTOLOGY_URL_wiki);
10 if (in_wiki == null) {

```

```

11     throw new IllegalArgumentException("File: " + ONTOLOGY_URL_wiki + "
12     not found");
13     model.read(in, "");
14     model.read(in_wiki, "");
15
16     // create a Jena query from the queryString
17     Query query = QueryFactory.create(inputQuery); // see as query example in
18     Listimi 25.
19
20     // create a Jena QueryExecution object that knows the query
21     // and the model over which the query will be run
22     QueryExecution qexec = QueryExecutionFactory.create(query, model);
23
24     // execute the query - get back a ResultSet
25     ResultSet resultSet = qexec.execSelect();
26     List resultVars = resultSet.getResultVars();
27

```

2. Nxjerrja dhe shfaqja e rezultateve – gjatë kësaj pjese bëhet iterimi nëpër të gjithë elementet e vendosura në bashkësinë e rezultateve `resultSet` (shih.rresh.3) dhe përcaktimi i nyjeve `RDFNode` nga ndryshoret e vendosura në listën e ndryshoreve `resultVars`, pastaj përcaktimi i literaleve apo resurseve nga nxjerrja e rezultateve dhe shfaqja e vlerave të tyre varësisht nga tipi i nyjës, siç shihet në Listimi 27.

| Listimi.27 | Nxjerrja dhe shfaqja e rezultateve nga SPARQL pyetësi. |
|------------|--|
|------------|--|

|    |   |
|----|---|
| 1  |   |
| 2  | <code>// iterate over the result set</code>   |
| 3  | <code>while (resultSet.hasNext()) {</code>  |
| 4  | <code>    QuerySolution querySolution = res</code>  |
| 5  | <code>    ultSet.nextSolution();</code>   |
| 6  | <code>    RDFNode nodeWebSite = querySolution.get((String) resultVars.get(0));</code>     |
| 7  | <code>    -----</code>  |
| 8  | <code>    RDFNode nodeUri = querySolution.get((String) resultVars.get(4));</code>         |
| 9  | <code>    Literal litWebSite = (Literal) nodeWebSite;</code>                              |
| 10 | <code>    ----</code>   |
| 11 | <code>    Resource resUri = (Resource) nodeUri;</code>                                    |
| 12 | <code>    -----</code>  |
| 13 | <code>    out.print("&lt;strong&gt;&lt;a href='http://wikipedia.org/wiki/' +</code>       |
| 14 | <code>    litWebSite.getString()+ " target='_blank'&gt;" + litWebSite.getString()+</code> |
| 15 | <code>    "&lt;/a&gt;&lt;/strong&gt;&lt;tr&gt;&lt;td&gt;");</code>                        |
| 16 | <code>    out.print("&lt;small&gt;" + nodeUri.isURIResource() +</code>                    |
| 17 | <code>    "&lt;/small&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;");</code>             |
| 18 |   |

3. Procesi i përfundimit të ekzekutimit të pyetësorit- pasi që të jetë përfunduar me sukses apo jo ekzekutimi i pyetësorit, ky proces përfundon gjithsesi përmes komandës `qexec.close()`.

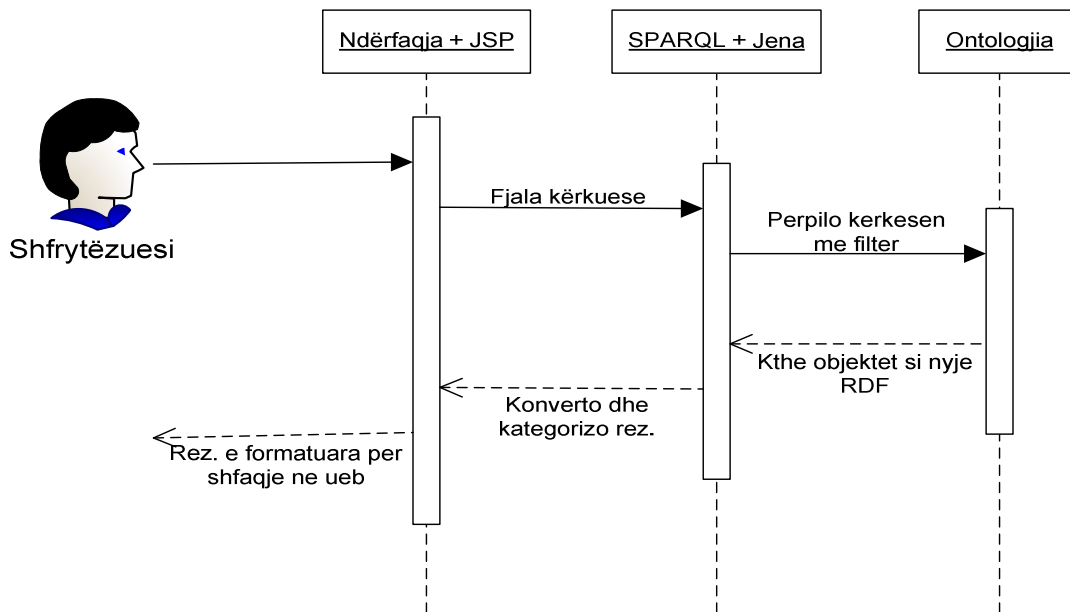
## 5.2.2 Ndërfaqja e SemanticSearchWikiApp

Ueb aplikacioni i zhvilluar është paraqitur në atë mënyrë që shfrytëzuesi ta ketë të lehtë përdorimin e tij. Kjo është mundësuar përmes zhvillimit të aplikacionit duke përdorur teknologjitë më të reja sa i përket uebit, konkretisht JSP, HTML dhe CSS.

Gjatë përdorimit të tij për qëllime testimi dhe të temës, aplikacioni është vendosur në ueb serverin e quajtur GlassFish [33].

Tek Figura 33 është paraqitur një diagram sekuencë i komplet procesit të kërkimit nga ana e shfrytëzuesit në aplikacion. Siç shihet në figurë shfrytëzuesi përmes uebit bën kërkimin duke vendosur

fjalën kyçe që ai dëshiron ta kërkoj, pastaj sekuenca kalon tek pyetësi SPARQL që së bashku me Jena bëjnë ekzekutimin e pyetësit duke e vendosur fjalën kyçe si filtër për kërkim. Pastaj pyetësi i ekzekutuar bën leximin e objekteve nga ontologjia dhe marrjen e atyre objekteve në formatin e nyjave RDF, pastaj përsëri përmes Jena bëhet strukturimi i nyjave dhe përmes html-së bëhet edhe shfaqja tek shfrytëzuesi në formatin që do të jetë i kuptueshëm për të, në këtë rast në formatin tabelar, për më shumë shikoni Figura 33 mbi rrjedhën e sekuençës për këtë proces.



**Figura 33 Diagrami i sekuençës për procesin e kërkimit në SemanticSearchWikiApp.**

Faqja kryesore e aplikacionit paraqet disa informata kryesore që i përkasin uebit semantik si dhe mundësinë e kërkimit direkt të ueb faqeve të vendosura në ontologji.

Në vijim janë paraqitur disa pamje nga ueb aplikacioni SemanticSearchWikiApp si dhe procedurat e përdorimit të tyre.

Në Figura 34 është paraqitur ndërfaqja e ueb aplikacionit që mundëson edhe kërkimin e faqeve nga ontologjia SemanticSearchOnto, duke lejuar shfrytëzuesit që të shënojnë një fjalë të çfarëdoshme në rubrikën për kërkim. Pastaj duke shfrytëzuar pyetësit SPARQL dhe Jena teknologjinë bëhet edhe kërkimi në ontologji. Pasi që kërkesa ekzekutohet, kthimi i rezultateve bëhet në formatin tabelar tek shfrytëzuesi, dhe listohen faqet të cilat e plotësojnë kushtin ( shih pyetësin nga Listimi 25).

Ky rezultat është prezantuar në Figura 35, në formë të katalogut, duke shfaqur listën e faqeve me të dhëna të rëndësishme për to, pastaj shfaq numrin total të faqeve të nxjerra nga kriteret e aplikuar, si dhe kohën e cila nevojitet për kthimin e këtyre rezultateve.



## The home of the Semantic Search Wiki

Search over Wikipedia dump pages, their links, calculated rank and original rank from Wikipedia, and more!

START SEARCHING  
Over thousands wiki pages



### SEMANTIC WEB

In addition to the classic "Web of documents" W3C is helping to build a technology stack to support a "Web of data," the sort of data you find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term "Semantic Web" refers to W3C's vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, OWL, and SKOS.

### Linked Data

The Semantic Web is a Web of data — of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF provides the foundation for publishing and linking your data. Various technologies allow you to embed data in documents (RDFa, GRDDL) or expose what you have in SQL databases, or make it available as RDF files.

[View more »](#)

### Vocabularies

At times it may be important or valuable to organize data. Using OWL (to build vocabularies, or "ontologies") and SKOS (for designing knowledge organization systems) it is possible to enrich data with additional meaning, which allows more people (and more machines) to do more with the data.

[View more »](#)

### Query

Query languages go hand-in-hand with databases. If the Semantic Web is viewed as a global database, then it is easy to understand why one would need a query language for that data. SPARQL is the query language for the Semantic Web.

[View more »](#)

Figura 34 Pamje nga faqja kryesore e aplikacionit SemanticSearchWikiApp.

Pasi që shfrytëzuesi vendos fjalën kyçe për kërkim (shih. Figura 34) atëherë sipas procesit të përshkruar në Figura 33 shihet se bëhet përpunimi i asaj kërkese dhe në fund bëhet kthimi i rezultatit të formatuar në formë tabelare, siç është paraqitur në katalogun e faqeve, konkretisht në Figura 35.



## Data Catalog

Search over catalog of Wiki Web-Sites.

[Home](#) / [Catalog](#)



2723 results found for "ba" (About 6850 ms).

#### Babel

[http://localhost/Institute#Web-Site\\_4480](http://localhost/Institute#Web-Site_4480)  
▼ 0.05151007 122 0.344460186517

#### Baq Semantic PageRank of Web-Site.

[http://localhost/Institute#Web-Site\\_8034](http://localhost/Institute#Web-Site_8034)  
▼ 0.0037981507 3 0.555642711783

#### Babel

[http://localhost/Institute#Web-Site\\_315](http://localhost/Institute#Web-Site_315)  
▼ 0.0011944072 122 0.846845487454

#### Babel

[http://localhost/Institute#Web-Site\\_4461](http://localhost/Institute#Web-Site_4461)  
▼ 0.0011944072 122 0.524140381566

#### Balete,\_Batangas

[http://localhost/Institute#Web-Site\\_7825](http://localhost/Institute#Web-Site_7825)  
▼ 9.0397155E-4 4 0.119632139401

Figura 35 Pamje nga katalogu i faqeve në SemanticSearchWikiApp.



---

Ajo që duhet cekur është që pastaj mundësia e kërkimit bëhet edhe tek katalogu i faqeve e jo vetëm në faqen kryesore. Në Figura 35 shihet se fjala kyçe e kërkimit është vendosur fjala “ba” dhe për këtë fjalë kemi 2723 rezultate të gjetura (duke ekzekutuar pyetësonin në Listimi 25) dhe renditja e tyre është bërë përmes vlerës së rangimit të faqes në mënyrë rënëse nga më e larta.

Kjo listë e paraqitur përmban informata mbi emrin e faqes, URI-në e faqes që gjendet në ontologji, pastaj vlerat numerike që përcaktojnë një faqe si rangimin semantik, si dhe numrin e lidhje të jashtëm dhe numrin rangues që Wikipedia ia jep faqes përkatëse.

Kërkimi në këtë ueb aplikacion bëhet në mënyrë semantike, jo duke marr për bazë vetëm fjalën kyçe që jepet si parametër, por edhe renditjen sipas rangut të faqes nga bashkësia e rezultateve të nxjerra nga ekzekutimi i pyetësonit.

## 6 Rezultatet dhe eksperimentet

Në këtë seksion do të paraqiten rezultatet dhe testimet e bëra në ontologjinë SemanticSearchOnto, e cila përmban një numër të konsiderueshëm faqesh nga baza e cunguar e Wikipedia-së (të ofruar nga Wikipedia) [35], më saktë aty gjenden 6104 ueb faqe dhe 2282 relacione të definuar ndërmjet faqeve. Në këtë ontologji ka edhe instanca të tjera të cilat në këtë eksperiment nuk do të jenë relevante të paraqiten.

Makina kompjuterike e cila është përdorur për kalkulimin e rregullave SWRL për llogaritjen e rangimit semantik (ku kalkulimi është simuluar për 5 iteracione siç shihet në Figura 29) dhe përpilimin e rezultateve dhe testimet posedon këto performanca: Processor: **Intel(R) Core(TM) i5 CPU M 560 @ 2.67GHz, 2667 Mhz, 2 Core(s), 4 Logical Processor(s)** dhe Installed Physical Memory (RAM): **4.00 GB**.

Para se të paraqesim ndonjë rezultat më parë të marrim një shembull që simulon kalkulimin e rangimit semantik për 5 iteracione.

### 6.1 Evaluimi kundrejt një bashkësie eksperimentale

Në Figura 36 është paraqitur relacioni ndërmjet faqeve (një shembull me tri instanca si një simulim i rastit të kalkulimit të rangimit semantik) duke paraqitur lidhjet e tyre.

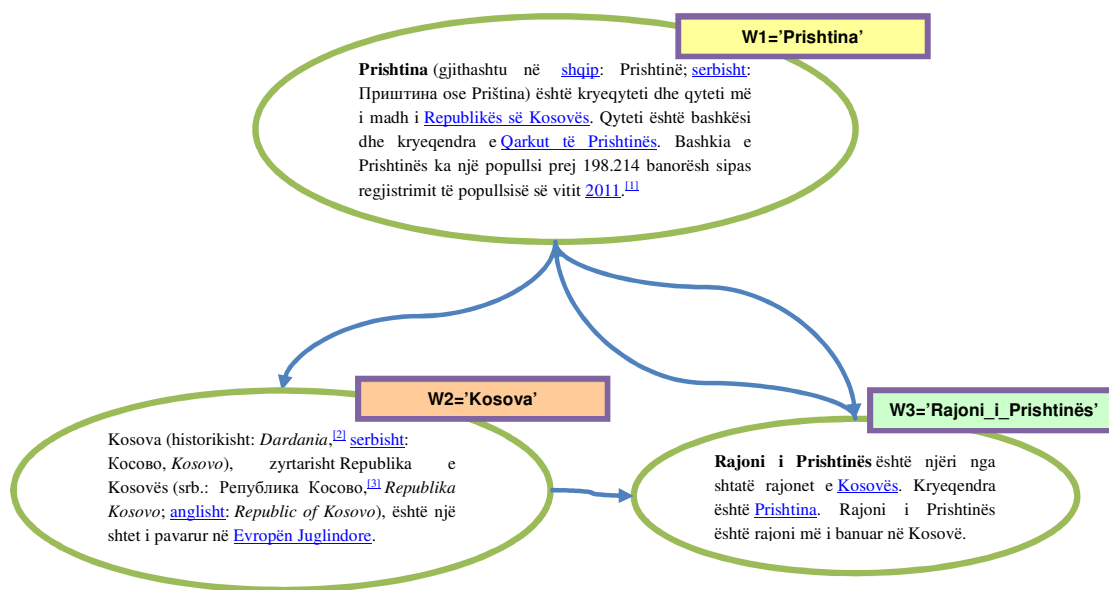


Figura 36 Relacionet ndërmjet faqeve (rasti me tri instanca).

Nëse aplikojmë ekuacionin (5) në shembullin tonë atëherë do të kishim:

$$PR(W3) = (1 - 0.85)/3 + 0.85 \left( \frac{PR(W1)}{2} + \frac{PR(W1)}{1} \right) \quad (7)$$

Kalkulimi i dhënë në ekuacionin (7) paraqet vetëm iteracionin e parë të kalkuluar për faqen W3. Duke u mbështetur në ekuacion shihet qartë se në peshën e faqes W3 marrin pjesë vetëm raporti i faqeve të cilat konsiderohen si lidhje të jashtme, kundrejt numrit total të lidhjeve të jashtëm që ato faqe kanë.

Ndërsa më poshtë është paraqitur edhe modaliteti i kalkulimit të rangimit semantik për faqet e dhëna në Figura 36.

Në këtë kalkulim kemi përdorur 5 iteracione dhe faktori *damping* 0.85. Vlerën fillestare të gjitha faqet e kanë 1.

Tabela 1 tregon raportin e lidhjeve ndërmjet faqeve të dhënë në Figura 36.

**Tabela 1 Raporti i lidhjeve të faqeve nga shembulli i dhënë në fig. 36.**

| Faqja | Lidhje të jashtëm: | PageRank-u fillestar |
|-------|--------------------|----------------------|
| W1    | W1                 | 1                    |
|       | W2                 |                      |
|       | W3                 |                      |
| W2    | W1                 | 1                    |
|       | W2                 |                      |
|       | W3                 |                      |
| W3    | W1                 | 1                    |
|       | W2                 |                      |
|       | W3                 |                      |

Faktori damping d  
0.85
Nr. Iteracioneve  
5

Procedura e mëposhtme jep detajet për kalkulimin e iteracionit të parë të rangut semantik nga shembulli paraprakë.

***PR(W1)***

$$\begin{aligned}
 PR(W1) &= 0.05 + 0.85(PR(W3)) \\
 &= 0.05 + 0.85 \\
 &= 0.9
 \end{aligned}$$

***PR(W1) = 0.9***

***PR(W2)***

$$\begin{aligned}
 PR(2) &= 0.05 + 0.85(PR(W1)/2) \\
 &= 0.05 + (0.85 * 0.5) \\
 &= 0.475
 \end{aligned}$$

***PR(W2) = 0.475***

***PR(W3)***

$$\begin{aligned}
 PR(W3) &= 0.05 + 0.85(PR(W1)/2 \\
 &\quad + PR(W2)) \\
 &= 0.05 + (0.85 * 1.5) \\
 &= 0.575
 \end{aligned}$$

***PR(W3) = 1.325***

Siç shihet nga elaborimi i kalkulimit të rangut në rastin e faqes W1 në kalkulim merr pjesë vetëm faqja W3, pasi që konsiderohet si lidhje e jashtme kundrejt faqes W1, dhe gjithashtu në këtë rast luan rol edhe raporti i lidhjeve të jashtëm nga faqja tek e cila referon faqja W1 (në rastin tonë nr. i lidhjeve të jashtëm të faqes W3 është 1). Dhe e njëjta procedurë ndiqet edhe tek faqet e tjera duke marruar pjesë ato faqe që konsiderohet të jashtëm kundrejt faqes që kalkulon rangun.

Pasi që është elaboruar edhe në formulimin matematik kalkulimi i iteracionit të parë të rangut semantik, në Tabela 2 është paraqitur edhe kalkulimi për të gjitha 5 iteracionit në mënyrë të detajuar (shih. Tabela 2)

**Tabela 2 Procedura e kalkulimit të PageRank-ut në mënyrë të detajuar.**

| Iteracioni              | W1           | W2           | W3           |
|-------------------------|--------------|--------------|--------------|
| 0                       | 1.000        | 1.000        | 1.000        |
| nga W1 në ..            | 0.050        | 0.425        | 0.425        |
| nga W2 në ..            | -            | 0.050        | 0.850        |
| nga W3 në ..            | 0.850        | -            | 0.050        |
| <b>PageRank Vlerat:</b> | <b>0.900</b> | <b>0.475</b> | <b>1.325</b> |
| 1                       | 0.900        | 0.475        | 1.325        |

|                         |              |              |              |
|-------------------------|--------------|--------------|--------------|
| nga W1 në ..            | 0.050        | 0.383        | 0.383        |
| nga W2 në ..            | -            | 0.050        | 0.404        |
| nga W3 në ..            | 1.126        | -            | 0.050        |
| <b>PageRank Vlerat:</b> | <b>1.176</b> | <b>0.433</b> | <b>0.836</b> |
| 2                       | 1.176        | 0.433        | 0.836        |
| nga W1 në ..            | 0.050        | 0.500        | 0.500        |
| nga W2 në ..            | -            | 0.050        | 0.368        |
| nga W3 në ..            | 0.711        | -            | 0.050        |
| <b>PageRank Vlerat:</b> | <b>0.761</b> | <b>0.550</b> | <b>0.918</b> |
| 3                       | 0.761        | 0.550        | 0.918        |
| nga W1 në ..            | 0.050        | 0.323        | 0.323        |
| nga W2 në ..            | -            | 0.050        | 0.467        |
| nga W3 në ..            | 0.780        | -            | 0.050        |
| <b>PageRank Vlerat:</b> | <b>0.830</b> | <b>0.373</b> | <b>0.841</b> |
| 4                       | 0.830        | 0.373        | 0.841        |
| nga W1 në ..            | 0.050        | 0.353        | 0.353        |
| nga W2 në ..            | -            | 0.050        | 0.317        |
| nga W3 në ..            | 0.715        | -            | 0.050        |
| <b>PageRank Vlerat:</b> | <b>0.765</b> | <b>0.403</b> | <b>0.720</b> |
| 5                       | 0.765        | 0.403        | 0.720        |

Siç shihet në Tabela 2 procedura e kalkulimit merr parasysh faktorin *damping* (1-d), dhe gjithashtu lidhjet e jashtëm që i ka faqja. Procedura e tillë me formula është paraqitur në Tabela 3.

**Tabela 3 Formulatat për kalkulimin e PageRank-ut.**

| Iteracioni              | W1                   | W2                         | W3                       |
|-------------------------|----------------------|----------------------------|--------------------------|
| 0                       | 1.000                | 1.000                      | 1.000                    |
| nga W1 në ..            | $(1-d)/n$            | $PR(W1)*d/2$               | -                        |
| nga W2 në ..            | -                    | $(1-d)/n$                  | $d*PR(W2)$               |
| nga W3 në ..            | $d*PR(W3)$           | -                          | $(1-d)/n$                |
| <b>PageRank Vlerat:</b> | $(1-d)/n + d*PR(W3)$ | $PR(W1)*d/2 + (1-d)/n + 0$ | $0 + d*PR(W2) + (1-d)/n$ |

Në Tabela 4 paraqiten rezultatet e PageRank-ut të dhënë në mënyrë të detajuar.

**Tabela 4 Rezultati i detajuar i PageRank-ut për 5 iteracione.**

| Iterationi | W1    | W2    | W3    |
|------------|-------|-------|-------|
| 0          | 1.000 | 1.000 | 1.000 |
| 1          | 0.900 | 0.475 | 1.325 |
| 2          | 1.176 | 0.433 | 0.836 |
| 3          | 0.761 | 0.550 | 0.918 |
| 4          | 0.830 | 0.373 | 0.841 |
| 5          | 0.765 | 0.403 | 0.720 |
| Iterationi | W1    | W2    | W3    |

Gjithashtu evaluimi i faqeve është dhënë edhe përmes grafikës (Figura 37), që përmes saj shihet që në iteracionin e 5 rangu i faqes W1 është më i madhi dhe konsiderohet që faqja W1 është më e rëndësishme,

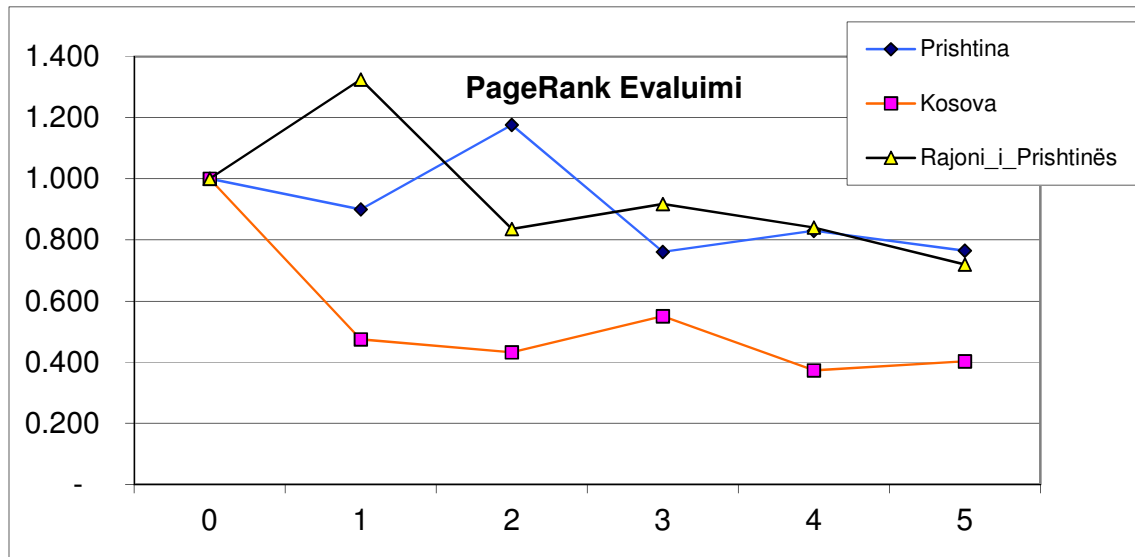


Figura 37 PageRank evaluimi (shembulli me 3 instanca).

dhe përcillet kjo kronologji ndërmjet faqeve (ekuacioni (8)).

$$PR(W1) > PR(W3) > PR(W2) \quad (8)$$

Në Tabela 5 është paraqitur rezultati final i kalkulimit të page rankut duke përdorur shembullin e dhënë në Figura 36.

| Faqja                          | W1    | W2    | W3    |
|--------------------------------|-------|-------|-------|
| PageRank-u inicial             | 0.900 | 0.475 | 1.325 |
| PgeRank-u final (5 iteracione) | 0.765 | 0.403 | 0.720 |

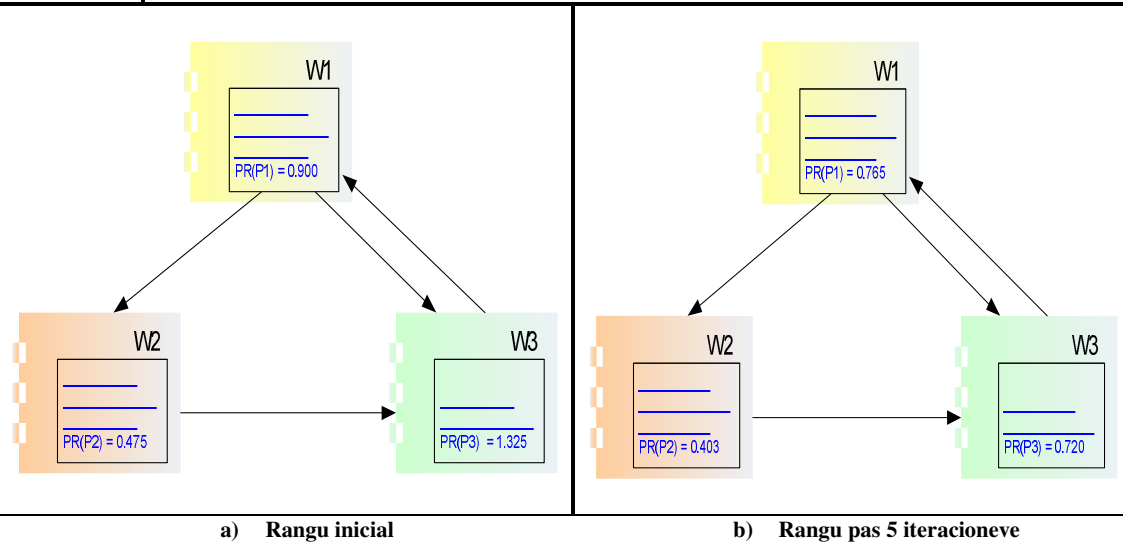
Ndërsa për kalkulimin semantik të këtyre vlerave kemi përdorur rregullat SWRL (të paraqitura në seksion të veçantë, konkretisht rregullat Rule-3 dhe Rule-4), gjë që edhe konfirmon se kalkulimi i bërë përmes këtyre rregullave është i saktë (shih rezultatet e paraqitura në Figura 38, duke krahasuar vlerën në iteracion 1 dhe atë përfundimtarë në iteracion e 5-të tek Tabela 5).

| Query   | Results  |          |                 |            |                 |          |            |          |   |     |           |            |        |   |       |            |            |                    |   |       |           |
|---|--|----------|-----------------|------------|-----------------|----------|------------|----------|---|-----|-----------|------------|--------|---|-------|------------|------------|--------------------|---|-------|-----------|
| <pre> prefix base: &lt;http://localhost/Institute#&gt; prefix wiki: &lt;http://localhost:2020/#&gt;  SELECT ?x ?WebSite ?OutLinks ?InitialPageRank ?PageRank WHERE {   ?x rdf:type portal:Web-Site.?x   portal:has-title ?WebSite.   FILTER regex(?WebSite, " ", "\r")   optional{?x base:hasTotalNrOutLinks ?OutLinks}.   optional{?x base:hasPR ?InitialPageRank}.   optional{?x base:hasWebPR ?PageRank}. } </pre> | <table border="1"> <thead> <tr> <th>x</th> <th>WebSite</th> <th>OutLinks</th> <th>InitialPageRank</th> <th>PageRank</th> </tr> </thead> <tbody> <tr> <td>Web-Site_1</td> <td>Pristina</td> <td>2</td> <td>0.9</td> <td>0.7646508</td> </tr> <tr> <td>Web-Site_2</td> <td>Kosova</td> <td>1</td> <td>0.475</td> <td>0.40270817</td> </tr> <tr> <td>Web-Site_3</td> <td>Rajoni_i_Pristinës</td> <td>1</td> <td>1.325</td> <td>0.7200517</td> </tr> </tbody> </table> | x        | WebSite         | OutLinks   | InitialPageRank | PageRank | Web-Site_1 | Pristina | 2 | 0.9 | 0.7646508 | Web-Site_2 | Kosova | 1 | 0.475 | 0.40270817 | Web-Site_3 | Rajoni_i_Pristinës | 1 | 1.325 | 0.7200517 |
| x   | WebSite  | OutLinks | InitialPageRank | PageRank   |                 |          |            |          |   |     |           |            |        |   |       |            |            |                    |   |       |           |
| Web-Site_1  | Pristina   | 2        | 0.9             | 0.7646508  |                 |          |            |          |   |     |           |            |        |   |       |            |            |                    |   |       |           |
| Web-Site_2  | Kosova   | 1        | 0.475           | 0.40270817 |                 |          |            |          |   |     |           |            |        |   |       |            |            |                    |   |       |           |
| Web-Site_3  | Rajoni_i_Pristinës   | 1        | 1.325           | 0.7200517  |                 |          |            |          |   |     |           |            |        |   |       |            |            |                    |   |       |           |

Figura 38 Rangu semantik inicial i dhe ai pas 5 iteracioneve nga shembulli ne Figura 36.

Siç shihet nga Tabela 5 dhe kronologjia e renditjes së rangut për faqet e dhëna, ato faqe që kanë më shumë lidhje të jashtëm, dhe gjithashtu lidhjet e jashtëm të tyre të kenë lidhje të jashtëm, rangui i faqes burimore është më i lartë në krahasim me të tjerat.

---

**Listimi.28** Gjendja e rangimit të faqeve në fillim dhe pas 5 iteracioneve.

Listimi 28 prezanton gjendjen iniciale (nën a) të rangut të faqeve, dhe gjendjen e rangut të arritur pas 5 iterimeve të kalkulimit të paraqitur nën b).

## 6.2 Evaluimi i rezultateve të rangimit kundrejt shënimeve të “Wikipedia dump”

Në vazhdim do të paraqiten rezultatet e rangimit të faqeve të Wikipedia-së, e cila është mapuar dhe përshtatur në modelin OWL sipas ontologjisë *SemanticSearchOnto* (Kapitulli 4). Rezultatet e prezantuara paraqesin rangimin semantik të shkallëzuar ndaj të dhënave reale të huazuara nga *Wikipedia dump*, duke iu kalkuluar rangimin semantik të tyre, si dhe duke modeluar relacionet ndërmjet faqeve.

Rezultatet janë nxjerrë duke përdorur pyetësorë SPARQL ndaj ontologjisë në fjalë.

Më poshtë tek Listimi 29 është paraqitur një pyetësor që liston faqet nga ontologjia, ku si fjalë kyçe gjatë kërkimit përdoret fjala kyçe “d”.

---

**Listimi.29** Pyetësori SPARQL për listimin e faqeve nga ontologjia të renditura sipas rangut të faqes.

```
1
2 prefix base: <http://localhost/Institute#>
3 prefix wiki: <http://localhost:2020/#>
4
5 SELECT ?WebSite ?OutLinks ?PR ?wikiR ?x
6 WHERE {
7   ?x rdf:type portal:Web-Site.
8   ?x portal:has-title ?WebSite.
9   FILTER regex(?WebSite, 'd', 'i' )
10  optional{?x base:hasTotalNrOutLinks ?OutLinks}.
11  optional{?x base:hasWebPR ?PR}.
12  optional{?x wiki:has-page_random ?wikiR}.
13 }
14 ORDER BY DESC(?PR)
15
```

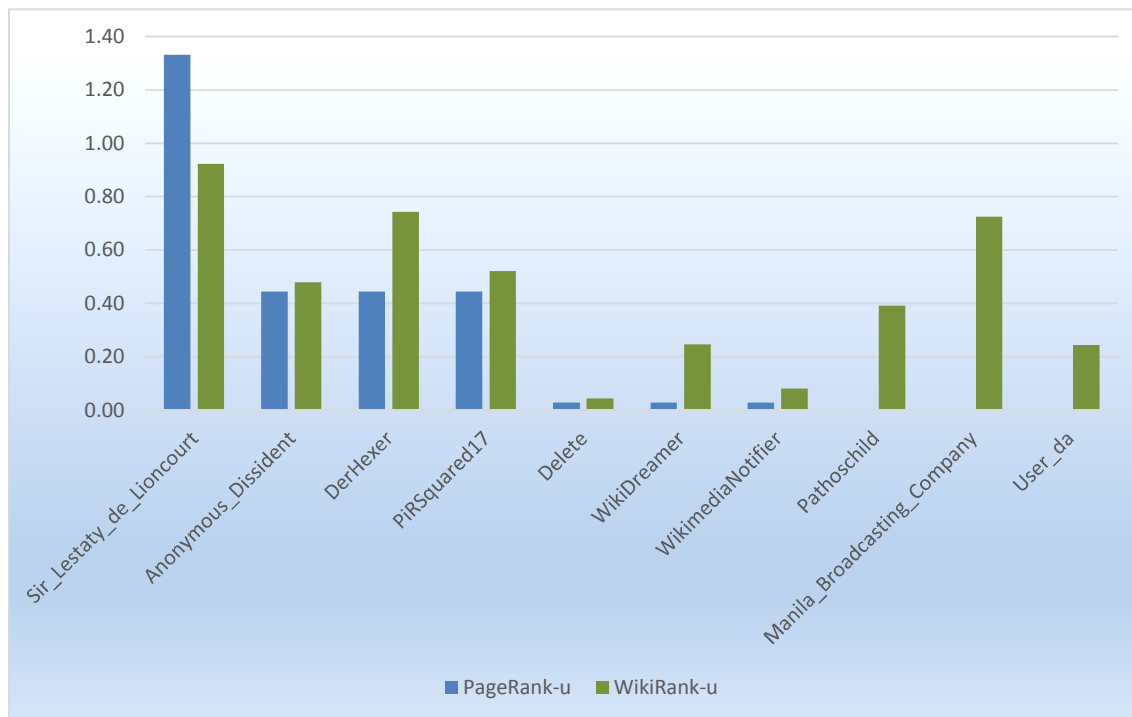
Renditja në rezultate është e bërë sipas vlerave të rangimit të faqeve në mënyrë rënëse nga më e larta.

Rezultati në formë tabelare i afektuar nga pyetësori i mësipërm është paraqitur në Tabela 6.

**Tabela 6 10 Web-Site në krye të gjeneruara nga SPARQL për fjalën kyçe "d".**

| #  | Web-Site-et    |                             | Vlerat numerike |                           |            |
|----|----------------|-----------------------------|-----------------|---------------------------|------------|
|    | #Web-Site      | Titulli                     | PageRank-u      | Nr. i lidhjeve të jashtëm | WikiRank-u |
| 1  | #Web-Site_5568 | Sir_Lestaty_de_Lioncourt    | 1.33120690      | 1                         | 0.92308330 |
| 2  | #Web-Site_6044 | Anonymous_Dissident         | 0.44379643      | 1                         | 0.47908174 |
| 3  | #Web-Site_5193 | DerHexer                    | 0.44379643      | 1                         | 0.74228692 |
| 4  | #Web-Site_7053 | PiRSquared17                | 0.44379643      | 1                         | 0.52065440 |
| 5  | #Web-Site_4701 | Delete                      | 0.02777373      | 2                         | 0.04369873 |
| 6  | #Web-Site_5738 | WikiDreamer                 | 0.02777373      | 2                         | 0.24669814 |
| 7  | #Web-Site_5791 | WikimediaNotifier           | 0.02777373      | 2                         | 0.08134166 |
| 8  | #Web-Site_5183 | Pathoschild                 | 0.00368612      | 3                         | 0.39164092 |
| 9  | #Web-Site_7834 | Manila_Broadcasting_Company | 0.00325000      | 3                         | 0.72492375 |
| 10 | #Web-Site_5451 | User_da                     | 0.00248443      | 7                         | 0.24381930 |

Rezultati i paraqitur në Tabela 6 është dhënë edhe në mënyrë grafike në Figura 39.



**Figura 39 Paraqitja grafike e 10 faqeve që prijnë sipas fjalës kyçe "d".**

Siç edhe mund të vërehet nga grafikoni, faqja "Sir\_Lestaty\_de\_Lioncourt" ka një vlerë ranguese më të lartë në krahasim me peshën që Wikipedia i ka vënë<sup>1</sup>. Ndërsa tek faqet e tjera (të këtij rezultati), rangimi semantik është i përafërt ose më i vogël sesa rangi i Wikipedia-së.

Por ajo që duhet theksuar në këtë rezultat është se, edhe nëse ke më shumë lidhje të jashtme nuk do të thotë domosdoshmërisht se ke rëndësi apo peshë më të lartë në rangim, përkundrazi, por peshë e faqet varet kryekëput nga peshë që kanë faqet të cilat konsiderohen si lidhje të jashtme, prandaj siç shihet në

<sup>1</sup> (WikiRank – diçka sikur Google Analytics që flet mbi popullaritetin e faqes Wiki).

Tabela 6 të rezultateve faqja “Sir\_Lestaty\_de\_Lioncourt “ me lidhje të jashtme 1 ka më shumë peshë sesa shembull faqja “User\_da” me 7 lidhje të jashtme.

Ndërsa me poshtë paraqitet raporti ndërmjet PageRank-ut të faqeve dhe peshës së faqes që Wikipedia ia ka vë (nga rezultati i pyetësorit në Listimi 29) në formë tabelare në Tabela 7.

**Tabela 7 Krahasimi i rezultateve përmes PageRank <-> WikiRank.**

| Të renditur sipas Rangut Semantik |                             |                | Të renditur sipas Wikipedia Rangut |                             |                |
|-----------------------------------|-----------------------------|----------------|------------------------------------|-----------------------------|----------------|
| #                                 | Titulli                     | PageRank-<br>u | #                                  | Titulli                     | WikiRank<br>-u |
| 1                                 | Sir_Lestaty_de_Lioncourt    | 1.33120690     | 1                                  | Sir_Lestaty_de_Lioncourt    | 0.92308330     |
| 2                                 | Anonymous_Dissident         | 0.44379643     | 3                                  | DerHexer                    | 0.74228692     |
| 3                                 | DerHexer                    | 0.44379643     | 9                                  | Manila_Broadcasting_Company | 0.72492375     |
| 4                                 | PiRSquared17                | 0.44379643     | 4                                  | PiRSquared17                | 0.52065440     |
| 5                                 | Delete                      | 0.02777373     | 2                                  | Anonymous_Dissident         | 0.47908174     |
| 6                                 | WikiDreamer                 | 0.02777373     | 8                                  | Pathoschild                 | 0.39164092     |
| 7                                 | WikimediaNotifier           | 0.02777373     | 6                                  | WikiDreamer                 | 0.24669814     |
| 8                                 | Pathoschild                 | 0.00368612     | 10                                 | User_da                     | 0.24381930     |
| 9                                 | Manila_Broadcasting_Company | 0.00325000     | 7                                  | WikimediaNotifier           | 0.08134166     |
| 10                                | User_da                     | 0.00248443     | 5                                  | Delete                      | 0.04369873     |

Siç mund të shihet nga Tabela 7, faqja “Sir\_Lestaty\_de\_Lioncourt” ruan pozicionin e saj në tabelë në të dy rastet e rangimit, pra edhe në rastin e rangimit semantik (PageRank-ut) është në pozitën e parë, por edhe përmes peshës së Wikipedia-së (WikiRank) është në pozitën e parë.

Pozitën e njëjte në renditje e ka ruajtur edhe faqja “PiRSquared17” e cila në të dy rastet vendoset në pozitën e katërt.

Ndërsa tek faqet e tjera të listuara renditja numerike ndryshon varësisht nga tipi i rangimit. Ky ndryshim është paraqitur edhe grafikisht në Figura 39.



---

## 7 Përfundimi dhe puna e ardhshme

Rangimi i faqeve luan një rol të rëndësishëm në vlerën dhe rëndësinë e faqes gjatë kërkimit nga ana e shfrytëzuesit. Ky lloj rangimi jo vetëm që ka gjetur zbatim në rëndësinë e informatës por edhe ka sjellë një konkurrencë të ndershme ndërmjet makinave kërkuese duke zgjeruar algoritmin e PageRank-ut me impakte të tjerë të cilët do të jenë të rëndësishëm, siç mund të jenë rangimi semantik i ueb faqeve, ndikimi i autorit në listimin e ranguar të faqes [11] por edhe ndikimi i rrjeteve sociale në rangimin e faqeve të kërkuara [10].

Përdorimi i rangimit semantik në listimin e faqeve ka mundësuar që për domena të ndryshëm, mënyra e kalkulimi të PageRankut varet nga metoda e përcaktimit të ndarjes së lidhjeve të jashtëm, siç kemi propozuar në mënyrën e lidhjeve të jashtëm që një faqe në Wikipedia i ka duke bërë krijimin e një lidhje të dyanshme për përcaktimin e asaj lidhje, pastaj përcaktimi i kësaj vetia na ka mundësuar që në mënyrë iterative të bëhet edhe kalkulimi i rangimit semantik të faqeve.

Në përfundimin tonë kemi arritur që të përcaktojmë se pesha e rangut të një faqe është drejtpërdrejt e lidhur me lidhjet e ndërsjella që faqja ka dhe peshën e tyre. Duke u bazuar në rezultatet e prezantuara shohim që faqja me më peshë është ajo e cila ka lidhje me faqet që kanë linqe të jashtëm me më peshë. Gjithashtu kemi treguar atë që duke shfrytëzuar peshën e lidhjeve të jashtëm që kanë faqet e Wikipedia-së, së bashku me rangimin, mundet të sjell rezultate të kënaqshme gjatë kërkimit.

Në të ardhmen kjo teknikë mund të zgjerohet edhe duke aplikuar ndonjë faktor tjetër përpos peshës së lidhjeve të faqeve që ka faqja e kërkuar, siç mund të jenë ndikimi i impakt faktorit të revistës tek e cila është referuar faqja, apo publikuar ajo në aplikimin e domenit të publikimeve, apo edhe peshën që ka autori i atij punimi apo faqeje (në rastin e rëndësisë së blloqeve individuale, forumeve të caktuara, postimeve në rrjete sociale, etj.), si dhe frekuencës që ka ai autor si një bashkë-autorë [10] duke e paraqitur krijuesin e faqes së Wikipedia-së si një instancë nga klasa `Person` e vokabularit `FOAF`.

---

## 8 Referencat

- [1]. Sergey Brin & Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine", In: *Seventh International World-Wide Web Conference (WWW 1998)*, April 14-18, Brisbane, Australia, 1998.
- [2]. Lawrence Page, Sergey Brin, Rajeev Motwani & Terry Winograd. "The pagerank citation ranking: Bringing order to the web". *Technical Report 1999-66, Stanford InfoLab*, November 1999.
- [3]. Arvind Arasu, Jasmine Novak, Andrew Tomkins & John Tomlin. "PageRank computation and the structure of the web: Experiments and algorithms". *Proceedings of the Eleventh International World Wide Web Conference*, Poster Track. Brisbane, Australia, pp. 107–117, 2002.
- [4]. Massimo Franceschet. "PageRank: Standing on the shoulders of giants". *ArXiv: 1002.2858 [cs.IR]*, 2010.
- [5]. Anne-Marie Vercoustre, James A. Thom, Jovan Pehceviski. "Entity Ranking in Wikipedia", *Dans the 23rd Annual ACM Symposium on Applied Computing*, 2008.
- [6]. Tim Berners-Lee, James Hendler, Ora Lassila. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities", *Scientific America*, pp: 34–43, 2001.
- [7]. Tim Berners-Lee. "Semantic Web - XML2000", <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>. (shikuar më 09.12.2013).
- [8]. Bijan Parsia & Peter F. Patel-Schneider. "Meaning and the Semantic Web", *ACM 1-58113-912-8/04/0005*, pp: 17-22, 2004.
- [9]. Lule Ahmedi, Lejla Abazi-Bexheti & Arbana Kadriu. "A Uniform Semantic Web Framework for Co-authorship Networks". *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, (pp. 958-965). Sydney, Australia 2011.
- [10]. Lule Ahmedi. "AuthorRank+FOAF: Ranking for Co-Authorship Networks on the Web". *ASONAM 2012*: 315-321.
- [11]. Lule Ahmedi, Lavdim Halilaj, Gëzim Sejdiu & Labinot Bajraktari. "Ranking Authors on the Web: A Semantic AuthorRank", *Lecture Notes in Social Networks, Springer*, 2014.
- [12]. Christian Bizer & Andy Seaborne. "D2rq: Treating non-rdf databases as virtual rdf graphs". *In 3rd International Semantic Web Conference, ISWC2004*, 2004.
- [13]. D2RQ documentation, <http://d2rq.org/> (shikuar më 15.02.2014).
- [14]. Christian Bizer & Richard Cyganiak. "D2R Server – Publishing Relational Databases on the Semantic Web", *In Poster at the 5th International Semantic Web Conference*, 2006.
- [15]. Tim Berners-Lee, Roy Fielding & Larry Masinter. "Uniform Resource Identifiers (URI): Generic Syntax". *The Internet Society, RFC 2396*, 1998.
- [16]. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler & François Yergeau. "Extensible Markup Language (XML) 1.0 (Fifth Edition)", *W3C Recommendation*, November 2008.
- [17]. Ding Ying, Fensel Dieter, Klein Michel & Omelayenko Borys. "The Semantic Web: Yet Another Hip?" *Data and Knowledge Engineering*, 2002.
- [18]. Graham Klyne & Jeremy J. Carroll eds. "Resource Description Framework (RDF): Concepts and Abstract Syntax". *W3C Recommendation*, February 2004.
- [19]. Dan Brickley & R. V. Guha eds. "RDF Vocabulary Description Language 1.0: RDF Schema". *W3C Recommendation*, February 2004.
- [20]. Deborah L. McGuinness & Frank van Harmelen eds. "OWL Web Ontology Language Overview." *W3C Recommendation, OWL Working Group*, 2004.
- [21]. Eric Prud'hommeaux & Andy Seaborne eds. "Sparql query language for RDF". *W3C working group*, October 3, 2009.
- [22]. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf & Mike Dean. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *W3C Member Submission 21 May 2004*.
- [23]. Bo Leuf & Ward Cunningham. "The Wiki Way: Quick Collaboration on the Web". *Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA*, 2001.
- [24]. Jimmy Wales. "Wikipedia and the free culture revolution", *OOPSLA/WikiSym Invited Talk*, Oct 2005.

- 
- [25]. Zoltan Gyongyi, Hector Garcia-Molina, & Jan Pedersen. "Combating web spam with trustrank". In *30th International Conference on Very Large Data Bases*, page 576587, 2004.
- [26]. Pai-Yen Chen, Hongqiang Xie, Sergei Maslov, & Sidney Redner. "Finding scientific gems with Google". Technical Report 0604130, arxiv/physics/, 2006.
- [27]. The AKT Reference Ontology. (2003), Advanced Knowledge Technologies, <http://www.aktors.org/publications/ontology/> (shikuar më 08.03.2014).
- [28]. Dan Brickley & Libby Miller. "FOAF Vocabulary Specification 0.98", *Namespace Document*, August 2010.
- [29]. Protégé Ontology Editor, Stanford University, <http://protege.stanford.edu/> (shikuar më 17.12.2013).
- [30]. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens & Chris Wroe. "A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0", *The University Of Manchester, Stanford University*, 2004.
- [31]. Apache Jena: A Semantic Web Framework, The Apache Software Foundation, <http://jena.apache.org/index.html> (shikuar më 08.03.2014).
- [32]. ARQ-A SPARQL Processor for Jena, <http://jena.apache.org/documentation/query/index.html> (shikuar më 08.03.2014).
- [33]. GlassFish Server, Java EE 7 Application Server, <https://glassfish.java.net/> (shikuar më 09.03.2014).
- [34]. SWRLJessTab, <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLJessTab> (shikuar më 07.01.2014).
- [35]. Wikimedia Downloads, <http://dumps.wikimedia.org/> (shikuar më 07.01.2014).

---

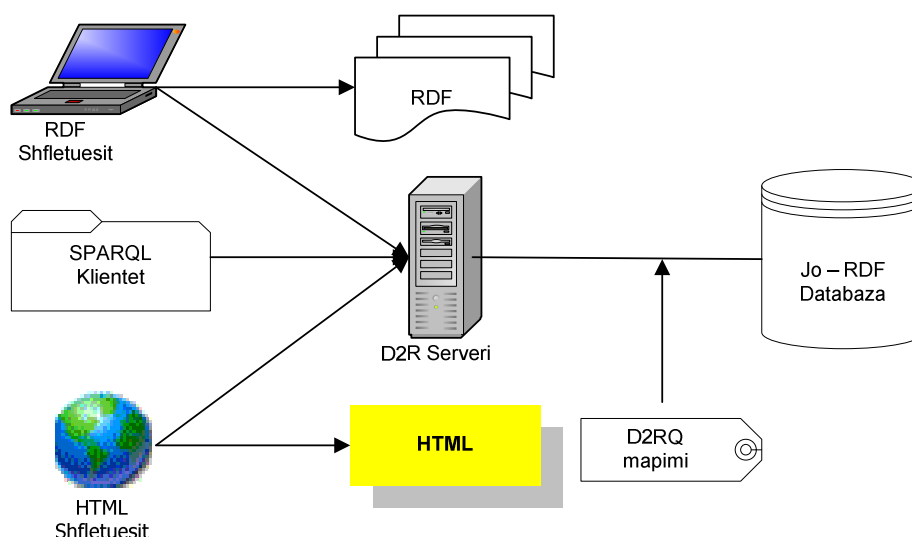
## 9 Shtojcat

### 9.1 Veglat e dobishme

Gjatë seksioneve të mëparshëm përshkruam në detaje disa nga rekomandimet e W3C-së të cilët janë zhvilluar për Uebin Semantik. Në mënyrë që uebi semantik të jetë i aplikuar, për këtë janë të nevojshme libraritë, dhe veglat të cilat bazohen në këto rekomandime dhe ju japin mundësi që zhvilluesit të mund të zhvillojnë aplikacione për Uebin Semantik.

Në këtë shtojcë do të jepet një përshkrim mbi veglat e përdorura për realizimin e këtij projekti. Duke filluar nga vegla dhe platforma e përdorur për migrim dhe konvertim të bazës së shënimeve relacionare tek ato të tipit RDF. Pastaj me një hyrje mbi veglën e cila manipulon me këtë lloj formati (pra RDF apo OWL), pastaj në fund me veglën për vizualizim dhe prezantim të rezultateve.

**D2R Serveri:** Pasi që uebi semantik është duke gjetur një zbatim të madh, atëherë është ardhur deri tek nevoja e përdorimit të të dhënave që nuk janë në formatin RDF (të dhëna relacionare). D2RQ [12, 13] është një gjuhë deklarative e cila përshkruan relacionin e mapimit ndërmjet skemave të të dhënave relacionare dhe ontologjive të tipit OWL/RDFS. Kjo platformë përdorë këtë procedurë të mapimit në mënyrë që të lejoj qasjen e të dhënave në formatin RDF për të dhëna që nuk janë RDF. Ndërsa D2R [14] server përdorë këtë gjuhë të mapimit, D2RQ, në mënyrë që të ekzekutoi mapimin.



**Figura 40 D2R- Serveri**

Përmes kësaj teknike specifikohet saktë se si janë të identifikuar resurset dhe si janë të gjeneruara vetitë e caktuara nga përmbajtja e bazës së të dhënave.

Objektet kryesore që shqyrtohen në D2RQ-në janë:

- *ClassMap* – paraqet një bashkësi entitetesh që përshkruhet përmes bazës së shënimeve, tek një klasë apo një grup klasash.
- *PropertyBridges* – specifikon saktë se si një resurse është krijuar, kjo mund të krijohet direkt nga vlera e bazës së shënimeve.

D2RQ përkrahë gjithashtu edhe mapimin me kushte duke mundësuar krijimin e entiteteve duke shfrytëzuar  $n:m$  relacione, si dhe lidhjen ndërmjet tabelave.

Përmes platëformës D2RQ mund të gjenerohet në mënyrë automatike struktura për mapimin e bazës së shënimeve nga tabelat, por ajo mundet edhe të optimizohet nga shfrytëzuesi sipas nevojave të tija.

---

Pastaj përmes këtij fajlli (që përmban instruksione për mapim), duke shfrytëzuar servern D2R bëhet gjenerimi i formatit RDF për çdo bazë të shënimeve, duke përdorur emrin e tabelave si emër të klasave në ontologji dhe emrat e kolonave si emra të vetive në ontologji.

Më poshtë do të jepen disa informata teknike rreth konfigurimit dhe ekzekutimit të D2R Serverit:

- Parakushtet e nevojshme për shfrytëzimin e D2R Server-t
  - o Instalimi i Java 1.5 ose version më i ri,
  - o Një databasë përkrahëse që do ta shfrytëzoi si drajver D2R Serveri. Ky server punon me Oracle, MySQL, PostgreSQL, Microsoft SQL Server, dhe secilën lloj të databasave që përkrahin SQL.
  - o Një shfletues të thjeshtë interneti (Internet Explorer, Chrome, Firefox, Opera ose Safari) në mënyrë që të përdoret shfletuesi i D2R Serverit (AJAX SPARQL Explorer).
- Në mënyrë që të shfrytëzohet D2R duhet që të shkarkohet libraritë nga faqja zyrtare, dhe pastaj pjesa e shkarkuar vendoset në një lokacion që ju e vendosni të përshtatshëm
- Shfrytëzimi i drajverit të bazës së shënimeve (duke vendosur fajllin JAR në direktoriumin e librarisë D2R). Nëse përdor MySQL dhe PostgreSQL janë të përfshira në D2R serverin, nuk është nevoja të merren.
- Hapi tjetër është krijimi/gjenerimi i fajllit mapues. D2R ofron mundësinë e shfrytëzimit të fajllit mapues i cili gjenerohet automatikisht, por në shumicën e kohës merret në konsideratë përditësimi i tij sipas nevojave.

Gjenerimi i këtij fajlli nëse bëhet në mënyrë automatike bëhet përmes komandës:

```
generate-mapping -o mapping-name.ttl -d driver.class.name  
-u db-user -p db-password jdbc:url:...
```

Por ky fajll bëhet zakonisht në mënyrë manuale sipas nevojave për ontologjinë.

Në mënyrë që ky fajll të bëhet i ekzekutueshëm duhet që përmes command prompt-it të ekzekutohet përmes komandës:

```
d2r-server mapping-name.ttl
```

Këtë ekzekutim duhet bërë në lokacionin tek i cili e keni vendos D2R serverin.

- Pasi që të jetë ekzekutuar kjo komandë, në mënyrë që të shihet rezultati i mapimit dhe çka është krijuar hapeni shfletuesin dhe adresoni tek `http://localhost:2020/`. Përmes këtij shfletuesi ju tani më mund të shikoni atë të cilën e keni mapu në fajll e cila lidhet me bazën e shënimeve dhe i paraqet të dhënat, por gjithashtu mund edhe të bëni në kohë reale kërkime specifike të të dhënave përmes SPARQL pyetësorëve.
- Dhe komanda e fundit por jo më pak e rëndësishme është gjenerimi i RDF/XML ose OWL ontologjisë dhe paraqitja e tresheve në fajllin .owl:

```
dump-rdf -f RDF/XML -b http://localhost:2020/ mapping-name.ttl  
> OutOnto.owl
```

Dhe pastaj kjo ontologji mund të hapet përmes veglave të përshtatshme për hapjen e tresheve RDF.

---

**Protégé:** Protégé [29] është një librari me kod të hapur e iniciuar si projekt nga Stanford Medical Informatics për menaxhimin dhe zhvillim e ontologjive [30]. Pjesa punuese e Protégé është e modeluar që të përkrah shtesa (ang. plug-ins), përmes së cilëve, varësisht prej domenit bëhet edhe shfrytëzimi i tyre.

Gjithashtu është edhe një platformë zhvilluese, tek e cila mund të bëhet modelimi dhe editimi i ontologjive, të cilat më pastaj mund të vendosen në tipa të ndryshëm si XML, OWL, RDF etj.

**Jena API:** Jena [31] është librari e cila ofron mundësinë që të zhvillohen aplikacione në Uebin Semantik. Duke shfrytëzuar libraritë e ofruara nga Jena, mundësohet që të thirren klasa përkatësisht interfejsa përmes gjuhës programuese JAVA për ontologji të tipave RDF, RDFS dhe OWL. Përdorimi i ontologjisë dhe modelit në ambientin e Jena-s mundësohet që të aplikohen edhe manipulime me ontologji sipas nevojës.

Në kuadër të Jena-së është edhe paradigma ARQ për ekzekutimin e pyetësorëve SPARQL.

Poashtu edhe Jena është një projekt me kodë të hapur që zhvillimi i saj ka filluar tek HP Labs Semantic Web.

## 9.2 Kodi burimor

Në këtë rubrikë janë paraqitur disa nga pjesët e kodit burimor të cilat janë kryciale për realizimin e punës së bërë në temë.

### 9.2.1 Pjesë kodi nga kalkulimi iterativ i PageRank-ut

Në Listimi 30 është paraqitur metoda në JAVA që bën kalkulimin iterativ të ueb faqeve në ontologji, duke lexuar nga ontologjia, dhe përmes rregullave SQWRL bën edhe ekzekutimin e këtij kalkulimi.

---

**Listimi.30** | ExecutePR(...) - Metoda për kalkulimin iterativë të rangut të ueb faqeve.

---

```
public void ExecutePR() throws SQWRLException, SWRLRuleEngineException,
InterruptedException, SWRLFactoryException, URISyntaxException {

    JTextStatus.setBackground(Color.white);

    int iterations = (Integer) spinnerIterations.getValue();

    if (iterations > 1) {
        iterations = iterations - 2;
    }
    if (iterations % 2 == 0) {
        iterations = iterations + 1;
    }
    try {

        try {

            URI outputURI = new URI(("file:/" + ONTOLOGY_URL.replaceAll(" ",
"%20")));
            JenaOWLModel owlModel =
ProtegeOWL.createJenaOWLModelFromURI(outputURI.toString());
            outputURI = new URI(("file:/" + Repository_Url.replaceAll(" ",
"%20")));
            LocalFolderRepository rep = new LocalFolderRepository(new
File(outputURI), true);
            owlModel.getRepositoryManager().addGlobalRepository(rep);
            outputURI = new URI(("file:/" + ONTOLOGY_URL.replaceAll(" ", "%20")));
```

```

SWRLFactory factory = new SWRLFactory(owlModel);
SWRLRuleEngine ruleEngine = SWRLRuleEngineFactory.create(owlModel);

String status = "Ontology Loaded";

factory.deleteImps();

SWRLImp mainRule;
mainRule = factory.createImp("hasOutLink(?wj, ?wrel) ^
wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj) ^ totalNrWeb-
Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"1/cj\", ?cj) ^ sqwrl:makeBag(?s, ?yj) ^
sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally, ?s) ^ swrlm:eval(?pri, \"(0.15 / n) + 0.85 *
ally\", ?n, ?ally) ^ hasWebPR(?wi, ?pri)");
ruleEngine.infer();

factory.deleteImps();
mainRule = factory.createImp("hasWebPR(?wj, ?prj) ^ hasOutLink(?wj,
?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj, ?cj) ^ totalNrWeb-
Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj, ?cj) ^ sqwrl:makeBag(?s, ?yj)
^ sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally, ?s) ^ swrlm:eval(?pri, \"0.15 / n + 0.85 *
ally\", ?n, ?ally) ^ hasWebPRTemp(?wi, ?pri)");
ruleEngine.infer();

System.out.println("Saving...");
Collection errors;

errors = new ArrayList();

owlModel.save(new File("SemanticSearchOnto-v.1.0.owl").toURI(),
FileUtils.langXMLAbbrev, errors);
System.out.println("File saved with " + errors.size() + " errors.");
int i = 0;
while (i < iterations) {

    if (i % 2 == 0) {
        Remove("hasWebPR");
    } else {
        Remove("hasWebPRTemp");
    }
    System.out.println("Iteracion" + (i + 3));

    owlModel.dispose();
    mainRule = null;
    owlModel = null;
    factory = null;
    ruleEngine = null;

    System.gc();
    java.lang.Runtime.getRuntime().freeMemory();
    java.lang.Runtime.getRuntime().gc();

    owlModel =
ProtegeOWL.createJenaOWLModelFromURI(outputURI.toString());
    owlModel.getRepositoryManager().addGlobalRepository(rep);

    factory = new SWRLFactory(owlModel);
    ruleEngine = SWRLRuleEngineFactory.create(owlModel);

    factory.deleteImps();
    if (i % 2 != 0) {
        mainRule = factory.createImp("hasWebPR(?wj, ?prj) ^
hasOutLink(?wj, ?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj,

```

```

?cj) ^ totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj, ?cj) ^
sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally, ?s) ^
swrlm:eval(?pri, \"0.15 / n + 0.85 * ally\", ?n, ?ally) ^ hasWebPRTemp(?wi, ?pri));
    } else {
        mainRule = factory.createImp("hasWebPRTemp(?wj, ?prj) ^
hasOutLink(?wj, ?wrel) ^ wiki:hasOutLinkValue(?wrel, ?wi) ^ hasTotalNrOutLinks(?wj,
?cj) ^ totalNrWeb-Sites(Constant1, ?n) ^ swrlm:eval(?yj, \"prj/cj\", ?prj, ?cj) ^
sqwrl:makeBag(?s, ?yj) ^ sqwrl:groupBy(?s, ?wi) ^ sqwrl:sum(?ally, ?s) ^
swrlm:eval(?pri, \"0.15 / n + 0.85 * ally\", ?n, ?ally) ^ hasWebPR(?wi, ?pri)");
    }

    ruleEngine.infer();

    errors = new ArrayList();
    owlModel.save(new File("SemanticSearchOnto-v.1.0.owl").toURI(),
FileUtils.langXMLAbbrev, errors);
    System.out.println("File saved with " + errors.size() + " errors.");
    i++;
}
Remove("hasWebPRTemp");

status = j textStatus.getText() + "\n" + " Rules Page Rank created";
j textStatus.setText(status);

status = j textStatus.getText() + "\n" + " Finished";
j textStatus.setText(status);

} catch (SWRLParseException ex) {
    j textStatus.setText(" Rule isn't defined in proper way");
    j textStatus.setBackground(Color.ORANGE);
}
} catch (OntologyLoadException ex) {
    j textStatus.setText(" Ontology couldn't found. Check Path.");
    j textStatus.setBackground(Color.ORANGE);
}
}
}
}

```

Ndërsa në Listimi 31 është paraqitur metoda e cila bën fshirjen/zëvendësimin e vlerave gjatë iterimeve, që varësisht nga iteracioni zëvendëson vlerën në vetinë e përcaktuar si parametër në metodë, toReplace .

### Listimi.31 Remove(...)-Metoda për zëvendësimin e vlerave duke shfrytëzuar një veti ndihmëse.

```

private void Remove(String toReplace) throws URISyntaxException {

    URI outputURI = new URI(("file:///\" + ONTOLOGY_URL.replaceAll(" ", \"%20\"));

    File f = new File(outputURI);

    FileInputStream fs = null;
    InputStreamReader in = null;
    BufferedReader br = null;

    StringBuffer sb = new StringBuffer();

    String textinLine;

    try {
        fs = new FileInputStream(f);
    }
}

```



```

in = new InputStreamReader(fs);
br = new BufferedReader(in);
Boolean replacing = false;
while (true) {
    textinLine = br.readLine();
    if (textinLine == null) {
        break;
    } else if (textinLine.contains("<" + toReplace)) {
        replacing = true;
    } else if (replacing) {
        if (textinLine.contains("</" + toReplace + ">")) {
            replacing = false;
            if (textinLine.contains("</rdf:Description>")) {
                sb.append("\n </rdf:Description>");
            }
        }
    } else {
        sb.append(textinLine + "\n");
    }
}

fs.close();
in.close();
br.close();

} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

}

try {
    FileWriter fstream = new FileWriter(f);
    BufferedWriter outobj = new BufferedWriter(fstream);
    outobj.write(sb.toString());
    outobj.close();

} catch (Exception e) {
    System.err.println("Error: " + e.getMessage());
}
}

```

## 9.2.2 Pjesë kodi nga ueb aplikacioni

Në Listimi 32 është paraqitur fajlli në JSP që përdorë paradigmen Jena + ARQ duke shënuar në të pyetësorët SPARQL për nxjerrje të rezultateve nga ontologjia dhe prezantimi i saj në HTML.

### Listimi.32 | JSP fajlli i cili bënë ekzekutimin e SPARQL – pyetësorit.

```

<!--
    Document   : Jena_Query
    Created on : Feb 26, 2014, 1:28:32 PM
    Author      : Gezimi
-->

<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<%@ page import="com.hp.hpl.jena.ontology.*" %>

```

```

<%@ page import="com.hp.hpl.jena.rdf.model.*" %>
<%@ page import="com.hp.hpl.jena.util.*" %>
<%@ page import="com.hp.hpl.jena.query.*" %>
<%@ page import="com.hp.hpl.jena.n3.IRIResolver" %>
<%@ page import="java.net.URI" %>
<%@ page import="edu.stanford.smi.protege.owl.jena.JenaOWLModel" %>
<%@ page import="edu.stanford.smi.protege.owl.ProtegeOWL" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    String ONTOLOGY_URL = "C:/Users/Gezimi/Desktop/New folder/SemanticSearchOnto-
v.1.0.owl";
    String ONTOLOGY_URL_wiki = "C:/Users/Gezimi/Desktop/New folder/wikipediaOnto.owl";
    String dataprfx = "prefix base: <http://localhost/Institute#>";
    String rdfprfx = "prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ";
    String rdfsprfx = "prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> ";
    String portalprfx = "prefix portal: <http://www.aktors.org/ontology/portal#> ";
    String wikiprfx = "prefix wiki: <http://localhost:2020/#> ";

    String QuerySearch = "";
    if (request.getParameter("Search") != null) {
        QuerySearch = request.getParameter("Search");
    }

    String inputQuery = dataprfx + rdfprfx + portalprfx + wikiprfx + ""
        + "SELECT ?WebSite ?OutLinks ?PR ?wikiR ?x "
        + "WHERE { "
        + "?x rdf:type portal:Web-Site."
        + "?x portal:has-title ?WebSite."
        + "FILTER regex(?WebSite, '" + QuerySearch + "', 'i' ) "
        + " optional{?x base:hasTotalNrOutLinks ?OutLinks}."
        + " optional{?x base:hasWebPR ?PR}."
        + " optional{?x wiki:has-page_random ?wikiR}."
        + "}"
        + "ORDER BY DESC(?PR) ";
    try {
        // Get the start time of the process
        long start = System.currentTimeMillis();

        OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM, null);
        OntDocumentManager dm = model.getDocumentManager();

        InputStream in = FileManager.get().open(ONTOLOGY_URL);
        if (in == null) {
            throw new IllegalArgumentException("File: " + ONTOLOGY_URL + " not found");
        }
        InputStream in_wiki = FileManager.get().open(ONTOLOGY_URL_wiki);
        if (in_wiki == null) {
            throw new IllegalArgumentException("File: " + ONTOLOGY_URL_wiki + " not
found");
        }
        dm.addAltEntry(ONTOLOGY_URL, ONTOLOGY_URL);
        model.read(in, "");
        model.read(in_wiki, "");

        // create a Jena query from the queryString
        Query query = QueryFactory.create(inputQuery);

        // create a Jena QueryExecution object that knows the query
        // and the model over which the query will be run
        QueryExecution qexec = QueryExecutionFactory.create(query, model);

```

```

try {
    // execute the query - get back a ResultSet
    ResultSet resultSet = qexec.execSelect();
    List resultVars = resultSet.getResultVars();
    int colCount = resultVars.size();

    out.print("<div class='container'><table>");
    int y = 0;
    // iterate over the result set
    while (resultSet.hasNext()) {
        out.print("<tr>");
        QuerySolution querySolution = resultSet.nextSolution();
        RDFNode nodeWebSite = querySolution.get((String) resultVars.get(0));
        RDFNode nodeOutLinks = querySolution.get((String) resultVars.get(1));
        RDFNode nodePR = querySolution.get((String) resultVars.get(2));
        RDFNode nodewikiR = querySolution.get((String) resultVars.get(3));
        RDFNode nodeUri = querySolution.get((String) resultVars.get(4));
        out.print("<td class='sqltabell'>");
        Literal litWebSite = (Literal) nodeWebSite, litOutLinks = (Literal)
nodeOutLinks, litPR = (Literal) nodePR, litwikiR = (Literal) nodewikiR;
        Resource resUri = (Resource) nodeUri;
        String sWebSite = "", sOutLinks = "0", sPR = "0.00", sUri = "", swikiR
= "";

        if (litWebSite != null) {
            sWebSite = litWebSite.getString();
        }
        if (litOutLinks != null) {
            sOutLinks = litOutLinks.getString();
        }
        if (litPR != null) {
            sPR = litPR.getString();
        }
        if (litwikiR != null) {
            swikiR = litwikiR.getString();
        }
        if (resUri != null & nodeUri.isURIResource()) {
            sUri = resUri.getURI();
        }
        y++;
        out.print("<strong><a href='http://wikipedia.org/wiki/' + sWebSite + ""
target='_blank'>" + sWebSite + "</a></strong><tr><td>");
        out.print("<small>" + sUri + "</small></td></tr><tr><td>");
        out.print("<blockquote><strong><i class=' icon-chevron-down'
title='Semantic PageRank of Web-Site.'></i>&nbsp;" + sPR + "&nbsp;");
        out.print(" <i class='icon-globe' title='Number of Out
Links.'></i>&nbsp;" + sOutLinks + "&nbsp;");
        out.print(" <i class='icon-random' title='The wikipedia rank
number.'></i>&nbsp;" + swikiR + "</strong></blockquote></td><tr></tr></tr>");
        out.print("</tr>");
    }
    // Get the end time of the process
    long end = System.currentTimeMillis();
    long elapsedTime = end - start;

    if (!QuerySearch.isEmpty()) {
        out.print("<strong>" + y + " results found for &#34;" + QuerySearch +
"&#34; (</strong><small>About " + elapsedTime + " ms</small><strong>).</strong>");
    } else {
        out.print("<strong>" + y + " results found (</strong><small>About " +
elapsedTime + " ms</small><strong>).</strong>");
    }
    out.print("</p><hr class='fancy-line'></hr>");
}

```

```

    } finally {
        qexec.close();
    }
} catch (Exception e) {
    out.print("<tr><td><xmp>" + e + "</xmp></td></tr>");
}
out.print("</table></div>");
out.print("<hr class='fancy-line'></hr>");

%>

```

Ndërsa në Listimi 33 është paraqitur sintaksa HTML nga fajlli JSP që shfaq katalogun e faqeve si dhe mundësinë e kërkimit direkt të faqeve varësisht nga fjala kyçe në kërkim (`request.getParameter("Search")`).

### Listimi.33 HTML sintaksa ne JSP fajllin e katalogut te faqeve.

```

<%--
Document   : Catalog
Created on : Mar 10, 2014, 10:06:49 PM
Author     : Gezimi
--%>
<!DOCTYPE html>
<html>
  <jsp:include page="Header.jsp" />
  <header class="masthead">
    <div class="jumbotron">
      <div class="container">
        <div class="page-header_new">
          <h1>Data Catalog</h1>
        </div>
        <div id="search-helper-message">
          Search over catalog of Wiki Web-Sites.
        </div>
      </div>
    </div>
  </header>

  <div class="container">
    <div class="toolbar">
      <ol class="breadcrumb">
        <li class="home"><a href="index.jsp"><i class="icon-home"></i><span>
Home</span></a></li>
        <li class="active">Catalog</li>
      </ol>
    </div>
    <div class="primary">
      <section class="module">
        <div class="module-content">
          <form id="dataset-search" class="search-form form-inline no-
padding col-md-12 col-lg-12" method="get" data-module="select-switch">
            <div class="input-group">
              <input type="search" id="search-header" title="Example
searches: Baseball, FIEK" data-strings='{ "targets":["Category",""]}' value="<%
if (request.getParameter("Search") != null) {
    out.print(request.getParameter("Search"));
} else {
    out.print("");
}
%>" name="Search" class="search-field form-control"

```

---

```
autocomplete="off" placeholder="Search catalog..." >
    <span class="input-group-btn">
        <button type="submit" class="search-submit btn btn-
primary">
            <i class="fa fa-search"></i>
            <span class="sr-only">Search</span>
        </button>
    </span>
</div>
</form>
<div class="results">
    <jsp:include page="Jena_Query.jsp" />
</div>
</div><!--/.container-->
</section>
</div>
</div>
<jsp:include page="Footer.jsp"/>
</html>
```