VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Dokumentų analizė ir palyginimas naudojant ontologijas

Analysis and comparison of documents using ontologies

Magistro baigiamasis darbas

Atliko:	Gita Kurklietytė	(parašas)
Darbo vadovas:	asist. Linas Būtėnas	(parašas)
Recenzentas:	Doc. dr. Vladas Tumasonis	(parašas)

Vilnius – 2008

Santrauka

Šiuolaikiniame pasaulyje informacijos kiekiai auga milžiniškais tempais, todėl atsiranda poreikis ją apdoroti ir sisteminti kompiuterio pagalba pagal jos prasmę. Ontologijos, dar besivystantis ir evoliucionuojantis produktas, yra vis plačiau naudojamos įvairių rūšių informacinėms sistemoms intelektualizuoti. Vienas iš tokių pavyzdžių galėtų būti automatinis elektroninių laiškų analizatorius, veikiantis ontologijų principu – sistema, atpažįstanti atėjusį laišką pagal jo prasmę ir galinti sugeneruoti jam atsakymą.

Šiame darbe aprašyta pagal išnagrinėtą konkrečią internetinių loterijų informacijos sritį suprojektuotos ontologijos ir jų interpretavimo taisyklės, veikiančios sukurtu žodžių dažnių bei klasės koeficientų palyginimo algoritmo principu. Taip pat suprogramuotas mechanizmas veikiantis ontologijų ir minėto algoritmo pagrindu, klasifikuojantis elektroninio laiško tekstą pagal jo prasmę, pritaikytas konkrečiai internetinių loterijų informacijos sričiai. Atlikta algoritmo klasifikavimo efektyvumo ir paklaidos analizė.

Raktiniai žodžiai: ontologijos, teksto klasifikavimas, teksto analizė, dokumentų palyginimas, algoritmas, xml.

Summary

Nowadays amounts of information are growing drastically. Now we have the new needs to manage this information according to its meaning with help of computer. Ontologies are still in evolution and progress phase, but this product is increasingly used to create intelligent information systems. One of samples can be an automatic analyzer for e-mails recognition working on the basis of ontologies. This is a system, which can recognize the new e-mail according to its meaning and send the corresponding reply.

In this research work are described an ontology, which was designed according to particular internet lotteries information domain, and the rules to ontology interpretation. These rules are working on the algorithm basis. The principles of computing are the comparison of word rates and the class coefficients. The algorithm was created to classify e-mails from above mentioned particular information domain. Furthermore, the ontology and the reading rules were realized in mechanism, which can classify the text according to its meaning. The efficiency and the classification error of this mechanism was analyzed and evaluated.

Keywords: ontology, text classification, text analysis, comparison of documents, algorithm, xml.

Turinys

Įvadas

Pagrindiniu šiandienos gyvenimo bruožu yra tai, kad pasaulis nuolat, nesustojamai ir vis sparčiau keičiasi. Naujos informacijos šaltiniai ir kiekiai katastrofiškai auga. Tvirtinama, kad per kiekvienus 18 mėnesių (t.y. kas pusantrų metų) pasaulyje sukauptos informacijos kiekiai padvigubėja. Taigi esminis šiandienos gyvenimo bruožas yra tai, kad mes gyvename katastrofiškai didėjančios informacijos pertekliaus fone.

Kompiuterijoje terminu ontologijos vadinamas tam tikros srities sąvokų visumos specifikavimas išreikštu pavidalu. Ontologijos yra vis plačiau naudojamos įvairių rūšių informacinėms sistemoms intelektualizuoti, be to, jos yra viena svarbiausių semantinio pasaulinio tinklo sudedamųjų dalių, kuris yra viena sparčiausiai besivystančių kompiuterijos sričių. Semantinis tinklas – ateities interneto tinklas, kaip globali duomenų bazė. Semantinio tinklo infrastruktūra leistų tiek žmonėms, tiek mašinoms daryti sprendimus kaip kategorizuoti informaciją ir ją panaudoti. Šiuo metu žiniatinklis remiasi daugiausia HTML kalba parašytais dokumentais. HTML yra tinkama kalba aprašymui, akcentuojant vizualinį atkūrimą, t.y. struktūrizuoto teksto masyvas, į kurį įsiterpia multimedijos objektai (paveiksliukai, interaktyvios formos). HTML turi ribotas galimybes klasifikuoti teksto blokus puslapyje, išskyrus, kai tai reikalinga dokumento struktūrai nustatyti ir suteikti jam norimą išvaizdą. Semantinis tinklas bando šiuos trūkumus pašalinti naudojant aprašomąsias technologijas RDF ir OWL bei į duomenis orientuotą, lanksčią žymėjimo kalbą XML. Šios technologijos yra sujungiamos taip, kaip būtų galima pateikti aprašymus, kurie papildytų arba pakeistų žiniatinklio dokumentų turinį. Taigi, semantiniai tinklai veda prie to, kad tinklo dokumentai bus apdorojami automatiškai. Semantinio tinklo kalba leidžia kurti ontologijas, kurios šiuo atveju yra žinynas, kur yra surašyti apibrėžimai, informacijos interpretavimo mechanizmai, žodynai su informacijos įvertinimu, t.y. informacijos interpretavimo biblioteka. Informacijos resursai sujungiami į vieną ar keletą ontologijų ir nurodoma, kokie apibrėžimai yra taikomi vienai ar kitai

informacijai. Logiškai apibrėžta sistema gali iš to daryti išvadas ir gauti papildomą informaciją, kuri nėra paminėta nei viename duotame informacijos vienete.

Ontologijos gali būti naudojamos ne tik globaliai, bet ir lokaliai – netgi įmonių vidinėse sistemose. Pasak šaltinio [Mas01] ontologijų naudojimo privalumai yra ne tik žinių struktūrizavimo palengvinimas naujose srityse, bibliotekoje saugomų komponentų pakartotinas naudojimas (išvengiant būtinumo kurti iš naujo), tarpusavio sąveikos tarp skirtingų komponentų palengvinimas, išsiaiškinant atitikmenis tarp jų naudojamų terminų bet ir intelektuali paieška apdorojant užklausas (t.y., automatiškai apibendrinant užklausas, surandant artimiausius dalinius atitikmenis).

Ontologijos kuriamos tam, kad būtų galima dalintis tam tikros srities informacijos struktūra tarp žmonių ir programų, tam, kad vienas ontologijas galima būtų panaudoti kuriant kitas, srities sąvokų ir sąryšių detalizavimui ir tikslinimui, tam, kad būtų atskiriamos tam tikros srities žinios nuo kitų žinių, taip pat naudojamos srities žinių analizei.

Ontologijos yra dar nebaigtas tyrinėti ir kurti produktas. Jis nuolat evoliucionuoja: naujos realizacijos yra kuriamos pagal augančius ir besikeičiančius poreikius bei reikalavimus. Bet kokiu atveju, ontologijos padeda atlikti reikšmingus informacijos apdorojimo uždavinius, tokius kaip teksto analizė, teksto generavimas ir samprotavimas apie teksto prasmę.

Taigi, informacijos kiekiai šiuolaikiniame pasaulyje auga beprotiškais tempais. Deja, šis augimas ženkliai pasunkino prieigą prie naudojamos ar ieškomos, netgi elektroninio formato, informacijos. Ieškant pagal raktinius žodžius gautuose paieškos rezultatuose beveik visada būna daug nereikalingos informacijos, kuri atsiranda dėl raktinių žodžių galimų dviprasmybių - tas pats žodis gali turėti kelias skirtingas reikšmes. Nurodyti paieškos sistemai, kuri reikšmė mus būtent domina, yra gana sudėtinga. Be to, paieškos sistemos naudotojas praranda dalį jam reikalingos informacijos, kuri yra logiškai susijusi su raktinio žodžio reikšme.

Milžiniški informacijos kiekiai apsunkina informacijos suvokimą ir apdorojimą. Beveik visa šiandien egzistuojanti informacija nėra aprašyta, todėl

yra neįmanoma rūšiuoti ją pagal reikšmę naudojantis kompiuterio pagalba, o daryti tai rankiniu būdu yra galima sakyti neįmanoma.

Gerai pasirinktos priemonės ir įrankiai leidžia ne tik optimizuoti darbą, bet sutaupyti nemažai laiko. Tiksli informacijos analizė įgalina priimti teisingus sprendimus, kas yra labai svarbu šiandieniniame informacijos pertekliaus pasaulyje. Priversti kompiuterį analizuoti informaciją taip, kaip analizuoja ją žmogaus smegenys – didžiulis pasiekimas ir laimėjimas kompiuterijos srityje. Dokumentų analizė naudojant ontologijas kaip tik ir sprendžia šias problemas.

Vienas iš darbo tikslų buvo išanalizuoti pagrindinius ontologijų elementus, bei sudarymo taisykles: sąvokas, esybių (reiškinių, daiktų) tipus, sąvokų hierarchijas, esybių tipų tarpusavio sąryšius, priklausomybes, aksiomas, taisykles, dėsningumus apie esybių tipus ir sąryšius, pavyzdinius atvejus. Išsinagrinėti galimas ontologijų rūšis, jų išreiškimo ir užrašymo būdus. Ontologijos gali būti užrašytos įvairiomis priemonėmis – tiek specialiai tam pritaikytomis, tiek tradicinėmis žinių vaizdavimo kalbomis. Išsiaiškinti ontologijų kūrimo būdus bei palyginti priemones ir įrankius.

Remiantis įgytomis teorinėmis žiniomis, buvo pasirinktos tinkamiausios priemonės ir metodai vienos konkrečios įmonės ateinančių elektroninių laiškų analizatoriui sukurti. Elektroninių laiškų tekstai būtų analizuojami remiantis ontologijomis. Analizės esmė – atpažinti naujai atėjusį elektroninį laišką ir išanalizavus jo turinį automatiškai jį priskirti kuriai nors ontologijoje esančiai klasei. Pagal priskirtą klasę galima sugeneruoti jam tinkantį atsakymą.

Kuriant ontologijų medį buvo atkreiptas dėmesys ne tik į jo tikslumą, bet ir į jo interpretavimo taisykles, kurios turi labai didelę reikšmę klasifikavimo efektyvumui. Ontologijų medis buvo sukurtas remiantis realiais elektroniniais laiškais, pagal šaltiniuose [NM01] ir [Den03] pasiūlytą metodologiją, kuri remiasi iteratyvumo dėsniu - išnagrinėta informacijos sritis buvo padalinta į savarankiškas klases, kurios vėliau buvo skaidomos į dar smulkesnes. Suprojektuotos ontologijos buvo užrašytos naudojantis „Protégé 3.3.1“ programa, o vėliau naudojama išeksportavus į XML formato failą.

Vienas iš pagrindinių darbo tikslų – sukurti efektyvų ontologijų interpretavimo algoritmą ir palyginti pagal jį skaičiuojant gautus elektroninių laiškų teksto klasifikavimo rezultatus. Tam tikslui buvo sukurtos trys programos naudojantis PHP, Javascript ir HTML programavimo kalbomis. Dvi iš jų klasifikuoja tekstą remiantis skirtingais algoritmais: pirmoji nuosekliai ieško šakoje reikšminių žodžių ir tikrina ontologijų klasės nuo aukščiausios iki žemiausios, antroji skaičiuoja klasių reikšminių žodžių dažnių ir hierarchijos koeficientų sandaugos reikšmes, juos lygina ir priskiria vienai ar kitai klasei. Trečioji programa skirta ontologijų apmokymui (žodžių dažnių skaičiavimui, kuriuos naudoja vienas iš anksčiau minėtų algoritmų). Visos trys programos naudoja anksčiau minėtą XML faile užrašytą ontologijų struktūrą.

1. Literatūros apžvalga

1.1. Ontologijų sąvoka

Žodis „ontologija“ graikų kalboje susideda iš dviejų žodžių: „būtis“ ir „žodis“, arba „kalba“. Pagal internetinę enciklopediją [Wik01] nuo senovės šis terminas yra naudojamas filosofijos skyriui apibūdinti, kuriame nagrinėjama būtis ir egzistencija, taip pat pagrindinės kategorijos, bandant išsiaiškinti, kokios ir kokių tipų esybės egzistuoja. Pagrindinis ontologijos klausimas filosofijoje — „Kas egzistuoja?“.

Tame pačiame šaltinyje teigiama, kad kompiuterijoje terminu „ontologijos“ vadinamas tam tikros srities sąvokų visumos specifikavimas išreikštu pavidalu. Šiuo atveju žodis „ontologijos“ naudojamas tik daugiskaitine forma. Ontologijos apibrėžia nagrinėjimo srities sąvokas, esybių (reiškinių, daiktų) tipus, sąvokų hierarchijas, esybių tipų tarpusavio sąryšius,

priklausomybes, aksiomas, taisykles, dėsningumus apie esybių tipus ir sąryšius [nebūtina dedamoji], pavyzdinius atvejus [nebūtina dedamoji].

Yra keletas skirtingų principų, pagal kuriuos yra skirstomos ontologijos. Pagal formalumą ontologijos skirstomos į neformalias (pvz., terminų katalogai) ir formalias, kurios savo ruožtu būna aksiomatizuotos (pvz.: formalios mokslų teorijos, taisyklių ir freimų rinkiniai ekspertinėse sistemose, duomenų bazių koncepcinių schemų specifikacijos), prototipais paremtos (terminologinės) ir mišrios. Formaliosiose ontologijose kategorijų sistemos aprašomos formaliai, vartojant tam tikrą matematinį formalizmą, pavyzdžiui, deskriptyvines logikas.

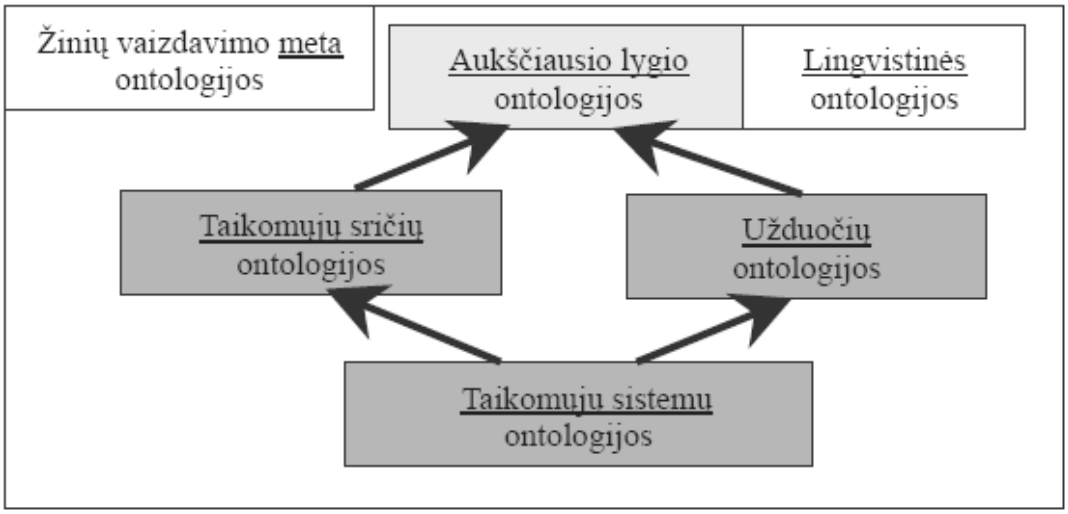
Pagal išreiškimo galią ontologijos skirstomos į „lengvasvores“ ontologijas (kurios išreiškia: sąvokas ir elementarius tipus, sąvokų hierarchiją, sąvokų sąryšius) ir „sunkiasvores“ ontologijas (kurios papildomai dar išreiškia ir: kardinalumo apribojimus, sąryšių klasifikaciją, galimybes manipuliuoti aksiomomis ir semantika, naudojant logikos formalizmus ir loginio išvedimo sistemas).

Straipsnyje [Gua98] yra išskirtos keturios ontologijų rūšys pagal paskirtį:

- Aukščiausio lygio ontologijos: aprašo bendriausias sąvokas (pvz. erdvė, laikas, objektas, įvykis, veiksmas, ir kt.), nepriklausomas nuo konkrečios problemos ar srities;
- Taikomųjų sričių ontologijos: aprašo konkrečių sričių žodynus (pvz. medicina, automobiliai), specializuodamos terminus, įvestus aukščiausio lygio ontologijose;
- Užduočių ontologijos: aprašo konkrečių užduočių ar veiklų žodynus (irgi specializuojant aukščiausio lygio ontologijų terminus);
- Taikomųjų sistemų ontologijos: aprašo konceptus, kurie yra tiek taikomųjų sričių, tiek užduočių ontologijų specializacija. Dažnai šie konceptai atitinka roles, kurias atlieka tam tikros taikomosios srities esybės vykdydamos tam tikrą veiklą.

Šią klasifikacinę schemą šaltinyje [Mas02] dar siūloma papildyti žinių vaizdavimo meta ontologijomis ir lingvistinėmis ontologijomis:

- Lingvistinės ontologijos: žodžiai kategorizuojami i daiktavardžius, veiksmažodžius, būdvardžius, prieveiksmius. Išreiškiami semantiniai sąryšiai tarp žodžių reikšmių: sinonimai, antonimai;
- Taikomųjų sričių ontologijos.



1 pav., Svarbiausios ontologijų rūšys

Viena iš svarbiausių ontologijų apibūdinimo ypatybių yra *sudėtingumas*. Paprastai yra skiriami tokie ontologijų sudėtingumo lygiai:

- Terminai
- Terminų apibrėžimai;
- Siauresnių/platesnių terminų sąryšiai;
- Neformalūs arba formalūs apibendrinimo (angl. „IS_A“) sąryšiai;
- Formaliai nurodyti konkretūs atvejai;
- Freimai (savybės);
- Galimų reikšmių apribojimai;
- Loginės sąlygos (suvaržymai, aksiomos);
- Inversijos, nepersidengimo (angl. „disjointness“), visumos - dalies (angl. „PART_OF“) sąryšiai ir kt.

2.1. Ontologijų kūrimas

Ontologijas sudaro klasės, kartais vadinamos koncepcijomis, klasės savybės (slotai), kurios aprašo klasių savybes ir atributus (kartais vadinamos vaidmenimis) bei jų apribojimai (facets). Ontologijos kartu su pavyzdžiais suformuoja žinių bazę. Egzistuoja labai nežymi riba kur baigiasi paprastos ontologijos ir prasideda žinių bazė. Klasės sudaro ontologijų pagrindą. Klasės gali turėti poklasius. Atitinkamai poklasių aukštesnės hierarchijos klasė vadinama superklase. Praktiškai ontologijų kūrimas susideda iš klasių nustatymo, klasių hierarchijos poklasiai-superklasės nustatymo, slotų ir jų apribojimų sudarymas ir slotų konkretiems pavyzdžiams užpildymas. Slotų reikšmės gali būti simbolių eilutė, skaičius, ar kurios kitos klasės reikšmės. Slotai taip pat gali turėti vienetinę reikšmę ar daugybines reikšmes.

Iš žiniatinklyje surasto straipsnio [Den03] galima daug sužinoti ne tik apie ontologijų panaudojimo galimybes, bet ir jų kūrimo būdus. Taigi, ontologijos paprastai yra kuriamos daugiau ar mažiau laikantis šių principų:

1. Įgyti srities žinių

   Susirinkti būdingus informacijos šaltinius ir juos ištirti. Nustatyti, kokie bus naudojami terminai formaliam dalykų aprašymui, užtikrinti jų suderinamumą ir neprieštaringumą. Šie apibrėžimai turi būti surinkti taip, kad jie galėtų būti lengvai išreikšti ontologijoms pasirinkta bendra kalba.

2. Ontologijos projektavimas ir realizacija.

   Bendros konceptualios srities struktūros projektavimas. Tai gali apimti srities principinių konkrečių sąvokų ir jų ypatybių, ryšių tarp sąvokų identifikavimą, abstrakčių principų, kaip organizavimo kriterijų, kūrimą, nustatymą ar įtraukimą kitų naudojamų ontologijų, išskyrimą ar pritaikymą kitų nuostatų, kurios priklauso nuo pasirinktos kūrimo metodologijos.

3. Apjungimas.

Nustatytų ir išaiškintų principų ir ryšių sujungimas tam, kad būtų tinkamai gaunami norimi rezultatai.

4. Tikrinimas

Sintaksinių, loginių ir semantinių prieštaringumų suderinimas tarp ontologijų elementų.

5. Diegimas.

Galutinis patikrinimas ir diegimas į numatytą aplinką.

Ontologijų konstravimas nėra griežtai linijinis procesas, galima atlikti iškart kelis darbus. Tai yra pakankamai iteratyvus procesas. Preliminarios (skeletinės) struktūros yra išplečiamos ir tobulinamos, papildomos naujais ryšiais.

Ontologijų kūrimo procesas gali apimti problemos specifikavimą, srities žinių įgijimą ir analizę, koncepcinį planą ir apjungti ontologijų iteratyvų konstravimą ir testavimą. Šiam procesui palengvinti yra tam tikros priemonės, skirtos automatizuoti dalį darbų.

Ontologijų kūrime paprastai galima išskirti tokius etapus: reikalavimų specifikavimas, konceptų visumos paruošimas, realizavimas ir įvertinimas. Reikalavimų specifikavimo fazėje apibrėžiama taikomoji sritis ir ontologijų paskirtis, naudotojai ir atsakingi už ontologijų priežiūrą asmenys, išsiaiškinama, ar nėra galimybių pasinaudoti jau esamomis ontologijomis, jas pritaikant esamoms reikmėms, integruojant. Ontologijos konceptų visumos paruošimo fazėje išvardinami nagrinėjamoje taikomojoje srityje naudojami terminai, apibrėžiamos konceptų klasės ir klasių hierarchija, apibrėžiamos klasių savybės (freimo slotai), apibrėžiami slotų aspektai (matiškumas, reikšmių tipai, klasės, kurioms priskiriami slotai, leistinos egzempliorių klasės), nurodomi klasių egzemplioriai.

Iš šaltinio [Mas02] sužinojau, kad formuluojant ontologijų reikalavimus būtina nurodyti ontologijų paskirtį, pobūdį, realizavimo priemones ir stadijas, numatomo panaudojimo ypatumus. Visų pirma kaip galima tiksliau apibrėžti, kokia yra numatoma ontologijų taikomoji sritis ir kokiu tikslu ontologijos bus kuriamos. Tam, kad būtų išsiaiškinta, kokiais principais bus kuriamos ontologijos, reikia apsiibrėžti, kokios rūšies ontologijos yra kuriamos, kokio sudėtingumo lygio

ontologijas numatoma kurti; koks numatomas elementų detalizuotumo lygis. Taip pat reikia susiplanuoti, kokie žinių šaltiniai bus panaudoti kuriant ontologijas; kokie žinių įgijimo metodai bus naudojami ontologijoms kurti ir kokios techninės priemonės numatomos naudoti: kokia ontologijų kūrimo aplinka, kokia kalba (ar kelios kalbos) bus naudojama ontologijoms specifikuoti. Reikia tiksliai nustatyti, kokie numatomi ontologijų realizavimo etapai ir kaip rezultatai bus dokumentuoti, patikrinti ir įvertinti. Tam, kad darbas būtų pilnai užbaigtas, reikia įvardyti, kas bus ontologijų naudotojai, kokie numatomi ontologijų panaudojimo scenarijai, kokios taikomosios sistemos naudosis sukurtuoju produktu ir kaip bus vykdoma priežiūra bei tolesnis vystymas.

Internete suradau žiniatinklį [BV01], kuriame galima pasitikrinti ir įvertinti savo sukurtas ontologijas. Deja, autoriai už teisingumą negarantuoja...

1.3. Ontologijų kūrimo metodikos ir kalbos

Dažniausiai naudojamos ontologijų kūrimo metodikos yra:

- M. Uschold'o metodika, paremta patirtimi kuriant „Įmonių ontologijas";
- M. Grüninger'io ir M.S. Fox'o (TOVE) metodika;
- METHONTOLOGY karkasas;
- KACTUS metodika;
- metodika SENSUS projekto pagrindu;
- On-To-Knowledge (OTK) metodika, sukurta Europos Sąjungos 5 bendrosios programos Informacinės visuomenės technologijų (ES IVT) projekte „Kontekstinio žinių valdymo priemonės, naudojančios evoliucionuojančias ontologijas".

Ontologijų išreiškimui dažniausiai naudojamos tokios ontologijų specifikavimo kalbos:

- Tradicinės:
    - CARIN, paremta Horno taisyklėmis ir deskriptyviomis logikomis;

- Flogic, integruojanti freimų logiką ir pirmos eilės predikatų skaičiavimą;
- LOOM, paremta pirmos eilės predikatų logika ir priklausanti KL-ONE kalbų šeimai;
- operacinė koncepcinio modeliavimo kalba OCML;
- Ontolingua, paremta keitimosi žiniomis kalba KIF. (Savo ruožtu, Ontolingua ontologijas galima eksportuoti Lisp , KIF, Generic-Frame, Loom, Clips, Epikit, HTML formatais);
- Express objektinė kalba informacijos modeliu specifikavimui, priklausanti ISO-10303 STEP („Standard for the Exchange of Product model data“) standartų šeimai.

- Specializuotos:
  - CycL (skirta visuotinai pripažįstamų žinių išreiškimui CyC ontologijose);
  - GRAIL (medicininėms ontologijoms GALEN);
  - naratyvinių dokumentų semantinio konteksto aprašymo kalba NKRL.
- Skirtos naudojimui internete:
  - paprastas HTML plėtinys ontologijoms SHOE;
  - keitimosi ontologijomis kalba XOL;
  - ontologijų žymėjimo kalba OML;
  - Resursų aprašymo karkaso RDF Schema;
  - Ontologijų išvedimo sluoksnis OIL;
  - Pasaulinio tinklo ištekliu semantinio žymėjimo kalba DAML+OIL;
  - Pasaulinio tinklo ontologijų kalba Web Ontology Language (OWL).

Kadangi pagrindinis šio darbo tikslas yra ištirti ontologijų interpretavimo galimybes, todėl toliau nebuvo gilinamasi į pačių ontologijų kūrimo metodikas. Tai yra labai plati sritis, ją galima išskirti ir analizuoti kaip atskirą darbą.

1.4. Ontologijų projektavimas

Kaip teigiama straipsnyje [NM01], nėra priimto vieno „teisingo“ kelio ar metodikos kaip kurti ontologijas. Čia aptariami bendri galimi rezultatai ir siūlomas ontologijų kūrimo procesas. Visų pirma reikia apibrėžti taisykles, kurios padeda priimti sprendimus projektuojant. Visų pirma taisyklė – nėra nei vieno „teisingo kelio, kaip sumodeliuoti sritį – visuomet yra alternatyvų“. Geriausias sprendimas beveik visada priklauso nuo srities, kurioje planuojama taikyti ontologijas ir galimo jų išplėtimo. Kūrimas visuomet būna iteracinis procesas. Sąvoka ontologijose turėtų būti apribota iki objektų (fizinių ar loginių) ir srities sąryšių. Tai panašiausia į daiktavardžius (objektai) ar veiksmažodžius (sąryšiai) sakiniuose, kurie aprašo sritį. Taigi, ontologijos yra realaus pasaulio modeliai, taigi principai joje turi atitikti realybę. Pirmiausia yra sudaroma grubi klasių struktūra, vėliau ji peržiūrima, tikslinama ir papildoma

Kuriant ontologijas reikia vadovautis tuo kam ji bus naudojama, apgalvoti jos naudojimosi, aiškumo, atitikimo tikrovei aspektus. Praktiškai ontologijos gali būti redaguojamos visą savo gyvavimo laiką.

Straipsnio [NM01] autoriai siūlo, prieš kuriant ontologijas visų pirma jas apibrėžti, t.y. atsakyti į tokius klausimus: kokią sritį turėtų apimti ontologijos? Kam jos bus naudojamos? Į kokio tipo klausimus jos turi atsakyti? Kas jas naudos?

Klasių hierarchijai kurti yra keletas galimų būdų: iš viršaus į apačią (iš pat pradžių apibrėžiamos pačios pagrindinės sąvokos, o paskui seka sąvokų specializavimas), iš apačios į viršų (iš pradžių apibrėžiamos labiausiai specifinės klasės, paskui jos sugrupuojamos į pagrindines, bendresnes sąvokas) ir kombinuotas (apima pirmąjį ir antrąjį hierarchijos kūrimo būdus). Kurį būdą pasirinkti priklauso tik nuo paties kūrėjo, nėra vieno būdo kažkuo išsiskiriančio ir geresnio už kitus.

Tolesniame žingsnyje apibrėžiamos klasės bei jų hierarchija, kurioje galioja tokia taisyklė:

Jei klasė A yra viršklasis klasės B, tai kiekvienas klasės B elementas yra taip pat ir klasės A elementas.

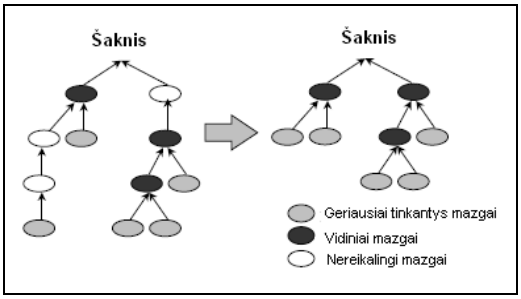
Kitais žodžiais tariant, klasė B yra klasės A „vaikas“, poklasiai paveldi superklasės savybes. Tačiau vien tik klasių iškeltiems klausimams atsakyti nepakanka. Taigi, klasių hierarchiją reikia papildyti savybėmis ir ryšiais tarp klasių. Klasių hierarchijos sudarymui svarbu, kad būtų išlaikyti sąryšiai: apibendrinimo (angl. is-a) ir agregavimo (angl. kind-of).

Straipsnyje [ČČ03] teigiama, kad viršutiniame lygmenyje apibrėžiamos pačios bendriausios sąvokos (esybė, įvykis, laikas, procesas ir pan.), paskui – dalykinės srities sąvokos (dalykinės srities ontologijos), šioje srityje vykstančių procesų aprašymo sąvokos (proceso ontologijos) ir pagaliau sprendžiamų uždavinių aprašymo sąvokos (problemos ontologijos). Šitaip įgyvendinamas N. Guarino pasiūlytas žinių apie dalykinę sritį ir žinių apie procesus nepriklausomumo principas. Procesai aprašomi vaidmenų terminais ir susiejami su problema priskiriant tiems vaidmenims vienas ar kitas dalykinės srities esybes.

Kad būtų išvengta klasių hierarchijos klaidų patartina klases vadinti tik daugiskaita ar tik vienaskaita. Taip pat negali būti klasių ciklų.

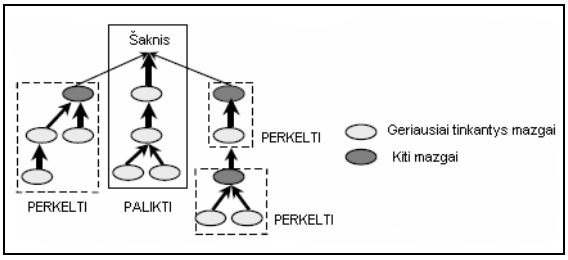
Šaltinyje [NM01] aprašomi būdai, kaip patikrinti klasių hierarchiją, tačiau informacija daugiau pritaikyta viena konkrečiam atvejui, pasigedau bendrų metodinių nurodymų.

Apie klasių hierarchiją ir šioje srityje sprendžiamas problemas daugiau informacijos radau straipsnyje [Yam01]**.** Čia rašoma apie tai, kaip konstruoti tam tikros srities ontologijas su keliais sąryšių tipais. Taip pat radau itin aktualios mano magistriniam tiriamajam darbui medžiagos - kaip surasti ir pašalinti nereikalingus klasių hierarchijos medžio mazgus. Šį procesą galima vykdyti laikantis tam tikrų taisyklių. Kaip sutrumpinti ontologijų medį pašalinant nereikšmines klases pavaizduota 2 paveikslėlyje.



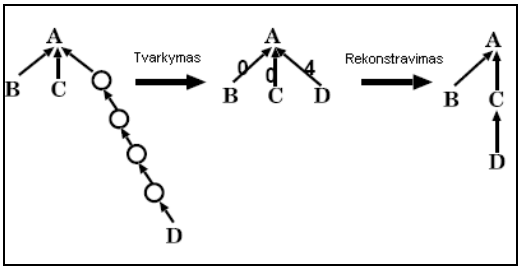
2 pav., Nereikalingų klasių pašalinimas

Po to, kai iš klasių hierarchijos medžių pašalinami nereikalingi mazgai (klasės), gali būti taikoma kelių rūšių analizė: sutapimo *(match result analysis)* ar „tvarkymo“ *(trimmed result analysis)*. Pirmos analizės atveju atskiriama, kurie keliai (šakos) yra sudaryti tik iš tinkamai priskiriamų mazgų ir kurie keliai (šakos) ne tik iš tinkamai priskiriamų. Pirmoji mazgų grupė gali būti perkeliama, abi grupės pavaizduotos 3 pav.



3 pav. Sutapimo analizės rezultatai

Jei klasių hierarchijos modelyje yra didelis skirtumas tarp pomedžių ilgių, atlikus „tvarkymo” analizę galima pakeisti to pomedžio struktūrą. Šios analizės rezultatai pavaizduoti 4 paveikslėlyje.



4 pav., „Tvarkymo” analizės rezultatai

Visa medžiaga, pateikta straipsnyje [Yam01] yra naudota srities ontologijų greito kūrimo aplinkoje DODDLE II, tačiau teorines ontologijų medžio analizės taisykles galima taikyti ir kitais atvejais.

1.5. Pakartotinis ontologijų panaudojimas

Neverta gaišti laiko tam, kas jau yra padaryta. Jau sukurtus dalykus tereikia pritaikyti savo darbui, išplėsti ar truputį paredaguoti. Pakartotinis ontologijų panaudojimas gali būti tiesiog privalomas, jei ruošiama sistema ateityje turės sąveikauti su kitomis, kurios jau turės savo ontologijas ir žodynus. Daugelis ontologijų jau yra pasiekiamos elektroninėje formoje ir gali būti importuotos į kitą aplinką. Pavyzdžiui, pakartotinai naudojamų ontologijų bibliotekos pasauliniame žiniatinklyje yra [OB01] ir [OB02].

1.6. Ontologijų kūrimui skirti įrankiai

Kuriant ontologijas svarbu pasirinkti tinkamą darbo priemonę – įrankį, su kurio pagalba būtų kaip galima greitai, tiksliai ir optimaliai kurti ontologijų schemas atskirai arba kartu su pavyzdžiais. Šaltinyje [Den03] radau gana išsamų sąrašą (daugiau nei 50), kuriame aiškiai, tiksliai ir glaustai aprašyta nemažai ontologijų kūrimui skirtų priemonių - redaktorių.

Išanalizavus [Den03] pateiktų ontologijų kūrimo priemonių sąrašą bei jų savybes, išskyriau keletą punktų, kuriais kaip kriterijais galima vadovautis renkantis tinkamiausią redaktorių:

- Kalba, skirta užrašyti (šifruoti) ontologijoms;
- Palaikymas žiniatinklyje ir naudojimas;
- Kitų ontologijų importavimo/jau sukurtų eksportavimo formatai;

- Grafinis būdas, kuriuo jau sukurtos ontologijos gali būti tiesiogiai kuriamos, derinamos ir redaguojamos;
- Kokio yra lygio automatinės ontologijų sintaksės, jų ryšių ir/ar loginio jų taisyklingumo tikrinimas;
- Savybės, kurios leidžia ir palengvina lygiagretų kelių ontologijų kūrimą;
- Galimybė lengvai palyginti ir apjungti nepriklausomai ir atskirai sukurtas ontologijas;
- Gebėjimas išskirti sąryšius pagal leksiką tarp ontologijos elementų (pvz., sinonimai) ir leksikinio turinio apdorojimas (pvz., paieška/filtravimas);
- Gebėjimas išskirti iš turinio adresuotą ontologijų informaciją ir galimybė patobulinti ontologijas.

2. Ontologijų, apibūdinančių internetines loterijas, kūrimas

2.1. Įmonės apibūdinimas

Kaip minėta literatūros apie ontologijas apžvalgoje, ontologijų kūrimas prasideda nuo smulkios ir išsamios informacijos srities analizės. Informacijos sritis – internetinė loterija, kurią organizuoja UAB „Olifėja”.

UAB „Olifėja“ buvo įkurta 1992 metų gruodžio 12 dieną. Bendrovė yra loterijų rinkos lyderė. Ji rengia tiesioginio ryšio (on-line) loteriją „PERLAS“ (Loterijos žaidimai „Keno“, „Jėga-2”, „Tuzinas“, „Teleloto“), momentinę loteriją „OLIMPAS“, telefoninę loteriją „LOTO 1634“ ir internetinę loteriją „ELOTERIJA“. „Olifėja“ yra populiariausių Lietuvoje TV šou žaidimų „Teleloto“ ir „Tuzinas auksinių“ organizatorė.

Visi bendrovės organizuojami žaidimai yra registruoti kaip loterijos, todėl jų prizų fondą kontroliuoja Valstybinė lošimų priežiūros komisija, prizams yra skiriama ne mažiau kaip 50 proc. bilietų nominalios vertės.

Pats naujausias UAB „Olifėja“ siūlomas produktas – internetinė loterija, pasiekiama adresu www.eloterija.lt. Ji startavo 2006 metų rugsėjo 5 dieną. Tai buvo pati pirmoji tokio pobūdžio loterija Lietuvoje. Visus programinius sprendimus ir žaidimus su vaizdo grafika UAB „Olifėja“ perka iš Suomių bendrovės „EGET“ (European Game & Entertainment Technology Ltd). „EGET“ yra viena iš lyderių Europos internetinių loterijų rinkoje. Jų sukurtos Loterijų sistemos yra įdiegtos Suomijoje („Fintoto“, „paf.fi“, „GAMarena“), Čilėje („Intralot“), Estijoje („Fortuuna.ee“) ir Lietuvoje.

Pasinaudojus UAB „Olifėjos“ ir „EGET“ loterijų organizavimo patirtimi Lietuvos rinkai buvo atrinkti ir adaptuoti tinkamiausi techniniai sprendimai ir žaidimai. Ši Loterijų sistema yra tiesioginio ryšio (on-line), taigi žaidėjai gali žaisti loterijas kada tik nori – 7 dienas per savaitę ir 24 valandas per parą. Jos saugumo lygis – itin aukštas. Norėdamas žaisti iš tikrų pinigų žaidėjas visų pirma turi užsiregistruoti Loterijų sistemoje ir patvirtinti savo tapatybę naudojantis bankine autentifikacija. Tokiu būdu yra užtikrinama, kad žaidėjas yra pilnametis, fizinis asmuo. Visi pateikti duomenys, naudojantis Loterijų sistema yra laikomi, saugomi ir tvarkomi vadovaujantis Lietuvos Respublikos Asmens Duomenų Teisinės Apsaugos Įstatymu. UAB „Olifėja“ turi Valstybinės Duomenų Apsaugos Inspekcijos leidimą tvarkyti asmens duomenis. Žaidėjo sąskaita Loterijų sistemoje papildoma naudojantis elektronine bankininkyste - atliekant pavedimus. Bet koks komunikavimas tarp Loterijų sistemos ir bankų yra vykdomas naudojant 128 bitų SSL koduotą tinklą ir COMODO išduotą sertifikatą.

Internetinių loterijų tinklalapyje yra 9 žaidimai, registruotis ir žaisti iš tikrų pinigų gali 4 bankų internetinės bankininkystės klientai: AB DnbNord banko, AB Sampo banko, AB SEB Vilniaus banko ir AB Hansa banko.

Ontologijos kuriamos pagal internetinės loterijos žaidėjų atsiųstus laiškus į klientų aptarnavimo centrą adresu info@eloterija.lt. Beveik visuose laiškuose klausiama apie kažkokias iškilusias problemas, klausimų spektras labai platus –

nuo klausimų apie žaidimų taisykles iki bankinių operacijų. Elektroninių laiškų analizatorius suteiktų galimybę atsakinėti į tokius laiškus automatiškai ir sutaupytų nemažai laiko.

2.2. Ontologijų srities apibrėžimas

Kaip ir buvo siūloma šaltinyje [NM01], prieš sudarinėjant ontologijas reikia atsakyti į keletą klausimų:

- Kokia yra sritis, kurią turi apibūdinti ontologijos?

Šiuo atveju sritis – internetinės loterijos.

- Kokiam tikslui yra kuriamos?

Gautų elektroninių laiškų apie internetines loterijas klasifikavimui.

- Į kokius klausimus turi atsakyti ontologijos?

Klausimų aibė labai plati – nuo klausimų apie žaidimų taisykles, iki bankinių operacijų.

- Kas naudosis ontologijomis?

Sukurtomis ontologijomis galėtų naudotis darbuotojai, rankiniu būdu atsakinėjantys į anksčiau minėtus elektroninius laiškus.

Ontologijos projektuojamos remiantis tuo, į kokius klausimus jos turi pateikti atsakymus. Šiuo atveju klausimų sąrašą sudaryti labai lengva turint realių laiškų pavyzdžių, nes kiekvienas laiškas yra klausimo pavyzdys.

2.3. Duomenų aibės apdorojimas

Tyrimui ir ontologijų projektavimui naudota medžiaga – visi į info@eloterija.lt pašto dėžutę nuo loterijos starto gauti elektroniniai laiškai. Gauta korespondencija tvarkoma naudojantis Pegasus Mail programa. Iš viso buvo gauta apie 3000 laiškų, juos išeksportavau į vieną failą tam, kad būtų lengviau

dirbti su laiškų aibe. Šie laiškai buvo naudojami ne tik kaip klausimai ontologijų sudarymui, bet ir vėliau, testuojant bei apmokant elektroninių laiškų analizatorių.

Kaip buvo minėta anksčiau, UAB „Olifėja“ turi licenciją saugoti asmens duomenis. Su saugomais duomenimis privalu elgtis vadovaujantis Lietuvos Respublikos Asmens Duomenų Teisinės Apsaugos įstatymu. Kai kurie žmonės rašydami laiškus nurodo savo asmens kodą ar banko sąskaitą. Jie mano, kad tai yra reikalinga sprendžiant pavedimų ar registracijos problemas. Iš tikrųjų tai nėra būtina, o reikalinga tik išskirtiniais atvejais. Klientų aptarnavimo centro darbuotojai informacijos paieškai naudojasi elektroninio pašto adresu, Žaidėjo identifikaciniu numeriu (Žaidėjo identifikacinis numeris – unikalus, registracijos metu automatiškai vartotojui suteiktas numeris, skirtas jį identifikuoti Loterijų sistemoje, *angl. PlayerID*), vardu ar pavarde. Asmens kodas iš vienos pusės yra asmens tapatybę patvirtinanti skaičių seka, iš kitos pusės – tiesioginė informacija apie asmenį. Asmens kodo struktūra ne tik atskleidžia asmeninę informaciją, bet yra unikali ir nekeičiama identifikavimo priemonė. Asmens kodo išviešinimas kelia grėsmę, kad atsiras galimybė jungti įvairiose informacinėse sistemose tvarkomus asmens duomenis, be to yra pavojus, kad asmens tapatybė elektroninėje erdvėje gali būti pasisavinta.

Taigi, visų pirma eksportuotame faile suradau tam tikrus simbolių junginius ir juos pakeičiau, kad bet kokiu atveju būtų išsaugotas asmens duomenų konfidencialumas

1 lentelė. Konfidencialių duomenų pakeitimas laiškų aibėje

Ieškomi duomenys	Jų formatas	Kuo buvo pakeista į:
Asmens kodas	nnnnnnnnnnn[1]	xxxxxxxxxxx
Banko sąskaitos numeris	LTnnnnnnnnnnnnnnnnn[1]	LTxxxxxxxxxxxxxxxxxx

Išeksportuotame laiškų faile palikau žaidėjų vardus ir pavardes, nors jie taip pat yra asmens duomenys. Deja, nesugalvojau būdo, kaip juos galima

[1] Čia n – bet koks sveikas skaitmuo nuo 0 iki 9

atskirti nuo visų žodžių. Mano nuomone, tai nėra tokios pat svarbos duomenys, kaip asmens kodas ar banko sąskaita.

Taip pat, kad būtų paprasčiau ir aiškiau, iš kiekvieno laiško pradžios pašalinau meta duomenis (informaciją apie laiško koduotę, versijas, kodavimą ir t.t.).

Dar viena problema, kuri iškilo eksportuojant laiškus į failą – lietuviškos raidės, jos tapo tam tikromis simbolių sekomis. Visas lietuviškas raides arba jas atitinkančias simbolių sekas keičiau atitinkamomis lotyniškomis raidėmis. Žinoma, atsirado problema kad du skirtingi žodžiai turės vieną išraišką Pavyzdžiui, šakės (vardininko linksnis) ir sakės (naudininko linksnis, vardininkas – sakė) tapo vienu žodžiu „sakes“. Žinoma, į tai bus atsižvelgta kuriant ontologijas, bet mano nuomone, tokių atvejų galimybė yra labai maža ir jie neturėtų turėti įtakos tyrimui, ontologijų konstravimui ar galutiniams rezultatams.

Tokiais būdais sutvarkiau išeksportuotą laiškų failą tolesniam nagrinėjimui ir tyrimui.

2.4. Svarbiausių klausimų išskyrimas

Nagrinėjant laiškų aibę pastebėjau, kad daugelis klausimų juose kartojasi. Taigi, iš visos laiškų aibės išskyriau dažniausiai pasitaikančias problemas. Jas pavaizdavau žemiau esančiame paveikslėlyje.

5 pav. Laiškų aibė

Smulkiau išskyriau kiekvieną sritį:

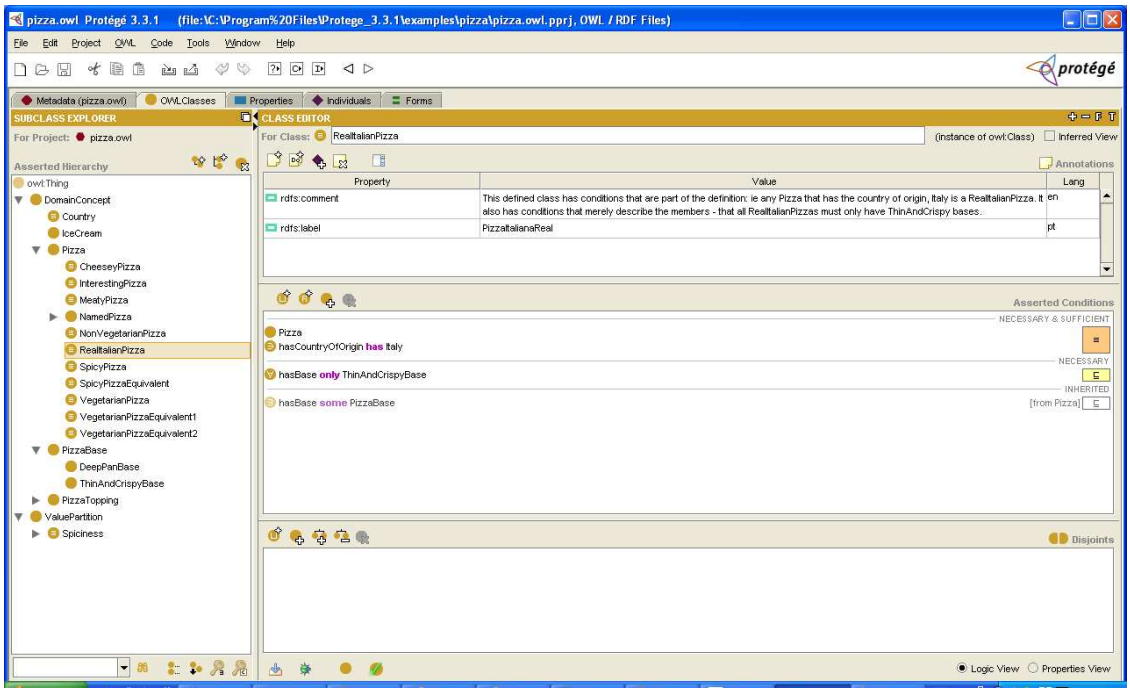
1. Registracija
    a) Įvedus neteisingai banko sąskaitą
    b) Įvedus jau egzistuojantį vartotojo vardą
    c) Registruojantis jau užregistruotam žaidėjui
2. Bankinės operacijos
    a) Nesėkmingas pavedimas į Žaidėjo sąskaitą
    b) Atgalinis pavedimas į banko sąskaitą
3. Žaidimai
    a) Žaidimų taisyklės
    b) Žaidimų laimėjimų lentelės
    c) Žaidimų apribojimai
4. Kompiuterinės problemos
    a) Nemato žaidimų (neįdiegta Flash programa)
    b) „Neteisinga sesija“
5. Saugumas
    a) Klausimai dėl legalumo
    b) Klausimai dėl asmens kodo reikalingumo
    c) Klausimai dėl asmens duomenų apsaugos
6. Prisijungimo problemos
    a) Pamiršo slaptažodį
    b) Pamiršo prisijungimo vardą
    c) Žaidėjo sąskaitos sustabdymas
7. Atsakymo nereikalaujantys laiškai
    a) Nepristatyti sugrįžę laiškai
    b) Pristatyto laiško pažyma
    c) Nepageidaujami elektroniniai laiškai (SPAM‘as)
    d) Retoriniai pasipiktinusiųjų laiškai
    e) Reklamų pasiūlymai
8. Jau kartą atsakyti laiškai (Tęsiamas dialogas).

Dažniausiai pasikartoja klausimai apie bankinius pavedimus į Žaidėjo sąskaitą, užmiršus slaptažodį ar prisijungimo vardą ir dėl laimėjimo persivedimo į banko sąskaitą.

Pagal išskirtas problemas vėliau bus generuojami raktiniai žodžiai (pavyzdžiai), kurie bus naudojami ontologijų kūrime.

2.5. Ontologijoms užrašyti pasirinktos priemonės

Atlikus programinių įrankių ontologijoms kurti analizę, pasirinkau sau tinkamiausią ontologijų kūrimo priemonę – programą „Protégé 3.3.1“, iš internetinio tinklalapio [PR01]. Kaip atrodo programos darbo laukas galite pamatyti žemiau esančiame paveikslėlyje. Taip pat radau labai gerą naudojimosi šia programa vadovą [HKR+04]. Čia ne tik išsamiai paaiškinamos naudojimosi programa galimybės, bet ir ontologijų projektavimo principai.



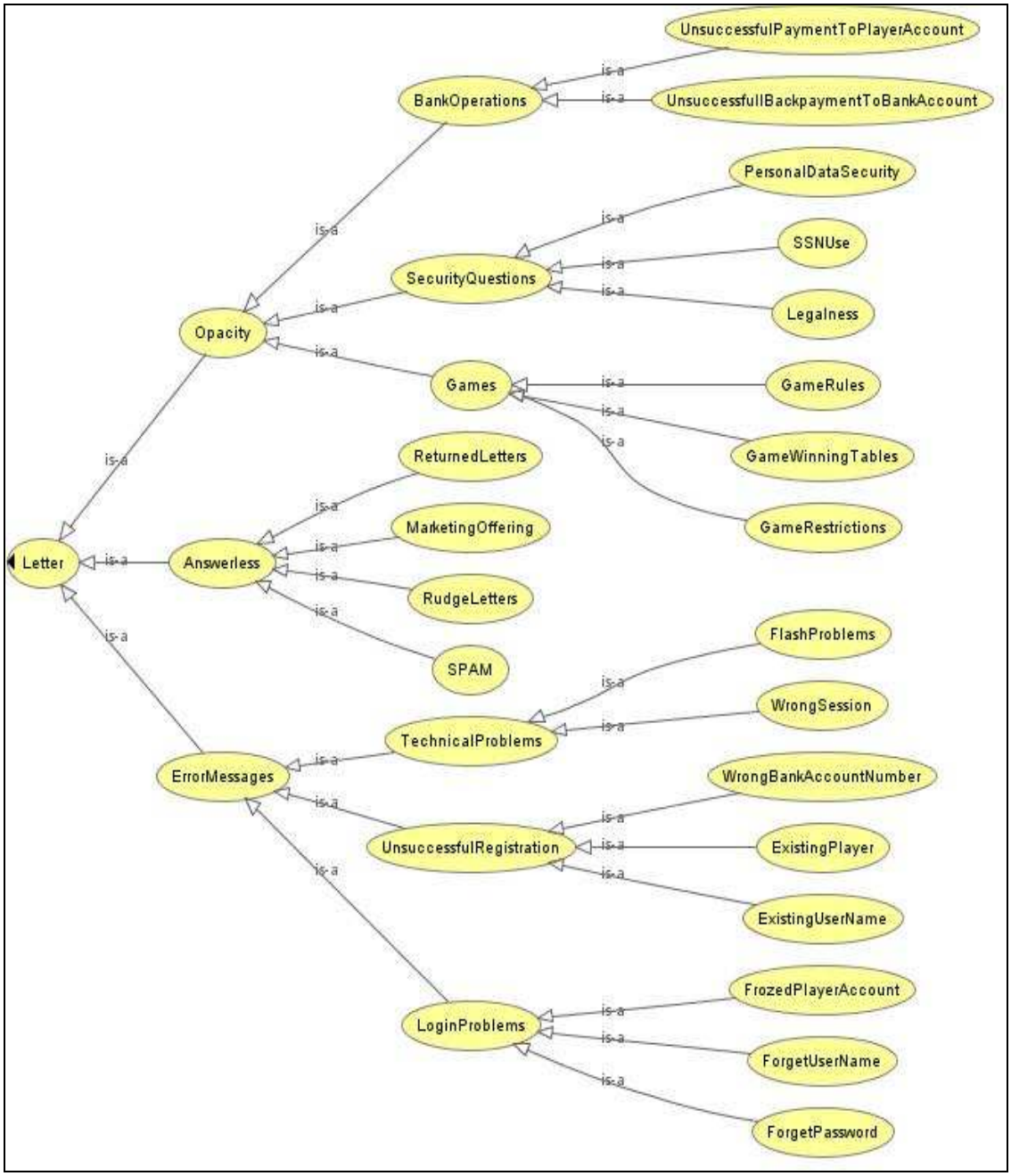
6 pav. Programos „Protégé“ darbinis langas

Vienas iš ontologijų kūrimo ir projektavimo programos „Protégé" pranašumų yra tas, kad ontologijas galima ne tik vaizduoti vizualiai, bet net grafiniame pavidale jas kurti ar redaguoti. Vizualiai matant klasių hierarchiją yra daug lengviau projektuoti. Tam tikslui reikėjo papildomai įdiegti programą „Graphviz 2.16", kurios dalis techninių komponentų yra naudojami „Protégé" programoje ontologijų grafiniam vaizdavimui, bei „Protégé" programos priedą „OWLViz". Šį priedą parsisiunčiau iš „Protégé" papildomų priedų bibliotekos, esančios internete adresu [PR03], o programą „Graphviz 2.16" iš [GR01]. „Graphviz" programa įprastai yra naudojama diagramų ir grafikų piešimui. Įdiegus abu naujus komponentus juos dar reikėjo suderinti tarpusavyje.

2.6. Ontologijų realizavimas

Projektuojant naudojausi konkrečiais ir realiais laiškų pavyzdžiais. Išskirtas dažniausiai pasitaikančias ir laiškuose minimas problemas sujungiau į tris bendras sritis: Klaidos pranešimo (ErrorMessages), tuomet kai žaidėjas susiduria su kažkokiomis problemomis ir gauna apie jas pranešimą, Neaiškumo (Opacity), kuomet žaidėjas teiraujasi jį dominančių klausimų, ir liko dar viena savarankiška sritis – Laiškai be atsakymo (Answerless), tai laiškai, į kuriuos nereikia atsakinėti.

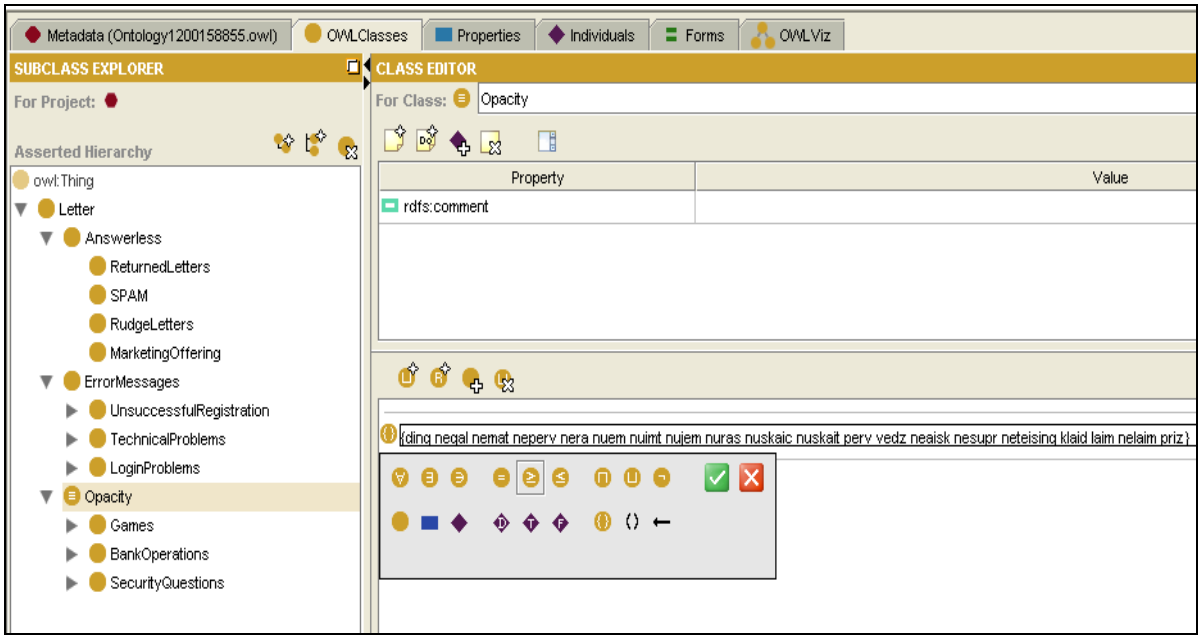
Dariau prielaidą, kad viename laiške užduodamas tik vienas klausimas (nagrinėjant laiškų aibę neradau nei vieno laiško, kuriame būtų rašoma apie kelias problemas). Taigi, vienai klasei priklausantys objektai negali priklausyti ir kitai klasei, t.y. jie yra disjunktai. Suprojektuota ontologijos grafinė išraiška yra pavaizduota 7 paveikslėlyje.



7 pav. Ontologijos

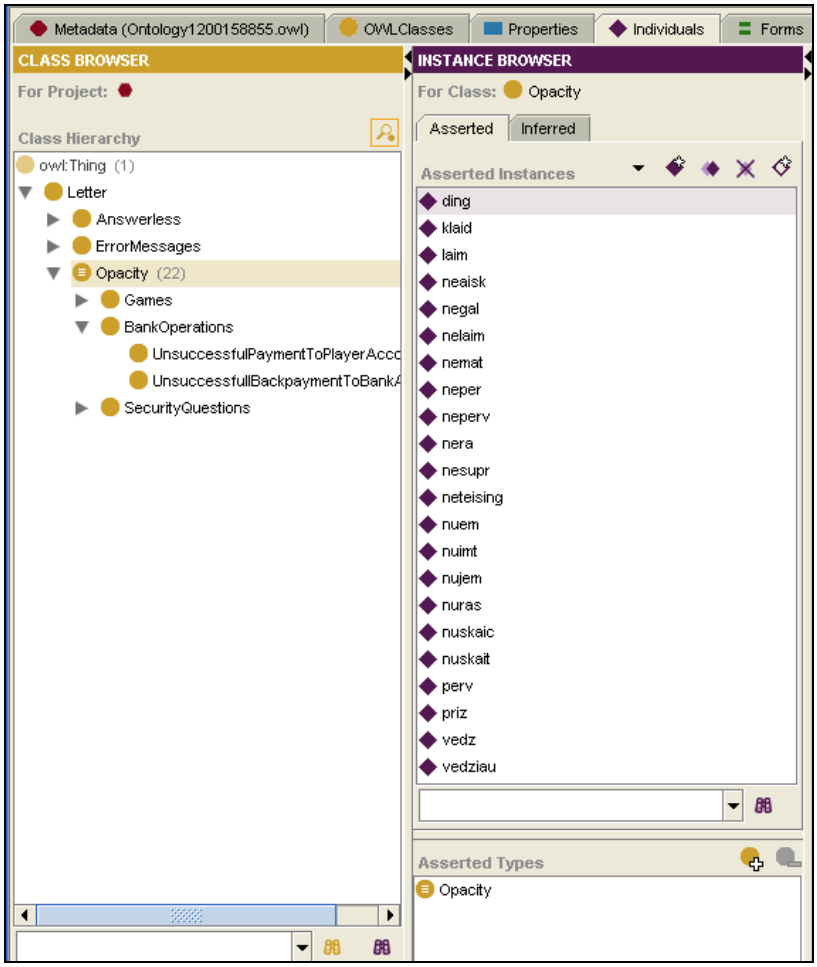
Paveikslėlyje pavaizduotos klasės ir tiesioginiai ryšiai tarp jų. Pavyzdžiui, klasė GameRules yra klasės Games poklasis. Savo ruožtu klasė Games yra klasės Opacity poklasis, kuri yra poklasis klasės Letter. Visų klasių superklasė yra owl:Thing, paveikslėlyje ji nepavaizduota vietos taupymo sumetimais.

Klasė Opacity apima visas klases, kuriose yra laiškai su užduotais klausimais, kurie kilo dėl neaiškumo ir nežinojimo. Ji turi tris poklasius: BankOperations, SecurityQuestions ir Games. Laiškas priklauso šiai klasei, jei jame yra bent vienas žodis iš šios aibės {ding; negal; nemat; neperv; nera; nuem; nuimt; nujem; nuras; nuskaic; nuskait; perv; vedz; neaisk; nesupr; neteising; klaid; laim; nelaim; priz}. Žodžiai imami be galūnių, kitu atveju būtų sudėtinga išvardinti visus žodžio linksnius ar formas. Kaip minėta anksčiau, visi laiškuose esantys lietuviški simboliai buvo pakeisti lotyniškomis raidėmis – daugelis internete rašo laiškus „šveplai“, tik maža dalis lietuviškai. Taigi, taip išvengėme daug darbo – rašyti tą patį žodį kelis kartus, su lietuviškais ir su lotyniškais simboliais.



8 pav. Klasės Opacity galimų reikšmių aibės apibrėžimas Protégé 3.3.1. programoje.

Norint apibrėžti tam tikras reikšmes klasėje, visų pirma jas reikia susikurti kaip tos klasės konkrečius pavyzdžius (*angl. Instances)*, tik po to jas galima naudoti kaip galimas reikšmes. Žemiau esančiame paveikslėlyje parodomas pavyzdys, kaip apibrėžiama tokia klasės pavyzdžių aibė.



9 pav. Klasės Opacity pavyzdžiai

Taigi, klasė Opacity pažymi neaiškumą. Jos poklasis BankOperations apima laiškus apie bankines operacijas, o konkretesnius atvejus apibrėžia UnsuccessfulPayment ir UnsuccessfulBackpayment – šios klasės yra paskutinė laiško atpažinimo grandis. UnsuccessfulPayment tipo laiškai rašomi tuomet, kai Žaidėjas per banko sistemą veda pinigus iš asmeninės banko sąskaitos į Žaidėjo sąskaitą ir jie dėl tam tikrų priežasčių ten nepatenka. Greičiausiai Žaidėjas netinkamai atlieka pavedimo operaciją – į Loterijų sistemą negrįžta per tam tikrą banko puslapyje esančią nuorodą „Grįžti pas pardavėją“. UnsuccessfulBackPayment – nekantrūs Žaidėjai rašo laiškus nesulaukdami, kada jiems iš Žaidėjo sąskaitos bus pervesti pinigai atgal į asmeninę banko sąskaitą. Tokius atgalinius pavedimus atlieka buhalterija rankiniu būdu kiekvieną ateinančią darbo dieną. Šiose klasėse yra žodžiai, kurie susiję su bankais.

Pavyzdžiui, {saskait; bank; zaid; pinig; lit; imok; pavedim; ved}. Priskiriant laiškus vienai iš šių klasių yra didžiausia klaidos tikimybė, nes žodžių aibės yra beveik identiškos. Pavyzdžiui, UnsuccessfulBackpayment klasėje nėra žodžio zaid, bet yra laim, nes laiškuose dažniausiai neužsimenama apie Žaidėjo sąskaitą, dažniausiai klausiama „Kada perves mano laimėjimą?“.

Klasė SecurityQuestions turi tris poklasius: PersonalDataSecurity, SSNUse ir Legalness. PersonalDataSecurity klasei priklauso laiškai, kuriuose abejojama Loterijų sistemos saugumu, nes joje saugomi asmens duomenys, klausiama kam to reikia, prašoma pašalinti šiuos duomenis ir panašiai. Šios klasės žodžių aibė yra tokia {asmen; konfid; duomen; vard; pavard; kod}. SSNUSE išskyriau kaip atskirą klasę, nors jos laiškai galėtų priklausyti ir PersonalDataSecurity klasei. Tačiau tai daugiau specifiniai klausimai apie asmens kodo saugojimą. Pavyzdžiui, klausimas: kodėl registruojantis iš banko į Loterijų sistemą persiunčiamas asmens kodas? Šios dvi klasės taip pat problematiškos, nes didelė tikimybė sumaišyti ir priskirti laišką ne tai, kuriai reikia, ypač jei laiškas bus parašytas nekorektiškai. Į Legalness klasę turėtų patekti laiškai, kuriuose teiraujamasi dėl loterijos legalumo. Į žodžių aibę įtraukti pasitaikančių žodžių šaknys, tokių kaip legalumas, licencija ir pan. Tokių laiškų būna nedaug, taipogi žodžiai specifiniai, taigi šansas apsirikti ir priskirti šios klasės laišką kitur yra nedidelis.

Kitas klasės Opacity poklasis – Games. Games turi tris poklasius GameRules, GameWiningTable ir GameRestrictions. Games klasės žodžiai yra visų žaidimų pavadinimai, netgi su galimomis jų klaidingomis interpretacijomis: Kiniškas Keno, Internetinis LOTO, Draugų LOTO, LOTO Plius, Triušių lenktynės, Krepšinis, Auksinis septynetas, Baltijos lobiai, Dvigubai arba Nieko. Žinoma, imamos tik šių žodžių šaknys be lietuviškų simbolių. GameRules klasei priklauso laiškai, kuriuose teiraujamasi konkrečiai apie žaidimų taisykles. Viena iš pagrindinių klasės žodžio šaknų – taisykl. GamingWinningTables – konkretūs klausimai apie žaidimų laimėjimų lenteles, laimėjimų dydžius ir statomas sumas. GameRestrictions – laiškai su klausimais apie žaidimų apribojimus. Žaidėjai gali užsidėti žaidimų ar piniginius apribojimus, t.y. nustatyti, kad tam tikrą žaidimą

sužais ne daugiau x kartų per tam tikrą pasirinktą laiko tarpą, arba nustatyti pinigų sumą, kurią gali išleisti loterijos bilietams per tam tikrą pasirinktą laiko tarpą.

Kitas klasės Letter poklasis – klasė Answerness. Tai laiškai, į kuriuos klientų aptarnavimo centras neprivalo siųsti atsakymo. Ši klasė turi 4 poklasius: ReturnedLetters, MarketingOffering, RudgeLetters, SPAM. ReturnedLetters – tai laiškai, kurie nepasiekė adresato. Šią klasę atpažinti itin lengva, nes kiekvienas toks laiškas prasideda standartiniu tekstu. Į MarketingOffering klasę patenkančius laiškus reikia persiųsti atsakingiems asmenims – tai būna reklamų pasiūlymai. Standartiniai žodžiai šiuose laiškuose yra pasiūlymas, reklama, galimybės, siūlyti, kaina, tinklalapis ir pan. Pati nemaloniausia laiškų klasė – RudgeLetters. Pasipiktinusių žmonių pikti laiškai. Dažniausi žodžiai: vagys, sukčiai, apgavikai, apgaulė, nesąmonė, banditai, keiksmažodžiai ir t.t. Tokius laiškus ontologijose atpažinti taip pat labai lengva, nes daugiau jokiose kitose laiškų klasėse tokių žodžių nebūna. Net pačiai buvo keista, kai nagrinėdama laiškus radau nedaug nepageidaujamų elektroninių laiškų – SPAM‘o. Tai ir yra paskutinis klasės Answerness poklasis. SPAM klasės žodžių aibė yra tokia {free; congratul; opport; win; won; diet; smoking; viagra; weight; degree; dvd; gambling; horoscope; ink; http; joke; tv; obligation; cash; cheap; credit; deal; great; money; debt; discount; income; insurance; loan; price; profit; promo; rate; shop; enlarge; girl; she; click}.

Paskutinis mano išskirtas klasės Letter poklasis yra ErrorMessages. Tai žaidėjų parašyti laiškai, po to kai jiems kažkas nepasiseka atlikti Loterijų sistemoje ir jie gauna kažkokį klaidos pranešimą. Klasė skaidoma į smulkesnes: TechnicalPoblems, UnsuccessfulRegistration ir LoginProblems. ErrorMessages klasei priklauso laiškai, kuriuose yra žodžių rašo, išmeta, pranešimas, langas ir pan. (aišku, tik šių žodžių šaknys, be lietuviškų simbolių). TechnicalProblems poklasis skaidomas į smulkesnius FlashProblems ir WrongSession poklasius. Į klasę FlashProblems patenka žaidėjų, kurie neturi įsidiegę kompiuteryje 7-tos ar dar naujesnės Flash programos versijos, laiškai. Tokį laišką atpažinti gana nesudėtinga, nes beveik jų visų tekstai yra vienodi. WrongSession klasės laiškus

atpažinti turbūt lengviausia. Beveik visi rašo laišką su tokiu pat tekstu „Ką daryti, rašo sesija baigėsi“. Taip pat daugiau nei vienoje klasėje nėra žodžio „sesija“.

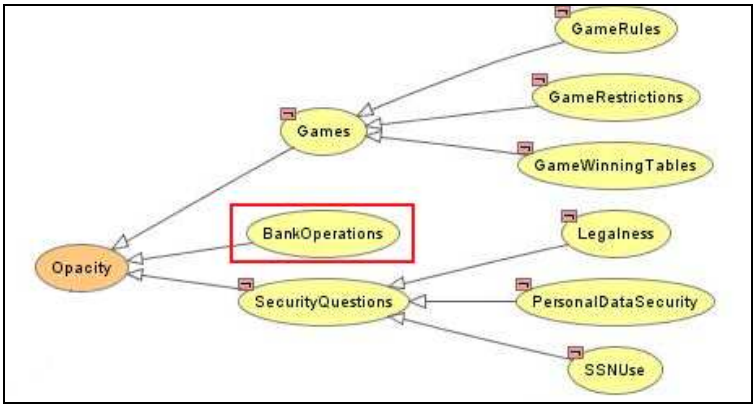
Antras ErrorMessages klasės poklasis yra UnsuccessfulRegistration (jo žodžių aibėje tėra vienas žodis - „registracija“), jam priklauso tų žaidėjų parašyti laiškai, kuriems nepavyko užsiregistruoti. Nesėkminga registracija gali būti trim atvejais, todėl šis poklasis išskirtas į 3 išsamesnius poklasius: WrongBanAccountNumber, ExistingPlayer ir ExistingUsername. WrongBankAccountNumber klasės laiškas parašomas tuomet, kai registruojantis neteisingai rašoma banko sąskaita: praleidžiami skaičiai ar raidės, dedami tarpai. ExistingPlayer klasei priklauso laiškai, kuriuos žaidėjas parašo tuomet, kai gauna pranešimą „Toks žaidėjas jau yra užsiregistravęs“. Taip atsitinka, kai registruotis bando jau užsiregistravęs žaidėjas. Deja, bet tokių būna... ExistingUsername – registracijos metu žaidėjas nori pasirinkti vartotojo vardą, kuris yra jau kažkieno pasirinktas ir užfiksuotas Loterijų sistemoje. Tokiu atveju žaidėjas gauna pranešimą „Toks vartotojo vardas jau užregistruotas“. Labai keista, bet nesuprantančių šio pranešimo yra ir ne vienas...

Paskutinis klasės ErrorMessage poklasis – LoginProblems. Šiai klasei priklauso užklausimai po nesėkmingo žaidėjo bandymo prisijungti prie Loterijų sistemos. Kodėl jam tai nepavyko galimos trys priežastys, jos yra išskirtos į tris atskiras klases. Pirmoji – FrozedPlayerAccount. Tokiu atveju žaidėjas gauna pranešimą su panašiu tekstu „Jūsų Žaidėjo sąskaita sustabdyta“. Dažniausiai tai ir parašo laiške, taip pat dažna frazė tokiuose laiškuose „kodėl negaliu prisijungti?“. Žaidėjo sąskaita užšaldoma tuomet, kai žaidėjas tris kartus iš eilės neteisingai suveda prisijungimo duomenis. Tokia tvarka taikoma saugumo sumetimais. Antra ir trečia klasės ForgetUsername ir ForgetPassword atitinkamai užmiršusiųjų vartotojo vardą ir slaptažodį klausimai. Beveik visi laiškai standartiniai: „Užmiršau/nebeatsimenu/pamiršau vartotojo vardą/slaptažodį“. Lengvai atpažįstamos klasės.

2.7. Sukurtų ontologijų analizė

Suprojektuotose ontologijose nėra tokios klasės, kuriai būtų galima priskirti jau kartą atsakytą laišką. Techniškai kol kas neįmanoma tokį laišką atskirti nuo kitų laiškų. Pavyzdžiui, pirmasis laiškas buvo klausimas apie nepavykusį pavedimą. Pagal sukurtas ontologijas jam būtų išsiųstas tinkamas atsakymas, kuris visuomet siunčiamas atsakyti laiškams iš klasės UnsuccessfulPayment. Tačiau jei žaidėjui vis tiek neaišku ir jis dar kartą paklaus kažką patikslinti, jo laiškas vėl bus priskirtas prie UnsuccessfulPayment laiškų klasės ir antrą kartą rašydamas jis gaus lygiai tokį pat atsakymą, kaip ir į pirmąjį laišką. Šiai problemai spręsti siūlyčiau ne keisti ontologijas, bet patobulinti laiškų atsakymo principą – kiekvienam atėjusiam naujam laiškui suteikti identifikacinį numerį ir atsakant į laišką jį įterpti į laiško temą. Tuomet kiekvieno atėjusio laiško tema būtų patikrinama, ar yra jame identifikacinis kodas. Jei yra, tuomet jis neturėtų būti atsakomas automatiškai. Šiuo atveju į jį būtinai turėtų atsakyti žmogus. Jei laiške identifikacinio numerio nėra – tuomet galima leisti jį atsakyti automatiškai pagal ontologijas.

Išskyriau ir problematines ontologijų sritis. Tai sritys, kuriose didžiausia klaidos tikimybė, t.y. didžiausia tikimybė apsirikti ir priskirti laišką ne tai klasei. Pavyzdžiui, labai sunku atskirti klases UnsuccessfulPayment ir UnssuccessfulBackpayment. Pirmos klasės laiškas „kodėl nepervedate pinigų į banko sąskaitą?“, antros „Kodėl nepervedate pinigų, nors iš banko sąskaitos nurašyti?“. Atrodo, kad klausimai tokie pat, bet iš tikrųjų jie skirtingi ir į juos tikrai negali būti to paties ar neteisingo atsakymo. Be to, tai dažniausiai pasitaikantys ir, galima sakyti, svarbiausi klausimai. Dėl šios priežasties BankOperations poklasius panaikinau. Šios klasės laiškai negali būti atsakinėjami automatiškai. Ontologijų medis su panaikintais BankOperations poklasiais pavaizduotas žemiau esančiame paveikslėlyje.

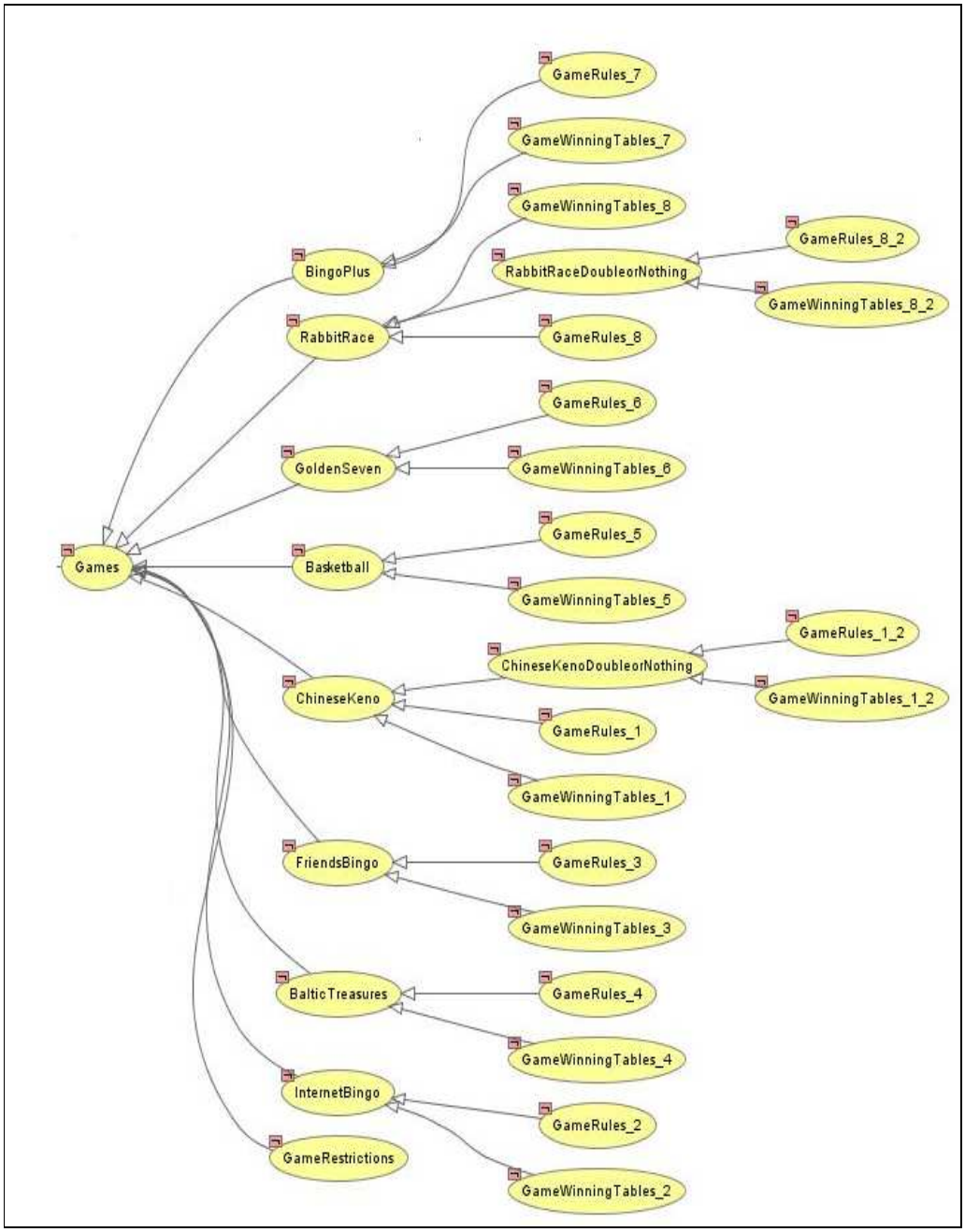


10 pav. Opacity klasė su pakoreguotu BankOperations poklasiu

Taip pat klaidos grėsmė yra klasėse BankOperations ir WrongBankAccountNumber, kadangi šių klasių laiškuose yra kalbama apie bankų sąskaitas, nors pirmoji klasė tiesiogiai susijusi su tuo, o antroji tėra dažnai pasitaikanti klaida registruojantis, t.y. registracijos metu neteisingai vedant banko sąskaitos numerį. Tačiau tai mažiau pavojinga sritis nei BankOperations poklasiai, nes WrongBankAccountNumber klasės laiškai beveik visada turi žodį „registracija“.

Panaši situacija yra ir su klasėmis ForgetUsername ir ExistingUsername – laiškuose kalbama apie vartotojo vardą, tačiau pirmoje laiškų klasėje yra užsimenama, kad negali prisijungti, nes pamiršo ar nebeatsimena, o antroje yra žodis „registruotis“ ar jo formos.

Klasė Games yra per daug bendra ir abstrakti, todėl ją reikia išskirti į smulkesnius poklasius. Šioje klasėje palikau neliestą GameRestrictions poklasį ir šalia jo pridėjau gimininges klases pagal žaidimų pavadinimus. Kiekviena žaidimo klasė turi du poklasius: žaidimo taisykles ir laimėjimo lentelę. Tie žaidimai, kurie turi papildomai dar antrą dalį, turi ir po trečią poklasį, kuris savo ruožtu turi savarankiškai laimėjimų lentelę ir taisykles. Tokie žaidimai yra „Triušių lenktynės“ (RabbitRace klasė) ir „Kiniškas Keno“ (ChineseKeno klasė). Toks suskirstymas daug išsamiau ir aiškiau apibūdina Games klasę. Games klasės pakeitimus galite pamatyti žemiau esančiame paveikslėlyje.



11 pav. Games klasė su pakeistais poklasiais.

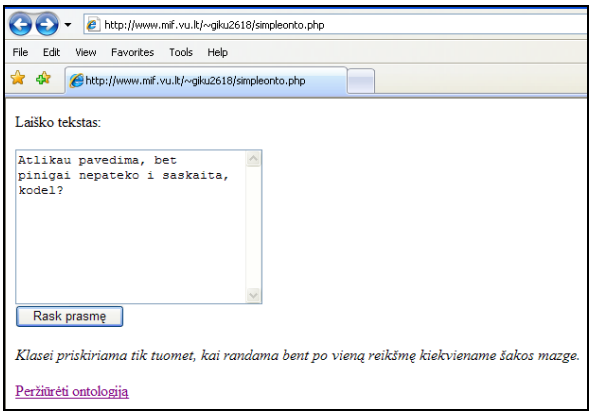
Galutinį suprojektuotą ir pakoreguotą ontologijų vaizdą galite pamatyti 1 priede. Šiose ontologijose pašalinti BankOperations poklasiai ir Games klasė papildyta naujais ir smulkesniais poklasiais.

3. Teksto analizės ir klasifikavimo pagal ontologijas būdai

3.1. Teksto klasifikavimas pagal klasių hierarchiją

Ontologijas galima laikyti tam tikros informacijos srities struktūra. Analizuojant ir klasifikuojant tekstą pagal prasmę ne mažiau svarbu yra tai, kaip interpretuojamos ontologijos. Pats paprasčiausias būdas interpretuoti ontologijas yra pagal jos klasių hierarchiją. Ontologijose radus žodžių iš teksto leistis šakomis žemyn, kol pasiekiama žemiausiai esanti klasė, kuriai jis ir priskiriamas.

Siekiant ištirti tokio algoritmo efektyvumą mano tiriamai informacijos sričiai parašiau programą, kuri būtent tokiu principu analizuoja tekstą. Programa parašyta naudojantis PHP, HTML ir JavaScript programavimo kalbomis, o ontologijų struktūra, pagal kurią klasifikuojamas laiško tekstas, yra išsaugota XML formato faile. Programos sąsajos su vartotoju fragmentą galite pamatyti žemiau esančiame 12 paveikslėlyje:



12 pav. Programos, klasifikuojančios laišką nuosekliai skaitant ontologijų medį, sąsajos su vartotoju langas

Programos veikimo principas: teksto lauke esantis tekstas nuskaitomas, tuomet po vieną žodį tikrinama, ar jis yra reikšminis, t.y., ar toks žodis yra įtrauktas į ontologijas. Visų pirma tikrinamos viršutinės ontologijų klasės, vėliau jų poklasiai ir t.t. Laiškas priskiriamas kokiai nors klasei tik tuomet, jei buvo rasta žodžių visuose virš jo esančiuose viršklasiuose, kitu atveju – laiškas lieka neklasifikuotas. Ontologijų, esančių XML faile, reikšmių nuskaitymui naudojau XML DOM – žymėjimo kalbos XML ir HTML dokumento objekto modelį. XML DOM yra nepriklausomas nei nuo platformos, nei nuo programavimo kalbos, ne tik suteikia prieigą prie dokumento, bet ir leidžia keisti jo struktūrą – pakeisti, pridėti ar pašalinti jo elementus.

Deja, atliekant bandymus su minėtomis ontologijų nuskaitymo taisyklėmis pasiekti laiškų klasifikavimo rezultatai nėra geri. Patikrinus su 100 laiškų, algoritmas nepriskyrė jokios reikšmės net 47 laiškams, o 12 priskyrė klaidingą reikšmę. Taigi, galime daryti išvadas, kad laiškų klasifikavimas tokiu būdu yra neefektyvus, tik 41 procentas.

3.2. Žodžių dažnių ir klasių hierarchijos koeficientų algoritmas

Ištyrus, kad laiško teksto klasifikavimui vien ontologijų klasių hierarchijos ir ryšių nepakanka, buvo ieškota kitų sprendimo kelių klasifikavimo paklaidai sumažinti. Visų pirma buvo atsisakyta minties nuosekliai pagal hierarchiją leistis ontologijų medžio šakomis nuo viršutinės klasės iki pat apatinių ieškant reikšminių žodžių – viršutinėms klasėms priklausantys žodžiai yra bendri, todėl dažnai jų laiškuose gali nebūti. Žemesnėse klasėse esantys žodžiai yra jau daugiau specifiniai ir pasitaikantys tos rūšies laiškuose. Remiantis šiuo principu, kiekvienam ontologijų medžio klasių lygiui priskirtas koeficientas pagal jų svarbą – kuo klasė yra žemiau, tuo joje esantys žodžiai daugiau apibūdina tos klasės laiškus ir turi didesnę reikšmę, taigi jos koeficientas turi būti didesnis nei aukščiau esančių klasių. Kuomet atsisakoma minties nuskaityti ontologijas nuosekliai,

atsiranda kita problema – tie patys žodžiai, pasikartojantys ne vienoje klasėje. Būtent dėl to klaidų tikimybė klasifikuojant laišką gali labai išaugti.

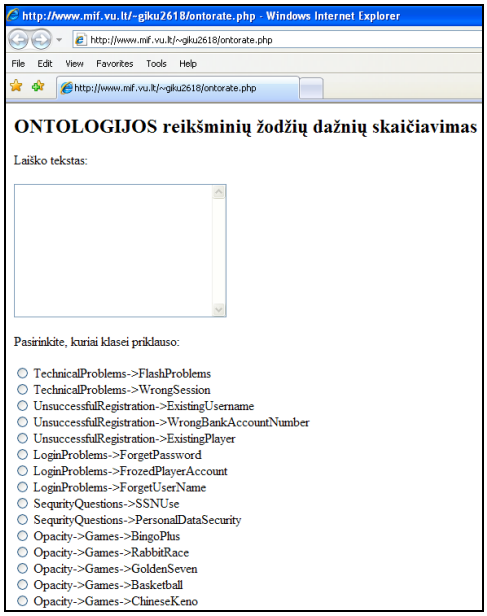
3.2.1. Žodžių dažnių apskaičiavimas

Remiantis padarytomis prielaidomis, prie kiekvieno ontologijose esančio žodžio buvo pridėtas dažnio koeficientas. Kiekvienam žodžiui jis turėtų būti skirtingas, net jei tas žodis kartojasi ir kitose klasėse. Pavyzdžiui, laiškuose apie bankinius pavedimus ir nesėkmingą registraciją neteisingai įvedus banko sąskaitos numerį, kartojasi žodžiai „bankas“ ir „sąskaita“. Tačiau laiškai apie pavedimus yra daug dažnesni, nei apie minėtą nesėkmingą registraciją. Taigi, šie du žodžiai bankinių pavedimų laiškų klasėje turi turėti didesnius svorius.

XML failas, kuriame saugomi ontologijų struktūra ir žodžiai, buvo papildytas žodžių dažnių duomenimis. Prie kiekvieno reikšminio žodžio buvo pridėtas atributas „rate“ su pradine reikšme. Pradinė kiekvieno dažnio reikšmė yra 1. Tam, kad dažnių reikšmės atitiktų realios informacijos srities laiškus, jos turi būti apskaičiuotos remiantis konkrečiais ir tikrais laiškų pavyzdžiais. Atlikti šiuos skaičiavimus rankiniu būdu užimtų nemažai laiko, todėl parašiau programą šiam procesui automatizuoti. Programa parašyta PHP ir JavaScript programavimo kalbomis. Ontologijų reikšminių žodžių ir jų dažnių nuskaitymui iš XML failo naudojama XML DOM – žymėjimo kalbos XML ir HTML dokumento objekto modelis.

Įvedus laiško tekstą ir iš sąrašo pasirinkus kuriai klasei jis priklauso, programa išrenka tai klasei priklausančius reikšminius žodžius. Atitinkamai pakoreguojamas ir XML failas su ontologijomis – konkrečios pasirinktos klasės rastų žodžių dažniai yra padidinami vienetu. Programos sąsajos su vartotoju fragmentas pavaizduotas 13 paveikslėlyje.

.



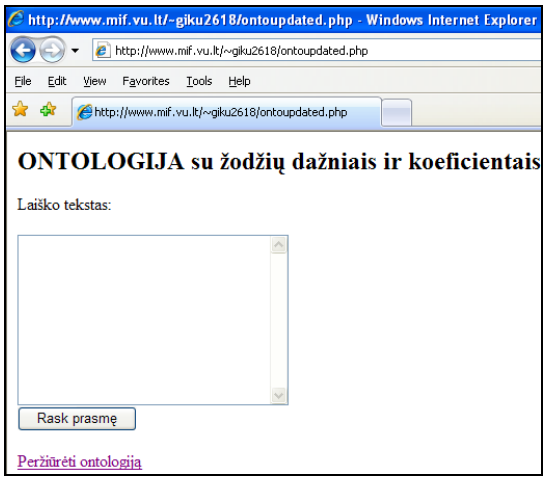
13 pav. Ontologijų reikšminių žodžių dažnių skaičiavimo programa

3.2.2. Algoritmo realizacija

Tam, kad laiškai būtų klasifikuojami efektyviau, buvo sugalvotas naujas algoritmas skaityti ir interpretuoti ontologijas, atsižvelgiant į reikšminių žodžių dažnius ir klasių hierarchijos koeficientus. Algoritmo esmė šįkart jau nebe kaip surasti bent po vieną reikšminį žodį kiekvienai klasei nuosekliai einant žemyn ontologijų šaka, bet atsižvelgti į rastų žodžių dažnių ir hierarchijos koeficientų sumas atskirai kiekvienai klasei. Algoritmas buvo įgyvendintas programoje. Programos veikimo principas: iš laiško teksto išrenkami reikšminiai žodžiai, atitinkamai iš ontologijų surandamos klasės, kuriose jie yra, taip pat ir tų žodžių dažniai jose. Tuomet skaičiuojama reikšmė pagal klases – susumuojami žodžių, esančių toje pat klasėje, dažniai ir padauginama iš klasės hierarchijos koeficiento. Kaip buvo minėta anksčiau, laiškas klasifikuojamas tuomet, jei pasiekia pačią paskutinę klasę ontologijų medyje. Šiuo atveju jei apatinė klasė turi ryšį su virš jos esančia klase (buvo rasta reikšminių žodžių ir klasėje, ir jos viršklasyje), tai viršklasio apskaičiuota reikšmė pridedama prie klasės reikšmės.

Iš visų apatinių ontologijų klasių, kurios ir klasifikuoja laišką, išrenkama ta, kuri turi didžiausią apskaičiuotą reikšmę.

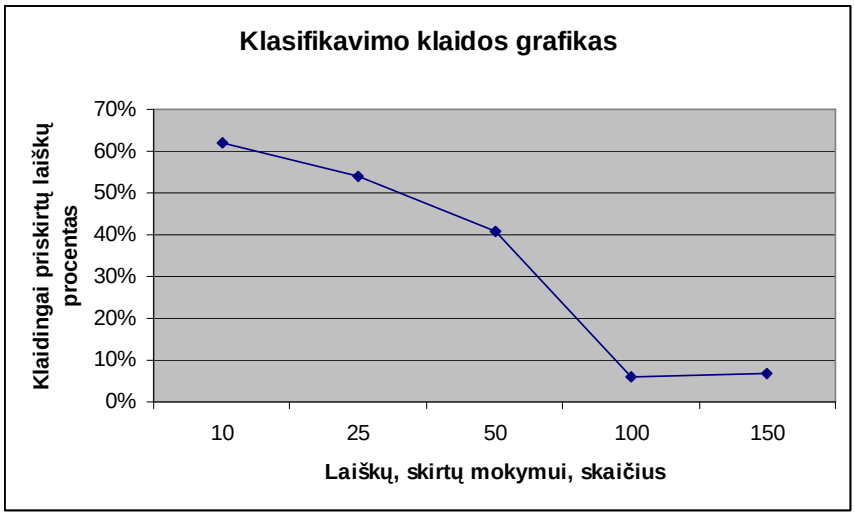
Programa įgyvendinta naudojant PHP, JavaScript ir HTML programavimo kalbas. Navigavimui ontologijų medžiu taip pat naudojau XML DOM dokumento objekto modelį. Jos sąsajos su vartotoju fragmentas pavaizduotas 14 paveikslėlyje.



14 pav. Programos, veikiančios žodžių dažnių
ir hierarchijos koeficientų algoritmo pagrindu, fragmentas

3.2.3. Sukurto algoritmo efektyvumo tyrimas ir analizė

Visų pirma žodžių dažnių ir klasių hierarchijos algoritmo tikslumas priklauso nuo ontologijų apmokymo, t.y. kiek laiškų yra naudojama klasių reikšminių žodžių dažniams apskaičiuoti. Taip pat labai svarbu, kad laiškai, naudojami mokymui, atitiktų realiai gaunamus laiškus, t.y., kad nebūtų naudojami tik vienos ar kelių klasių laiškai. Paėmus labai mažą laiškų aibę mokymui, padidinti kai kurių klasių dažniai iškreipia klasifikavimo rezultatus ir klasifikavimo klaida būna gan didelė. Kaip klaidos tikimybė priklauso nuo apmokymui skirtų laiškų skaičiaus pavaizduota 15 paveikslėlyje.



15 pav. Klasifikavimo klaidos priklausomybės nuo apmokytų laiškų skaičiaus grafikas

Atsitiktinai parinkus nepakankamą laiškų aibę apmokymui gali pasitaikyti viena ar kelios per daug dominuojančios laiškų klasės. Tokiu atveju situaciją galima taisyti sumažinant klasės hierarchijos koeficientą. Tuomet padauginus žodžių dažnių sumas, bendra klasės reikšmė taip drastiškai nepadidės ir neišsiskirs iš kitų.

Šio algoritmo klasifikavimo rezultatai yra žymiai geresni, žinoma jei ontologijų žodžių dažniai yra tinkamai apskaičiuoti. Šiuo atveju neklasifikuotų laiškų telieka apie 2 procentai, kai ankstesniu atveju turėjome daugiau nei pusę. Klasifikavimo paklaida – apie 5 procentai, tai dukart mažiau nei gauta skaičiuojant ankstesniuoju algoritmu nuosekliai apeinant šakas.

Rezultatai ir išvados

Šio darbo metu buvo ištirta UAB „Olifėja” informacijos sritis, susijusi su elektronine loterija www.eloterija.lt. Buvo apdoroti ir paruošti duomenys ontologijoms kurti – visi adresu info@eloterija.lt nuo loterijos starto atėję laiškai. Buvo pašalinta nereikalinga ir nereikšminė informacija, tam, kad nebūtų

pažeistas Asmens duomenų apsaugos įstatymas, į nereikšmines simbolių sekas pakeisti asmens ir kiti konfidencialūs duomenys, taip pat visi lietuviški simboliai pakeisti atitinkamais lotyniškais.

Pagal apdorotus duomenis buvo išskirtos probleminės sritys ir suprojektuotos pirminės ontologijos, jos klasių hierarchija ir ryšiai, apibrėžti kiekvienos klasės galimi pavyzdžiai ir žodžių aibės, pagal kurias elektroninis laiškas priskiriamas vienai ar kitai klasei. Pavyzdžiuose ir žodžių aibėse naudojau žodžius be galūnių, taip vienu žodžiu galima apimti visas jo formas ar linksnius.

Toliau atliekant tyrimą buvo nustatytos klasės, kuriose yra didžiausia tikimybė apsirikti, t.y. klasei klaidingai priskirti laišką. Šių klasių sritys buvo papildomai smulkiau panagrinėtos ir galiausiai perprojektuotos.

Buvo sugalvoti sprendimai, kaip patobulinti patį laiškų atsakymo mechanizmą – kiekvienam naujai atėjusiam laiškui galima suteikti unikalų identifikacinį numerį ir įrašyti jį laiško temoje. Jei žaidėjui vis tiek kyla neaiškumų, jis dar kartą atsako į jam atsiųstą laišką. Tas laiškas galėtų būti atpažintas kaip jau kartą atsakytas radus identifikacinį numerį, esantį laiško temoje. Tai reiškia, kad reikia giliau panagrinėti jo problemą ir atsakyti laišką rankiniu būdu. Kitu atveju jam būtų antrąkart išsiųstas standartinis atsakymas.

Kita problema – laiškai, kuriuose yra gramatinių klaidų. Deja, visų žodžių su galimais klaidų variantais įtraukti į ontologijas neįmanoma. Paprasčiausiai galima tikėtis, kad laiške bus dar kitų teisingų reikšminių žodžių, pagal kuriuos laišką bus galima priskirti kokiai nors konkrečiai klasei.

Nemažiau svarbus aspektas yra ir klasifikuojamo teksto interpretavimo pagal ontologijas taisyklės. Nuo to, kokiu būdu yra analizuojamas tekstas, labai priklauso klasifikavimo paklaida. Deja, klasifikuojant tekstą pagal ontologijų klasių hierarchiją ir ryšius nebuvo pasiekta gerų rezultatų, todėl buvo ieškoma kitų ontologijų interpretavimo būdų.

Kad ontologijos kuo labiau atitiktų informacijos sritį, jos reikšminiai žodžiai buvo papildyti ir jų pasitaikymo realiuose laiškuose dažniais. Dažniai gaunami apmokius ontologijas realiais laiškais, iš anksto pažymint kuriai klasei jie priklauso. Klasifikavimo paklaidai didelę reikšmę turi apmokymui naudojamų

laiškų kiekis ir kokybė, t.y. visi laiškai neturi priklausyti tik vienai ar tik kelioms klasėms. Taip pat buvo atkreiptas dėmesys į klasių hierarchiją, kuo bendresni ir mažiau apibūdinantys žodžiai yra klasėje, tuo mažesnę svarbą jie turi. Taigi, buvo sukurtas algoritmas kaip ieškoti teksto prasmės atsižvelgiant ir į reikšminių žodžių dažnius, ir į klasių hierarchiją. Tinkamai suskaičiavus ir parinkus parametrus šiuo būdu sukurtas elektroninių laiškų analizatorius klasifikuoja tekstą kelis kartus geriau ir tiksliau.

Šio magistro darbo metu įrodyta, kad ontologijos – galingas ir veiksmingas įrankis, padedantis išreikšti informaciją pagal jos prasmę. Tai yra dar besiplečianti ir besivystanti informacinio pasaulio dalis, pasižyminti ypatingu lankstumu ir lengvu pritaikymu įvairiose srityse. Buvo ištirta, kad galima pasiekti itin gerų rezultatų, tekstų analizei, palyginimui ir klasifikavimui naudojant ontologijas. Sukurtos programinės priemonės gali būti panaudojamos ir kitos informacijos srities tekstams klasifikuoti, reikalingas tik pagal specialius reikalavimus užrašytas ontologijų XML formato failas, bei keli minimalūs programos pakeitimo darbai.

Šaltinių sąrašas

[BV01] Sean Bechhofer, Raphael Volz. Ontologijų vertinimo įrankis internete URL: http://www.mygrid.org.uk/OWL/Validator

[ČČ03] Donatas Čiukšys, Albertas Čaplinskas, straipsnis „Ontologijų naudojimo ypatumai kuriant moderniąsias informacines sistemas“, 2003, Vilnius.

[Den03] Michael Denny, 2003. Ontology Building: A Survey of Editing Tools. URL: http://www.xml.com/pub/a/2002/11/06/ontologies.html

[GR01] Programa „Graphviz“ URL: http://www.research.att.com/sw/tools/graphviz/

[Gua98] N. Guarino. Formal Ontology and Information Systems, Italy, 1998.

[HKR+04] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe, A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools, 2004 URL: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf

[Yam01] Takahira Yamaguchi, straipsnis “Acquiring conceptual relationships from Domain-Specific texts”, Shizuoka, Japan

[Mas01] Prezentacija „Ontologijos ir semantinis pasaulinis tinklas”, Saulius Maskeliūnas URL: http://eta.ktl.mii.lt/~mask/varia/Ontologijos%20ir%20semantinis%20pasaulinis%20tinklas.ppt

[Mas02] Saulius Maskeliūnas, „Modernių informacinių sistemų ontologijos ir paslaugų reikalavimų formulavimas“, “Informacinės technologijos 2006”, Kaunas

[NM01] Natalya F. Noy, Deborah L. McGuiness, Stanford. Ontology Development 101: A guide to creating your first ontology.
URL: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

[OB01] Ontolingua ontologijų biblioteka pasauliniame žiniatinklyje.
URL: http://www.ksl.stanford.edu/software/ontolingua/

[OB02] DAML Ontologijų biblioteka pasauliniame žiniatinklyje.
URL: http://www.daml.org/ontologies/

[PR01] Programa „Protégé“
URL: http://protege.stanford.edu/download/download.html

[PR03] Programos „Protégé“ papildomų komponentų biblioteka internete
URL: http://www.co-ode.org/downloads/plugins-3.x.php

[Wik01] Internetinė enciklopedija
URL: http://lt.wikipedia.org/wiki/Ontologija_(informatika)

1 priedas

Galutinis ontologijos projektas.