

## DALYKINĖS SRITIES ONTOLOGIJA GRĮSTA INTELEKTUALIOJI GEDIMŲ DIAGNOSTIKOS SISTEMA

Lina Tankelevičienė, Rimantas Sakalauskas

Šiaulių universitetas, Matematikos ir informatikos fakultetas

### Įvadas

*Semantinis pasaulinis tinklas* – tai naujos kartos pasaulinis interneto tinklas, kuriame informacija suprantama ne tik vartotojams-žmonėms, bet ir programiniams agentams ar kitokiems programiniams moduliams, kurie, kaip ir vartotojai, gali prisidėti prie informacijos atrinkimo, platinimo, kūrimo. Tokių galimybių sudarymo pagrindas – informacija – turi būti analizuojama ne tik sintaksiniam, bet ir semantiniame lygmenyje, o analizės rezultatai turi būti pateikiami formalia arba pusiau formalia kalba. Pagrindiniai semantinio pasaulinio tinklo komponentai yra resursai ir juos aprašantys metaduomenys, ontologijos, programiniai agentai bei pasaulinio tinklo paslaugos.

Ontologijos dažnai apibrėžiamos kaip bazinis semantinio pasaulinio tinklo komponentas. Kalbant apie dalykinės srities ontologijas (angl. *domain ontology*), nagrinėjama tik tam tikra realaus pasaulio dalis, pavyzdžiui, mechanika, mokslinės publikacijos, universiteto infrastruktūra ar pan. Pagrindiniai dalykinės srities ontologijoms keliami reikalavimai yra šie [1]: 1) pateikti bendrą supratimą apie tam tikrą dalykinę sritį, praskaidrinti srities žinių struktūrą; 2) leisti komunikuoti žmonėms ir programinei įrangai, sumažinant terminijos naudojimo nevienareikšmiškumą; 3) palengvinti žinių platinimą ir pakartotinį naudojimą. Dėl šių tikslų ontologijos, kaip inžinerinis produktas, vis plačiau naudojamos įvairiose srityse, taip pat ir e. mokymos(si) srityje. Įvairios informacinės komunikacinės technologijos leidžia paįvairinti mokymą(si), padaryti jo valdymą efektyvesnį. Keičiasi ne tik technologijos, bet ir edukacinės nuostatos į mokymosi organizavimo principus ir būdus. Šiuo metu, akcentuojant mokymosi visą gyvenimą svarbą, dažnai minimas probleminis, patirtinis mokymas(-is). Patirtinis mokymas(-is) yra susijęs tiek su asmeniniu, tiek su profesiniu tobulėjimu [2]. Patirtis įgyjama per praktikavimąsi ir eksperimentavimą. Ontologija grindžiama patariančioji sistema leistų įgyti daugiau patyrimo, kai nesama realių eksperimentų. Tai ypač svarbu besikeičiančioje visuomenėje, kai, norėdami prisitaikyti prie naujų sąlygų, turime gebėti greitai suprasti ir išspręsti kilusias problemas. Kompiuterinės sistemos negali ir turbūt negalės atlikti visų mokytis padedančio asmens funkcijų, tačiau ontologijų

kūrimas ir jų pagrindu išreikštas reikalingų žinių aprašymas sudaro prielaidas realizuoti efektyvesnę sąveiką tarp studijuojančiojo ir elektroninės mokyimo(si) medžiagos, taip perduodant dalį dėstytojo funkcijų programiniams moduliams.

Darbe pateikta išsami dalykinės srities ontologijų panaudojimo studijų procese galimybių studija [3]. Darbe analizuojamas samprotavimo mechanizmų realizavimas, norint pasiekti personalizaciją, adaptyvų informacijos išgavimą [4]. Šiame darbe nagrinėjama situacija, kai ontologija ir informacinių resursų kiekis yra didelės apimties. Darbe [5] akcentuojamas patirtinio mokymo(si) realizavimas, taikant dalykinės srities ontologijas medicinos srityje, poreikis. Ontologija, kaip vienareikšmė srities terminų ir galiojančių taisyklių specifikacija, leistų išvengti subjektyvumo, kuris atsiranda mažai struktūrizuotuose klinikiniuose tyrimuose. Darbe pristatyta sukurta eksperimentinė sveikatos stiprinimo klausimais patariančioji sistema [6]. Šiame straipsnyje aprašytos sistemos paskirtis – teikti rekomendacijas, kokius veiksmus atlikti / neatlikti atsižvelgiant į bendrą sveikatos būklę (pavyzdžiui, širdies ligą) ir, iš dalies, į esamą padėtį (pavyzdžiui, pulsą).

**Šio straipsnio tikslas** – išanalizavus esamas semantinio pasaulinio tinklo technologijas, pasiūlyti dalykinės srities ontologija grindžiamos patariančiosios sistemos kūrimo metodiką ir pristatyti pagal šią metodiką sukurta eksperimentinę realizaciją, skirtą problemoms, kilusioms dėl BIOS klaidų, spręsti.

**Tyrimo metodai:** mokslinės ir techninės literatūros analizė, modeliavimas, eksperimentas.

### Semantinio tinklo technologijos

Sąvoka *ontologija* skirtinguose informacijos šaltiniuose apibrėžiama nevienodai. Apibrėžimas dažnai parodo autoriaus požiūrį į numatomą ontologijos pagrindinę paskirtį [7], pavyzdžiui:

- *Ontologija yra konceptualios sistemos, kuri yra charakterizuojama specifinėmis loginėmis savybėmis, reprezentacija.* Šis apibūdinimas pabrėžia teiginių, galiojančių dalykinėje srityje, rinkinio egzistavimą.

- *Ontologija yra konceptualizacijos sinonimas.* Šiame akcentuojama, kad kalbama apie abstraktų, supaprastintą pasaulio dalies vaizdą.
- *Ontologija yra specializuota žinių bazių rūšis.* Šiame apibrėžime teigiama, kad ontologija yra inžinerinis artefaktas.

Bendru atveju dalykinės srities ontologija specifikuoja šiuos elementus: sąvokas, esybių (daiktų bei reiškinių) tipus, sąvokų hierarchijas, esybių tipų tarpusavio sąryšius, galiojančias taisykles, dėsningumus apie esybių tipus ir sąryšius, pavyzdinius atvejus.

Ontologiją formaliai galima aprašyti rinkiniu  $\langle C, R, I, A \rangle$  [7], kur:

- *C* yra konceptų (klasių) aibė. Konceptai nusakoma, apie ką bus nagrinėjamas turinys. Taip pat konceptų aibė apibrėžia dalykinėje srityje vartojamą terminiją.
- *R* yra ryšių aibė. Ryšiai reprezentuoja asociacijas tarp srities konceptų. Dažniausiai yra naudojami specializaciją-generalizaciją žymintys ryšiai (angl. *is-a*). Pagal šiuos ryšius sudaromos konceptų taksonominės struktūros. Taip pat gali būti pritaikomi agregavimą žymintys ryšiai (angl. *part-of*) bei kitokie ryšiai. Ryšiai paprastai būna binariniai, jų pirmas argumentas vadinamas sritimi (angl. *domain*), o antras – rangu (angl. *range*). Pavyzdžiui, trejete  $\langle \text{Dokumentas} \rangle \langle \text{yraSukurtas} \rangle \langle \text{Autorius} \rangle$  *Dokumentas* yra sritis, *Autorius* yra rangas, *yraSukurtas* yra ryšio pavadinimas. Atributai išskiriami kaip atskiras ryšio tipas; jie iš esmės skiriasi tuo, kad rangas yra ne konceptas, o paprastas duomenų tipas. Pavyzdžiui, trejete  $\langle \text{Autorius} \rangle \langle \text{turi} \rangle \langle \text{Pavardė} \rangle$  *Pavardė* yra paprasta simbolių eilutė.
- *I* yra egzempliorių aibė. Jei ontologija pilna, tai bet kuris aprašomos realaus pasaulio dalies objektas priklauso vienai ontologijoje deklaruotai klasei.
- *A* yra aksiomų aibė. Aksiomomis išreiškiami visada teisingi teiginiai. Jos naudojamos naujai informacijai išgauti.

Šiuo metu populiariausia ontologijų kalba yra OWL (angl. *Web Ontology Language*). Tai vieningas žinių apie tai, kas egzistuoja realiame pasaulyje, pateikimo internete formatas. Nuo 2004 m. OWL yra rekomenduojama W3C konsorciumo [8]. OWL turi 3 dialektus: 1) OWL Lite – pritaikyta pačioms paprasčiausioms užduotims atlikti ir turi mažiausias išraiškos galimybes; 2) OWL DL – tinkamiausia daugeliui projektuojamų ontologijų, vadovaujasi aprašomąja logika; 3) OWL Full – labiausiai išplėtotas dialektas, turintis geriausias žinių išraiškos priemones, tačiau neteikiantis

garantijos dėl samprotavimų proceso baigtinumo. Kiekvienas iš šių dialektų yra prieš tai buvusio plėtinys.

RDF (angl. *Resource Description Framework*) yra kalba metaduomenims aprašyti semantiniame tinkle, ji taip pat rekomenduota W3C konsorciumo [9]. RDF kalboje sąryšius tarp resursų nusakome trejetais  $\langle \text{subjektas} \rangle \langle \text{predikatas} \rangle \langle \text{objektas} \rangle$ . Tokių trejetų aibė sudaro RDF grafą, kuris yra kryptinis ir žymėtasis. Atskiru trejetu konstatuojamas konkretus faktas, o pilnas RDF grafas reprezentuoja faktų, galiojančių tam tikroje dalykinėje srityje, visumą.

Informacijai iš RDF grafų išgauti reikalingos užklausų kalbos. Technologiniu požiūriu, užklausomis grįstų samprotavimų standartas yra SPARQL (angl. *Simple Protocol and RDF Query Language*) – tai 2006 m. sukurta ir 2008 m. sausio mėnesį W3C konsorciumo rekomenduota ontologijų užklausų kalba [10].

SPARQL palaiko 4 užklausų formas:

- SELECT gražina visus arba dalį kintamųjų, kurie buvo susieti užklausos šablone.
- CONSTRUCT gražina RDF grafą, sukonstruotą pagal užklausos šablone paminėtus kintamuosius.
- ASK nustato, ar egzistuoja faktinė informacija pagal užklausą. Gražinama loginė reikšmė.
- DESCRIBE gražina RDF grafą, kuris aprašo rastus resursus.

SPARQL kalba yra kilusi iš RDQL kalbos [11], RDQL užklausos technologiškai lengvai transformuojamos į SPARQL užklausas.

SPARQL įgalina užrašyti tiek paprastas užklausas, išreiškiamas paprastu grafo šablonu  $\langle \text{subjektas} \rangle \langle \text{predikatas} \rangle \langle \text{objektas} \rangle$ , kur atskiras trejeto elementas gali būti kintamasis, tiek sudėtingesnes, naudojančias agreguotus šablonų rinkinius ir papildomus ribojimus.

SPARQL yra nepriklausoma nuo kalbos ir naudojamos platformos, tačiau, renkant konkretų SPARQL užklausas realizuojantį įrankį, reikia atsižvelgti į kitas naudojamas technologijas.

### Dalykinės srities ontologija grįstos patariančiosios sistemos kūrimo metodika

Dalykinės srities ontologija gali būti naudojama skirtingais programinės įrangos gyvavimo ciklo etapais ir atlikti skirtingą paskirtį IS kūrimo ar eksploatavimo metu. Išsamus ontologijos panaudojimo suskirstymas pateikiamas darbe [2], kuriame išskiriami trys aspektai, arba dimensijos:

- *Laiko dimensija:* ontologija naudojama kūrimo metu ir vykdymo metu.

- **Struktūros dimensija:** IS gali būti ontologiją suprantanti ir ontologija grįsta.
- **Integravimo dimensija:** ontologija egzistuoja kaip architektūrinis sistemos elementas arba kaip informacinis resursas.

Mūsų siūlomoje metodikoje dalykinės srities ontologija naudojama vykdymo metu. IS yra ontologija grįsta, kadangi ontologija yra privalomas sistemos struktūrinis-architektūrinis komponentas.

Tokia dalykinės srities ontologija grįstos patariančiosios sistemos kūrimo metodika apima šiuos žingsnius:

1. Dalykinės srities modeliavimas ir reprezentavimas dalykinės srities ontologija.
2. Faktinės informacijos surinkimas ir talpinimas dalykinės srities ontologijoje arba atskiroje žinių bazėje.
3. Užklausų, kuriose naudojami dalykinės srities ontologijos elementai, realizavimas.
4. Vartotojo sąsajos projektavimas ir realizavimas.
5. Kitų programinės sistemos dalių projektavimas ir realizavimas.

Toliau išsamiau aprašomi 1-as ir 3-ias žingsniai, pateikiamos rekomendacijos. Su kitais žingsniais susiję praktinės realizacijos aspektai apibūdinami praktinį eksperimentą aprašančiame skyrelyje.

### Dalykinės srities modeliavimas ir reprezentavimas dalykinės srities ontologija

Pakankamai siauros apimties dalykinės srities ontologijai kurti rekomenduojame tokią žingsnių seką [12]. Čia kiekvieno žingsnio aprašymas papildomas rekomendacijomis:

1. Srities, paskirties, apimties, naudojimo konteksto nustatymas. Nors dalykinės srities modeliavimas išskiriamas kaip visiškai atskira veikla, ir ontologiją stengiamasi modeliuoti pakankamai universaliai, kad tiktų naudoti skirtingose IS, esant bendram atvejui, negalima atsiriboti nuo jos pagrindinės paskirties. Turi būti randami atsakymai į tokius klausimus [13]: a) kokią sritį aprėps ontologija? b) kam ši ontologija bus naudojama? c) į kokius klausimus informacija, esanti ontologijoje, turi pateikti atsakymus, t. y., kokios planuojamos tipinės užklausos? d) kas naudos ir vystys šią ontologiją? Srities, paskirties, apimties, naudojimo konteksto nustatymas daro įtaką ontologijos apimčiai, detalumo lygiui, kalbos vaizdavimo parinkimui, ontologijos ekspresyvumo lygio parinkimui, ryšių tipų įvairovei.
2. Svarbių dalykinės srities konceptų išskyrimas ir taksonominių struktūrų sudarymas. Gali būti

naudojami geriausi, plačiai pripažinti srities vadovėliai, techninė literatūra, ekspertų (dėstytojų) žinios, standartai ir rekomendacijos. Naudojamas „iš viršaus žemyn“ projektavimo principas.

3. Kitų ryšių tipų (pavyzdžiui, agregavimo, priklausomybės) apgalvojimas, reikalingų ryšių tipų išskyrimas. Galima rinktis lengvasvorės (angl. *lightweight*) arba sunkiasvorės (angl. *heavyweight*) ontologijos tipą. Reikia ieškoti „aukso vidurio“ tarp išraiškingumo ir išsprendžiamumo. Kuo mažesnis išraiškingumas, tuo efektyviau realizuojami samprotavimų mechanizmai. Tai svarbu sistemose, kur, remiantis ontologija, ne ieškoma sudėtingų problemų sprendimų, o padedama vartotojui atlikti paprastus veiksmus. Taip pat šiame žingsnyje turi būti suliejamos atskiros taksonominės struktūros, apibrėžiant esminius ryšius tarp hierarchijose aukščiausiai esančių konceptų. Labai svarbu, kad žinių struktūra būtų semantiškai teisinga. Tipinė klaida yra hierarchijų naudojimas, kai ryšys yra ne specializacijos / generalizacijos tipo. Darbe [14] aiškiai išskirti du ontologijų tipai: schema ir temomis grįstos ontologijos. Temomis grįstose ontologijose gali būti susieti konceptai, kurių sąsajos tiesiog intuityvios, pavyzdžiui *Afrika* ir *Nilas*, tačiau, norint realizuoti efektyvius samprotavimų mechanizmus, reikia naudoti schema grįstą ontologiją, kurioje semantikos apibrėžimo pagrindas yra aibių teorija, ir todėl egzempliorių aibė atspindi klasės semantiką.
4. Klasių savybių sąrašo sudarymas ir savybių reikšmių ribojimų nustatymas. Sudarant klasių savybių sąrašus, vėl reikia grįžti prie klausimo „Į kokius klausimus ontologijoje esanti informacija turi pateikti atsakymus?“
5. Egzempliorių išskyrimas. Nedidelės apimties ontologijai rekomenduojama egzempliorius aprašyti toje pačioje ontologijoje.

Daugiausia dėmesio reikia skirti 2-am ir 3-iam žingsniam, nes nuo jų labiausiai priklauso ontologijos kokybė. Šie du žingsniai apima svarbiausius ontologijos modeliavimo ir konstravimo etapus: 1) dalykinės srities žinių išgavimą – specifikaciją; 2) konceptualios struktūros projektavimą – konceptualizaciją [15].

Dalykinės srities reprezentacijos atskyrimas tokia forma, kad ją galėtų naudoti skirtingos IS, leidžia pakartotinai naudoti tiek šią specifikaciją, tiek kurti šabloninius įrankius darbui su konkrečia ontologija. Reprezentacijai rekomenduojama naudoti standartizuotus produktus, šiuo atveju, OWL ontologijų užrašymo kalbą.

## Samprotavimų, kuriuose naudojami dalykinės srities ontologijos elementai, mechanizmų realizavimas

Ontologija pati savaime yra statinis IS komponentas ir jokių procesų nevykdo. Dėl šios priežasties reikalingi programiniai moduliai, kurie vykdytų vartotojo užklausas ir, remdamiesi dalykinės srities žiniomis, atsakytų į vartotojo keliamus klausimus. Samprotavimuose vartojami skirtingi ontologijos elementai [16]: konceptai, egzemplioriai, ryšiai.

Skiriami du samprotavimų virš ontologijos būdai [17]:

- *Užklausomis grįsti samprotavimai.* Šių samprotavimų idėja yra kilusi iš reliacinių duomenų bazių technologijos. Naudojama struktūrizuota užklausų kalba. Pagrindinis tikslas dažnai būna išgauti nustatytą šabloną atitinkančius duomenis.
- *Logika grįsti samprotavimai.* Šis samprotavimų būdas gali būti panaudotas, kai ontologija yra užrašyta naudojant aprašomąją logiką.

Užklausomis grįsti samprotavimai yra paprastesni, efektyvesni nesudėtinguose uždaviniuose.

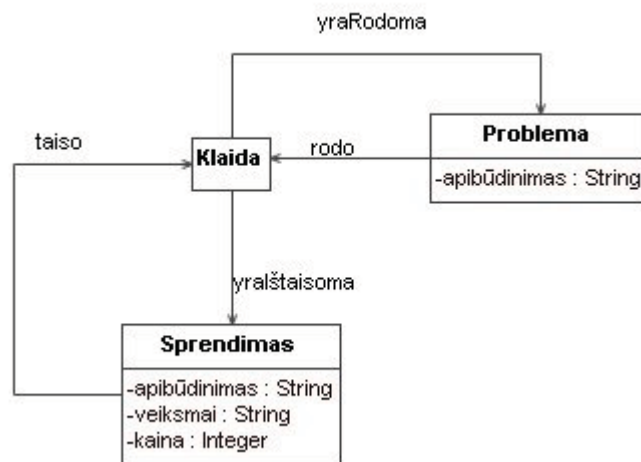
Todėl šis samprotavimų būdas dažniausiai pasirenkamas, kai ontologija naudojama vykdymo metu. Mes rekomenduojame užklausomis grįstus samprotavimus.

## Dėl BIOS klaidų kylančių problemų sprendimas: sistemos projektavimo ir kūrimo specifika

### Dalykinės srities modeliavimas

Tyrimams pasirinkta dalykinė sritis – kompiuterio sisteminės klaidos ir dėl jų kylančių problemų sprendimas. Šio praktinio eksperimento atveju dalykinė sritis susiaurinta apsiribojimu dėl BIOS klaidų kylančių problemų sprendimo. BIOS (angl. *Basic Input / Output System*) – tai sisteminė programinė įranga, saugoma kompiuterio atmintyje. Viena iš jos funkcijų yra kompiuterio tikrinimo testo POST (angl. *Power On Self Test*) atlikimas, kai, įjungus kompiuterį, tikrinami svarbiausi jo komponentai.

Bazinės ontologijos klasės pavaizduotos 1 pav. (Pastaba: realioje ontologijoje konceptams, ryšiams ir egzemplioriams įvardinti vartojama anglų kalba).



1 pav. Bazinės klasės ir ryšiai tarp jų

Šios klasės nusako dalykinę sritį, kurią natūralia kalba galima būtų apibūdinti taip: „Kartais vartotojo kompiuteris dėl jam nežinomų priežasčių nepradeda funkcionuoti taip, kaip priklauso. Įjungus kompiuterį, atliekamo POST testo metu sistema pateikia garsinius signalus. Pagal šiuos garsinius signalus ir / arba pranešimus ekrane galima spręsti apie įvykusią sisteminę klaidą. Dažniausiai problema būna susijusi su BIOS klaida. Vienareikšmiškai identifikavus klaidą, galima išspręsti problemą (pačiam arba kreipiantis į specialistus; pastaba: pritaikyta tik Award BIOS turintiems įrenginiams).

Kaip pavaizduota 1-ame paveiksle, konkreti problema, t. y. konkretaus tipo garsinis signalas, leidžia vienareikšmiškai identifiкуoti klaidą. O konkreti klaida (pavyzdžiui, *neprijungta klaviatūra*) yra ištaisoma atlikus tam tikrus konkrečius veiksmus (*garantavus klaviatūros prijungimą prie pagrindinės plokštės*).

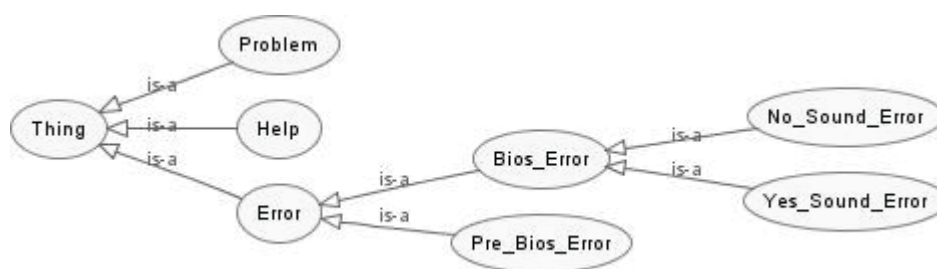
Ontologijai modeliuoti pasirinkta sistema PROTEGE 4.0 [18]. Ryšiai modeliuoti taip, kad ontologija atitiktų schema grįstos ontologijos principus. Ontologija yra lengvasvorė (ALIF klasės). 1 lentelėje pateikiamos atskiros ontologijos failo *errors.owl* dalys, atspindinčios, kaip ontologijoje deklaruojami skirtingi elementai.



1 lentelė. Ištraukos iš ontologijos failo *errors.owl*, iliustruojančios skirtingų tipų elementų užrašymą

Ontologijos elementas	Pavyzdys
Antraštė	<pre>&lt;!DOCTYPE rdf:RDF [   &lt;!ENTITY owl "http://www.w3.org/2002/07/owl#" &gt;   &lt;!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" &gt;   &lt;!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" &gt;   &lt;!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" &gt;   &lt;!ENTITY errors "http://cokolis25.puslapiai.lt/errors.owl#" &gt;   &lt;!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" &gt; ]&gt;</pre>
Klasė	<pre>&lt;owl:Class rdf:about="#Error"/&gt;</pre>
Poklasis	<pre>&lt;owl:Class rdf:about="#Bios_Error"&gt;   &lt;rdfs:subClassOf rdf:resource="#Error"/&gt; &lt;/owl:Class&gt;</pre>
Ryšys	<pre>&lt;owl:ObjectProperty rdf:about="#isShownBy"&gt;   &lt;rdfs:domain rdf:resource="#Error"/&gt;   &lt;rdfs:range rdf:resource="#Problem"/&gt; &lt;/owl:ObjectProperty&gt;</pre>
Inversyvus ryšys	<pre>&lt;owl:ObjectProperty rdf:about="#shows"&gt;   &lt;rdfs:range rdf:resource="#Error"/&gt;   &lt;rdfs:domain rdf:resource="#Problem"/&gt;   &lt;owl:inverseOf rdf:resource="#isShownBy"/&gt; &lt;/owl:ObjectProperty&gt;</pre>
Savybė (funkcinė)	<pre>&lt;owl:DatatypeProperty rdf:about="#description"&gt;   &lt;rdf:type rdf:resource="&amp;owl;FunctionalProperty"/&gt;   &lt;rdfs:domain rdf:resource="#Bios_Error"/&gt;   &lt;rdfs:range rdf:resource="&amp;xsd:string"/&gt; &lt;/owl:DatatypeProperty&gt;</pre>
Egzempliorius ir jo savybės reikšmė	<pre>&lt;Problem rdf:about="#variant_3"&gt;   &lt;description rdf:datatype="&amp;xsd:string"&gt;Trys trumpi&lt;/description&gt; &lt;/Problem&gt;</pre>
Egzempliorius ir jo dalyvavimas sąryšyje	<pre>&lt;No_Sound_Error rdf:about="#code_0"&gt;   &lt;isShownBy rdf:resource="#variant_0"/&gt; &lt;/No_Sound_Error&gt;</pre>

Kiekviena ontologijos klasė yra *Protege* integruotos klasės *Thing* poklasis. Sukonstruota klasių taksonomija pavaizduota 2 pav.:



2 pav. Ontologijos klasių taksonomija

### Samprotavimų realizacija ir eksperimentinės sistemos įvertinimas

Suprojektuota ir realizuota eksperimentinė patariančiosios sistemos versija. Užklausoms realizuoti pasirinktas įrankis RAP [19], kadangi jis pritaikytas PHP ir MySQL pagrindu kurtoms IS.

Eksperimentinė sistema leidžia įvardinti problemą, pasirenkant ją iš galimų sąrašo. Vartotojas negali įvesti užklauso lietuvių kalba, nes sistemoje nėra integruotų natūralios kalbos atpažinimo mechanizmų; tačiau parinkto užklauso pateikimo formato privalumas yra tas, kad vartotojui pateikiamos visos galimos BIOS problemų išraiškos. Dėl šios priežasties sistemoje negalima užklausa apie problemą, kuriai sprendimas nenumatytas. Pasirinkimų sąrašas sudaromas dinamiškai, t. y. iš ontologijoje esančių klasės *Problema* egzempliorių savybės *apibūdinimas* reikšmių (3 pav):

```
$query1 = '
PREFIX ns1:
<http://cokolis25.puslapiai.lt/errors.owl#>
SELECT ?kodas ?klaida
WHERE { ?x ns1:aprasymas ?kodas.
?x ns1:klaidos ?klaida
}
ORDER BY ASC(?kodas)';
```

3 pav. Užklausa siekiant dinamiškai formuoti problemų sąrašą iš ontologijos elementų

Įvykdžius kitą SPARQL užklausa, problemos sprendimas pateikiamas vartotojui (4 pav.).

```
Užduotas klausimas:
Kompiuterio garsų signalų seka:
Vienas ilgas ir trys trumpi

Žemiau pateiktas atsakymas į Jūsų klausimą:
Klaviatūros inicializacijos klaida

Galimas gedimo šalinimo variantas:
Patikrinkite klaviatūros sujungimą su pagrindine plokšte
```

4 pav. Rezultatų pateikimo pavyzdys

Be tiesioginio problemos ir jos sprendimo susiejimo, galima generuoti ataskaitas pagal egzempliorių savybių reikšmes (5 pav.).

- Sąrašas gedimų, kurių pataisymas nekainuoja
- Sąrašas gedimų, kurių pataisymas gali kainuoti
- Sąrašas gedimų, kai neveikia maitinimo blokas
- Sąrašas gedimų, kai neveikia klaviatūra

5 pav. Ataskaitos pasirinkimas

Galimos generuoti ataskaitos turi būti apmąstomos dar prieš modeliuojant dalykinę sritį. Ataskaitos generavimo užklauso pavyzdys pateikiamas 6 paveiksle.

```
SELECT ?kiek ?kodas ?name
WHERE ( ?x, ?name)
( ?x, ?kiek)
( ?x, ?y)
( ?y, ?kodas)
AND ?name ~ "/"^Klaviat/"
USING ns1 FOR
```

6 pav. Ataskaitos užklauso pavyzdys

Sukurtą sistemą numatyta plėsti šiomis kryptimis:

- Dalykinės srities ontologijos apimties plėtimas. Bus nagrinėjamos ne tik BIOS klaidos.
- Bendras žinių kūrimo realizavimas per vartotojų autorizacijos mechanizmų sudarymą. Šiuo metu visi vartotojai yra vieno lygio ir turi tik užklauso formulavimo teisę. Planuojama skirti dviejų tipų vartotojus. Aukštesniojo lygio vartotojai galės modifikuoti, plėsti ontologiją numatytose ribose per naršyklinę vartotojo sąsają. Tokiu būdu bus palengvinami tiek žinių išgavimo iš ekspertų, tiek jų pateikimo paprastam vartotojui procesai.

Dalykinės srities ontologijos projektavimas, kūrimas, resursų žymėjimas jos pagrindu yra sunkus darbas, reikalaujantis tiek daug intelektinių išteklių, tiek ir daug darbo laiko sąnaudų. Nepaisant to, sukurta aukštos kokybės ontologija gali būti panaudota skirtingose sistemose, ir tokiu būdu pasiekiamas pakartotinis naudojimas. Taip pat dalykinės srities ontologijos integravimas į IS leidžia sumažinti IS eksploataavimo metu reikalingų ekspertų darbo laiko sąnaudų, nes dalį veiksmų gali atlikti sistema.

### Išvados

1. Mokslinės / techninės literatūros analizė rodo, kad semantiniame pasauliniame tinkle galintys funkcionuoti įrankiai, technologijos, kalbos,

- standartai vis tobulėja ir darosi vis labiau suderinami, o pagrindinis nepakankamos semantinio pasaulinio tinklo plėtros veiksnys yra iniciatyvų kurti pasaulinio lygmens dalykinės srities ontologijas ir atlikti jomis grįstą interneto resursų žymėjimą trūkumas.
2. Atliktas eksperimentas parodė, kad Lietuvoje įprastam LAMP technologijų komplektui (Linux, Apache, MySQL, PHP) egzistuoja tinkami ir suderinami su šiomis technologijomis semantiškumą realizuojantys įrankiai (pavyzdžiui, Protege ir RAP).
  3. Straipsnyje pasiūlytą metodiką galima pritaikyti bet kuriai gerai struktūrizuotai dalykinei sričiai, tiek realizuojant probleminio / patirtinio mokymosi principus e. mokymo(si) sistemose, tiek paramos paslaugų teikimo prekės ar produkto vartotojui sferoje.
  4. Metodikoje numatytas automatizuotas sudėtingų sprendimų priėmimo procesas; šios metodikos pagrindiniai privalumai:
    - a) dalykinės srities ir resursų atskyrimas sudaro galimybes pakartotinai naudoti dalykinės srities reprezentaciją skirtingiems funkciniais tikslams;
    - b) semantinė paieška, grindžiama dalykinės srities ontologija, palengvina atsakymų į vartotojo užklausas realizavimą.
5. Samara K., Patel D., Patel S., 2007, An Ontology Based Knowledge Experiential Learning Framework. *Journal of Knowledge Management Practice*, Vol. 8. No. 3. <<http://www.tlinc.com/articl138.htm>>.
  6. Izumi S., Kuriyama D., Itabashi G., Togashi A., Kato Y., Takahashi K., 2006, An Ontology-based Advice System for Health and Exercise. *Proceedings of Internet and Multimedia Systems and Applications*. P. 95–100.
  7. Guizzardi G., 2005, Ontological Foundations for Structural Conceptual Models. *PhD dissertation*. Telematica Instituut Fundamental Research Series, Vol. 015. Enschede, the Netherlands.
  8. OWL Web Ontology Language Reference, 2004, <<http://www.w3.org/TR/owl-ref/>>.
  9. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. <<http://www.w3.org/TR/rdf-concepts/>>.
  10. SPARQL Query Language for RDF. <<http://www.w3.org/TR/rdf-sparql-query/>>
  11. RDQL – A Query Language for RDF. <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>.
  12. Dzemydiene D., Tankeleviciene L., 2008, On the development of domain ontology for distance learning course. In Sakalauskas L., Weber, G. W., Zavadskas E. K. (Eds.), *The 20th International Conference EURO Mini Conference on Continuous Optimization and Knowledge-Based Technologies EurOPT-2008*. Vilnius: Technika. P. 474–479.
  13. Noy N. F., McGuinness D. L., 2001, *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory, Technical Report KSL-01-05.
  14. Kiryakov A., 2006, Ontologies for knowledge management. In Davies J., Studer R., Warren P. (Eds.), *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Chichester: John Wiley & Sons, Ltd.
  15. Uschold M., Gruninger M., 1996, Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. Vol. 11. No. 2. P. 93–155.
  16. Brusse R., Pokraev S., 2007, Reasoning on the Semantic Web. In Cardoso J. (ed), *Semantic Web Services: Theory, Tools and Applications*. London: IGI Global. P. 110–133.
  17. Walton C. D., 2007, *Agency and the Semantic Web*. Oxford, New York: Oxford University Press.
  18. The Protégé Ontology Editor and Knowledge Acquisition System. <<http://protege.stanford.edu/>>
  19. RAP – RDF API for PHP V0.9.6. <http://sites.wiwiw.de/suhl/bizer/rdfapi/index.html>

## Literatūra

1. Ruiz F., Hilera J. R., 2006, Using Ontologies in Software Engineering and Technology. In Calero C., Ruiz F., Piattini M. (Eds.), *Ontologies for Software Engineering and Software Technology*. Berlin / Heidelberg: Springer.
2. Teresevičienė M., Gedvilienė G., Zuzevičiūtė V., 2006, *Andragogika*. Kaunas: Vytauto Didžiojo universiteto leidykla.
3. Allert H., Markannen H., Richter C., 2006, Rethinking the Use of Ontologies in Learning. In Memmel M., Burgos D. (Eds), *Proceedings of the 2nd International Workshop on Learner-Oriented Knowledge Management and KM-Oriented Learning (LOKMOL 06)*. P. 115–125.
4. Blanco-Fernandez Y., Pazos-Arias J. J., Gil-Solla A., Ramos-Cabrer M., Lopez-Nores M., 2008, Semantic Reasoning: A Path To New Possibilities of Personalization. In *Proceedings of 5th European Semantic Web Conference (ESWC2008)*, Berlin / Heidelberg: Springer. P. 720–735.

## INTELLIGENT SYSTEM FOR FAULT DIAGNOSTICS BASED ON DOMAIN ONTOLOGY

*Lina Tankelevičienė, Rimantas Sakalauskas*

### Summary

In this paper, we analyse semantic web technologies for representation and management of domain knowledge. We present an approach for the development of an intelligent advice system for fault diagnostics based on domain ontology. The approach encapsulates both a separation of conceptualisation of domain and factual data, and realisation of reasoning mechanisms over domain ontology. We present an experimental system for supporting identification of BIOS-related computer problems. The system was developed using the proposed approach. Our approach can be used in different domains, both in e-learning context, and in the field of support services delivery for users of a product or process.

**Keywords:** domain ontology, intelligent system, problem based learning.

## DALYKINĖS SRITIES ONTOLOGIJA GRĮSTA INTELEKTUALIOJI GEDIMŲ DIAGNOSTIKOS SISTEMA

*Lina Tankelevičienė, Rimantas Sakalauskas*

### Santrauka

Straipsnyje analizuojamos semantinio pasaulinio tinklo technologijos dalykinės srities žinioms užrašyti ir valdyti. Pristatoma dalykinės srities ontologija grįstos patarančiosios sistemos kūrimo metodika. Metodika apima dalykinės srities konceptualizacijos ir faktinės informacijos atskyrimą bei samprotavimų virš dalykinės srities ontologijos mechanizmų realizavimą. Remiantis šia metodika, sukurta eksperimentinė sistema, skirta padėti identifikuoti su BIOS klaidomis susijusias problemas bei jas spręsti. Metodiką galima pritaikyti bet kokiai dalykinei sričiai, tiek realizuojant probleminio mokymosi principus e. mokymo(si) sistemose, tiek paramos paslaugų teikimo prekės ar produkto vartotojui sferoje.

**Prasminiai žodžiai:** dalykinės srities ontologija, patariančioji sistema, probleminis mokymasis(is).

Įteikta 2008-10-07