

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Justinas Stubrys
Informatikos specialybės II kurso dieninio skyriaus studentas

**INTELEKTINĖS INTERNETO NARŠYMO SISTEMOS
TYRIMAI IR TAIKYMAI**

**THE RESEARCH AND APPLICATION OF INTELLIGENT WEB BROWSING
SYSTEM**

MAGISTRO DARBAS

Darbo vadovas:
Prof. L. Sakalauskas

Recenzentas:
Lekt.G. Felinskas

Šiauliai, 2011

Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąrašę pateiktus informacinius šaltinius bei savo tyrimų duomenis“

Darbo autoriaus _____
(vardas, pavardė, parašas)

Darbo tikslai ir uždaviniai

Tikslas

Išnagrinėti intelektinių naršymo sistemų architektūras, bei palyginti interneto paieškos sistemas. Išanalizuoti problemas su kuriomis susiduria vartotojai naršydami internete ir pateikti šių problemų sprendimo būdus. Išanalizuoti duomenų gavybos metodus naudojamus semantiniame tinkle.

Uždaviniai

- Pateikti intelektinės naršymo sistemos architektūrą;
- Palyginti keletą interneto naršymo sistemų;
- Surinkti sistemos testavimo rezultatus ir išvadas;
- Išanalizuoti problemas su kuriomis susiduria vartotojai naršydami internete;
- Išanalizuoti semantinio interneto sandarą;
- Pateikti semantinius sprendimus pagerinančius interneto naršymo našumą;

Darbo vadovo _____
(vardas, pavardė, parašas)

Turinys

IVADAS	7
1. NARŠYMO SISTEMOS	9
1.1 Įvadas	9
1.2 Naršyklės struktūra	10
1.3 Naršymo sistemų istorija.....	11
1.4 Kas yra intelektinė naršymo sistema?.....	12
2. INFORMACIJOS TURINIO ŽODŽIŲ SUDARYMAS NAUDOJANT PROGRAMINĘ ĮRANGĄ STATISTICA	14
2.1 Teksto gavybos metodai.....	14
2.2 Gerai patikrintų metodų naudojimas ir teksto gavybos rezultatų suvokimas	15
2.3 “Juodųjų dėžių” panašumas su teksto gavybos koncepcija.....	15
2.4 Teksto gavyba kaip dokumentų paieška	16
3. INTELEKTUALIOS INTERNETO NARŠYMO SISTEMOS ARCHITEKTŪRA	17
3.1 Sistemos anatomija	17
3.2 Google architektūros apžvalga	17
3.3 Pagrindinės duomenų struktūros	18
3.3.1 Saugykla	19
3.3.2 Dokumento indeksas	19
3.3.3 Įrašų sąrašas.....	19
3.4 Internetinių duomenų indeksavimas	20
3.5 Paieška	21
3.5.1 Rangavimo sistema	21
3.5.2 Grįžtamasis ryšys	22
3.6 Rezultatai ir charakteristikos	23
4. PAIEŠKOS VARIKLIAI.....	26
4.1 Geriausia paieškos sistema.....	26

4. 2 Paieškos sistemų palyginimas	27
4. 3 Hakia	27
4. 4 WolframAlpha	28
4. 5 Eksperimentas.....	28
5. INTERNETO NARŠYMAS	33
5. 1 Interneto naršymo problemos.....	33
5. 2 Tinklapio architektūros tobulinimas	35
5. 3 Vartotojų analizavimo sistema ir vartotojo profilis.....	37
5. 4 Tinklapio stebėjimo sistema ir „tinklapio duomenų bazė“	38
6. SEMANTINIS INTERNETAS (WEB3) KONCEPCIJA	40
6. 1 Įvadas	40
6. 2 Semantinio interneto veikimas	40
6. 3 Semantinio žiniatinklio sandara	41
6. 4 Web3 analizė	42
6. 4. 1 XML ir RDF žymėjimas.....	43
6. 4. 2 Kaip atpažinti tinkamą URI?	44
6. 4. 3 Kalbos ir žodynai: RDFS, OWL.....	45
7. ONTOLOGIJŲ ŽYMĖJIMAS NAUDOJANT NEURONINIO TINKLO METODĄ.....	50
7. 1 Įvadas	50
7. 2 Ontologijų žymėjimas	50
7. 3 IAC Neuroninis tinklas	51
7. 4 Realizavimas	52
7. 5 OAEI testo procedūra	55
7. 6 Metodo naudos įvertinimas	55
7. 6. 1 Duomenų rinkiniai	55
7. 6. 2 Vertinimo kriterijai	56
7. 6. 3 Eksperimentiniai metodai ir rezultatai	56
IŠVADOS	58

TERMINŲ ŽODYNĖLIS.....	59
LITERATŪRA.....	60
Santrauka	63
Summary	63
Priedai	64

IVADAS

Pasaulinis kompiuterių tinklas WWW įgavo nepaprastai didelį populiarumą dėka tokių interneto naršyklių kaip Explorer, Firefox ir Netscape. Dėl itin spartaus interneto tinklo augimo komercija WWW pagrindu tapo nauja konkurencinga verslo šaka, kurios pagrindiniai tikslai yra: padidinti pajamas, supaprastinti verslo procesus ir pakelti produktyvumą.

Lyginant su tradicine verslo operacijų aplinka, WWW turi daug privalumų, tokių kaip, pasaulinė komunikacija, hyper nuorodos, interaktyvus darbas, daugialypės terpės ir laiko juostų skirtumų panaudojimas. Vis dėl to neriboti išteklių ir daug laisvės įtakojo keletą naršymo problemų, tokių kaip paklydimas, nenašus naršymas ir svarbiausių puslapių ar naujo turinio praradimas. Tai sukelia daug problemų WWW pagrindu sukurtoje komercijoje. Iš tikrųjų, pavojinga turėti tokius individualiai pritaikomus interaktyvius bruožus, tam kad įgyti konkurencinius privalumus nesuskaičiuojamame kiekyje internetinių svetainių, nes tai gali sudaryti prielaidas nesankcionuotam naudojimui ar tinklo užteršimui internetinėmis šiukšlėmis [1].

Temos aktualumas ir svarba. Nežiūrint į tai, kad pasaulinis kompiuterių tinklas pastaraisiais metais smarkiai išpopuliarėjo, tačiau lieka dar neišspręstų problemų su kuriomis susiduria vartotojai ir net patys tinklapių kūrėjai. Automatiniai paieškos varikliai, kurie pasikliauna klaviatūros atitikmenimis, paprastai gražina per daug prastos kokybės atitikmenų. Sąveika pasauliniame tinkle vis dar ribojama dėl įvairiarūšės informacijos.

Nuo 2004 m. kalbama apie WEB 2.0 interneto technologiją, kai nemažą turinio dalį kuria nebe svetainių savininkai, bet patys lankytojai. Pradėtas kurti naujos kartos žiniatinklis arba „maštantis internetas“, kitaip dar vadinamas WEB 3.0, kuris pasak interneto autoritetų išstums „Google“.

Tam, kad būtų sukurta intelektualinė interneto naršymo sistema, reikalingas intelektualus internetas. Semantiniame žiniatinklyje interneto turinys gebėtų keistis ir suprastų koks jis buvo anksčiau, tuomet nebereikėtų greitųjų nuorodų. Naršyklė galės nukreipti vartotoją tiesiai į norimo turinio informacijos puslapį.

Darbo tikslo ir darbo temos sąsaja. Šio magistro darbo tikslas – išnagrinėti intelektinių naršymo sistemų architektūras, bei palyginti interneto paieškos sistemas. Išanalizuoti duomenų gavybos metodus naudojamus semantiniame tinkle. Pateikti intelektualių agentų pavyzdžius ir apžvelgti jų veikimą. Šie darbo tikslai glaudžiai ar net tiesiogiai susiję su darbo tema, nes visa tai, sudaro intelektinę interneto naršymo sistemą.

Darbo tikslų ir uždavinių sąsaja. Atsižvelgiant į darbo tikslus, pagal tai išskelti ir darbo uždaviniai.

Darbo uždaviniai:

1. Pateikti intelektinės naršymo sistemos architektūrą;
2. Palyginti keletą interneto naršymo sistemų;
3. Surinkti sistemos testavimo rezultatus ir išvadas;
4. Išanalizuoti problemas su kuriomis susiduria vartotojai naršydami internete;
5. Pateikti sprendimus padedančius išspręsti išanalizuotas problemas;
6. Išanalizuoti semantinio interneto sandarą;
7. Pateikti semantinius sprendimus pagerinančius interneto naršymo našumą;

Visos darbo užduotys atspindi keliamus tikslus. Nagrinėjamos temos, kurios sudaro arba yra susijusios su intelektine naršymo sistema.

Paiškinimai dėl naudotų tyrimo metodų. Darbe pateikiama interneto naršymo sistemos sandara ir paskirtis. Pristatoma programinė įranga, kuri geba išnagrinėti tekstą ir palengvina darbą sudarant dokumento turinio raktinius žodžius, pagal kuriuos vartotojui ir net pačiai paieškos sistemai būtų lengviau dirbti. Lyginami trys žiniatinklio paieškos varikliai. Pateikiama intelektinė interneto svetainės architektūra, kuri naršymą sistemoje gali palengvinti kiekvienam vartotojui pagal jo poreikius.

Išnagrinėta nauja semantinio interneto technologija, kuri galimai pakeis visą esamo žiniatinklio veikimo principą. Pateikiama semantinės duomenų paieškos sistemos architektūra. Išanalizuotas ontologijų žymėjimo metodas, pagerinantis sąsajų žymėjimo tikslumą tarp internetinių duomenų. Metodas paremtas vienu iš duomenų gavybos metodu – neuroninio tinklu.

1. NARŠYMO SISTEMOS

1.1 Įvadas

Pasaulinis kompiuterių tinklas WWW – tai informacinė erdvė, kurioje esami informacijos elementai vadinami ištekliais. Šie ištekliai atpažįstami universalių identifikatorių pagalba – URI.

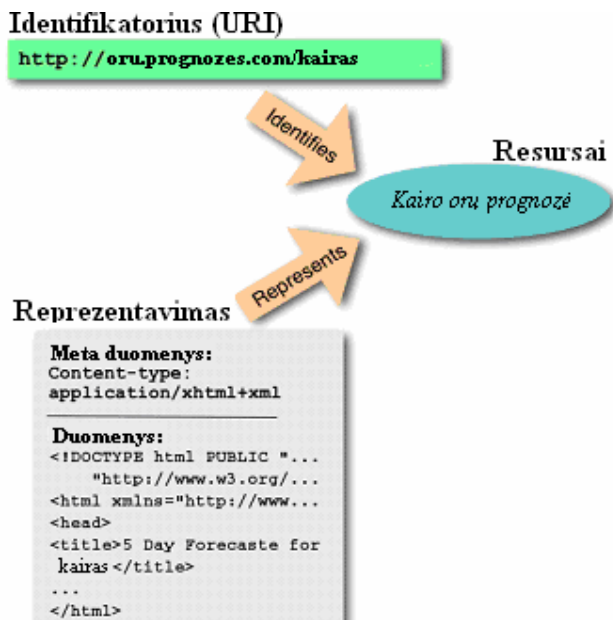
Interneto naršyklė – tai programinė įranga skirta surasti, atvaizduoti ir analizuoti resursus pasauliniame kompiuterių tinkle. Informacijos resursai yra identifikuojami universaliu resursų identifikatoriumi. Tai gali būti internetinio puslapio paveikslėliai, video vaizdas ar kitas panašus turinys. Greitosios nuorodos esamos internetiniame puslapyje suteikia galimybę vartotojams lengvai orientuoti savo naršykles bei į norimus informacijos resursus.

Žemiau pateikiamas kelionės scenarijaus pavyzdys, apibūdinantis internetinių agentų elgesį tinkle. Šie agentai, yra ne kas kita, kaip programinė įranga veikianti informacinėje erdvėje. PĮ agentai apima: serverius, internetinius vurus, naršykles ir daugialypės terpės vartotojus.

Pavyzdys

Planuodamas kelionę į Egiptą keliautojas spaudoje pastebi, kad orų prognozę į vykstančią regioną gali peržiūrėti šioje nuorodoje 'http://oru.prognozes.com/kairas'. Keliautojas supranta, kad šis adresas yra URI ir jis greičiausiai norės perskaityti orų prognozę savo naršyklėje. Kai smalsuolis įveda minėtą URI į savo interneto naršyklę:

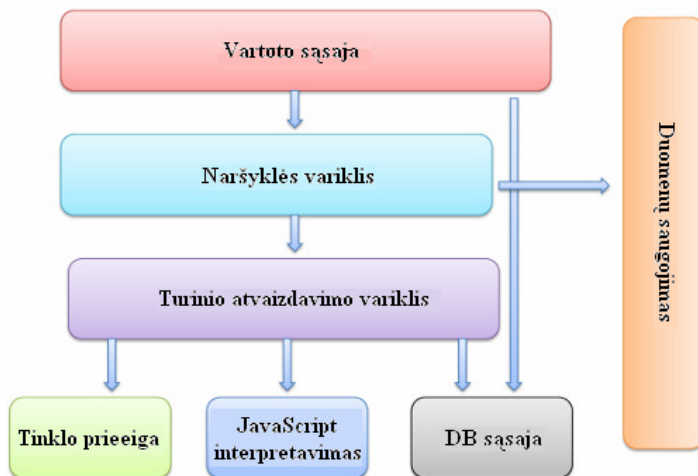
1. Naršyklė iš karto nustato, kad vartotojas įvedė URI.
2. Naršyklė vykdo informacijos paiešką pagal nurodytą užklausa per “http” protokolo URI schemą.
3. Atsakingi už internetinį puslapį “weather.example.com/cairo” asmenys, suteikia informaciją pateiktai užklausiai.
4. Naršyklė interpretuoja atsaką, pateiktą serverio XHTML kompiuterine kalba, tuomet sistema vykdo papildomą paiešką susijusią su užklausa.
5. Naršyklė pateikia surastą informaciją hipertekstinėmis nuorodomis į susijusią informaciją. Keliautojas gali naudotis šiomis nuorodomis, kad surastų daugiau papildomos informacijos apie Egipto sostinę – Kairą.



Pav. 1 Naršyklės užklauso vykdomo schema [2]

Pav.1 iliustruoja sąryšius tarp identifikatoriaus – internetinės nuorodos, resursų – orų prognozės internetinio puslapio ir duomenų apibrėžiančių tą puslapį. [2]

1. 2 Naršyklės struktūra



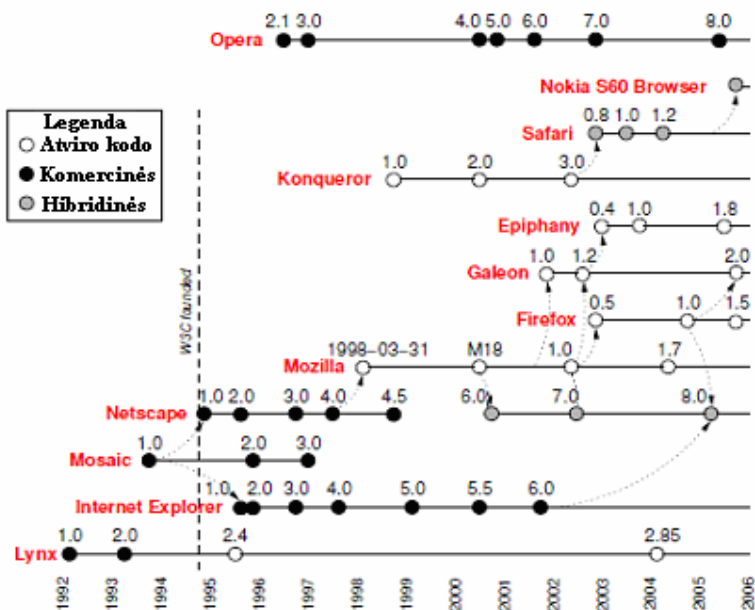
Pav. 2 Naršyklės pagrindiniai komponentai [3]

Naršyklės pagrindiniai elementai yra:

1. Vartotojo sąsaja – įtraukia adreso įvedimo laukelį, pirmyn/atgal mygtukus, žymėjimo (angl. bookmark) meniu ir t.t. Trumpiau tariant, apima kiekvieną naršyklės dalį išskyrus pagrindinį langą, kuriame matomas užklauso puslapis.
2. Naršyklės variklis – tai sąsaja skirta naršyklės variklio užklausoms ir valdymo komandoms vykdyti.
3. Naršyklės turinio atvaizdavimo variklis – atsakingas už užklauso turinio vaizdavimą. Pvz. jei užklauso turinys yra HTML, jis atsakingas už HTML ir CSS gramatinį nagrinėjimą ir išnagrinėto turinio atvaizdavimą ekrane.
4. Tinklo prieigos sistema – skirta užklausoms tinkle vykdyti, tokioms kaip HTTP.
5. Antrinė vartotojo sąsaja – skirta pagrindiniams objektams vaizduoti, tokiems kaip pasirinktinio įvedimo laukai ir langai. Naudoja vartotojo sąsajos operacinės sistemos metodus. Kitaip dar vadinama duomenų bazės vartotojo sąsaja.
6. JavaScript interpretavimo sistema. Naudojama gramatiniam JavaScript kodų nagrinėjimui ir vykdymui.
7. Duomenų saugojimas. Naršyklei reikia išsaugoti visus duomenų rūšis kietame diske. [3]

1.3 Naršymo sistemų istorija

Žemiau pateikiamas paveikslėlis vaizduojantis interneto naršyklių istoriją ir evoliuciją.



Pav. 3 Naršymo sistemų istorija [4]

Iš paveikslėlio galime matyti, kurios sistemos yra komercinės, o kurios nemokamos. Informacija pateikta tik iki 2006 m., bet nuo to laiko neįvyko didelių persilaužimų šioje rinkoje, todėl naršyklių sąrašas iš esmės nepasikeistų.

1.4 Kas yra intelektinė naršymo sistema?

Intelektinė interneto naršymo sistema – tai sistema, kuri stebi jūsų naršymą internete, siūlo daugiau internetinių svetainių susijusių su jūsų vykdoma paieška, neapsunkindama vartotojo darbo prašydama papildomai įvesti raktinių rodžių.

Intelektinė interneto naršymo sistema – tai sistema, kuri reaguoja į vartotojo veiksmus ir pasiūlo jam tinkamą informaciją, ir taip kiekvieną kartą kiekvienam vartotojui individualiai, žinoma jei yra ką pasiūlyti. [5]

Intelektualus agentas – tai sistema, kuri padeda ir funkcionuoja interneto vartotojų naudai. Intelektualūs agentai dirba atlieka darbą vietoj vartotojo, taip sutaupydamas laiką, kurį vartotojai praleistų naršydami ieškant naudingos informacijos. Agentai gali atlikti pasikartojančias užduotis, dalykus kuriuos pamiršo vartotojas, apibendrinti duomenis, mokytis iš vartotojo veiksmų ir pateikti rekomendacijas. Visi šie veiksmai suteikia naršymo sistemai galimybę vadintis intelektine interneto naršymo sistema.

Įsivaizduokime, mes stebime vartotojo naršymą internete ir matome jo žiūrimus puslapius, puslapis P1 tarp daugelių. Jis paspaudžia nuorodą, kurios pagrindą sudaro žodis „delfinai“, nuoroda nukreipia vartotoją į puslapį P2 apie Majamio delfinus – futbolo komandą NFL lygoje. Tuomet vartotojas greitai spragtelė atgal į puslapį P1 ir jame praleidžia antrą nuorodą su pavadinimu „laisvi delfinai“, tada paspaudžia trečią nuorodą su pavadinimu „banginiai“. Perskaito puslapį P3, tada seką į nuorodą su pagrindo žodžiu „banginis“ ir „orka“, kol jis pasiekia puslapį P5, kuris sudarytas iš „banginiai“ ir „jūros pasaulis“.

Šiuo atveju galime manyti, kad vartotojas domisi banginiais ir delfiniais, bet ne futbolu. Tai yra todėl, kad matėme vartotojo lankomus puslapius, kurie įtraukė šiuos pirminius žodžius. Galime charakterizuoti kokie žodžiai yra „informacijos turinio“ žodžiai, vykdomos sesijos kontekste. Taip pat įmanoma nustatyti kaip dažnai vartotojas seka nuorodą, kuri įtraukia tam tikrą raktinį žodį. Galime naudotis informacijos turinio puslapiais, kad nustatytume, kurie internetiniai puslapiai bus mums naudingi. Turbūt rastume informacijos turinio puslapių, naudodami informacijos turinio žodžius, kaip užklausos terminus paieškos sistemoje (pvz. Google).

Rekomendavimo sistema kaip tik yra sudaryta šiuo pagrindu. Darbo metu surenka naršymo savybes apie kiekvieną žodį, kuris pasirodo vartotojo vykdomoje sesijoje, tada naudoja klasifikavimo algoritmą, kad nuspėtų, kuris iš žodžių labiausiai tikėtinas būti

informacijos turinio žodžiu. Tuomet skirtingais metodais, naudojant informacijos turinio žodžių rezultatų rinkinį, ieško informacijos turinio puslapių .

Iš aukščiau minėto pavyzdžio apie vartotojo naršymą internete, galime pateikti bendrą naršyklės architektūrą su agentų sistemomis. Taip atrodytų naršyklės architektūros pakitimai norint realizuoti vartotojo stebėjimo funkciją ir rekomenduoti jam svarbią informaciją.



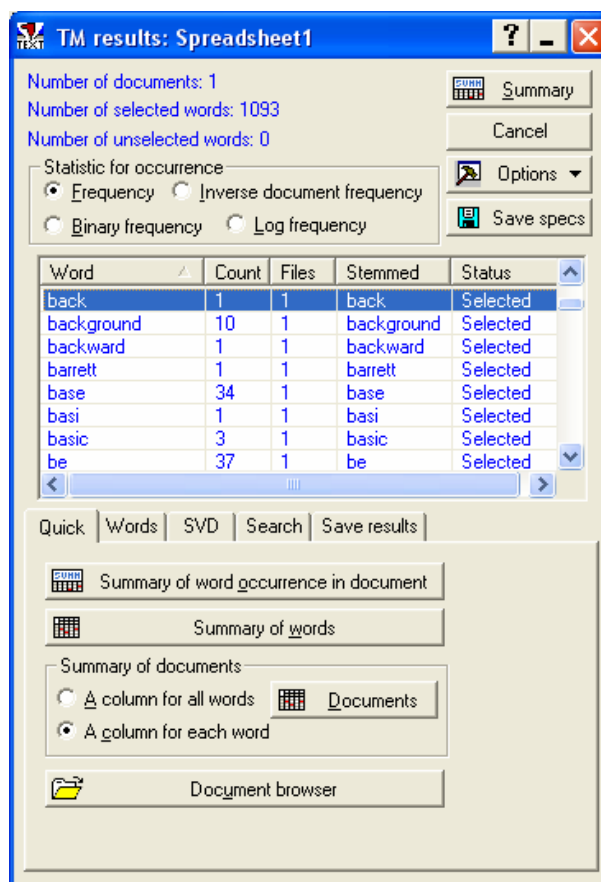
Pav. 4 Naršyklės architektūra su agentais

Intelektiniai sprendimai turėtų būti realizuoti tarp vartotojo sąsajos ir naršyklės variklio. Stebėjimo sistema stebi vartotojo naršymą žiniatinklyje. Informaciją perduoda naršyklės varikliui, kuris išsaugo vartotojo veiksmų istoriją saugykloje. Tuomet rekomendavimo agentas per naršyklės variklį geba pasiūlyti vartotojui jam reikalingą informaciją.

2. INFORMACIJOS TURINIO ŽODŽIŲ SUDARYMAS NAUDOJANT PROGRAMINĘ ĮRANGĄ STATISTICA

2.1 Teksto gavybos metodai

Teksto gavybos metodai – tekstinės informacijos apdorojimas į automatiškai išgaunamą informaciją, realizuotas STATISTICA *Teksto gavybos ir dokumentų apžvalgoje*, gali būti apibendrintas kaip procesas sunumeruojantis tekstą. Žemiausiame lygmenyje, programa sunumeruos visus žodžius rastus nurodytame dokumente ir suskaičiuos juos visus tam, kad sudaryti dokumentų ir žodžių lentelę. Pvz. Matrica dažniausiai pasikartojančių skaičių kiekviename dokumente. Šis pagrindinis procesas skirtas neįtraukti tam tikrų dažnai pasitaikančių žodžių, tokiu kaip „the” ir „a”. Taip pat sujungti skirtingas gramatines formas to paties žodžio kaip "keliavimas", „keliavo“, „kelionė“. Visi statistiniai standartai ir duomenų gavybos technikos gali būti taikomi, siekiant išgauti matmenis ar žodžių grupes, dokumentus arba identifikuoti svarbius žodžius ir terminus, kurie geriausiai atspindi kitus rezultatus. [6]



Pav. 5 Išanalizuotas dokumentas

Išanalizavus dokumentą galime pasirinkti, kaip atvaizduoti duomenis. Kaip pavyzdžiui pateikti lentelę su žodžiais ir kiek kartų jie pasikartoja arba lentelę su kiekvieno žodžio indeksu.

File summary (Spreadsheet1)						
Size	Number of words	Indexed text 1	Indexed text 2	Indexed text 3	Indexed text 4	Indexed text 5
IMMO_EN.pdf	3371	518	univ	immo	vag	immobil emul

Pav. 6 Suindeksuotų žodžių lentelė

Words summary (Spreadsheet1)			
	Frequency	Number of documents	Stemmed word
backward	1	1	backward
barrett	1	1	barrett
base	34	1	base
basi	1	1	basi
basic	3	1	basic
be	37	1	be
beat	1	1	beat
beauty	1	1	beauti
becaus	15	1	becaus
becom	7	1	becom
been	6	1	been
befor	6	1	befor
began	1	1	began
begin	1	1	begin
behavior	16	1	behavior

Pav. 7 Rezultatai su anglų kalbos kamienų filtravimu

2. 2 Gerai patikrintų metodų naudojimas ir teksto gavybos rezultatų suvokimas

Kai tik sudaroma duomenų matrica iš tam tikrų dokumentų ir žodžių rastų šiuose dokumentuose, STATISTICA pateikia išsamų analizės metodų rinkinį, tolimesniam šių duomenų apdorojimui. Čia svarbiausia dokumentinis metodas išgauti žinias iš esamų duomenų ir tai, kad struktūros “konceptija” STATISTICA *teksto gavyba ir dokumentų apžvalga* yra nesunkiai panaudojama ir suprantama. Kitaip tariant, propaguojamas prieinamumas lengvai suprantamų algoritmų ir grupavimo metodų naudojimas, faktoriinė ar prognozuojanti duomenų gavyba tam, kad būtų galima panaudoti informaciją gautą iš dokumentų.

2. 3 “Juodųjų dėžių” panašumas su teksto gavybos koncepcija

Būdas realizuotas STATISTICA *teksto gavyba ir dokumentų apžvalga* yra priešingybė kitoms komercinėms teksto gavybos programoms, kurios remiasi savais algoritmais, kad

galimai išgautų „idėjas“ iš teksto ir teigtų, kad galima reziumuoti didelius kiekius tekstinių dokumentų automatiškai, išlaikant pagrindinę ir svarbiausią esmę tuose dokumentuose. Kol yra begalė algoritmų susijusių su „prasmės iš dokumentų“ išgavimu, tol šis technologijos būdas liks dar labai ankstyvoje stadijoje ir siekis tiekti prasmingą automatizuotą didelės apimties dokumentų suvestinę, gali likti visam laikui nepajėgus.

Mintis tai, kad STATISTICA pateikia rinkinį gerai apibūdintų algoritmų ir lengvai suprantamų analizės metodų, suteikia analitikams galimybę suvienyti informaciją esančią tekste, teksto gavybos analizavimo standarte.

Apsvarstykime konkretų pavyzdį: pabandykime skirtingas automatines kalbos vertimo paslaugas prieinamas internete, kurios gali išversti iš visas teksto pastraipas iš vienos kalbos į kitą. Tada išverskime truputį teksto, nors keletą žodžių, iš savo gimtos kalbos į kitą kalbą ir atgal, ir pažvelkite gautus rezultatus. Beveik visada, bandymas išversti netgi trumpą sakinį į kitas kalbas ir atgal, išlaikant pradinę sakinio reikšmę yra sudėtingas, programa pateikia klaidingus, o ne tikslus rezultatus. Tai atspindi automatiškai verčiamų tekstų prasmės sudėtingumą.

2. 4 Teksto gavyba kaip dokumentų paieška

Yra kitas taikymo būdas, kuris dažniausiai apibūdinamas ir minimas kaip „teksto gavyba“ – automatinė paieška didelės apimties dokumentuose, sudarytuose iš raktinių žodžių ar raktinių frazių. Tai sritys, kaip pvz. populiarūs internetinės paieškos varikliai, kurie buvo išvystyti per paskutinį dešimtmetį, kad suteiktų veiksmingą priėjimą prie internetinių puslapių su tam tikru turiniu. Tai akivaizdžiai svarbus paraiškos būdas su daugybe naudos, bet kuriai organizacijai, kuriai reikia ieškoti didelės apimties dokumentų saugyklose, sudarytose iš įvairių kriterijų. Šis būdas skiriasi nuo funkcionalumo esamo STATISTICA *teksto gavyba ir dokumentų apžvalga*, kur pradinė funkcija yra „suskaiciuoti“ tekstą tam, kad paversti jį prieinamą dideliems kiekiams analitinių procedūrų esamų STATISTICA sistemoje. Kaip bebūtų, pažymėkime, tai, kad teksto gavybos rezultatų dialogas taip pat turi skirtingų funkcijų į sunumeruotą terminų sąrašo „užklausa“, gautą iš įvesties dokumentų; šios funkcijos pateikia efektyvų kelią didelės apimties skaičių paieškai įvesties dokumente, kuris buvo sunumeruotas programos pagalba.

3. INTELEKTUALIOS INTERNETO NARŠYMO SISTEMOS ARCHITEKTŪRA

Pasaulinis kompiuterių tinklas metė naujus iššūkius informacijos paieškoje. Informacijos kiekis internete auga stebėtinais tempais, kaip ir skaičius naujų nepatyrusių vartotojų internetinio „meno“ nagrinėjime. Žmonės dažniau naršys internete naudodami esamas nuorodų grafus, kurias pateikia Yahoo ar kiti paieškos varikliai. Žmonių sukurti paieškos varikliai efektyviai pateikia populiarias temas, bet yra subjektyvūs, brangiai sukurti ir išlaikomi, lėtai tobulinami ir negali apimti visų ezoterinių temų. Automatiniai paieškos varikliai, kurie pasikliauna klaviatūros atitikmenimis, paprastai gražina per daug prastos kokybės atitikmenų. Dar blogiau, kiti interneto vartotojai deda pastangas, kad pritrauktų žmonių dėmesį, naudodami priemones, kurios skirtos suklaidinti automatizuotą paieškos variklį. Larry Page kartu su Sergey Brin sukūrė plataus masto paieškos variklį, kuris išsprendžia daug egzistuojančių sisteminių problemų. Pasirinktas sistemos vardas Google.

3.1 Sistemos anatomija

Pirmiausia, yra pateikiama diskusija apie paieškos variklio architektūrą. Vėliau pristatoma keletas išsamių apibūdinimų apie svarbias duomenų struktūras. Galiausiai, svarbiausios aplikacijos: gavyba, indeksavimas ir paieškas bus išnagrinėta smulkiau.

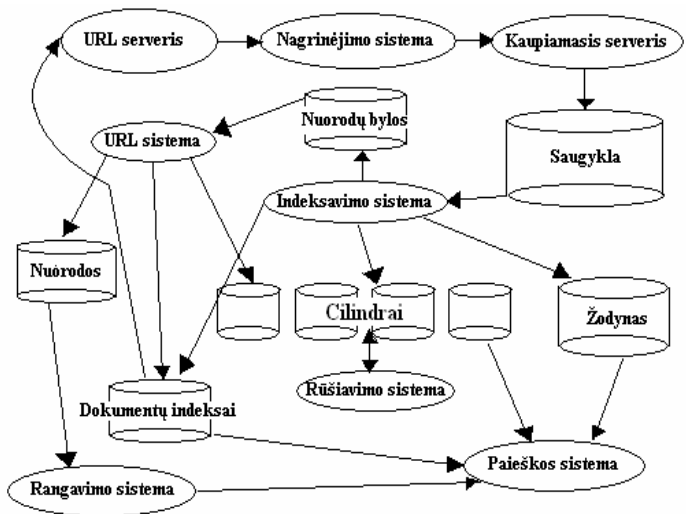
3.2 Google architektūros apžvalga

Šiame skyriuje pateikiama aukšto lygmens sistemos veikimo apžvalga, kaip pavaizduota paveikslėlyje Pav.8

Kiti skyriai apžvelgs aplikacijas ir duomenų

struktūras, kurios nepaminėtos šioje skiltyje. Beveik visa Google realizuota C arba C++ kalba geresniam efektyvumui užtikrinti ir geba dirbti su bet kuria Solaris ar Linux platforma.

Google sistemoje internetinė paieška (internetinių puslapių siuntimas) yra vykdomas keletos išskirstytų kompiuterinių sistemų pagalba. Egzistuoja URL serveris, siunčiantis URL sąrašą, kuris yra įtraukiamas į paieškos sistemą. Internetiniai puslapiai, kurie yra įtraukti, vėliau siunčiami į kaupiamąjį serverį. Kaupiamasis serveris suspaudžia, apdoroja informaciją



Pav. 8 Aukšto lygio Google architektūra

ir patalpina interneto puslapius į saugyklą. Kiekvienas puslapis turi susietą identifikuojantį skaičių, vadinamą dokumento ID, kuris yra priskiriamas, kai naujas URL yra gramatiškai išnagrinėtas iš internetinio puslapio. Indeksavimo funkcija yra vykdoma indeksavimo sistemos ir rūšiavimo sistemos pagalba. Indeksavimo sistema atlieka daug funkcijų: skaito iš saugyklos, išplėčia dokumentus ir gramatiškai juos išnagrinėja. Kiekvienas dokumentas yra konvertuojamas į žodžių grupes, vadinamus įrašus. Įrašai apibūdina žodžius, jų vietą dokumente, apytiksliai šrifto dydžio reikšmę ir iš didžiosios raidės rašomus žodžius. Indeksavimo sistema suskirsto šiuos įrašus į „cilindrų“ rinkinius, taip yra sukuriama iš dalies eilės tvarka surūšiuotas indeksas. Indeksavimo sistema vykdo kitą labai svarbią funkciją. Ji gramatiškai išnagrinėja visas nuorodas į kiekvieną internetinį puslapį ir patalpina svarbią informaciją apie juos į greitosios nuorodos bylas. Ši byla talpina pakankamai informacijos, kad atskirtų kiekvienos nuorodos tekstą ir kur kiekviena nuoroda nukreipiama „iš ir į“.

Universali resursų sprendimo priėmimo sistema (*angl.* URL) skaito greitosios nuorodos teksto bylą, tuomet konvertuoja giminingus URL į absoliučius URL ir verčia į dokumentų ID. Tada patalpina nuorodos tekstą į priekinį indeksą, susijusį su dokumento ID, kuri pažymėjo greitoji nuoroda. Taip pat sistema generuoja nuorodų duomenų bazę, kurioje yra dokumentų identifikavimo poros. Duomenų bazės nuoroda naudojama apdoroti visų dokumentinių puslapių rangą.

Rūšiavimo sistema paima cilindrus, kuriuos surūšiuo dokumentų identifikatorius ir perrikiuoja juos žodžių identifikatoriaus pagalba, tam kad sugeneruotų invertuotus indeksus. Rūšiavimo sistema taip pat sukuria žodžių identifikavimo sąrašą ir suskirsto juos į invertuotus indeksus. Programa pavadinimu „DumpLexicon“, sujungia šį sąrašą kartu su leksikonu, kuri pagamino indeksavimo sistema ir sugeneruoja naują žodyną, kuri naudoja paieškos sistema. Paieškos sistema paleidžiama internetinio serverio pagalba ir naudoja leksikoną sukurtą programos „DumpLexicon“ kartu su invertuotais indeksais ir puslapių rangavimo sistema, kad atsakytų į užklausas.

3. 3 Pagrindinės duomenų struktūros

Google duomenų struktūros yra optimizuotos taip, kad plati dokumentų kolekcija galėtų būti išnagrinėta, suindeksuota ir surasta nedidelių išteklių pagalba. Nepaisant, kad procesorių našumas ir didelis kiekis įvesties, išvesties įrenginių greitis dramatiškai padidėjo per paskutinius metus, bet paieška diske vis dar reikalauja apie 10ms darbo užbaigimui. Google sistema yra suprojektuota taip, kad kiek įmanoma mažiau laiko užimtų paieška diske, ir tai turėjo didelę įtaką projektuojant duomenų struktūras.

3.3.1 Saugykla

Saugykla talpina pilną, kiekvieno internetinio puslapio HTML kodą. Kiekvienas puslapis suspaudžiamas naudojant programinės įrangos biblioteką zlib, skirtą duomenų suspaudimui. Dokumentai saugykloje saugomi vienas po kito ir yra patalpinami pradžioje, dokumentų identifikatoriaus pagalba, kaip pavaizduota Pav.9. Saugykla nereikalauja jokių kitų duomenų struktūrų, tam kad jos būtų panaudotos patekti į ją. Tai gelbsti nuo duomenų išderinimo, padaro žymiai lengvesnį sistemos plėtojimą ir vystymąsi. Galima atstatyti visas kitas duomenų struktūras iš saugyklos, taip pat ir bylą, kuri talpina naršymo sistemos klaidas.

Saugykla: 53,5 GB = 147,8 GB išplėsta

Sinchr	Ilgis	Suspaustas paketas
Sinchr	Ilgis	Suspaustas paketas

...
Paketas suspaustas ir patalpinas į saugyklą

DokID	ekodas	url ilgis	psl ilg.	url	puslapis
-------	--------	-----------	----------	-----	----------

Pav. 9 Saugyklos duomenų struktūra

3.3.2 Dokumento indeksas

Dokumento indeksas saugo informaciją apie kiekvieną dokumentą. Jis yra fiksuoto pločio ISAM indeksas, kurį tvarko dokumentų identifikatorius. Informacija laikoma kiekviename įrašė apima esamą dokumento statusą, nuorodą į saugyklą, dokumento kontrolę ir įvairias statistikas. Jei dokumentas buvo išnagrinėtas, jis taip pat turi nuorodą į kintamos apimties bylą, pavadintą dokumento informacija, kurioje yra dokumento URL ir pavadinimas. Kitu atveju nuoroda nukreipia į URL sąrašą, kuris talpina tik URL.

3.3.3 Įrašų sąrašas

Įrašų sąrašas atspindi konkrečių žodžių įvykių sąrašą, tam tikrame dokumente, įskaitant išsidėstymą, šriftą ir žodžius rašomus iš didžiosios raidės informaciją. Įrašų sąrašas užima didžiąją dalį vietos, naudojamos indeksuose. Dėl to, yra svarbu atvaizduoti juos kaip įmanoma efektyviau. Laikomasi kelių alternatyvų šifruojant poziciją, šriftą ir žodžius rašomus iš didžiosios raidės – paprastas šifravimas (trigubais skaičiais), kompaktiškas šifravimas (ranka optimizuotą bitų pasiskirstymą), ir Huffman šifravimas. Pabaigoje pasirinktas optimizuotas šifravimo būdas, nes jis reikalauja daug mažiau vietos, negu paprastas šifravimas ir daug mažiau bitų valdymo nei Huffman kodavimas. Įrašų detalės pateiktos paveikslėlyje pav 10.

Google sistemos kompaktiškas kodavimas naudoja du baitus kiekvienam įrašui. Yra dvi įrašų rūšys: įmantrūs ir paprasti įrašai. Įmantrūs įrašai apima įrašus esančius URL, pavadinimą, nuorodos tekstą ar meta etiketes. Paprasti įrašai apima visą kitą informaciją. Paprasti įrašai susideda iš žodžių bitų rašomų iš didžiosios raidės, šrifto dydžio ir 12 bitų

esančių žodžio pozicijos dokumente. Šrifto dydis yra pateiktas santykinai su likusiais dokumentais, naudojant tris bitus. Įmantrus (aukščiausias) įrašas, susideda iš žodžių rašomų iš

didžiosios raidės bitų, šrifto dydžio nustatyto į 7, kad pažymėti jog yra aukščiausias įrašas, 4 bitai užšifruoja aukščiausią įrašo tipą, ir 8 bitai skirti pozicijai.

Nuorodos įrašai, 8 pozicijos

bitai yra atskiriami į 4 bitus pozicijai nuorodoje ir 4 bitus dokID sumaišymui, kur pasirodo

nuoroda.

Visų įrašų sąrašo ilgis yra saugomas prieš patį įrašą. Kad išsaugoti vietas, įrašų sąrašo ilgis yra sujungiamas su žodžio ID priekiniame indekse ir dokumento ID invertuotame indekse. Tai riboja jį atitinkamai iki 8 ir 5 bitų.

Įrašas: 2 baitai

Paprastas:	talpa:1	aukščiausias įrašas:3	pozicija:12	
Įmantrus:	talpa:1	aukščiausias įrašas = 7	tipas: 4	pozicija: 8
Nuorodos:	talpa:1	aukščiausias įrašas = 7	tipas: 4	maiša: 4 pozicija: 4

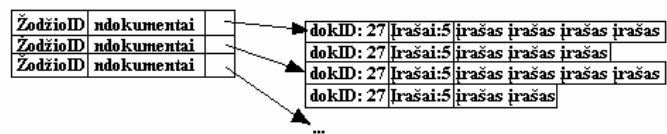
Priekiniai cilindrai: Viso 43 GB

DokID	ŽodžioID: 24	Įrašai: 8	Įrašai	Įrašai	Įrašai	Įrašai
	ŽodžioID: 24	Įrašai: 8	Įrašai	Įrašai	Įrašai	Įrašai
	Nulinis žodžioID					
DokID	ŽodžioID: 24	Įrašai: 8	Įrašai	Įrašai	Įrašai	Įrašai
	ŽodžioID: 24	Įrašai: 8	Įrašai	Įrašai	Įrašai	Įrašai
	Nulinis žodžioID					

...

Žodynas: 293MB

Invertuoti cilindrai: 41 GB



Pav. 10 Priekiniai, atvirkštiniai indeksai ir žodynas

3. 4 Internetinių duomenų indeksavimas

- **Gramatinis nagrinėjimas** – bet kuri gramatinio nagrinėjimo sistema, sukurta dirbti visame internete, privalo turėti didelį kiekį masyvų su galimų klaidų sąrašais. Jie apima nuo HTML rašybos žodžių iki nulinių kilobaitų esančių žymos viduryje, simbolius nepriklausančius ASCII ir didelę įvairovę klaidų. Maksimaliam greičiui pasiekti, sistema naudoja lanksčios leksikos analizatorių generavimui, kuris, apsirūpina savo nuosavu steku.

- **Dokumentų indeksavimas į cilindrus** – kai kiekvienas dokumentas yra gramatiškai išnagrinėtas, jis yra užšifruojamas į tam tikrą cilindrų skaičių. Kiekvienas žodis konvertuojamas į žodžio ID, naudojant atmintyje esančia maišos lentelę – žodyną. Nauji žodyno papildymai yra prijungiami prie bylos, kai tik žodžiai yra konvertuoti į žodžio ID, jų paplitimas esamame dokumente išverčiamas į įrašų sąrašą ir įrašomas į priekyje esančius cilindrus. Pagrindinis sunkumas su indeksavimo frazių paralelizavimu yra tas, kad žodynas turi būti dalinamas. Užuoat dalinti žodyną, panaudotas rašymo algoritmas visiems papildomiems žodžiams, kurie nebuvo pagrindiniame žodyne. Žodynas sudarytas iš 14 milijonų žodžių. Šiuo būdu daug indeksavimo sistemų gali dirbi lygiagrečiai, tada maža papildomų žodžių logaritminė byla gali būti vykdoma tik vienos galutinės indeksavimo sistemos pagalba.

- **Rūšiavimas** – tam, kad sugeneruotų invertuotus indeksus, rūšiavimo sistema paima kiekvieną priekyje esantį cilindrą ir surūšiuoja jį pagal žodžio ID, kad sukurtų invertuotą cilindrą pavadinimui ir nuorodų įrašams, taip pat ir pilnai invertuotą teksto cilindrą.

3.5 Paieška

Paieškos tikslas – efektyviai tiekti kokybiškus paieškos rezultatus. Dauguma plataus masto komercinių paieškos variklių atrodo jau padarė didelį progresą į priekį, kalbant apie efektyvumą. Dėl to, Google skyrė daugiau dėmesio paieškos kokybei šiame tyrime, taip pat tikėtina, kad priimti sprendimai yra priklausomi nuo komercinės apimties. Google užklauso analizės procesas pateiktas Pav. 11

Kad nustatytų atsako laiko limitą, kai tam tikras skaičius (dabar 40,000) atitinkamų dokumentų yra rastas, paieškos sistema automatiškai pereina į 8 žingsnį. Tai reiškia, yra galimybė, kad optimalūs rezultatai būtų gražinti. Praeityje buvo rūšiuojami įrašai priklausomai nuo puslapio rango, kuris atrodė galintis pagerinti situaciją.

3.5.1 Rangavimo sistema

Google palaiko žymiai daugiau informacijos susijusios su interneto dokumentais, lyginant su tipiškais paieškos varikliais. Kiekvienas įrašų sąrašas įtraukia poziciją, šriftą ir iš didžiosios raidės rašomų žodžių informaciją. Papildomai, įrašai skaidomi pagal nuorodos tekstą ir dokumento puslapio rangą. Sudėtinga sujungti visą šią informaciją į rangą. Suprojektavimo rangavimo funkcija veikia, taip kad, joks atskiras faktorius neturėtų per daug įtakos jos darbui. Pirmiausia, įsivaizduokime paprasčiausią atvejį – vieno žodžio užklausa. Siekiant suranguoti dokumentą su vieno žodžio užklausa, Google sprendžia iš to dokumento įrašų sąrašo ieškomam žodžiui. Google mano, kad kiekvienas įrašas yra vienas iš keletos skirtingų tipų (pavadinimas,

1. Gramatiškai išnagrinėja užklausa.
2. Konvertuoja žodžius į žodžio ID.
3. Kreipiasi į dokumentų sąrašą esantį nepilname kiekvieno žodžio cilindre.
4. Skanuoja dokumentų sąrašą, kol randa dokumentą, kuris atitinka visus paieškos terminus.
5. Apskaičiuoja to dokumento užklauso rangą.
6. Jei esama nepilnuose cilindruose ir dokumentų sąrašo pabaigoje, kreipiamasi į dokumentų sąrašo pradžią pilno cilindro kiekvieną žodžį ir grįžtame prie 4 žingsnio.
7. Jei nesame, bet kurio dokumento sąrašo pabaigoje, grįžtame prie 4 žingsnio. Surūšiuoti dokumentus, taip kad atitiktų pagal rangą ir grįžti atgal.

Pav. 11 Google užklauso analizė

nuoroda, URL, paprasto teksto didelis šriftas, paprasto teksto mažas šriftas,...), kurių kiekvienas turi savo tipo svorį. Tipo svoris sudaro vektorių, suindeksuotą pagal tipą. Google nustato įrašų skaičių kiekvienam tipui įrašų sąrašė. Kiekvienas skaičius konvertuojamas į skaičiaus svorį. Skaičiaus svoriai tiesiškai auga su skaičiavimais. Naudojama skaičių svorių vektorius skaliarinė sandauga su tipo svorių vektoriumi, kad apskaičiuoti dokumento informacijos taškus. Galiausiai, informacijos taškai sujungiami su puslapio rangu, kad gautų galutinį dokumento rangą.

Daugybinių žodžių paieškoje situacija yra labiau komplikauta. Dabar daugybiniai įrašų sąrašai turi būti peržvelgti iš karto, įrašai pasitaikantys dokumente vienas šalia kito yra įvertinami aukščiau, nei tie kurie pasitaiko toli vienas nuo kito. Įrašai iš daugybinių įrašų sąrašų yra parenkami į poras taip, kad šalia esantys įrašai parenkami kartu. Kiekvienam įrašų atitikčių rinkiniui, suskaičiuojamas artumas. Artumas remiasi tuo, kaip toli vienas nuo kito įrašai randasi įrašų dokumente arba nuorojoje, bet yra suskirstyti į 10 skirtingų verčių, pradedant nuo frazės „ne per arti“. Įvertis paskaičiuojamas netik kiekvienam sąrašo tipui, bet ir kiekvienam tipui ir artumui. Kiekvieno tipo ir artumo pora turi tipo-artumo svorį. Įverčiai konvertuojami į skaičiavimo svorius tuomet paimami skaičiavimo svorių ir tipo-artumo svorių skaliarinė sandauga, kad apskaičiuoti informacijos taškus. Visi šie skaičiai ir matricos gali būti rodomi paieškos rezultatuose, naudojant specialų derinimo būdą. [7]

Lentelė nr. 1 Puslapiai su atitinkamais rango svoriais

Internetinis puslapis	Puslapio rangas
Google Search	1,366,035,153
World Wide Web Consortium	62,096, 461
US Government website	8,468,953
Adobe - Adobe Reader Download	2,875,348
National Portal of India	283,473

Lentelė nr.1 vaizduoja penkias populiariausias interneto svetaines, kurių rangas yra didžiausias. Iš pateiktų rezultatų matome, kad populiariausia ir aukščiausią rangą turi internetinis puslapis yra Google. Neveltui antrą vietą užima „World Wide Web Consortium“ (W3C), nes tai kompanija sukūrusi naujos kartos web koncepciją, kuri turi apie 40 garsiausių prekinių ženklų kaip partnerius. [8]

3. 5. 2 Grįžtamasis ryšys

Rangavimo funkcija turi daug parametrų tokių kaip tipo-svoriai ir tipo-artumo-svoriai. Aiškinti teisingas šių parametrų reikšmes yra sudėtinga. Tam, kad tai atlikti, Google turi vartotojo atsiliepimų mechanizmą kiekviename paieškos variklyje. Patikimas vartotojas gali

pasirinktinai įvertinti visus gražintus rezultatus. Šis atsiliepiamas yra išsaugomas. Tuomet, kai sistemos rangavimo funkcija yra modifikuojama, galima matyti šio pokyčio įtaką visuose buvusiuose paieškose, kurias rangavo vartotojai. Nors tai toli gražu nėra tobula, bet tai suteikia Google kūrėjams idėjų kaip pokytis rangavimo funkcijoje įtakoja paieškos rezultatus.

3.6 Rezultatai ir charakteristikos

Pats svarbiausias paieškos variklio vertinimo matas – jo paieškos rezultatų kokybė.

Patirtis su Google įrodė tai, kad sistema gali pateikti geresnius rezultatus nei dauguma

komercinių paieškos variklių daugumai paieškų. Kaip parodyta pavyzdyje, kuris iliustruoja puslapio rangą, nuorodos tekstą ir artumą. Pav.12 pateikta Google sistemos paieškos rezultatai su raktiniu žodžiu „bill clinton“. Šie rezultatai demonstruoja keletą Google savybių. Rezultatai sugrupuojami serverio pagalba. Tai labai padeda, kai reikia kruopščiai patikrinti rezultatų rinkinius. Tam tikras rezultatų skaičius yra iš whitehouse.gov domeno, kuris yra vienas iš labiausiai pagrįstai tikėtinų atliekant tokią paiešką. Šiuo metu, dauguma pagrindinių komercinių paieškos variklių negražina jokių rezultatų iš whitehouse.gov, žymiai mažiau pateikia teisingų. Atkreipkime dėmesį, kad nėra pirmojo rezultato pavadinimo. Tai yra dėl to, kad jis nebuvo išnagrinėtas nagrinėjimo sistemos. Vietoj to, Google rėmėsi nuorodos tekstu, kad nustatytų ar šis


Užklausa: bill clinton

<http://www.whitehouse.gov/>

100.00%  (no date) (0K)


<http://www.whitehouse.gov/>

[Office of the President](#)

99.67%  (Dec 23 1996) (2K)


http://www.whitehouse.gov/WH/EOP/OP/html/OP_Home.html

[Welcome To The White House](#)

99.98%  (Nov 09 1997) (5K)


<http://www.whitehouse.gov/WH/Welcome.html>

[Send Electronic Mail to the President](#)


99.86%  (Jul 14 1997) (5K)

http://www.whitehouse.gov/WH/Mail/html/Mail_President.html


<mailto:president@whitehouse.gov>

99.98% 

<mailto:President@whitehouse.gov>


99.27% 

[The "Unofficial" Bill Clinton](#)

94.06%  (Nov 11 1997) (14K)


<http://zpub.com/un/un-bc.html>

[Bill Clinton Meets The Shrinks](#)

86.27%  (Jun 29 1997) (63K)


<http://zpub.com/un/un-bc9.html>

[President Bill Clinton - The Dark Side](#)

97.27%  (Nov 10 1997) (15K)

<http://www.realchange.org/clinton.htm>

[\\$3 Bill Clinton](#)

94.73%  (no date) (4K)

<http://www.gateway.net/~tjohnson/clinton1.html>

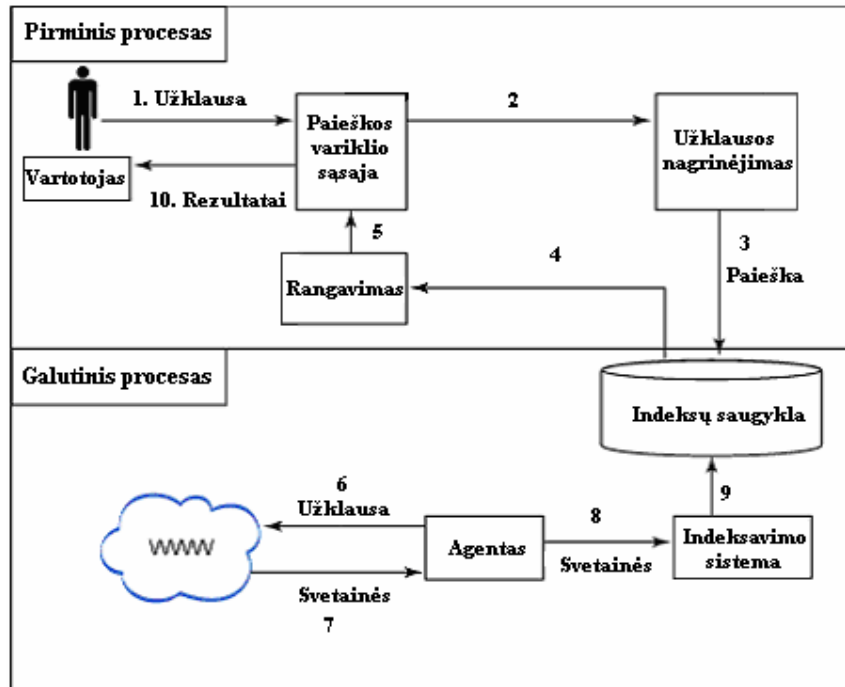
užklauskos atsakymas buvo geras. Panašiai, pentas rezultatas yra elektroninio pašto

Pav. 12 Google paieškos rezultatai

adresas, kuris, be abejo, neišnagrinėtas. Tai taip pat yra nuorodos teksto rezultatas.

Visi šie rezultatai yra pagrįstai aukštos kokybės puslapiai ir galiausiai kontrolė, nei viena nuoroda nebuvo sugadinta. Tai yra dėl to, kad visi jie turi aukštą puslapio rangą. Puslapio rangai yra raudoni procentiniai dydžiai kartu su (išilgai) stulpelinėmis diagramomis. Galiausiai, nėra nei vieno rezultato apie Bill nei apie Clinton ar apie Clinton nei apie Bill. Tai yra todėl, kad sistemos kūrėjai didelį dėmesį skyrė žodžių paplitimo artumui nustatyti. [9]

Taigi, išnagrinėjus paieškos variklio detalią architektūrą, galime sudaryti bendrą vartotojo užklausos procesų grafiką.



Pav. 13 Bendra paieškos sistemos architektūra

Pradinis procesas

1. Pirmiausia vartotojas įveda užklausos raktinį terminą;
2. Užklausa išnagrinėjama, lingvistinės nagrinėjimo sistemos;
3. Išnagrinėta ir suindeksuota užklausa pateikiama indeksų saugyklai;
4. Saugykloje duomenys atrenkami ir dokumentų sąrašas pateikiamas rangavimo sistemai, kuri suskirsto dokumentus pagal rangą;
5. Rezultatai pateikiami vartotojui;

Galinis procesas

- 6-7 Agentas išnagrinėja interneto svetaines;
- 8. Svetainės suskirstomos indeksavimo sistemos pagalba;
- 9. Svetainių sąrašas patalpinamas saugykloje.

4. PAIEŠKOS VARIKLIAI

Paieškos variklis – tai sistema, kuri ieško informacijos visame žiniatinklyje ir FTP serveriuose. Paieškos rezultatai dažniausiai pateikiami rezultatų sąrašais. Pateikta informacija gali būti sudaryta iš internetinių puslapių, paveikslėlių, informacijos ir kitos rūšies bylų. Kai kurie paieškos varikliai vykdo duomenų gavybą, kad pagerintų paieškos duomenų kokybę.

4.1 Geriausia paieškos sistema

Interneto platybėse galima rasti įvairiausių paieškos variklių, kurie vykdys paiešką žiniatinklyje pagal užklauso raktinius žodžius. Bet iškyla vartotojiškas klausimas: „Kuris paieškos variklis labiausiai tenkina mano poreikius?“

Štai keletas patarimų, kurie galėtų pagelbėti vartotojui pasirinkti paieškos variklį labiausiai tenkinantį jo poreikius:

Pagrindinių paieškos variklių sąrašo pateikimas. Visi paieškos varikliai: vaizdo paieškos, knygų, dokumentų paieškos ir daugiau, galime rasti interneto svetainėje <http://websearch.about.com>. Tereikia naršyklėje įvesti minėtą adresą ir paieškos laukelyje pateikti užklausa: ”Search engine list” ir spragtelėti aukščiausiai esančią nuorodą. [10]

- Dešimt populiariausių metodų, pateikiančių daugialypes terpes internete. Panaudokime šiuos paieškos raktinius žodžius minėtoje svetainėje: „Multimedia on the web”, kad surastume muzikos, video įrašus ar kitas daugialypes terpes. Sistema pateikia sąrašą paieškos variklių, kurie skirti multimedijos paieškoms internete.

- Šimtas paieškos variklių, užklausa „100 Search Engines”, pateikia šimtą šiai dienai geriausių paieškos variklių sąrašas.

- Dešimt populiariausių paieškos variklių knygoms rasti, užklausa ”Book search engines”, pateikiama knygų paieškos sistemos. Kiekvienas vartotojas tikrai išsirinks jam tinkamą paieškos sistemą, ieškodamas naujų, naudotų ar audio knygų.

- Google paieškos komandos užklausa: „Google cheat sheet”, panaudoję šią užklausa rasime sąrašą su komandų sintaksėmis, kurias panaudoję minėtoje paieškos sistemoje galime sukonkretinti vykdomą paiešką ir taip išgauti tikslius paieškos rezultatus.

Visas šias paieškos sistemas galime rasti minėtoje interneto svetainėje, panaudoję minėtas užklausas. Tai turėtų sutaupyti nemažai laiko pradantiems ”internautams”, kurie dar nėra išsamiai susipažinę su internetu ir jo galimybėmis. Sistema pateikia gan platų sąrašą paieškos sistemų, kurios vykdo specializuotą paiešką. Yra tik viena problema su kuria gali susidurti vartotojai – tai, kad sistemos paieškos užklauso turi būti vykdomos anglų kalba.

Beabejo, šias užklausas kaip raktinius žodžius galime pateikti ir kituose populiariuose paieškos varikliuose, bet rezultatai gali būti skirtingi ir ne tokie konkretūs.

4. 2 Paieškos sistemų palyginimas

Šiame skyrelyje palyginsime tris skirtingos struktūros paieškos variklius: Google, Hacia ir WolframAlpha. Google paieškos variklis paremtas puslapių rangavimo ir indeksavimo algoritmais. Hacia paremtas semantiniais ontologijų sprendimais. WolframAlpha žinių skaičiavimo paieškos variklis, kaip pagrindą naudojantis skaičiavimo paketą Mathematica. Kadangi Google architektūra detaliai išnagrinėta prieš tai buvusiam skyriuje, jos aprašymo nebetalpinama. Pristatomi du konkuruojantys paieškos varikliai:

4. 3 Hacia

Hacia – tai semantinės technologijos paieškos variklis, naudojantis QDEX technologiją, kaip naujos alternatyvos algoritmą, semantiniam puslapių rangavimui. Algoritmas susideda iš ontologinės semantikos, kompiuterinės lingvistikos ir matematinių skaičiavimų. Variklis sukurtas 2004 metais.

Semantinės technologijos suteikia galimybes, kurios yra naujos paieškos aplinkoje. Jos yra :

- Kategorizavimas įprastiniame paieškos variklyje.

Trumpoms užklausoms kaip vėžys ar Lietuvos prezidentai, Hacia pateikia paieškos rezultatus suskirstytus kategorijomis. Kiekviena galerija turi vidutiniškai 10 kategorijų, tai lygu 10 užklausų įprastame paieškos variklyje.

- Lygiagretumas

Svarbus gebėjimas paieškos varikliui, lygiagretumas leidžia Hacia varikliui pateikti rezultatus naudojant ekvivalentiškumo sąlygas tokias kaip: „gydyti = vaistai“. Šie asociacijos metodai naudojami su tinkamais reikšmių dviprasmiškumo pašalinimo funkcijomis.

- Generalizacija

Dauguma užklausų paieškos variklyje reikalauja generalizacijos galimybių. Kaip pvz. užklausoje „kuris automobilis nuvažiuoja daugiausiai kilometrų“, terminas „automobilis“ turėtų būti atpažintas kaip pagrindinis terminas prekiniam ženklui pateikti, kaip Toyota, Ford, Volkswagen. Paieškos rezultatai turėtų apimti automobilių koncerno prekinį ženklą, o ne raktinį žodį automobilis.

- Paryškinimas

Sudėtingoms ilgomis užklausoms, HAKIA paryškina susijusias frazes, kurios labiausiai atitinka užklauso reikšmę.[11]

4.4 WolframAlpha

WolframAlpha – tai paieškos sistema, paremta žinių skaičiavimo metodu. Šis variklis naudoja visiškai naują būdą išgaunantį žinias ir pateikiantį atsakymus į užklauso. Metodas remiasi ne tik paieška internete, bet ir atlieka dinaminis skaičiavimus paremtus didžiuliais duomenų, algoritmų ir metodų kiekiais.

Pagrindinis sistemos tikslas – paversti visas sistemingas žinias greitai suskaičiuojamas ir prieinamas kiekvienam vartotojui. Sistema siekia surinkti visus objektyvius duomenis, apdoroti kiekvieną gerai žinoma modelį, metodą ir algoritmą. Taip padaryti juos prieinamus sistemos skaičiavimams, kad pateiktų vienuose informaciją, kuri patikimai išreikštų aiškų atsakymą į faktinę užklauso.

Visa WolframAlpha sistema sukurta remiantis sisteminiu paketu Mathematica. Ši sistema vaidina esmines roles paieškos vaikyje. Mathematica naudoja bendrą simbolių kalbą, ja aprūpina darbalaukį, kuriame yra pateikiama visa WolframAlpha įvairiarūšė informacija ir realizuoti visi sistemos gebėjimai. [12]

Visos trys sistemos iš išorės per daug nesiskiria, visos turi: paieškos įvedimo laukelį ir vykdymo mygtukas.

Tam, kad tiksliau įvertume sistemas sudarysime vertinimo kriterijų lentelę susidedančią iš: rezultatų pateikimo laiko, rezultatų į užklauso pateikimo tikslumo ir rezultatų pateikimo formos.

4.5 Eksperimentas

Tarkime norime sužinoti Lietuvos gyventojų populiaciją. Į užklauso laukelį įvedame: „What is the population of Lithuania?“. (liet. „Kas yra Lietuvos sostinė“)

1. Google pirmiausia pateikė aiškų atsakymą: „Lithuania Population“ spėjimas yra 3,555,179“

Antroje nuoroje pateikė nuorodą: <http://en.wikipedia.org/wiki/Lithuania>, kurios aprašyme parašyta: the *population of Lithuania* stands at 3349900. Bet atsidarius patį puslapį trumpesniame valstybės aprašyme teigiama, jog Lietuvos populiacija 2011m. 3,225,694.

HAKIA taip pat pateikė nuorodą į Wikipedia: http://en.wikipedia.org/wiki/Demographics_of_Lithuania, bet nuorodos apraše neatsispindėjo gyventojų skaičius (reikėjo spragtelėti nuorodą). Nuoroje pateikiami Lietuvos demografiniai duomenys iki 2010 metų. 2010m 3,329,000 gyventojų Tik iš šeštos nuorodos

aprašo matome: The population of Lithuania in 2003 was estimated by the United Nations at 3,444,000.

WolframAlpha pateikė išsamų atsakymą pateikė tam pačiam lange. 3,26million 2010m.

Taigi visos trys sistemos pateikė skirtingus atsakymus. Nors Google ir greičiausiai susidorojo su užduotimi, bet pateikti rezultatai nebuvo tiksliausi. Hacia pateikė nuorodą, kurią atidarius sužinojome norimą informaciją. WolframAlpha šiek tiek užtruko kol pateikė atsakymą, bet atsakymas buvo konkretus ir aiškus.

Įvertinimas: 0-0-1

2. Su pirma užduotimi paieškos sistemos susitvarkė nesunkiai. Užduotį šiek tiek pasunkiname ir į paieškos lauką įvedame: „What is temporary capital of Lithuania?“ (*liet.* „Kas yra Lietuvos laikinoji sostinė“)

Google ir Hacia pateikia į tas pačias Wikipedia nuorodas apie Kauną: http://en.wikipedia.org/wiki/Temporary_capital_of_Lithuania . Taip pat iš nuorodų aprašymo galime perskaityti, kad laikinoji Lietuvos sostinė tarpukariu buvo Kaunas.

WolframAlpha sistema pateikė klaidingus rezultatus, pateikdama visą informaciją apie Vilnių.

Įvertinimas: 1-1-0

3. Sekančiai užklausi pasirinktas ne geografinis ar istorinis terminas, bet nesudėtingas juokelis, klausimas: „Why did the chicken cross the road?“ (*liet.* „Kodėl viščiukas perėjo kelia?“).

Visos trys sistemos su užduotimi nesunkiai susitvarkė.

Google ir Hacia pateikė nuorodas į tas pačias Wikipedia nuorodas: http://en.wikipedia.org/wiki/Why_did_the_chicken_cross_the_road%3F , kurių aprašymuose matomas teisingas atsakymas: „To get to the other side“ (*liet.* „Kad nusigautų į kitą kelio pusę“).

Sistema WolframAlpha iškart pateikė aiškų atsakymą: „To get to the other side“.

Kadangi visos trys sistemos su užduotimi puikiai susitvarkė, įvertinimas vienodas.

Įvertinimas:1-1-1

4. Sekančios eksperimentinės užklauskos užduotis, panaudoti lietuviškus simbolius kaip raktinius žodžius užklausoje: „What is Šiaulių Šiauliai?“ (*liet.* „Kas yra Šiaulių Šiauliai?“)

Google pateikdama nuorodą į krepšinio klubo Šiaulių Šiauliai internetinę nuorodą, su aprašymu, kad tai krepšinio klubas.

Hacia pateikė nuorodą į Wikipedia apie Šiaulių miestą.

WolframAlpha užduoties neįveikė, nepateikė jokio susijusio atsakymo, net ir nenaudojant Lietuviškų simbolių užklausoje.

Geriausiai šioje užduotyje susitvarkė Google.

Įvertinimas: 1-0-0

5. Klausimas iš tikslųjų mokslų: „How far away is the moon?“. (*liet.* „Kaip toli mėnulis?“)

Google pateikė nuorodą: http://coolcosmos.ipac.caltech.edu/cosmic_kids/AskKids/moondist.shtml, iš kurios aprašo matome, kad mėnulis nuo žemės nutolęs 384400 km. Atidarius nuorodą galime perskaityti, kad atstumas nuo Žemės iki mėnulio keičiasi jam judant orbitoje.

Hakia pirmoje nuorodoje pateikė <http://www.myspace.com/mfaofficial> svetainę, kurioje pateikiama „MOON FAR AWAY“ multimedijos svetainė. Tik trečioje nuorodoje pateikiama informacija apie mėnulio atstumą nuo Žemės.

WolframAlpha pateikia aiškų rezultatą 365300km, atsižvelgiant, kad atstumas gali keistis šie duomenys nėra visiškai tikslūs, bet į pateiktą užklausą aiškiai atsakyta.

Įvertinimas:1-0-1

6. Kad įvertintume paieškos sistemas lingvistinės semantikos prasme, pateikiame užklausą: „Is the Eiffel tower bigger than Pisa tower?“ (*liet.* „Ar Eifelio bokštas aukštesnis nei Pizos bokštas?“)

Google pirmoje nuorodoje pateikė informaciją apie Eifelio bokšto aukštį, bet nieko neužsiminė apie Pizos bokštą. Tačiau antroje nuorodoje: http://www.trueknowledge.com/q/is_the_eiffel_tower_higher_than_the_tower_of_pisa, pateiktas aiškus atsakymas „Yes“.

Hakia šiek tiek nustebino pateiktama nuorodą: <http://www.chacha.com/question/what-is-bigger-the-eiffel-tower-or-the-leaning-tower-of-pisa>, kuria atidarius parašyta : „The Eiffel Tower is 990 feet tall & the Leaning Tower of Pisa is 182 feet tall. So the Eiffel Tower is much taller“. (*liet.* „Eifelio bokštas yra 900 pėdų aukščio, o Pizos bokštas 182 pėdų aukščio. Taigi Eifelis daug aukštesnis“).

WolframAlpha nepateikė tikslaus atsakymo į klausimą, bet pateikė išsamia informaciją apie Eifelio bokštą.

Taigi, kad įvertintume semantikos prasmę, pateikiame ta pačią užklausą iš kitos pusės: „Is the Pisa tower bigger than Eiffel tower?“ (*liet.* „Ar Pizos bokštas aukštesnis nei Eifelio bokštas?“).

Google

pateikė

nuorodą:

http://www.trueknowledge.com/q/is_the_tower_of_pisa_taller_than_the_eiffel_tower, kurioje aiškus atsakymas į klausimą – NO.

Hakia pateikė: <http://www.chacha.com/question/what-is-bigger-the-eiffel-tower-or-the-leaning-tower-of-pisa> tą pačią nuorodą į prieš tai užduotą klausimą, kurioje tas pats atsakymas „The Eiffel Tower is 990 feet tall & the Leaning Tower of Pisa is 182 feet tall. So the Eiffel Tower is much taller“. (liet. „Ar Pizos bokštas aukštesnis nei Eifelio bokštas?“).

WolframAlpha pateikė tą pačią informaciją apie Eifelio bokštą, jokių duomenų apie Pizos bokštą.

Sudėtinga įvertinti pirmas dvi sistemas kurios iš esmės atsakė teisingai. Tačiau Google pateikė aiškius atsakymus taip arba ne.

Įvertinimas: 1-0,5-0

Bendri rezultatai:

Lentelė nr 2. Eksperimento rezultatai

Nr	Google	Hakia	WolframAlpha
1	0	0	1
2	1	1	0
3	1	1	1
4	1	0	0
5	1	0	1
6	1	0,5	0
Viso	5	2,5	3

Iš atliktų užklausų eksperimento matome, kad visi trys varikliai gali konkuruoti tarpusavyje. Kaip ir tikėtasi populiariausias paieškos variklis gavo didžiausią įvertinimą. Google turi įvairių užsienio kalbų palaikymą, stiprus informacijos rangavimo algoritmas pateikia daug informacijos. Pagrindinis minusas būtų toks, kad sistema pateikia gal būt per daug įvairiausių duomenų į užklausą, todėl neradus norimos informacijos pirmuose nuorodose darosi painu atskirti naudingą informaciją. Ne toks populiarus variklis Hakia niekuo nenustebino ir neaplenkė konkurentų. Nors ieškomos informacijos paryškimas šiek tiek palengvina atsirinkti norimus duomenis, tačiau rezultatų kokybė dar nėra labai aukšto lygio.

Šio variklio semantinių algoritmų panaudojimas dar nėra išvystytas iki Google lygio. Didžiasias WolframAlpha minusas – skirtingų užsienio kalbų nepalaikymas. Taip pat šio variklio žinių bagažas dar nėra pilnai užbaigtas. Nepaisant to ši sistema pateikia išsamiai konkretų atsakymą apie užklausą. Jei norima atlikti matematinės operacijas, sužinoti tikslus duomenis ar konkretų atsakymą iš gamtos mokslų ar ekonominės srities, šis paieškos variklis puikiai tenkins vartotojų poreikius.

Lentelė nr. 3 Galutiniai sistemų įvertinimo rezultatai

	Rezultatų pateikimo laikas	Rezultatų į užklausas tikslumas	Rezultatų pateikimo forma	Bendra
Google	10	8	8	26
Hakia	10	6	8	24
WolframAlpha	8	4	10	22

Pirmų dviejų paieškos varikliu rezultatų pateikimo laikas yra geriausias. Rezultatai pateikiami akimirksniu. WolframAlpha šiek tiek užtrunka kol pateikia rezultatą. Iš atlikto testo matome, kad tiksliausiai į užklauso rezultatus atsakė Google. Pirmų dviejų variklių rezultatų pateikimo forma gana panaši, pateikiamos nuorodos į internetines svetaines. Trečioji sistema pateikia aiškius rezultatus tame pačiame lange, todėl jai skiriamas aukščiausias įvertinimas.

5. INTERNETO NARŠYMAS

5.1 Interneto naršymo problemos

Pasak, Hsiangchu Lai, Tzyy-Ching Yang, prieš įgyvendinant intelektinius sprendimus sistemoje, pirmiausia išnagrinėjamos pagrindinės naršymo problemos, su kuriomis susiduria lankytojai. Tie patys lankytojai suskirstomi į dvi grupes:

- Nauji lankytojai – kurie į puslapį ateina pirmą kartą ir jiems puslapio struktūra ir turinys prieš tai nebuvo žinomi.
- Grįžtantys lankytojai – kurie ateina į puslapį nebe pirmą kartą ir kiekvieno apsilankymo metu jie tikisi surasti ką nors naujo.

Taip pat patį apsilankymą galima suskaidyti į dvi dalis:

- Betikslis naršymas – kai lankytojas į tinklapį užklysta atsitiktinai.
- Tikslingas naršymas – kai vartotojas tinklapyje ieško jam reikiamos informacijos.

Toliau pateiktos apibendrinamosios lentelės (4 ir 5). *Lentelė Nr.4* apibūdina internetinio naršymo problemas, o *lentelė Nr.5* sumuoja ištirtus reikalavimus interneto puslapiui, kad pagerintų tinklapio naršymą. Toks sistematiškas pavaizdavimas gali tapti pagrindu, darant tinklapiui vartotojo sąsajos architektūrą, kuri gali pateikti išsamią vartotojo sąsaja visų tipų vartotojams, nepriklausomai nuo to kaip dažnai jie lankosi tinklapyje, kiek yra su juo susipažinę ir ar rado tinklapį atsitiktinai ar tikslingai.[13]

Lentelė nr. 4. Naršymo problemos

Naršymo tikslas		Tinklapių pažinimas	
		Vartotojas nesusipažinęs	Vartotojas susipažinęs
Betikslis naršymas		<ul style="list-style-type: none"> Laiko švaistymas 	<ul style="list-style-type: none"> Laiko švaistymas
		<ul style="list-style-type: none"> Pasiklydimas 	<ul style="list-style-type: none"> Pasiklydimas
		<ul style="list-style-type: none"> Praleidžiami svarbiausi puslapiai 	<ul style="list-style-type: none"> Praleidžiami svarbiausi puslapiai
		<ul style="list-style-type: none"> Išsėina iš tinklapių, neperžiūrėjęs visų puslapių 	<ul style="list-style-type: none"> Praranda susidomėjimą dėl naujovių nebuvimo Praleidžia kažką svarbaus
Tikslingas naršymas	Reguliarus		<ul style="list-style-type: none"> Nenašus naršymas Praleidžiama kažkas svarbaus Praranda susidomėjimą puslapiu dėl neinformatyvaus turinio
		<ul style="list-style-type: none"> Neefektyvi paieškos sistema 	<ul style="list-style-type: none"> Neefektyvi paieškos sistema
		<ul style="list-style-type: none"> Nepatenkinami paieškos rezultatai 	<ul style="list-style-type: none"> Nepatenkinami paieškos rezultatai
	Specialus		

Lentelė nr. 5 Reikalavimai tinklapiui

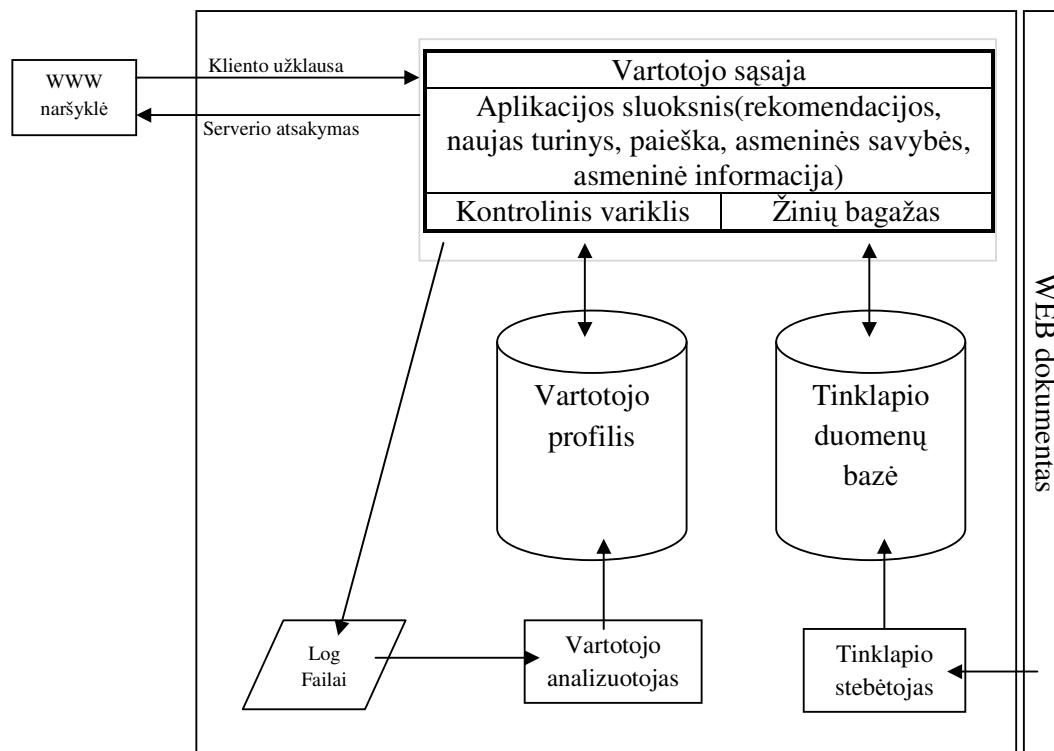
Naršymo tiklas		Tinklapio pažinimas		
		Vartotojas nesusipažinęs	Vartotojas susipažinęs	
Betikslis naršymas		<ul style="list-style-type: none"> Bendras tinklapio struktūros žemėlapis 	<ul style="list-style-type: none"> Bendras tinklapio struktūros žemėlapis 	
		<ul style="list-style-type: none"> Pėdsakų žemėlapis 	<ul style="list-style-type: none"> Pėdsakų žemėlapis 	
		<ul style="list-style-type: none"> Pagrindinių nustatymų laukas 	<ul style="list-style-type: none"> Pagrindinių nustatymų laukas 	
		<ul style="list-style-type: none"> Populiariausių puslapių ar produktų rekomendavimas 	<ul style="list-style-type: none"> Populiariausių puslapių ar produktų rekomendavimas 	
		<ul style="list-style-type: none"> Vertingas turinys 	<ul style="list-style-type: none"> Vertingas turinys 	
		<ul style="list-style-type: none"> Lankstumas 	<ul style="list-style-type: none"> Lankstumas 	
		<ul style="list-style-type: none"> Naudingiausio turinio paryškinimas 	<ul style="list-style-type: none"> Naudingiausio turinio paryškinimas Naujo turinio paryškinimas Detalus pagrindinis puslapis 	
Tikslingas naršymas	Reguliarus	/		<ul style="list-style-type: none"> Detalus pagrindinis puslapis
				<ul style="list-style-type: none"> Naujo turinio paryškinimas
				<ul style="list-style-type: none"> Informacijos filtravimas
	Specialus	<ul style="list-style-type: none"> Bendras tinklapio struktūros žemėlapis 	<ul style="list-style-type: none"> Bendras tinklapio struktūros žemėlapis 	
		<ul style="list-style-type: none"> Intelektualus paieškos variklis 	<ul style="list-style-type: none"> Intelektualus paieškos variklis 	
<ul style="list-style-type: none"> Numatytų raktažodžių sąrašas 		<ul style="list-style-type: none"> Numatytų raktažodžių sąrašas 		
	<ul style="list-style-type: none"> Populiariausių tinklapių nuorodų rekomendavimas 	<ul style="list-style-type: none"> Populiarių puslapių nuorodos Einamoji asmeninė būseną 		

5. 2 Tinklapio architektūros tobulinimas

Remiantis lentele nr. 5 galime teigti, kad pagrindinės reikalavimų charakteristikos tinklapiui yra: pateikti vertingą turinį, tinklapio struktūros žemėlapi ir suteikti galimybę vartotojams patogiau ir sparčiau naršyti internete. Tačiau, to nepakanka, kad aprūpintų interneto vartotojus nuostabia naršymo aplinka, nebent tinklapis turėtų įdiegtus intelektualius gebėjimus, tokius kaip pavaizduoti lentelėje nr.6. Tam, kad tinklapis turėtų intelektualius gebėjimus, jis turėtų įgyti keletą skirtingų rūšių intelektualių agentų.

Lentelė nr 6. Intelektualūs agentai pagerinantys naršymą

Reikalingi intelektualiniai gebėjimai		Reikalingi intelektualūs agentai
Lankytojams nesusipažinusiems su tinklapiu	<ul style="list-style-type: none"> • Rekomenduoti populiariausias nuorodas 	Rekomendacijų agentas
	<ul style="list-style-type: none"> • Paryškinti vertingiausią turinį 	
	<ul style="list-style-type: none"> • Pateikti galimus paieškos raktažodžius 	Paieškos agentas
	<ul style="list-style-type: none"> • Vykdyti intelektualią paiešką 	
Lankytojams susipažinusiems su tinklapiu	<ul style="list-style-type: none"> • Rekomenduoti populiariausius puslapius 	Rekomendacijų agentas
	<ul style="list-style-type: none"> • Paryškinti vertingiausią turinį 	
	<ul style="list-style-type: none"> • Pateikti galimus paieškos raktažodžius 	Paieškos agentas
	<ul style="list-style-type: none"> • Tiekti intelektualią paiešką 	
	<ul style="list-style-type: none"> • Paryškinti naują turinį 	Naujo turinio agentas
	<ul style="list-style-type: none"> • Pateikti individualų turinį 	Asmeninių savybių agentas
	<ul style="list-style-type: none"> • Stebėti asmeninius vartotojo veiksmus 	Asmeninės būklės agentas



Pav. 14 Sistemos architektūra su realizuotais intelektiniais sprendimais [13]

Šioje architektūroje pateikti penkių rūšių agentai: rekomendacijų agentas, naujo turinio agentas, paieškos agentas, asmeninių savybių agentas ir asmeninės būklės agentas. Tam, kad būtų sudarytos sąlygos šiems agentams gerai veikti, būtina įdiegti vartotojų analizatorių skirtą vartotojų veiksmų stebėjimui. Taip pat tinklapio stebėtoją tam, kad prižiūrėtų vartotojų duomenų bases.

5.3 Vartotojų analizavimo sistema ir vartotojo profilis

Versle yra žinoma, kad kuo labiau yra suprantami kliento poreikiai, tuo geresnes paslaugas jam galima pasiūlyti. Tas pats galioja ir tinklapyje, tik šiuo atveju klientas yra lankytojas. Todėl agentai turi patenkinti lankytojo poreikius. Tam pasiekti pirmiausia reikia rinkti ir analizuoti lankytojo elgseną. Taigi „vartotojo analizuotojas“ ir vartotojo profilis tam ir yra skirtas. Vartotojo profilis yra duomenų bazė, kurioje kaupiama informacija apie vartotoją. Vartotojo informaciją galima suskirstyti į kintamą ir nekintamą.

Nekintama informacija – visa informacija apie vartotojo charakteristiką (pvz: vardas, amžius, lytis, darbas, miestas) ir asmeninius tinklapių nustatymus. Tokia informacija keičiasi retai.

Kintama informacija keičiasi pastoviai, kai tiks vartotojas apsilanko tinklapyje. Kintama informacija gali susidėti iš:

- Pirkimų istorija;
- Naršymo istorija, produkto peržiūros istorija;
- Įvertinimų istorija;
- Apsilankymo laikas ir trukmė.

Visi šie duomenys gali būti analizuojami ir taip tampa pagrindinis informacijos šaltinis apie unikalų vartotoją.

5. 4 Tinklapio stebėjimo sistema ir „tinklapio duomenų bazė“

Sudominti ir suprasti vartotoją yra tik pradžia. Galutinis tikslas yra išsiaiškinti kuo domisi ir kokio tipo yra vartotojas. Taip pat papildomai tinklapis turi pasirūpinti, kad naršymas būtų efektyvus ir kiek įmanoma sumažinti riziką, kad vartotoją dominanti informacija nepasimestų. Todėl reikia pilnai suprasti ir pažinti visą tinklapio turinį. Tinklapio duomenų bazė pilnai atlieka šią funkciją. Ji išsaugo informaciją apie kiekvieną tinklapio elementą (naujieną, produktą, tekstinį puslapį, straipsnius). Pats tinklapio elementas ir elementų ryšiai yra du didžiausi informacijos tipai duomenų bazėje. Apie tinklapio elementus išsaugoma informacija tokia kaip: URL, paskutinio keitimo data, pavadinimas, santrauka ir raktiniai žodžiai. Kita svarbi informacija, elementų ryšiai nurodantys santykius tarp jų. Santykių tipas gali būti medis, grafikas. Tokia informacija labai padeda gerinant paieškos funkcijas.

Taigi tinklapio stebėjimo sistema stebi tinklapio elementų pasikeitimus ir visus pakitimus išsaugo atnaujindama tinklapio duomenų bazę.

Lentelė nr. 7 Intelentualių agentų tikslai ir uždaviniai

Rekomendacijų agentas	
Tikslas	Išanalizavus tinklapio, turinį pateikti produktų ar paslaugų rekomendacijas
Uždaviniai	Išnagrinėti log bylas
	Rekomenduoti "karštas nuorodas", kurios turi didžiausią rangą.
	Rekomenduoti skirtingas nuorodas, skirtingų poreikių grupių vartotojams
	Paryškinti naujausia arba svarbiausia svetainės informaciją
	Analizuoti kuo domisi skirtingų poreikių vartotojai ir pateikti jiems rūpimą informaciją
Naujo turinio agentas	
Tikslas	Atkreipti vartotojų dėmesį į naujus produktus
Uždaviniai	Išgauti naujai atsiradusia informaciją iš duomenų bazės ir paryškinti ją visiems vartotojams
	Išgauti naują informaciją iš duomenų bazės, kuri būtų dar nežinoma konkrečiam vartotojui ir ją paryškinti.

Paieškos agentas	
Tikslas	Vykdyti išsamia produktų ar paslaugų paiešką
Uždaviniai	Aprūpinti vartotoją galimu raktinių žodžių sąrašu
	Patarti vartotojui atsižvelgiant į jo užklauso terminus
	Atsiminti vartotojo vykdytas paieškas ir pateikti pasiūlymus tolimesnėje paieškos sesijoje
Asmeninių savybių agentas	
Tikslas	Atsižvelgiant į vartotojo naršymą, siūlyti jam asmeniškai pritaikytą naršymo aplinką
Uždaviniai	Analizuoti vartotojo veiksmus svetainėje
	Analizuoti svetainės duomenų bazę ir vartotojo profilį ir pateikti vartotojui asmeniškai pritaikytą svetainės naršymo aplinką
Asmeninės informacijos agentas	
Tikslas	Tiekti atnaujintą vartotojo profilio informaciją kiekvienam registruotam lankytojui
Uždaviniai	Stebėti visas svetainės transakcijas ir atsižvelgiant į tai atnaujinti vartotojo profilio informaciją

6. SEMANTINIS INTERNETAS (WEB3) KONCEPCIJA

6.1 Įvadas

Pasaulinis kompiuterių tinklas www yra plačiai naudojamas, kaip universali aplinka informacijos keitimuisi. Tačiau sąveika pasauliniame tinkle vis dar ribojama dėl įvairiarūšės informacijos.

Tam, kad būtų sukurta intelektualiai interneto naršymo sistema, reikalingas intelektualus internetas. Jei besikeičiantis interneto turinys gebėtų suprasti koks jis buvo anksčiau, tuomet nebereikėtų greituju nuorodu. Naršyklė galėtų nukreipti vartotoją tiesiai į norimo turinio informacijos puslapį.

Ontologija – oficiali, tiksli koncepcijų kūrimo specifikacija, buvo pasiūlyta kaip problemos sprendimo būdas ir viena iš pagrindinių sudedamųjų dalių Web3 technologijoje. Populiarėjant ontologijoms, ontologijų funkcijoms, siekimas rasti semantinių panašumų tarp panašių elementų skirtingose ontologijose, pritraukė daug mokslinių tyrimų dėmesį iš įvairių sričių. Skirtingos technikos buvo išbandytos ontologijų žymėjime, pvz. analizuojant lingvistinių elementų informaciją ontologijose, nagrinėjant ontologijas kaip struktūrinius grafus, naudojant euristines taisykles ar taikant mašinų mokymo metodus.

Semantinis internetas – tai grupė metodų ir technologijų padedančių mašinoms suprasti informacijos esančios internete „semantikos“ prasmę. Semantinio tinklo terminas sukurtas Tim Berners-Lee. Jis apibūdino semantinį internetą taip: internetas, kurio duomenys gali būti apdoroti mašinų tiesiogiai ir netiesiogiai.

Semantinis tinklas – tai tokių duomenų tinklas, kurių prasmę ir reikšmę gali suprasti kompiuterinės programos, kad galėtų tiksliau juos vykdyti, apdoroti. [14]

6.2 Semantinio interneto veikimas

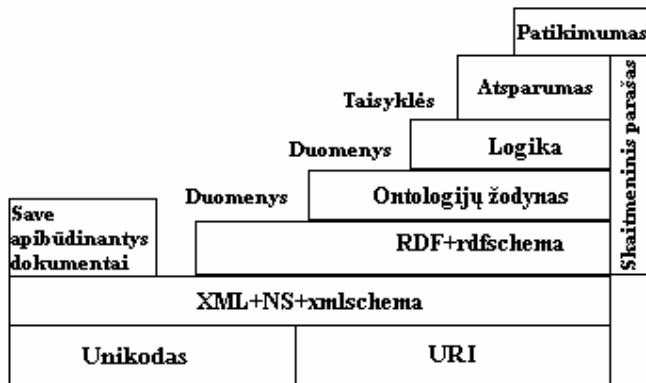
Semantinis internetas pasiūlys kompiuteriams pagalbą „skaityti“ ir naudoti internetą. Pagrindinė mintis, tai nesudėtingi metaduomenys įtraukti į internetinius puslapius, kurie esamą internetą pavers perskaitoma mašina. Tai nesuteiks dirbtinio intelekto ir nepadarys kompiuteriu sąmoningų, bet suteiks mašininų įrankių surasti, pakeisti ir interpretuoti informaciją. Tai esamo interneto praplėtimas, bet ne pakeitimas.

Tai skamba gana abstrakčiai. Kol keletas internetinių svetainių jau naudoja semantinio interneto koncepcijas, daugelis nereikalingų įrankių tėra kūrimo stadijoje. Šiame darbe apžvelgsime įrankių koncepcijas.

Tarkime jūs nusprendėte žiūrėti filmą ir pavalgyti. Esate nusiteikęs žiūrėti komediją ir užkąsti aštraus meksikietiško maisto. Įsijungiate savo kompiuterį, pasileidžiate interneto

naršyklę ir tiesiai Google paieškoje ieškote kino teatro, filmo ir restoranų informacijos. Norite žinoti kokie filmai yra rodomi kino teatruose jūsų mieste. Taip praleidžiate šiek tiek laiko skaitydami kiekvieno filmo siužeto santraukas kol išsirenkate jums tinkamą. Taip pat norite peržiūrėti lankytojų atsiliepimus apie restoranus, kuriuose jie lankėsi. Bendrai jūs aplankote pusę tuzino internetinių svetainių, kol išeinatė pro duris laukan. [15], [16]

6.3 Semantinio žiniatinklio sandara



Pav. 12 Semantinio žiniatinklio sluoksniai

Semantinio žiniatinklio sluoksniai:

1. URI sudaro galimybes kreiptis į subjektus, o Unikodas yra simbolių keitimosi standartas.
2. XML duomenų struktūrų bei jų turinio aprašomoji kalba. Pagrindinė XML kalbos paskirtis yra užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų, dažniausiai sujungtų internetu. Pagrindinis XML kalbos vienetas yra *elementas*.
3. RDF tai pirmas sluoksnis, kuriame informacija tampa suprata kompiuteriams. RDF kalba skirta meta duomenų apdorojimui. Taip pat užtikrina suderinamumą tarp taikomųjų programų, kurios keičiasi „mašinoms“ suprantama informacija internete. RDF dokumentai susideda iš trijų subjektų: resursai, savybės ir žiniaraščio. Resursais gali būti paprasčiausi tinklapiai. RDF kalboje resursai visuomet adresuojami URI pagalba. RDF savybės – tai specifiniai atributai, įvairios charakteristikos. Žiniaraščiais ar objektais vadinamos objekto atributo vertė.
4. Ontologijų žodynas – tai tam tikras ontologijų sąrašas, palengvinantis intelektinės sistemos darbą, leidžiantis palyginti turimas ontologijas su naujai aptiktomis.

5. Logikos sluoksnis sudarytas iš loginių aksiomų. Pritaikant logines išvadas ontologiniuose sprendimuose, galima išgauti naujų žinių iš informacijos, kuri gali būti netiesiogiai susijusi su nagrinėjamu objektu.

6. Atsparumo ir patikimumo sluoksniai. Šie sluoksniai sukurti tam, kad patikrintų semantinio tinklo elementų tinkamumą ir taip kaip teisingai tinklas apdoroja informaciją. [17], [18]

6.4 Web3 analizė

Tarkime mes norime internete nusipirkti “žvaigždžių karų trilogiją” ir mes turime keletą būtinų kriterijų savo pirkiniai. Pirmiausia, norime plačiaekranio vaizdo, DVD ir norime rinkinuko, kuris turi papildomą diską papildomai kaip bonusą. Visų antra, norime žemiausios galimo kainos. Ir galiausiai norime mokėti mažiausią kainą už siuntimo išlaidas, bet tuo pačiu nenorime laukti ilgai siuntinio.



Pav. 13 Kriterijai perkant DVD filmą internetu [19]

Šiuo atveju mums reikės patikrinti skirtingus mažmeninius prekybininkus, lyginant pardavimo kainas ir siuntimo terminus. Taip pat galime peržvelgti internetinį puslapį, kuris palygins kainas ir siuntimo išlaidas keletos prekybininkų vienu metu. Bet kuriuo atveju reikės nemažai virtualaus pasivaikščiojimo, kad rasti reikiamą pirkimo pasiūlymą.

Naudojant semantinį internetą turėsime kitą variantą. Galėsime įvesti savo pageidavimus į kompiuterizuotą agentą, kuris vykdys paiešką internete, suras jums geriausią

variantą ir vykdys jūsų užsakymą. Agentas galėtų atsidaryti finansinę programinę įrangą jūsų kompiuteryje ir įrašytų kiekį pinigų kuriuos išleido, taip pat galėtų pažymėti datą, kada DVD turėtų atkelti. Agentas taip pat mokytųsi mūsų įpročių ir pomėgių. Taigi jei mes turėjome neigiamos patirties perkant iš konkrečios svetainės, agentas nenaudos šios svetainės vėl.

Agentas atliks, tai nežiūrėdamas į paveikslėlius ar skaitydamas apibūdinimus kaip daro žmogus. Sistema ieškodama per meta duomenis, kurie aiškiai identifikuos ką agentas turi žinoti. Meta duomenys yra paprasti mašinų skaitomi duomenys, kurie apibūdina kitus duomenis. Semantiniame internete, meta duomenys yra nematomi žmonėms naršanties internetinius puslapius, bet aiškiai matomi kompiuteriams. Meta duomenys taip pat gali vykdyti sudėtingesnę, koncentruotą paiešką internete, pateikiant daugiau tikslių rezultatų .[19]

6. 4. 1 XML ir RDF žymėjimas

Tarkime norime šį sakinį padaryti suprantamą kompiuteriui: Anakiną Skywalkeris yra Luko Skywalkerio tėtis.

Mums nesunku suprasti šio sakinio reikšmės – Anakiną ir Luką Skywalker'iai abu yra žmonės ir yra giminystės ryšiai tarp jų. Mes žinome, kad tėtis yra vienas iš tėvų ir, kad Lukas Skywalker'is yra sūnus. Bet kompiuteris to nesupranta be papildomos pagalbos. Leisti kompiuteriui suprasti ką šis sakinytis reiškia, mums reikia pridėti mašinoms suprantamą informaciją, kuri apibūdina kas yra Anakin'as ir Luke'as ir kokie jų giminystės ryšiai. Tai prasideda dviejų įrankių pagalba: XML ir RDF.

XML yra žymėjimo kalba tokia kaip hyper teksto žymėjimo kalba HTML. HTML valdo informacijos, kuria matome internete pasirodymą. XML papildo, bet ne pakeičia HTML pridėdama etiketes, kurios apibūdina duomenis. Šios etiketės yra nematomos paprastiems interneto vartotojams, bet suprantamos kompiuteriams. Etiketės jau yra naudojamos internete ir esami botai, kurie renka duomenis iš paieškos variklių, ir galintys suprasti etiketes.

RDF daro tai, kaip parašyta jo pavadinime – naudojant XML etiketes jis aprūpina darbalaukį, kad apibūdintų resursus. Kalbant RDF terminais, beveik viskas pasaulyje yra resursai. Ši sistema suporuoja resursus (bet kokius daiktavardžius, kaip Anakin'as Skywalker'is ar "Žvaigždžių karų trilogija") su specifiniais elementais ar su lokacijos vieta internete, taip kad kompiuteris aiškiai supranta koks tai resursas. Aiškiai identifikuojami resursai apsaugo kompiuterius nuo sumaišymo Anakin'ą Skywalker'į su Sebastian'u Shaw ar su Hayden Christiansen ar su originalia vieno žmogaus "Star Wars" trilogija.

Kad tai įgyvendinti, RDF naudoja trigubą sistemą parašytą kaip XML etiketės, tam kad išreikštų šią informaciją kaip grafą. Ši sistema susideda iš trijų dalykų: subjekto, savybių ir objekto, kurie yra kaip subjektas, veiksmažodis ir tiesioginio objekto sakinytis. Kai kurie šaltiniai vadina juos subjektu, predikatu ir objektu.



Pav. 14 Triguba RDF sistema

Šiame pavyzdyje kompiuteris žino, kad yra du objektai šiame sakinyje ir, kad yra giminystės ryšys tarp jų. Bet kompiuteris nežino kokie objektai yra ar kaip jie susiję vienas su kitu. [20]

6. 4. 2 Kaip atpažinti tinkamą URI?

Net ir su PĮ „Framwork“, kuri aprūpina XML ir RDF, kompiuteriui reikalingas labai aiškus, specifinis kelias, kad suprastų kokie tai resursai. Tam įgyvendinti, RDF naudoja URI, kad nukreiptų kompiuterį į dokumentą ar objektą, kuris atspindi resursus. URI gali pažymėti bet ką internete ir taip pat gali nurodyti objektus, kurie nėra interneto dalis, tokius kaip prietaisai kompiuterizuotuose namuose. El. pašto siuntimo protokolas, FTP ir „Telnet“ adresai yra keletas kitų URI pavyzdžių.

Kaip pavyzdys, naudojami veikėjo puslapiai oficialioje Žvaigždžių karų internetinėje svetainėje kaip jų URI.



Pav. 15 Informacijos elementų trejetukas

RDF programavimo kalba šis pavyzdys turėtų atrodyti taip:

```
<rdf:RDF>
<rdf:Description
rdf:about="http://www.starwars.com/databank/character/anakynskywalker/index.html">
<rdf:type rdf:resource=
" http://www.howstuffworks.com/example/RDF/relationship#fatherof "/>
<depiction rdf:resource=
"http://www.starwars.com/databank/character/lukeskywalker/index.html "/>
</rdf:Description>
</rdf:RDF>
```

Dabar kompiuteris žino, kad subjektas ir objektas yra Anakin Skywalker, jis yra esybė reprezentuota pirmo URI ir Luke Skywalker yra esybė reprezentuota antro. Pastebime, kad vidurinė URI mūsų trejetuke, skirta ypatybėms, nepažymi Žvaigždžių karų puslapio. Vietoj to, ji pažymi dokumentą esantį „HowStuffWorks“ serveryje. Jei tas puslapis realiai egzistavo, tai jis būtų mūsų XML informacijos vardas.

Skirtingai nuo HTML, kuris naudoja etikečių standartą pabrėžimui, XML neturi standartinių etikečių. Tai yra naudinga, nes leidžia kūrėjams sukurti unikalias etiketes skirtas specifiniams tikslams.

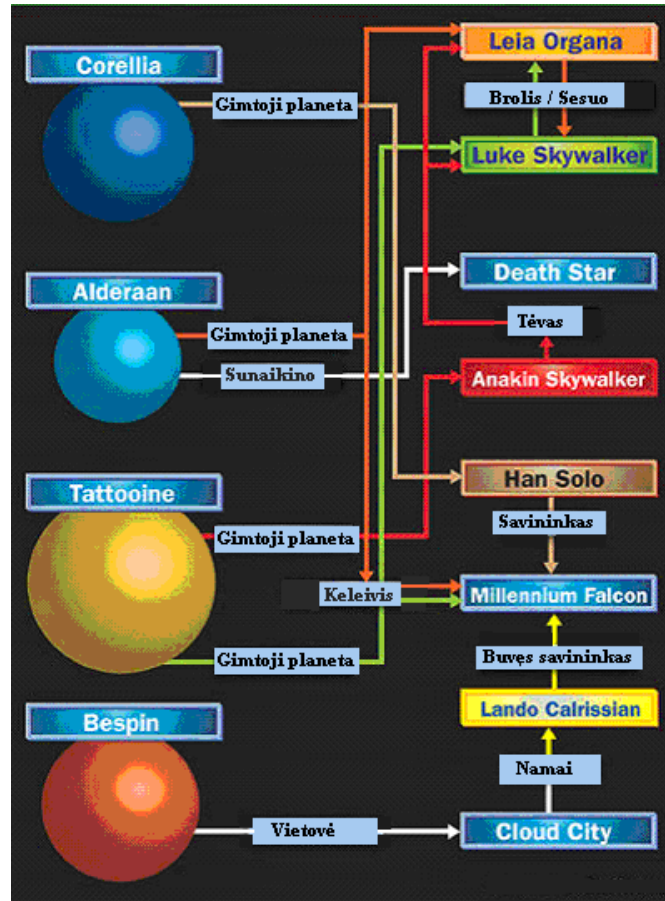
6. 4. 3 Kalbos ir žodynai: RDFS, OWL

Dar viena kliūtis semantiniam internetui yra ta, kad kompiuteris neturi žodyno, kokį turi žmonės. Mes naudojame žodyną visą savo gyvenimą, tai turbūt lengva mums pastebėti ryšius tarp skirtingų žodžių ar konceptų taip suprasti konteksto reikšmes. Deja, bet kas negali įteikti kompiuteriui žodyno, kalendoriaus ar enciklopedijų rinkinio ir leisti kompiuteriui mokytis pačiam. Tam, kad suprastų, ką reiškia žodis ir koks jo sąryšis yra tarp kitų žodžių, kompiuteris privalo turėti dokumentus, kurie apibūdina visus žodžius ir logiką, kad sudarytų reikiamus ryšius.

Semantiniame internete, tai ateina iš dokumento, kurį apibūdina XML ir RDF, taip pat ir iš ontologijų. Tai yra du tarpusavyje susiję įrankiai padedantys kompiuteriui suprasti žmogišką žodyną. Ontologija – nesudėtingas žodynėlis, kuris apibūdina objektus ir kaip jie susiję vienas su kitu. Schemata yra informacijos organizavimo metodas. Kaip ir su RDF etiketėmis, prieiga į schematą ir ontologijas yra įtraukta į dokumentus kaip meta duomenys. Dokumento kūrėjas privalo deklaruoti, kurios ontologijos yra paminėtos pradžioje dokumento.

Schemata ir ontologijos įrankiai naudoti semantiniame internete įtraukia:

- RDF žodyno kalbos apibūdinimo schema – RDFS prideda klases, subklases ir ypatybes į resursus, kuriant pagrindinę kalbą. Kaip pavyzdys, resursai Dagobah yra klasės planeta subklasė. Dagobah ypatybės gali būti – pelkėta. Dagobah – planeta vaizduota filme Žvaigždžių karai.
- Ontologijų kūrimo kalba – OWL, sudėtingiausias lygmuo, formalizuojantis ontologijas, apibūdinantis ryšius tarp klasių ir naudojantis logiką, kad padarytų išvadą. Sistema gali konstruoti naujas klases kaip pagrindą naudojant turimą informaciją.



Pav. 16 Nedidelio kiekio resursų ir ryšių pavyzdys, kurie galėtų būti rasti Žvaigždžių karų ontologijoje [19]

Pabandykime įsivaizduoti, kaip turėtų atrodyti paieškos architektūra minėtam, Žvaigždžių karų filmui.

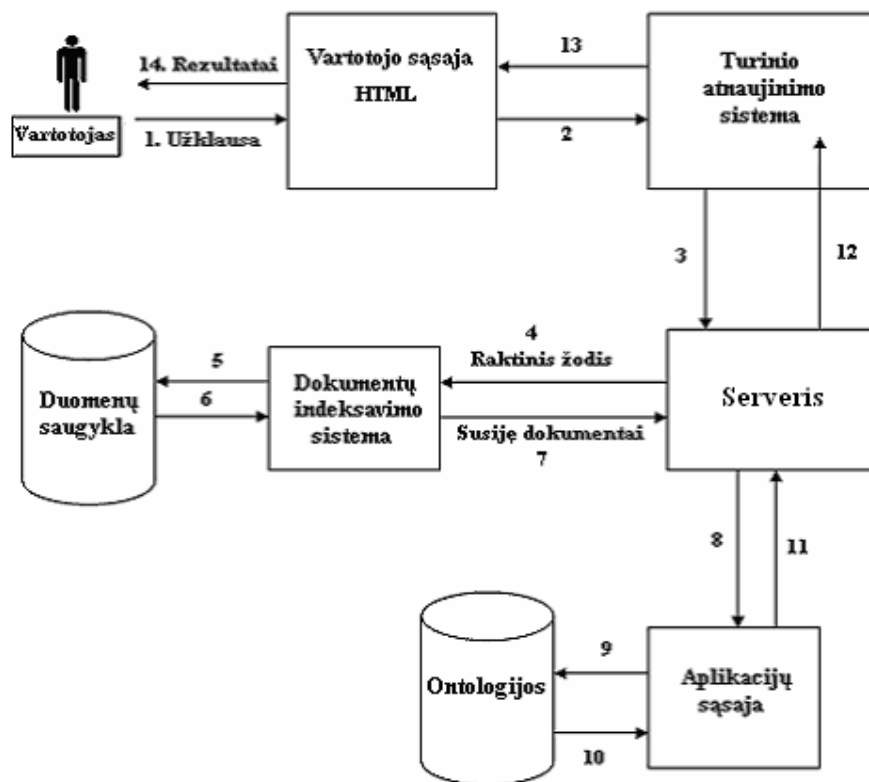
Pirmiausia, pagal nurodytus ryšius reiktų ontologijas, kurios apibūdintų duomenis.

Kaip pvz. Planeta Bespin.

Klasė	Bespin
Ypatybės	Cloud City
Gyventojas	Lando Calrissian

Klasė	Lando Calrissian
Buvęs savininkas	Millenium Falcon
Gyvena	Bespin

Taip turėtų būti sudaromos ontologijos kiekvienam duomenų tipui (klasei). Taip pat privalo būti įdiegta aplikacijų sąsaja, kuri gebėtų suprasti RDF duomenų tipus. Toliau panaudojami įrankiai, realizuoti įprastose paieškos varikliuose: vartotojo sąsaja, duomenų saugykla, indeksavimo sistema, serveris.



Pav. 17 Semantinio paieškos variklio architektūra

1. Vartotojas pateikia užklausą paieškos sistemoje;
- 2-3. Paieškos sistemoje naudojama turinio atnaujinimo sistema, kuri kreipiasi į serverį, kad gautų dokumentus, kuriuose pasirodo užklaustos terminas. Sistema naudojama tam, kad nereiktų perkrauti viso puslapio iš naujo, o tik pateikti atnaujintus duomenis;
4. Raktinis žodis pateikiamas indeksavimo sistemai;
5. Indeksavimo sistema kreipiasi į duomenų saugyklą;

6-7. Duomenų saugykla pateikia sąrašą dokumentų, kuriuose pasirodo užklauso terminas. Indeksavimo sistema pagal per serverį pateikia dokumentų sąrašą vartotojui, kuriuose užklauso terminas pasirodė. Vartotojui pateikiamas sąrašas dokumentų, kuriuose pasirodė raktinis žodis;

8. Jei užklausoje įvedamas raktinis žodis iš peržiūrėto dokumento (klasės vardas). Tuomet serveris kreipiasi į aplikacijų sąsają, kad gautų ontologijų tipus;

9. Aplikacijų sąsaja kreipiasi į ontologijų saugyklą, naudodama ontologijų RDF sintaksės trigubą sistemą;

10-14. Rezultatai (klasių sąrašas) pateikiami vartotojui;

Pvz. jei pateiksime užklausą į dokumentą, kuriame saugoma informacija apie „Žvaigždžių karus“, sistema pateiks dokumentą kuriame saugoma ši informacija. Jei atsidarę dokumentą įvesime terminą iš gauto dokumento pvz. „Lukas Skywalkeris“, sistema pateiks duomenis iš ontologijų klasių, nes viena iš klasių bus Lukas Skywalkeris.

Taigi pirmiausia sistema vykdytų dokumentų paiešką, o tik po to pagal poreikį vykdytų gilesnę dokumento analizę, kurioje duomenys suskirstyti į ontologijas.

Problema su ontologijomis yra ta, kad jas yra labai sudėtinga sukurti, realizuoti ir prižiūrėti. Priklausomai nuo jų apimties, jos gali būti milžiniškos, apibūdinančios didelį kiekį sąvokų ir priklausomybės ryšių. Kai kurie kūrėjai dėl šių sunkumų, labiau mėgsta susikoncentruoti į logiką ir taisykles, nei į ontologijas. Nesutarimas dėl šių taisyklių vaidmens turėtų būti vieni iš potencialių spąstų semantiniam internetui.

7. ONTOLOGIJŲ ŽYMĖJIMAS NAUDOJANT NEURONINIO TINKLO METODĄ

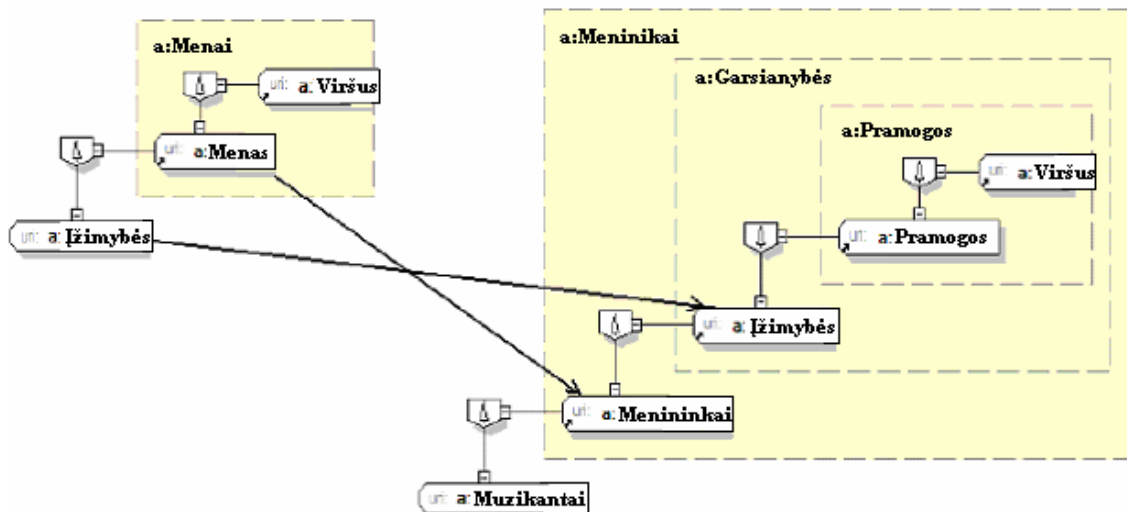
7.1 Įvadas

Ontologijų žymėjimas – tai vienos ontologinės sąvokos (koncepto) priskyrimas ir tų sąvokų reikalavimas iš kitos ontologijos. Tai gali būti naudinga, pvz. kai reikia pasirinkti vieną tinkamą iš keletos ontologijų atlikti užduočiai. [21]

Ontologijų žymėjimas siekia rasti semantinius atitikimus tarp skirtingų ontologijų panašių elementų. Ontologijų žymėjimas yra ypač svarbus, norint pasiekti semantines sąveikas pasauliniame tinkle. Atsižvelgiant į tai, kad šie, visur esantys, ryšiai (pvz. hierarchiniai apribojimai RDFS kalboje) yra įtakojami ontologinių charakteristikų ir jų atvaizdavimo, ir jie egzistuoja pačiuose ontologijose, apribojimų tenkinimas tapo intriguojančia moksline problema ontologijų žymėjimo srityje. Šioje dalyje pateikiamas neuroniniu tinklu grįstas metodas, kuris padės surasti globaliai optimalius sprendimus, kurie kuo daugiau gali tenkinti ontologijų apribojimus. Eksperimento rezultatai OAEI bandymo testo metu, jog rezultatai daug žadantys. Tai žymiai pagerina preliminarius žymėjimo rezultatus.

7.2 Ontologijų žymėjimas

Ribojimų tenkinimas tapo intriguojančia problema ontologijų žymėjime dėl to, kad ontologijų ribojimai vyrauja visur. Kaip pvz. hierarchinėje RDFS kalboje ryšius neleidžiama žymėti kryžminėmis linijomis, tokios aksiomos kaip: „*owl:sameAs*“ ir „*owl:equivalentClass*“, OWL kalboje atspindi skirtingus ryšius tarp skirtingų elementų ir taisyklės SWRL kalboje reikštų ar patvirtintų, kai kurias savybes, kurios nėra prieinamos tiesiogiai.



Pav. 19 Nesudėtingas veiksmų planas ontologijų braižyme

Pav. 19 vaizduoja nesudėtingą realios aplinkos ontologijų žymėjimo atvejį tarp Google direktorijų taksonomijos ir Yahoo direktorijos. Pavyzdyje kryžminis žymėjimas (pvz. žodis „įžymybės“ nurodo į „įžymybes“ ir „mes“ žymi į „menininkai“) gali būti neteisinga išėiga kaip galutinis rezultatas, grįstas tik ontologijų leksikos analize. Kad panaikinti tokius neteisingus žymėjimus ir taip pagerinti ontologijų žymėjimo kokybę, svarbu rasti optimalią konfigūraciją, kuri galėtų labiausiai tenkinti ontologijų ribojimus. [22], [23]

Ieškojimas konfigūracijų, kurios tenkintų ribojimus kaip įmanoma labiau, vadinama ribojimų tenkinimo problema (*angl.* CSP). [24]

7.3 IAC Neuroninis tinklas

Ribojimų tenkinimo problemos paprastai yra sprendžiamos paieškos būdais, pvz. grįžimas tuo pačiu keliu, ribojimų dauginimo ir vietinės paieškos metodais. Tinklas paprastai susideda iš tam tikro skaičiaus konkuruojančių mazgų, sujungtų tarpusavyje.

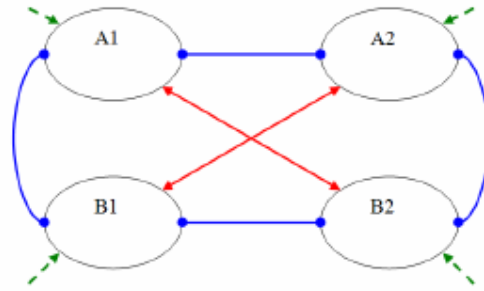
Anot Simon Dennis, IAC neuroninis tikslas susideda iš keletos konkuruojančių mazgų, kurie yra sujungti kiekvienas su kitu. Kiekvienas mazgas pateikia hipotezę. Ryšys tarp dviejų mazgų atspindi ribojimus tarp jų hipotezių. Jei dvi hipotezės palaiko viena kitą, tuomet ryšys tarp jų yra teigiamas (pvz. aktyvus); o, jei dvi hipotezės prieštarauja viena kitai, tada tarpusavio ryšys yra neigiamas (pvz. konkuruojantis). Kiekvienas ryšys yra susijęs su svoriu, kuris yra proporcingas ribojimo stiprumui. Mazgo aktyvavimas yra apibūdinamas keturių šaltinių: savęs pradinė aktyvacija, įėiga iš savo gretimo mazgo, savo šališkumo ir keletos išorinių įėigų. IAC neuroninio tinklo mechanizmas gali būti iliustruotas naudojant šį nesudėtingą pavyzdį.[25]

Tarkime turime du tinklus nr.1, nr.2 ir du ribojimus:

1. Kiekvienas tinklas gali turėti vieną reikšmę, A arba B.
2. Abiejų tinklų reikšmės yra skirtingos.

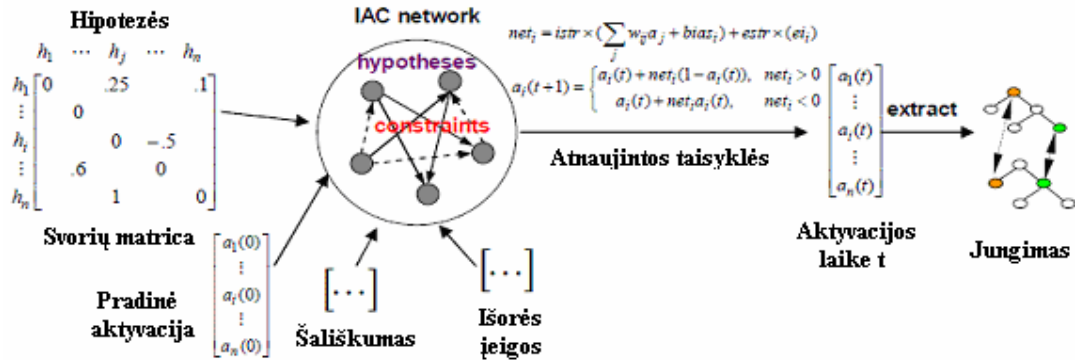
Taip pat turime keturias hipotezes: A tinkle 1 (HA1); B tinkle 1 (HB1); A tinke 2 (HA2); B tinkle 2 (HB2). Remiantis dvejais ribojimais žinome, kad yra du neigiami ryšiai ir vienas teigiamas ryšys kiekvienai hipotezei.

1. H_{Ai} yra prieš H_{Bi} , ir atvirkščiai ($i=1$ arba 2)
2. H_{x1} yra prieš H_{x2} , ir atvirkščiai ($x=A$ arba B)
3. H_{Ai} palaiko H_{Bj} , ir atvirkščiai ($i, j = 1$ arba 2 , ir $i \neq j$)



Pav. 20 Nesudėtingas IAC neuroninis tinklas

Pav.20 vaizduoja nesudėtingą pavyzdį, kuriame kiekvienas mazgas vaizduoja hipotezę, užapvalinta linija ir rodyklės priekyje vaizduoja neigiamą ryšį ir teigiamą ryšį tarp hipotezių atitinkamai. Brūkšninė linija su rodykle priekyje vaizduoja nedidelius stimulus kiekviename mazge esančius išorėje. Tarkime, kad neigiami svoriai yra pusė teigiamų svorių ir visi mazgai yra neaktyvūs pradžioje. Nors įeiga iš trijų mazgų esančių šalia bus atšaukta, tačiau maža sužadavimo įeiga iš išorės aktyvuos mazgą. Visi mazgai gali būti atnaujinti paeiliui atsiktine tvarka. Galiausiai, bet kuris A1 ir B2 arba B1 ir A2 bus aktyvūs ir tinklas pasieks stabilią būseną.



Pav. 21 IAC neuroninis tinklas ontologijų žymėjimo kontekste [26]

7.4 Realizavimas

Pav. 21 vaizduoja IAC neuroninio tinklo įgyvendinimą ontologijų žymėjimo kontekste.

Pasak, Ming Mao, Yefei Peng, 2007 mazgo būseną ir sąryšių svoriai apibūdinami šalia esančių mazgų pagalba. Tinklo būseną – tai kiekvieno mazgo esančio tinkle būseną.

Ontologijų žymėjimo kontekste, mazgas esantis IA tinkle, vaizduoja hipotezę, kuri simbolizuoja sąvoką C_{1i} ontologijoje O_1 , kuri gali būti sujungta su sąvoka C_{2j} esančia ontologijoje O_2 . Pradinė mazgo aktyvacija yra panašumas tarp (C_{1i}, C_{2j}) sąvokų. Mazgo aktyvacija gali būti atnaujinta, naudojant paprastą taisyklę, kur a_i žymi mazgo i aktyvaciją, parašytą kaip h_i , net_i – reiškia mazgo tinklo įeigą.

$$a_i(t+1) = \begin{cases} a_i(t) + net_i(1 - a_i(t)), & net_i > 0 \\ a_i(t) + net_i a_i(t), & net_i < 0 \end{cases}$$

Formulė nr.1

Reikšmė net_i atsiranda iš trijų šaltinių, pvz. iš sąvokos esančios šalia, iš sąvokos šališkumo ir iš sąvokos išorinių įeigų, kaip aprašyta antroje formulėje žemiau. Kur w_{ij} žymi ryšio svorį tarp h_i ir h_j , a_j reiškia mazgo n_j aktyvaciją, $bias_i$ vaizduoja h_i šališkumą, $istr$ ir $estr$ žymi vidinės ir išorinės įeigos parametrus. Pažymėkime, kad ryšių matrica nėra simetrinė ir mazgai nebūtinai turi būti sujungti vienas su kitu. Pvz. $w_{ij} \neq w_{ji}$, $w_{ii} = 0$.

$$net_i = istr \times \left(\sum_j w_{ij} a_j + bias_i \right) + estr \times (input_i)$$

Formulė nr.2

Be to, ryšiai tarp mazgų esančių tinkle vaizduoja ribojimus tarp hipotezių. Pvz. ribojimas „tik 1 į 1“ žymėjimas yra leidžiamas: rezultatas yra neigiamas ryšys tarp mazgų (C_{1i}, C_{2j}) ir (C_{1j}, C_{2k}) , kur $k \neq j$. Daugiau, dvi sąvokos sutampa jei jų „vaikai“ sutampa, rezultatas – teigiamas ryšys tarp taškų (C_{1i}, C_{2j}) ir (C_{1k}, C_{2l}) , kur C_{1k} ir C_{2l} yra sąvokos C_{1i} ir C_{2j} atitinkamai. Galiausiai, sąryšių painiava gali būti labai didelė dėl sudėtingų ribojimų.

Taigi, visi panašumai kiekvienos žymėjimo poros atspindi hipotezių žymėjimo patikimumą. Tai gali būti tiesiogiai panaudota kaip pradinė aktyvacija, išorinės įeigos ar mazgo tendencingumas, IAC tinkle.

Dabar mes realizavome 12 ribojimų (lentelė nr. 8). Svoriai svorių matricoje atitinka pirminius ribojimų atitikmenis, kurie šiuo metu yra nustatyti kaip 1 teigiamam ribojimui ir -1 neigiamam ribojimui. Kiekvieno mazgo pirminė aktyvacija yra nustatoma į bendrą panašumą (C_{1i}, C_{2j}) iš praeitų procesų. Kiekvieno mazgo tendencingumas (bias) yra nustatomas kaip 0. Išorinė įeiga nustatoma į kiekvienos hipotezės patikimumą. Dabar išorinė vienareikšmiškos hipotezės įeiga, kuri turi didžiausią panašumą savo vaizduojamoje eilutėje ir stulpelyje

panašumų matricioje, nustatoma kaip 10, priešingu atveju 0. Mazgo aktyvacija gali būti atnaujinta taisykle pavaizduota paveiksle, kur a_i reiškia mazgo i aktyvaciją, parašytą kaip n_i , net_i reiškia tinklo mazgo įeigą. Kai tik tinklas pradeda veikti, jis gali būti sustabdytas po n ciklų arba tam tikrame taške, kuris yra norimų ribojimų tenkinimo laipsnis, laike t . [26]

Minėtame realizavime, leidžiame tinklui sustoti, kai jo pokyčio reikšmė pasiekia kažkokią tenkinimo būseną (pvz. $\leq 1\%$). Galiausiai, Lentelėje nr. 8 nors neigiamų ribojimų skaičius yra mažesnis, nei teigiamų ribojimų skaičius, neigiamų ryšių ir teigiamų ryšių santykis nėra mažas dėl to, kad kiekvienas mazgas esantis tinkle turės didelį kiekį neigiamų ryšių nustatytų pagal ribojimus, kad „tik 1-1 žymėjimas yra leidžiamas“.

Lentelė nr. 8 Ribojimai naudoti IAC metode

Nr.	Suvaržymai	Ryšys
1	Tik 1 – 1 žymėjimas yra leistinas	neigiamas
2	Joks kryžminis žymėjimas nėra leidžiamas	neigiamas
3	Jei „vaikų“ elementai atitinka, tuomet jų tėviniai elementai taip pat atitinka.	teigiamas
4	Jei tėviniai elementai sutampa tuomet „vaikų“ elementai taip pat sutampa	teigiamas
5	Jei $e1i$ atitinka $e2j$, tuomet $e1i$ sutampa su $e2t$, kur $e1i$ ir $e1s$, $e2j$ ir $e2t$ yra vienos rūšies ontologijos.	teigiamas
6	Jei savybės elementai sutampa tuomet jų domeno elementai taip pat sutampa	teigiamas
7	Jei savybės elementai sutampa tuomet jų srities elementai taip pat sutampa	teigiamas
8	Jei klasės elementai sutampa tuomet jų tiesioginiai savybės elementai sutampa	teigiamas
9	Jei savybės elementai sutampa tuomet jų pagrindinės klasės elementai sutampa	teigiamas
10	Jei klasės du elementai sutampa tuomet visi klasėje esantys elementai sutampa	teigiamas
11	Jei atskiri elementai sutampa tuomet jų pagrindinės klasės elementai sutampa	teigiamas

12	Du elementai sutampa jei jų „owl:SameAs“ arba „owl:equivalentClass“ ar „owl:equivalentProperty“ elementai sutampa	teigiamas
----	---	-----------

7.5 OAEI testo procedūra

Prieš pradėdant kalbėti apie IAC neuroninio tinklo modelį, trumpai apžvelgsime visą metodo procedūrą. Duota ontologijų žymėjimo užduotis pvz. OAEI testas Nr. 248-266, pirmiausia gramatiškai išnagrinėjamos ontologijos, naudojant PĮ Jena ir sugrupuojame juos panaikinant trumpuosius žodelius. Toliau, matuojamos trys panašumų rūšys pvz. atstumo panašumai, profilių panašumai ir struktūrų panašumai, kiekvienai ontologijai. Po to, kai suskaičiuojamos harmonijos kiekviename panašume, tuomet sudaromi trys panašumai jų harmonijose. Naudojant panašumų sudarymą, galime aktyvuoti IAC neuroninį tinklą, kad ieškotų optimalių konfigūracijų, kurios geriausiai tenkina ontologijų ribas. Galiausiai, išgaunami žymėjimo rezultatai naudojant paprastus paveldėjimo algoritmus.[27]

7.6 Metodo naudos įvertinimas

7.6.1 Duomenų rinkiniai

Duomenų jungimo užduotis buvo suprojektuota iš Google ir Yahoo direktorių, kaip pavaizduota pav.19. Duomenų rinkiniai vaizduojami kaip taksonomijos, kur internetinių direktorių mazgai yra klasės ir tarpusavio ryšių klasifikavimas vykdomas jungiant juos komanda: `rdfs:subClassOf` relation.

Pagrindinė duomenų rinkinių metodologijos užduotis yra ženkliai sumažinti paieškos erdvę. Vietoj to, kad atlikti visą lyginimo užduotį, kuri yra labai sudėtinga ir didelės apimties, sistema naudoja pusiau automatinių atmetimo būdą tam, kad ženkliai susiaurintų paieškos erdvę. Kaip pvz. sistema peržvelgė tik 2265 atitikmenų, ne visas galimus variantus.

Kad įvertintume šį metodą, naudojame bandymo testus Nr. 248-266 iš OAEI ontologijų lyginimo kampanijos. Testai įtraukia vieną informacinę ontologiją, kuri apibūdina labai siaurą bibliografinį domeną, ir 15 dirbtinai sukurtų ontologijos testų, kiekvienas, kuris šalina įvairią informaciją iš informacinės ontologijos taip, kad būtų galima įvertinti kaip algoritmas veikia, kai trūksta kai kurių informacijos rūšių.

Priežastis kodėl pasirinkti OAEI bandymo testai Nr. 248-266 yra: 1.OAEI bandymo testai tapo autoritetingi testų ontologijų žymėjimo srityje. 2. Šie bandymo testai yra atviri ir todėl gali būti naudojami išsamiam įvertinimui.3. Testai Nr. 248-266 yra patys sunkiausi testai tarp daugelio bandymo testų. Visų dalyvių rezultatai šiuose testuose yra gana žemesni

nei jų rezultatai kituose bandymo testuose. Todėl šių testų pagerinimas gali daug prisidėti prie bendro visų rūšių ontologijų žymėjimo metodų gerinimo. [28]

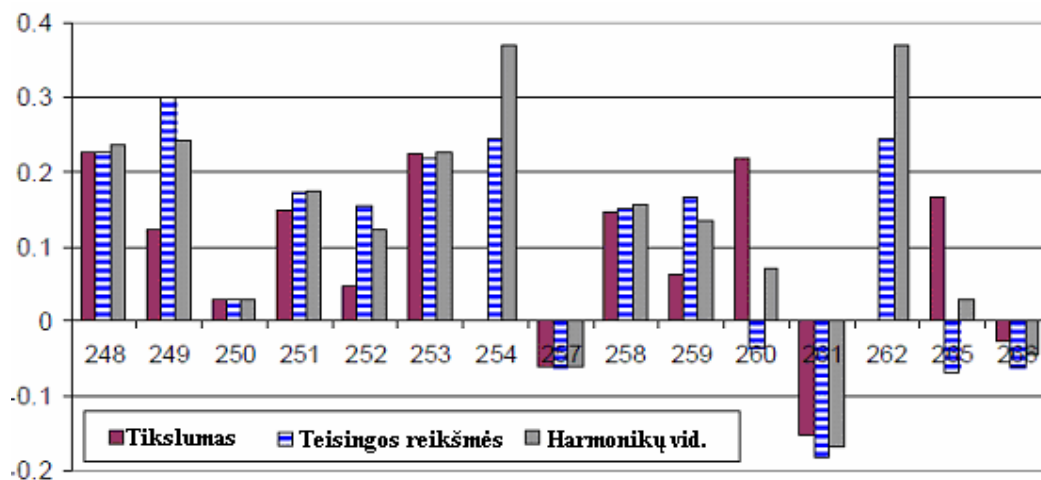
7. 6. 2 Vertinimo kriterijai

Sekame vertinimo kriterijus iš OAEI kampanijos, skaičiuojant tikslumą (angl. precision) (pvz. santykis teisingai rastų žymėjimų tarp visų rastų žymėjimų), santykis tarp teisingai rastų žymėjimų visuose teisinguose teigiamuose žymėjimuose (angl. recall) ir svertinų harmonikų vidurkiai (angl. f-measure) per kiekvieną bandymo testą.

7. 6. 3 Eksperimentiniai metodai ir rezultatai

Eksperimento metodologija: yra duoti preliminarūs žymėjimo rezultatai, aktyvuojamas IAC neuroninis tinklas kiekviename OAEI bandymo teste Nr. 248-266. Parametro vertė tinkle yra nustatoma kaip apibūdinta paragrafe 7.4. Tuomet, leidžiama tinklui pradėti dirbti pačiam ir jis sustabdomas, kuomet jo delta (jei reikšmė testo metu pakinta mažiau nei 1%) reikšmė yra mažesnė nei 1%.

Eksperimento rezultatai Pav.21 parodo neuroninio tinklo grįsto ribojimų tenkinimo metodas pagerina tikslumą ir teisingų rezultatų skaičių iš 15, 12 testų išskyrus nr.257, nr.261 ir nr.266. Didžiausias funkcijos pagerinimas (pvz. 0,37) įvyksta nr.254 ir nr.262. Tuo tarpu, jokia lingvistinė informacija nėra galima #261 teste, ir todėl nėra jokių lingvistinių analizių, kuriomis mes galime remtis. Lentelė nr.9 pateikia mūsų metodo pagerinimą visuose nr. 248-#66 testuose. Jie yra 13%, 24%, ir 19%.



Pav. 22 IAC neuroninio tinklo metodo rezultatai, OAEI bandymo testuose Nr.248-266 [23]

Lentelė nr. 9 Bendras metodo pagerinimas testuose nr.248-266

Testas	Tikslumas	Teisingi rezultatai	Harmonikų vidurkis
Prieš NT	.76	.54	.63
Po NT	.88	.67	.76
NT pagerinimas	13%	24%	19%

NT – neuroninis tinklas

Harmonikų vidurkis yra lygus $F = \frac{2 \times \text{tikslumas} \times \text{teisigi_rez}}{\text{tikslumas} + \text{teisingi_rez}}$

Tikslumas yra teisingų rezultatų skaičius padalintas iš visų gražintų rezultatų.

IŠVADOS

Išnagrinėjus intelektines interneto naršymo sistemas, galime drąsiai teigti, kad pagrindinis intelektinės naršymo sistemos elementas yra paieškos variklis.

Pateiktas programinės įrangos – Statistica metodas, kurios pagalba žodžiai ir terminai, esantys dokumente suskaičiuojami ir suindeksuojami eilės tvarka. Atsižvelgus į išnagrinėtą paieškos sistemos architektūrą, šis programinės įrangos įrankis galėtų puikiai tikti, kuriant paieškos sistemas.

Išnagrinėjus internetinės svetainės, vykdančios paieškos sistemų radimo funkcijas, nustatyta, kad sistema puikiai atlieka paieškos variklių vadovo užduotis – geba surasti paieškos sistemą labiausiai tenkinančią kiekvieno interneto vartotojo poreikius.

Atlikus trijų žiniatinklio paieškos sistemų užklausų bandymo testus, galime teigi, jog Google paieškos sistema geriausiai atlieka savo darbą iš visų minėtų paieškos sistemų. Gauti rezultatai tik patvirtino, jog Google neveltui yra populiariausia paieškos sistema.

Išnagrinėjus problemas su kuriomis susiduria vartotojai naršydami internete nustatyta, kad dažniausiai nesklaidumų patiria vartotojai su aiškiu naršymo tikslu reguliariai lankydami norimose interneto svetainėse. Atsižvelgus į pastebėtas naršymo problemas, pateikti intelektinių sistemų sprendimai galintys išspręsti tokias problemas kaip: paklydimas ar nenaši duomenų paieška.

Išsiaiškinus semantinio tinklo veikimo principą sudaryta semantinės paieškos variklio architektūra, pagal kurią realizuota sistema gebėtų vykdyti paiešką ontologijų saugykloje. Išsiaiškinus pagrindinę semantinio interneto plėtros problemą – ontologijų žymėjimą, pateiktas neuroniniu tinklu grįstas ontologijų jungimo metodas, skirtas rasti optimalų ontologijų žymėjimo sprendimą. Iš ontologijų testavimo kampanijos eksperimento rezultatų pastebime, kad taikomas metodas aiškiai pagerino preliminarių žymėjimo rezultatų vykdymą.

TERMINŲ ŽODYNĖLIS

1. URL (angl. Uniform Resource Locator) – universalus informacijos šaltinio adresas
2. URI (angl. Uniform Resource Identifier) – universalus informacijos šaltinio vardas
3. CSP (angl. Constraint satisfaction problems) – ribojimų tenkinimo problemos
4. HTML (angl. *Hyper text Markup Language*) – Hiperteksto žymėjimo kalba
5. DNS (angl. Domain Name System) – Srities vardų struktūra
6. ISAM (angl. Indexed sequential access method) – Nuoseklaus indeksavimo metodas, naudojamas greitam duomenų indeksavimui, greitai duomenų gavymui
7. ASCII (angl. American Standard Code for Information Interchange) – Amerikietiškas informacijos mainų koduotės standartas
8. NFS (angl. network file system) – tinklo duomenų (bylų) sistema
9. OWL (angl. Web Ontology Language) – Internetinių ontologijų programavimo kalba
10. XML (angl. Extensible Markup Language) – programavimo kalba, rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba
11. DVD (angl. digital video disk) – optinis diskas informacijai skaitmenine forma įrašyti, saugoti
12. FTP (angl. File Transfer Protocol) – Bylų perkėlimo protokolas,
13. OAEI (angl. Ontology Alignment Evaluation Initiative) – Ontologijų palyginimo vertinimo iniciatyva
14. IAC (angl. Interactive Activation and Competition) – Interaktyvios aktyvacijos ir konkurencijos tinklai, kitaip neuroniniai tinklai
15. Zlib – programinės įrangos biblioteka, skirta duomenų kompresijai
16. IR (angl. Information retrieval) – informacijos paieška
17. PĮ – programinė įranga
18. SWRL (angl. Semantic web rule language) – semantinio interneto taisyklių kalba
19. RDF (angl. *Resource Description Framework*) – resursų deskriptorius, naudojamas
20. WWW (angl. *World Wide Web*) – pasaulinis kompiuterių tinklas
21. QDEX (stands for Query Detection and Extraction) – užklausų atpažinimo ir nagrinėjimo sistema

LITERATŪRA

1. Hsiangchu Lai, Tzyy-Ching Yang. A system architecture for intelligent browsing on the Web. National Sun Yat-sen University, Kaohsiung 80426, Taiwan, p. 1-2.
2. Naršymo sistemos [žiūrėta 2011-04-10]. Prieiga per internetą: <<http://www.w3.org/TR/webarch/>>
3. Naršyklės struktūra [žiūrėta 2011-04-10]. Prieiga per internetą: <<http://taligarsiel.com/Projects/howbrowserswork1.htm>>
4. Architecture and evolution of the modern web browser Alan Grosskurth, Michael W. Godfrey David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada
5. Intelektinė interneto naršymo sistema [žiūrėta 2011-01-10]. Prieiga per internetą: <http://code.entersources.com/f/WebFuret--An-Intelligent-Web-Browser_2_4513_0.aspx>
6. Statistiniai teksto gavybos metodai [žiūrėta 2011-04-10]. Prieiga per internetą: <<http://www.statsoft.com/textbook/text-mining/#transforming>>
7. Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Manuscript in progress 1998.
8. Puslapių rangavimas [žiūrėta 2011-04-10]. Prieiga per internetą: <<http://www.searchenginegenie.com/pagerank-10-sites.htm>>
9. Paieškos variklio anatomija [žiūrėta 2011-01-10]. Prieiga per internetą: <<http://infolab.stanford.edu/~backrub/google.html>>
10. Paieškos vadovas [žiūrėta 2011-05-10]. Prieiga per internetą: <<http://websearch.about.com>>
11. Paieškos sistema Hakia [žiūrėta 2011-05-10]. Prieiga per internetą: <<http://data.hakia.com/ip-091907.html>>
12. Paieškos sistema WolframAlpha [žiūrėta 2011-05-10]. Prieiga per internetą: <www.wolframalpha.com>
13. Hsiangchu Lai, Tzyy-Ching Yang. A system architecture for intelligent browsing on the Web. National Sun Yat-sen University, Kaohsiung 80426, Taiwan, p. 221-225.
14. Tim Berners-Lee. *Weaving the Web The Original Design and Ultimate Destiny of the World Wide Web*. Psl.6
15. Web3 technologijos veikimas [žiūrėta 2011-01-10]. Prieiga per internetą: <<http://computer.howstuffworks.com/web-304.htm>>

16. An Overview of W3C Semantic Web Activity. Eric Miller and Ralph Swick
Bulletin of the American Society for Information Science and Technology – April/May 2003
Volume 29, No. 4 psl. 8-11
17. Gerd Stumme, Andreas Hotho, Bettina Berendt. Semantic Web Mining State of
Art and Future Directions psl.3-7
18. P. Patel-Schneider and J. Siméon. Building the semantic web on xml. psl 147–
161, 2002
19. Semantinio interneto veikimas [žiūrėta 2011-01-10]. Prieiga per internetą:
<<http://computer.howstuffworks.com/semantic-web.htm>>
20. Resource Description Framework: Metadata and Its Applications K. Selcuk
Candan, Huan Liu, and Reshma Suvarna, Department of Computer Science & Engineering
Arizona State University Tempe, AZ 852875406
21. D. Zhang and W.S. Lee. Learning to integrate web taxonomies. Journal of Web
Semantics, 2(2):131–151, 2004
22. Google direktorijos [žiūrėta 2011-05-10]. Prieiga per internetą:
<<http://directory.google.com/>>
23. Yahoo direktorijos [žiūrėta 2011-05-10]. Prieiga per internetą:
<<http://dir.yahoo.com/>>
24. Ming Mao, Yefei Peng, Michael Spring. Neural Network based Constraint
Satisfaction in Ontology Mapping. Proceedings of the Twenty-Third AAAI Conference on
Artificial Intelligence (2008)
25. The Interactive Activation and Competition Network: How Neural Networks
Process Information , Simon Dennis, University of Queensland, 1998.
26. The PRIOR+: Results for OAEI Campaign 2007 Ming Mao, Yefei Peng
University of Pittsburgh, Pittsburgh, PA, USA. p.2-4
27. OAEI kampanija 2007 [žiūrėta 2011-05-10]. Prieiga per internetą:
<<http://oaei.ontologymatching.org/2007/>>.
28. Results of the Ontology Alignment Evaluation Initiative 2007. Jérôme Euzenat,
Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondřej Šváb, Vojtěch
Svátek, Willem Robert van Hage and Mikalai Yatskevich. p.14-20
29. Stuart J. Russell, Peter Norvig. Artificial Intelligence A Modern Approach. 1995.
p36-40
30. Stuart J. Russell, Peter Norvig. Artificial Intelligence A Modern Approach. 1995.
p41-50

31. Agentų klasės [žiūrėta 2011-01-10]. Prieiga per internetą:
<http://en.wikipedia.org/wiki/Intelligent_agent>

32. Layered Learning in Multi-Agent Systems Peter Stone 1998 School of Computer Science Carnegie Mellon University Pittsburgh

SANTRAUKA

Interneto naršymo sistemos sumanymas priklauso nuo įvairiausių aspektų: sistemos architektūros, intelektinių sistemų panaudojimo įvairovės ir net nuo pačio interneto struktūros. Šie aspektai padaro interneto naršymo sistemą našesne ir labiau patrauklia vartotojui.

Pagrindinis darbo tikslas – išnagrinėti intelektinių naršymo sistemų architektūras. Palyginti paieškos variklius. Išsiaiškinti su kokiais problemomis susiduria vartotojai naršydami internete. Pateikti internetinio puslapio architektūrą, kuri palengvintų vartotojų naršymą tinkle. Išnagrinėti trečios kartos interneto technologiją. Pritaikyti duomenų gavybos metodą, kuris pagerintų naršymo charakteristikas.

SUMMARY

Internet browser intelligence depends of lot of factors: system architecture, number of intelligent systems usage and even of internet structure.

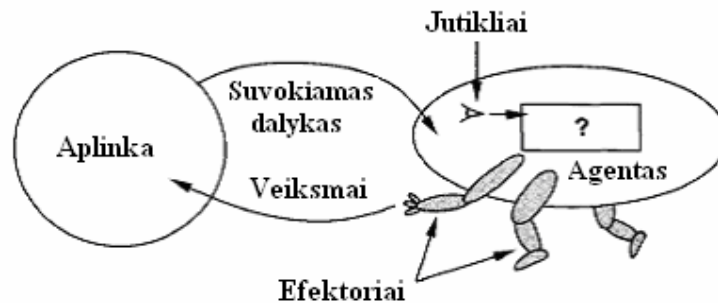
The main purpose of this work is to explore intelligent web browsing systems architecture. To compare several web search engines. Find out the user problem which happens during browsing the web. To offer system architecture which would help users to browse the web. Implement the data mining method that would improve browsing characteristics.

PRIEDAI

INTELEKTUALŪS AGENTAI

Sumanus agentas

Pasak, Stuart J. Russell, Peter Norvig, agentas – bet kas, kas gali būti matomas kaip suvokiantis savo aplinką daviklių pagalba ir veikiantis toje aplinkoje per vykdymo sistemą. Žmogus agentas turi akis, ausis ir kitus organus kaip daviklius. Rankos, kojos, burna ir kiti kūno organai kaip veikimo sistema, priemonės. Automatinis agentas turi kameras ir infraraudonųjų spindulių sistemas aplinkos stebėjimui, kaip daviklius ir įvairius variklius-motorus, kaip vykdymo sistemas. Programinės įrangos agentas turi šifruotą bitų eilutę kaip savo suvokiamą dalyką ir veiksmus [29]. Bendra agento schema pavaizduota Pav. 1.



Pav. 1 Agento sąveikavimas su aplinka daviklių ir vykdymo sistemų pagalba

Sąryšiai tarp agentų architektūros ir programų gali būti apibendrinti taip:

Agentas = *architektūra* + *programa*

Pavyzdys

Pirmiausia, reiktų apibendrinti konkrečią aplinką. Apžvelgsime automatinio taksi vairuotojo projektavimo darbą. Turėtume paaiškinti, kad tokia sistema šiuo metu yra už egzistuojančių technologijų galimybių ribų. Visos vairavimo užduotys neturi ribų, nėra jokių apribojimų naujoms aplinkybių kombinacijoms, kurios gali netikėtai atsirasti.

Pirmiausia turime galvoti apie suvokiamus dalykus, veiksmus, tikslus ir aplinką, kurioje bus taksi automobilis. Visi šie dalykai pateikti lentelėje nr. 1:

Lentelė nr.1 Agentas ir jo uždaviniai

Agento tipas	Suvokiami dalykai	Veiksmai	Tikslai	Aplinka
Taksi vairuotojas	Kameros, spidometras, GPS, mikrofonas	Vairavimas, greičio didinimas, kalbėjimas su keleiviu	Saugumas, greitis, komfortas važiuojant, maksimalus uždarbis	Keliai, eismas, pėstieji, klientai

Kokius vykdymo duomenis norėtume, kad mūsų automatizuotas vairuotojas pasiektų? Pageidautina, kad taksi nuvyktų į teisingą atvykimo vietą, sunaudojant mažiausiai degalų, sugaištų kuo mažiau laiko, nepažeistų kelių eismo taisyklių ir netrukdytų kitiems vairuotojams. Užtikrintų saugumą ir keleivio komfortą, gautų didesnę pelną. Akivaizdu, kad keletas iš šių tikslų yra priešaringi, taigi abipusiai suderinamumai bus įtraukti.

Galiausiai, turėtume nuspręsti kokia vairavimo aplinka laukia taksi vairuotojo. Ar jis važiuos tik vietiniuose keliuose ar ir greitkeliuose? Ar tai bus pietų Kalifornija, kur praktiškai niekada nebūna sniego ar tai bus Aliaska? Ar vairuoti reiks dešiniąja eismo juosta ar sistema galės vairuoti ir kairiąja, tuo atveju jei norėsime dirbi Britanijoje ar Japonijoje? Akivaizdu, kuo aplinkoje yra daugiau apribojimų, tuo lengviau ją suprojektuoti.

Dabar turime nuspręsti kaip sukurti tikrą programą, kad realizuoti žymėjimą iš suvokiamo dalyko į tam tikrus veiksmų. Žinome, kad skirtingi automobilio vairavimo aspektai siūlo skirtingus agentų programų tipus. Mes apžvelgsime keturis agentų programų tipus.

- Paprastas refleksinis agentas;
- Agentas, kuris stebi aplinką ;
- Tikslu siekiantis agentas;
- Naudos siekiantis agentas;

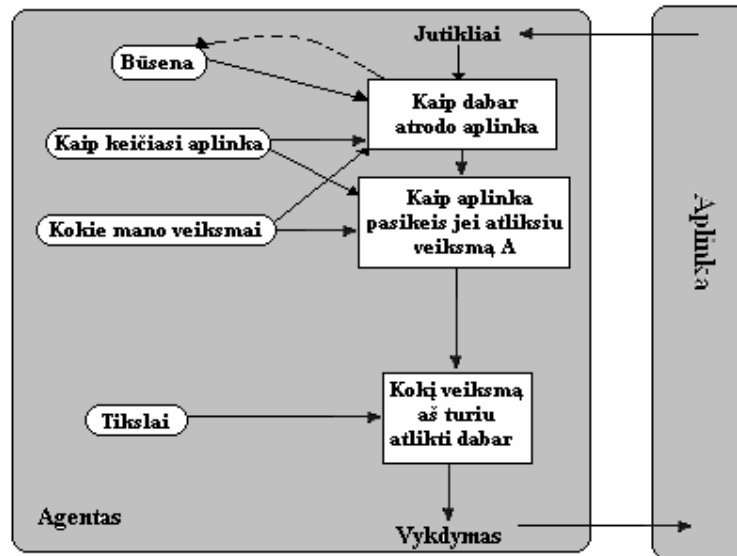
Paprasto reflekso agentai

Regima įvestis iš paprasto foto aparato pasirodo 50 megabaitų per sekunde dažnumu (25 kadrai per sekundę, 1000 x 100 pikselių su 8 bitų spalvomis ir 8 bitų ryškumo informacija). Informacinėje lentelė už valandos bus 260x60x50M įrašų.

Tačiau, galime reziumuoti lentelės dalis, pažymint dažniausiai pasitaikančias įvesties/išvesties asociacijas. Kaip pvz. jei priekyje esantis automobilis stabdo ir jo stabdžių žibintai užsidega, tuomet vairuotojas pastebėjęs tai, pradeda stabdyti. Kitaip tariant, keletas veiksmų yra atlikta vizualioje įvestyje, kad sukurtų sąlygas, vadinamas “priekyje esančio automobilio stabdymas”. Šią sąsają mes vadiname sąlygos-veiksmo taisykle užrašoma taip: „jei priekyje esantis automobilis stabdo, tuomet inicijuojamas stabdymas“.

Žmonės taip pat turi tokias sąsajas, kai kurios iš jų yra išmoktos reakcijos vairavime ir kai kurios yra įgimti refleksai, tokie kaip akių mirksėjimas, kaip kas patenka į akį.

Pav. 1 pateikia paprasto reflekso agento struktūrą, vaizduodama kaip sąlygos-veiksmo taisyklės leidžia agentui sudaryti ryšį iš suvokiamo veiksmo. Naudojamos stačiakampės figūros pažymėti einamąją vidinę agento būseną apsisprendimo procese ir užapvalintas figūras antrinei informacijai atvaizduoti.



Pav. 2 Paprasto refleksinio agento struktūra

Tokie agentai gali būti realizuoti labai veiksmingai, bet jų pritaikymo erdvė yra labai siaura.

Agentas stebėtojas

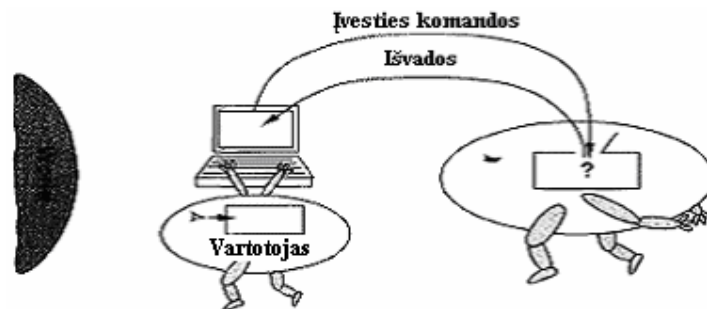
Prieš tai minėtas paprastas refleksinis agentas veiks tik jei teisingas sprendimas bus priimtas pagrindiniame einamajame suvokime. Jei automobilis esantis priekyje yra naujas modelis ir turi centralizuotai įmontuotą stabdymo žibintą, tuomet bus galima pasakyti ar tai yra stabdymas iš vieno vaizdo. Deja, senesni modeliai turi skirtingus galinius žibintus, stabdymo žibintus ir posūkių signalo žibintus, ir ne visada įmanoma pasakyti ar mašina stabdo.

Vadinasi net ir paprastoje stabdymo taisyklėje, mūsų vairuotojas privalo išlaikyti tam tikros rūšies vidines būsenas, tam kad gebėtų pasirinkti veiksmą. Čia, vidinė būsena nėra labai išsiplėtusi – jai reikalinga tik prieš tai gautas kadrą iš kameros, kad nustatytų, kada dvi raudonos šviesos iš automobilio kraštų užsižiebs ar išsijungs tuo pačiu metu.

Apžvelkime sekančius labiau aiškesnius atvejus: laikas nuo laiko, vairuotojas žvilgteli į galinio matymo veidrodėlį, kad matytų gale važiuojančių automobilių padėtį. Kai vairuotojas nežiūri į veidrodėlį, automobiliai esantys gretimoje juostoje nėra matomi (t.y. būsenos kuriuose jie dalyvauja ir nedalyvauja yra nepastebimos), bet tam, kad nuspręstų dėl eismo juostų pakeitimo – manevravimo, vairuotojui reikia žinoti ar yra gretimais važiuojančių automobilių.

Iškyla problema šiame aprašytame pavyzdyje, nes davikliai neužtikrina prieigos prie galutinės aplinkos būsenos. Tokiu atveju, agentui gali tekti išlaikyti kai kurias vidinių būsenų informacijas, tam kad galėtų atskirti aplinkos būsenas, kurios generuoja tą patį suvokiamo dalyko įvestį, bet vis dėlto yra ženkliai skirtingos. Čia, „ženkliai skirtingos“ reiškia, kad skirtingi veiksmai yra būdingi dviejuose būsenose.

Atnaujinant šių vidinių būsenų informaciją, laikui bėgant, reikalinga dviejų rūšių informacija, kad būtų galima užkoduoti ją agento programoje. Pirmiausia, mums reikalinga informacija apie tai, kaip aplinka plečiasi, vystosi, keičiasi nepriklausomai nuo agento. Kaip pvz. bandydamas lenkti automobilis, paprastai bus arčiau lenkiamo automobilio, nei akimirka prieš tai. Antra, mums reikalinga informacija apie tai, kaip agentas savo veiksmais įtakoja aplinką. Kaip pvz. kad, kai agentas keičia važiavimo juostą į kairę, atsiranda tarpas tarp (bent jau laikinas) juostų, kuris buvo prieš. Kitas pvz. po penkių minučių vairavimo į šiaurę, greitkelyje vienas automobilis, paprastai, bus toliau nutolęs į šiaurę, nei buvo prieš penkias minutes.



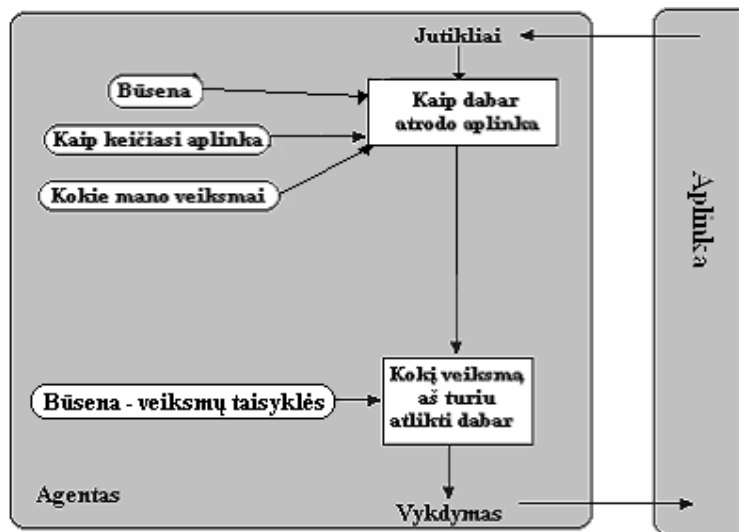
Pav. 3 Sakiniai kuriais kreipiamasi į aplinką, prie kurios agentas neturi nepriklausomo prieigos

Tarkime mes pakeičiame naudotoją pavyzdyje Pav. 3 į kameros pagrindu kurtą regėjimo apdorojimo sistemą, kuri siunčia įėjimo sakinius į mąstančią sistemą. Tai nesudaro jokio skirtumo. Net ir kompiuteris, dabar jau turi „tiesioginę prieigą“ į aplinką, išvados vis dar gali vykti per tiesiogines sakinių sintaksės operacijas, be jokios papildomos informacijos apie jų numanomą prasmę.

Pav.2 pateikia refleksinio agento struktūrą, vaizduodama kaip esamas suvokiamas dalykas yra susijęs su prieš tai buvusiomis būsenomis, kad sugeneruotų einamos būsenos atnaujintą apibūdinimą.

Tikslo siekiantys agentai

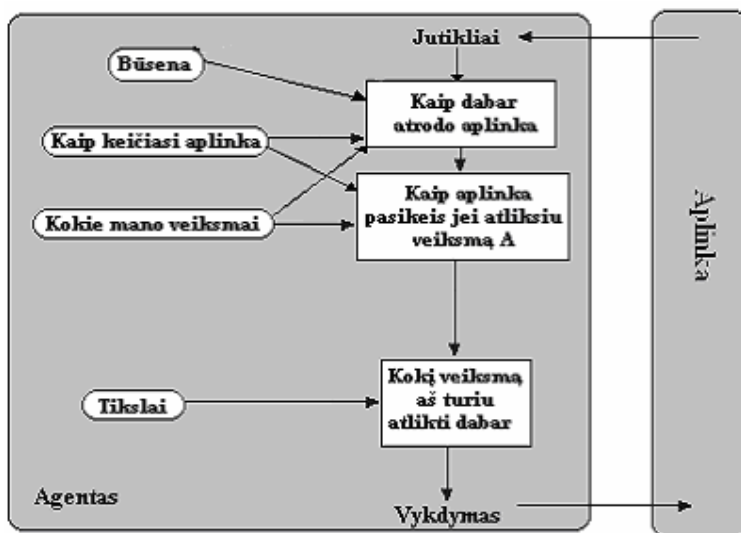
Žinojimas apie esamą aplinkos būseną, ne visada padeda nuspręsti ką daryti. Kaip pvz. kelio sankryžoje, automobilis gali pasisukti dešinėn, kairėn arba važiuoti tiesiai. Teisingas sprendimas priklauso nuo to, kur automobilis bando nuvažiuoti. Kitaip tariant, taip kaip esamos būsenos apibūdinimas, agentui reikalinga tam tikros rūšies tikslinė informacija, kuri apibūdina norimas situacijas. Kaip pvz. būvimas keleivio vietoje. Agento programa gali palyginti tai su informacija, kurioje yra galimų veiksmų rezultatai, (ta pati informacija, kuri buvo naudota vidinių būsenų atnaujinimui refleksiniame agente) tam, kad pasirinktų veiksmus, kurie pasiekia tikslą. Kartais, tai bus sudėtingiau, kai agentas turi atidžiai peržiūrėti ilgas posūkių sekas, kad rastų kelią tikslui pasiekti. Paieška ir planavimas yra dirbtinio intelekto sub-laukai, skirti surasti veiksmų sekas, kurie pasieks agento tikslą.



Pav. 4 Refleksinis agentas su vidine būseną

Pastebime, kad šios rūšies sprendimo-sudarymo taisyklės iš esmės skiriasi nuo sąlygos-veiksmo taisyklių apibūdintų anksčiau, tai įtraukia abu ateities apsvarstymus: “Kas nutiks jei bus taip ir taip?” ir “Ar tai padarys mane laimingą?” Refleksinio agento konstrukcijoje, ši informacija nėra atvirai naudojama, nes kūrėjai iš anksto apdorojo teisingus veiksmus skirtingiems atvejams. Refleksinis agentas stabdo, kai pamato užsidegusius stabdymo žibintus. Tikslinio pagrindo agentas paprastai, geba mąstyti, jei priekyje esančio

automobilio žibintai užsidegė, jis sumažins greitį. Laikui bėgant aplinka paprastai keičiasi, vienintelis veiksmas, kuris leidžia išvengti susidūrimo yra stabdymas. Taip pat, tiksliniai agentai pasirodo mažiau veiksmingi, jie yra labiau lankstūs. Jei pradeda lyti, agentas geba atnaujinti savo žinias kaip efektyviai jo stabdžiai veiks lyjant. Tai automatiškai įtakos visus tiesiogiai susijusius veiksmus, pasikeitimui, kad prisitaikyti prie naujų sąlygų. Refleksiniam agentui, mes turime perrašyti didelį kiekį sąlygos-veiksmo taisyklių. Taip pat, tiksliniai agentai, yra labiau lankstesni, atsižvelgiant į tai, kad reikia pasiekti skirtingus tikslus. Paprastai nustatant naują krypties tikslą, galime panaudoti tikslinio pagrindo agentą, kad susidoroti su naujomis funkcijomis. Refleksiniai agentai vyrauja, kai reikia pasukti ir kada važiuoti tiesiai, bet veiks tik su viena kryptimi. Jie visi turi būti pakeisti, kad važiuotų kur nors kitur. Pav. 4 vaizduoja tikslinio pagrindo agento struktūrą.



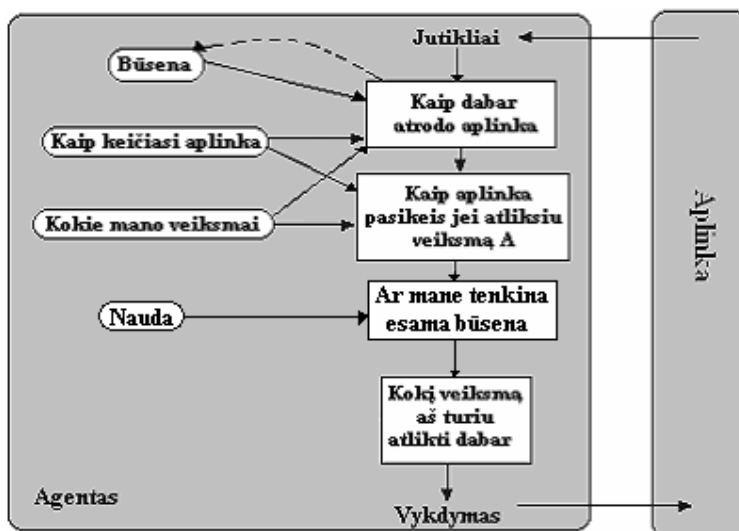
Pav. 5 Agentas su aiškiais tikslais

Naudos siekiantis agentas

Vien tikslų nepakanka realiai, tam kad sugeneruotume aukštos kokybės funkcijas. Kaip pvz. yra daug veiksmų sekų, kurios nukreips taksi automobilį reikiama kryptimi, tuo būdu pasiekiant tikslą, bet keletas iš jų yra greitesni, saugesni, labiau patikimi arba pigesni, nei kiti. Tikslai atspindi tik paprastus skirtumus tarp “laimingo” ir “nelaimingo” būsenų, tuo tarpu daugiau bendrų veiksmų matavimas turėtų sudaryti sąlygas skirtingų aplinkos būsenų palyginimui, pagal tai kiek tiksliai agentą padarytų laimingu, jei jie būtų pasiekti. Todėl “laimingas” neskamba labai moksliskai, įprasta terminologija sako, kad jei viena aplinkos būseną yra labiau mėgiama nei kita, tuomet tai įgyja didesnę naudą agentui.

Nauda dėl to ir yra funkcija, kuri pažymi būseną realiam skaičiui, kuris apibūdina susijusį laimingumo laipsnį. Galutinė naudos funkcijos specifikacija, skiria racionalius sprendimus į dvi atvejų rūšis, kur tikslas patenka į bėdą. Pirmiausia, kai yra priešaringi tikslai, tik keletas iš jų gali būti pasiekti (pvz. greitis ir saugumas), pagalbinė funkcija tiksliai nusako tinkama kompromisą. Antra, kai yra keletas tikslų, kuriuos agentas gali siekti, nė vienas iš jų negali būti pasiektas užtikrintai. Naudos funkcija parodo kelią, kuriame tikslo pasisekimo tikėtinumai gali būti pasvertas prieš tikslų svarbą.

Agentas, kuris turi detalią naudos funkciją, gali atlikti racionalius sprendimus, bet gali palyginti palaikančias pagalbines funkcijas, įgytas naudojant skirtingus veikimo būdus. Tikslai, net jei ir paprasti, suteikia galimybę agentui išsirinkti veiksmą iš karto, jei tai atitinka keliamą tikslą. Bendra naudos siekiančio agento struktūra pavaizduota pav. 6.



Pav. 6 Naudos siekiančio agento struktūra

Sekantys šeši ryšiai yra laikomi kaip naudingumo teorijos aksiomos. Jie tiksliai nusako akivaizdžiausius semantinius ryšius pirmenybei teikiamiems dalykams ir loterijoms.

- **Trazivytivumas:** Galioja bet kokioms trijoms būsenoms, jei agentas tenkina A ir C sąlygas.

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

- **Tęstinumas:** Jei bet kokia būsena B yra tarp A ir C, tada egzistuoja tikimybė p, kuriai racionalus agentas bus nesvarbus, kad gauti B tikėtinumui ir loterijai, kuris gauna A su tikimybe p ir C su tikimybe 1-p.

$$A \succ B \succ C \Rightarrow \exists p [p, A; 1 - p, C] \sim B$$

- **Pakeičiamumas:** Jei agentas neįtakoja dviejų loterijų A ir B, tai agentas yra nešališkas tarp dviejų kompleksinių loterijų, kurios yra tos pačios išskyrus, tai kad B yra pakeistas į A viename iš jų. Tai neturi jokios įtakos tikimybėms ir kitiems rezultatams loterijose.

$$A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-p, C]$$

- **Monotoniškumas:** Tarkime yra dvi loterijos, kurios turi vienodus rezultatus A ir B. Jei agentas tenkina A ir B, tuomet agentas privalo tenkinti loteriją, kuri turi didesnę tikimybę A ir atvirkščiai

$$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1-p, B] \succ [q, A; 1-q, B])$$

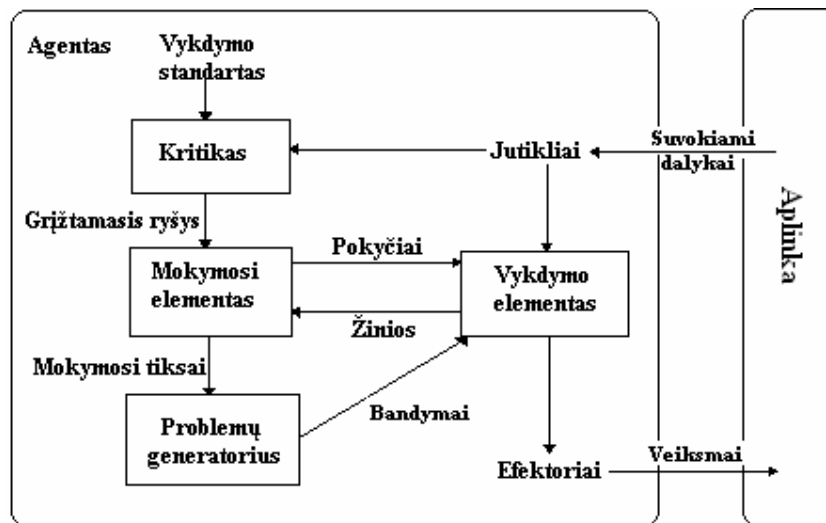
- **Skaidrumas (angl. Decomposability):** Sudėtingos loterijos gali būti sumažintos į paprastesnes naudojant tikimybės taisykles. Tai vadinama „neįdomaus lošimo“ taisykle, nes ji sako, kad agentas netenkina vieno loterijos, nes ji turi daugiau pasirinkimo galimybių nei kita.

$$[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$$

Pastebime, kad naudingumo aksiomų teorija nieko nesako apie naudingumą. Jos kalba tik apie pirmenybes. Pirmenybė (preference) laikoma pagrindine racionalaus agento savybe.

Mokymosi agentas

Mokymosi agentas gali būti padalintas į keturis konceptualius komponentus, pateiktus paveiksle 6. Svarbus skirtumas yra tarp mokymosi elemento ir vykdymo elemento. Mokymosi elementas yra atsakingas už patobulinimus, vykdymo elementas – atsakingas už išorinių veiksmų parinkimą. Vykdymo elementas yra tai ką mes anksčiau laikėme visą agentą. Mokymosi elementas pasiima žinių apie mokymosi elementą ir keletą atsiliepimų apie tai kaip agentas veikia ir nustato, kaip vykdymo elementas turi būti pakeistas, kad veiktų geriau ateityje. Mokymosi elemento konstrukcija labai priklauso nuo vykdymo elemento struktūros. Bandant suprojektuoti agentą, kuris mokosi tam tikrų gebėjimų. Pirmas kylantis klausimas nėra: "Kaip aš išmoksiu tai?", bet "Kokios rūšies vykdymo elementą mano agentas turi vaidinti, kuomet jis išmoko kaip tai atlikti?".



Pav. 7 Mokymosi agento struktūra

Kritikas yra sukurtas tam, kad pasakytų mokymosi elementui, kaip gerai agentas veikia. Kritikas naudoja nustatytą vykdymo standartą. Tai yra būtina, nes suvokti dalykai tarpusavyje netiekia jokios informacijos apie agento pasisekimą. Kaip pvz. šachmatų programa gali suvokti dalyką, kuris nurodo, kad jo oponentui yra šachas ir matas, bet jam reikalingas vykdymo standartas, kad sužinoti, kad tai yra naudingas dalykas. Pats suvokimas taip nepasako. Svarbu tai, kad vykdymo standartas yra nustatytas matas, kuris yra konceptualiai agento išorėje, kitaip agentas galėtų pritaikyti savo vykdymo standartus atitinkamai pagal savo elgesį.[30]

Kitos protingų agentų klasės

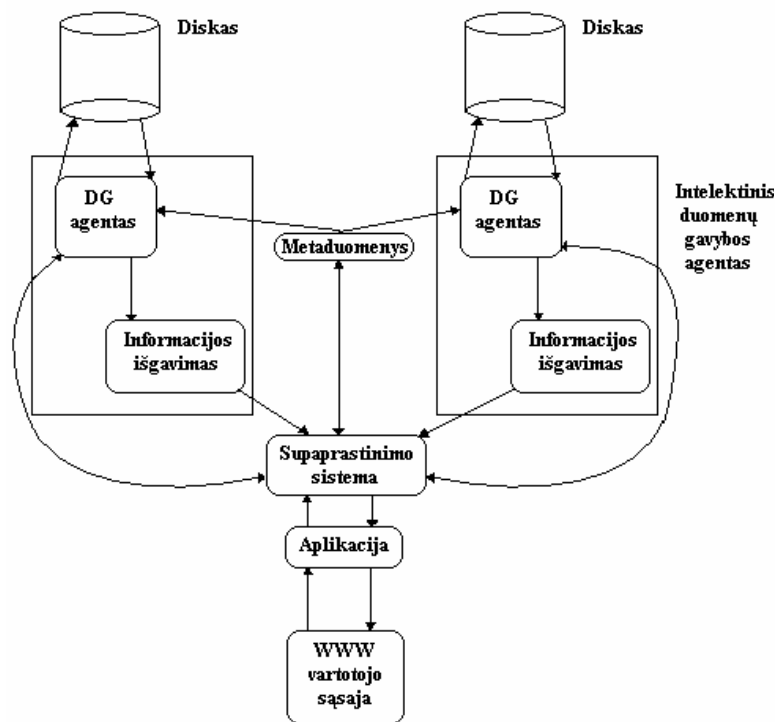
Remiantis kitais šaltiniais, keletas sub agentų, kurie gali būti protingųjų agentų dalis ar pats tikrasis protingas agentas yra:

- Sprendimų agentai (*angl.* Decision Agents) – orientuoti į sprendimų priėmimą.
- Įvesties agentai (*angl.* Input Agents) – vykdo ir supranta daviklių įvesties informaciją. Kaip pvz. neuroninio tinklo grįstas agentas.
- Apdorojimo agentai (*angl.* Processing Agents) – sprendžia problemas tokias kaip kalbos atpažinimas.
- Erdviniai agentai (*angl.* Spatial Agents) – susiję su fiziniu tikru pasauliu.
- Pasaulio agentai (*angl.* World Agents) – įtraukia visų kitų agentų klasių kombinacijas, kad leistų savarankiškai veikti.
- Fiziniai agentai (*angl.* Physical Agents) – fizinis agentas yra esybė, kuri suvokia sensorių pagalba ir veikia aktuatorių pagalba.

- Laikini agentai (*angl.* Temporal Agents) – laikinas agentas gali naudoti laiku grįžtą informaciją, kad pasiūlytų komandas ar duomenų veiksmus, kompiuterio programai ar žmogui ir naudoja programos įeigoje suvokiamus dalykus, kad suderintų savo sekančią funkciją.[31]

Duomenų gavybos agentas

Duomenų gavybos agentas – programinė įranga, kurios pagrindinis tikslas efektyviau vykdyti informacijos išgavimą. Tai yra vienas iš protingų agentų tipas, kuris veikia duomenų saugykloje ir atlieka bene sunkiausia darbą. Jis darbo tikslai įtraukia, ieškojimą aiškių sąveikos ryšių tarp skirtingų duomenų dalių. Šis agentų tipas gali aptikti pagrindinius tendencijų pasikeitimus, taip pat gali aptikti naują susijusia informaciją. Jei randama nauja informacijos dalis, agentas bandys išpėti galutinį vartotoją apie naujai aptiktą informaciją.



Pav. 8 Lygiairečiai veikiančių duomenų gavybos agentų struktūra

Kaip pavyzdžiui, įmonė gali sukurti agentą, kuris analizuos ekonomines tendencijas. Jei agentas nustatys, kad vartotojai tampa labiau konservatyvūs, sistema išpės vadovybė apie pasikeitimus. Turėdamą šią informaciją, vadovybė gali geriau suplanuoti, kaip geriau

pagaminti produktą, jį pateikti rinkai ir jį parduoti. Agento pagalba visas šis procesas vyks žymiai veiksmingiau.

Agentų hierarchijos – Multi-agentų sistemos

Tam, kad aktyviai įgyvendinti jų funkcijas, šiandien protingi agentai yra paprastai surinkti į hierarchinę struktūrą talpinančią daug „sub-agentų“. Protingieji sub-agentai vykdo ir atlieka žemesnio lygmens funkcijas. Bendrai paėmus, protingą ir sub-agentą, sukurta sistema vadinama galutinė sistema, kuri geba atlikti sudėtingus uždavinius ar įgyvendinti tikslus su funkcijomis ir atsakymais, kurie atspindi intelekto formą. [32]

Santrauka

Agentas – tai kas nors kas suvokia aplinką ir veikia joje. Mes išskaidome agentą į architektūrą ir agento programą.

Idealus agentas – tas, kuris visuomet imasi laukiamų (numatytų) veiksmų, siekiant padidinti savo efektyvumo mastą, atsižvelgiant į suvokiamo dalyko rezultata, kurį stebėjo iki šiol.

Agento programa planuoja iš suvokiamo dalyko į veiksmų atlikimą, kol atnaujinama vidinė būseną.

Aplinka, kuri yra sudėtingesnė už kitas, taip pat, yra nesuprantama, neapibrėžta, neatsitiktinė, dinaminė ir tolydi yra pati įdomiausia ir reikalaujanti daugiausiai pastangų.