

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Jovita Vyšnauskaitė

**Nuo dalykinės srities nepriklausomas TOLOG
užklausų sudarymas**

Magistro darbas

Darbo vadovas

prof. Rita Butkienė

Kaunas, 2012

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Jovita Vyšnauskaitė

**Nuo dalykinės srities nepriklausomas TOLOG
užklausų sudarymas**

Magistro darbas

Recenzentas
doc. Tomas Blažauskas

2012-05-24

Darbo vadovas
doc. Rita Butkienė

2012-05-24

Atliko
IFM-0/4 gr. Stud.
Jovita Vyšnauskaitė

2012-05-24

Kaunas, 2012

Turinys

DOMAIN INDEPENDENT DEVELOPMENT OF TOLOG QUERIES	5
TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS.....	6
1 ĮVADAS.....	7
2 NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO ANALIZĖ	8
2.1 Tyrimo objektas, sritis, problema ir numatomas sprendimas	8
2.2 Tyrimo tikslas ir uždaviniai	8
2.3 Analizės tikslas.....	8
2.4 Analizės metodai	9
2.5 Tyrimo objekto analizė	9
2.5.1 Teminis tinklas.....	9
2.5.2 XTM kalba.....	16
2.5.3 Įrankis Ontopia	18
2.5.4 TOLOG užklauso	25
2.5.1 Natūralios kalbos užklauso.....	27
2.5.2 Natūralios kalbos klausimų analizė TOLOG užklauso kalbos kontekste	28
2.5.3 Įrankis LifeRay Portal.....	40
2.6 Sistemos vartotojų analizė.....	41
2.6.1 Vartotojų aibė, tipai ir savybės	41
2.6.2 Vartotojų tikslai ir problemos	41
2.7 Esamų sprendimų analizė.....	41
2.8 Siekiamas sprendimas	44
2.9 Nuo dalykinės srities nepriklausomo TOLOG užklauso sudarymo analizės išvados.....	46
3 NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO SISTEMOS REIKALAVIMAI	47
3.1 Sistemos reikalavimų specifikacija	47
3.1.1 Sistemos funkciniai reikalavimai.....	47
3.1.2 Sistemos nefunkciniai reikalavimai.....	56
3.2 Dalykinės srities modelis	58
3.3 Sistemos reikalavimų analizės apibendrinimas.....	61
4 NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO SISTEMOS PROJEKTAS.....	62
4.1 Sistemos architektūros projektas	62
4.1.1 Sistemos loginė architektūra	62
4.1.2 Sistemos vartotojo sąsajos navigavimas	63
4.2 Detalus sistemos projektas	63
4.2.1 Teminio tinklo temų ir asociacijų įvardinimo taisyklės.....	65
4.2.2 Struktūrizuotos natūralios kalbos užklauso sudarymo taisyklės	65
4.3 Sistemos elgsenos modelis.....	66
4.4 Sistemos realizacijos modelis	70
5 NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO SISTEMOS REALIZACIJA	71
5.1 Sistemos realizacijos ir veikimo aprašymas.....	71
5.1.1 Demonstracinio teminio tinklo sukūrimas.....	71
5.1.2 Laikinos duomenų bazės sukūrimas teminio tinklo duomenims saugoti	71
5.1.3 Vartotojo sąsajos sukūrimas	71

5.1.4	Struktūrizuotos natūralios kalbos užklausų transformavimo į TOLOG užklausas algoritmo sudarymas.....	74
5.2	Sistemos testavimo modelis	77
5.3	Sistemos testavimo duomenys ir rezultatai	78
5.4	Realizacijos rezultatai	83
6NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO SISTEMOS EKSPERIMENTINIS TYRIMAS		84
6.1	Eksperimento planas	84
6.1.1	Jungtukai IR.....	84
6.1.2	Jungtukai ARBA.....	89
6.1.3	Neiginiai NE	94
6.1.4	IR, ARBA, NE junginiai.....	96
6.1.5	Abstrakcijos	98
6.2	Eksperimento rezultatai.....	102
7NUO DALYKINĖS SRITIES NEPRIKLAUSOMO TOLOG UŽKLAUSŲ SUDARYMO IŠVADOS		103
8LITERATŪROS SĄRAŠAS		104
PRIEDAS 1		106
PRIEDAS 2		109

Domain Independent Development of TOLOG Queries

Summary

The main goal of this work is to make abilities to search for any subject area thematic network using close to natural language form fulfill it. The main point came from various good reasons - users who are looking for information using thematic network, needs user-friendly interface. the consumer is aware to TOLOG queries, so the question submission must be close to natural language.

Seeking for solution in this study was analyzed the thematic networks, natural language queries, TOLOG queries, potential users of the system, their needs and existing simiral solutions. This work solved the question how a thematic network must be formed where would be abilities to natural language search, because there was no clear-cut rules. The analysis showed that there are no tools that would have user interface ant would give abilities to form the queries in form of thematic network, which would be similar to natural language and would be independent to the subject area. This is one of the main points of work to implement user-friendly interface form thematic network queries ant natural language questions transformation to TOLOG queries algorithm.

This work represents ant explains the algorithm, that transforms natural language queries to TOLOG queries. moreover, the experiment shows how many varioaus natural language queries could be provided using created system ant what TOLOG requests could be automatically generated.

Terminų ir santrumpų žodynis

Datalog – tai užklausų ir tasyklių kalba dedukcinėms duomenų bazėms, kuri taip pats yra Prolog poaibis.

Ginseng (angl. A guided input natural language search engine) – semantinių užklausų web vartotojo sąsaja.

Liferay – Java priemonėmis sukurta atviro kodo turinio valdymo sistema.

LTM (angl. Linear Topic Maps) - tiesinis teminio tinklo kūrimo formatas.

MySQL (angl. "My Sequel") – užklausų kalba skirta reliacinėms duomenų bazių valdymo sistemoms.

Omnigator – Ontopia modulis, leidžiantis naršyti po kiekvieną sukurta teminį tinklą.

Ontopia – teminių tinklų programų kūrimo ir palaikymo įrankis.

Ontopoly – Ontopia Topic Maps redaktorius.

OWL (angl. Ontology Web Language) - internetinių sistemų ontologijų kūrimo kalba.

Prolog – tai bendros parkirties loginio programavimo kalba, kuri susijusi su dirbtiniu intelektu ir kompiuterine lingvistika.

Portletas - grafinės sąsajos programinės įrangos komponentas tvarkomas web portaluose.

Querix - tokia natūralios kalbos vartotojo sąsaja, kurioje galima pateikti ir vykdyti pilnus angliškus klausimus.

RDF (angl. Resource Description Framework) – žinių ir informacijos valdymo standartas, aprašantis interneto išteklius.

SPARQL (angl. Protocol and RDF Query Language) - užklausų kalba, skirta RDF.

SQL (angl. Structured Query Language) – užklausų kalba skirta reliacinėms duomenų bazių valdymo sistemoms.

Tolog – loginė teminio tinklo užklausų kalba.

TM (angl. Topic Maps) – žinių atvaizdavimo modelis (liet. Teminis Tinklas)

XTM (angl. XML Topic Maps) - į XML panašus teminio tinklo kūrimo formatas.

WordNet – anglų kalbos leksikos duomenų bazė.

1 Įvadas

Kaip žinia, žmonija sparčiai žengia į priekį informacijos kompiuterizavimo srityje. Kuo daugiau informacijos, tuo didesnis jos poreikis. Taigi kokios nors srities informaciją reikia ne tik sutalpinti atitinkamoje duomenų saugykloje, bet ir sugebėti ją greitai pasiekti. Dideliame informacijos sraute kiekvienam prireikia atsirinkti sau reikalingą informaciją. Jeigu yra poreikis patogiai paieškai, tai visada atsiranda tas, kas kuria atitinkamas priemones, tobulina jas ir pasiūlo į rinką.

Taigi kalbant konkrečiau, vartotojams reikia patogaus, greito ir nesudėtingo būdo pateikti klausimą teminiam tinklui (duomenų talpykla) ir gauti tikslų atsakymą. Vartotojui siekiama suteikti galimybę šiuos klausimus užrašyti kuo natūralesne ir suprantamesne kalba. Kalba – tai daugeliui įprastas, savaime suprantamas dalykas ir vis daugiau technologinių procesų yra siejama su kalba ir jos apdorojimu. Tai reiškia, kad semantinės natūralios kalbos technologijos kas dieną tampa vis aktualesnės. Net ir verslas persikelia į virtualią erdvę - internetą, kuris be semantinių technologijų nieko vertas. Pasaulis jau seniai rūpinasi semantiniu kalbos vaizdavimu, kurį būtų galima padaryti suprantamą žmogui ir kompiuteriui. Norint virtualioje erdvėje kalbėti lietuvių kalba, reikia įsijungti į pasaulio bendruomenę, kuri jau šiandien taiko semantines technologijas daugelyje aktualių sričių ne tik teoriškai moksle, bet ir praktiškai, pavyzdžiui versle.

Tam, kad išsiaiškinti, kokios priemonės ir budai reikalingi šiai problemai spręsti, būtina visapusiškai ir nuodugniai išstudijuoti pačią problemą, tyriamuosius objektus atitinkamos srities aspektu ir jau esamus sukurtus sprendimus. Analizė reikalinga, kad pagrįstume siekiamo sprendimo tinkamumą ir privalumus.

2 Nuo dalykinės srities nepriklausomo TOLOG užklausų sudarymo analizė

2.1 Tyrimo objektas, sritis, problema ir numatomas sprendimas

Tyrimo objektas - teminis tinklas ir užklausos jame, ribota natūralia kalba užrašytų užklausų transformavimas į TOLOG užklausas.

Tyrimo sritis - ribota natūralia kalba užrašytų užklausų transformavimo į TOLOG užklausas automatizavimas.

Problema:

- nėra aiškiai suformuluotų taisyklių kaip turi būti suformuotas teminis tinklas, kuriame būtų įmanoma paieška natūralia kalba;
- nėra įrankių, kurie turėtų sąsają, leidžiančią formuluoti užklausas teminiame tinkle tokia forma, kuri artima natūraliai kalbai, ir kuri nepriklauso nuo dalykinės srities;
- vartotojas neturi žinių apie *TOLOG* sintaksę, todėl jam būtina tokia vartotojo sąsaja, kuri suteiktų draugišką ir suprantamą aplinką formuoti artimas natūraliai kalbai užklausas.

Sprendimas – sukurti priemones, kurios sudarytų teminį tinklą taip, kad informacijos paiešką jame būtų galima atlikti formuluojant užklausas natūralia kalba.

2.2 Tyrimo tikslas ir uždaviniai

Tyrimo tikslas – sudaryti galimybę pateikti užklausą bet kokios dalykinės srities teminiam tinklui natūraliai kalbai artima forma ir ją įvykdyti.

Uždaviniai:

1. Išanalizuoti teminio tinklo (Topic Map) užrašymo kalbą XTM ir jo kūrimą įrankiu Ontopia.
2. Išanalizuoti teminio tinklo užklausų kalbą TOLOG.
3. Sudaryti teminį tinklą (Topic Map) XTM kalba, kad iš pasirinktos formos užklausų natūralia kalba būtų galima suformuoti veikiančią TOLOG užklausą.
4. Sudaryti teminio tinklo temų ir asociacijų įvardinimo taisykles.
5. Sudaryti natūralios kalbos užklausos formulavimo taisykles.
6. Programiškai realizuoti užklausos natūralia kalba formulavimą ir transformavimą į TOLOG užklausą.
7. Eksperimentiškai iširti programinę realizaciją.

2.3 Analizės tikslas

Šio darbo analizės tikslas yra sudaryti galimybę pateikti užklausą bet kokios dalykinės srities

teminiam tinklui natūraliai kalbai artima forma ir ją įvykdyti. Šiam tikslui pasiekti, reikia atlikti tokius uždavinius - išanalizuoti:

1. Teminis tinklas ir jo sudarymas;
2. *XTM* kalba ir jos privalumai lyginant su kitomis;
3. Įrankis *Ontopia*, jo alternatyvos;
4. *TOLOG* užklauso;
5. Natūralios kalbos užklauso;

2.4 Analizės metodai

Analizės metu naudojamas paprastas palyginimo metodas. Palyginimui organizuoti panaudota schema: *tekstas-po-teksto*. Jis pritaikomas analizuojant teminių tinklų kūrimo įrankius, kalbas bei esamus sprendimus.

Tam, kad būtų objektyvus palyginimas, naudojami vienodi palyginimo kriterijai visiems lyginamiems objektams (4 ir 5 lentelės).

2.5 Tyrimo objekto analizė

Tyrimo objektai yra teminis tinklas, užklauso parašytos natūralia kalba ir *TOLOG* užklauso. Jiems susieti naudojami tam tikri įrankiai, kalbos, vartotojo sąsaja. Visi šie ryšiai trumpai apžvelgiant apibūdinami taip: Pradedant analizuoti teminį tinklą paaiškėja, kad tokie teminiai tinklai gali būti sudaryti kelių kalbų pagrindu. Teminio tinklo peržiūrai ir bandymams sukurta nemažai įrankių. Įrankiams užklauso pateikiamos kažkokia tam įrankiui pritaikyta užklauso kalba. Užklauso natūralia kalba teminiui tinklui pateikiamos per vartotojo sąsają.

Šiame darbe pasirinkta yra *XTM* kalba, įrankis *Ontopia*, kuriame užklauso užrašomos *TOLOG* kalba.

Pirmiausia reikia atsakyti į klausimą – kas tai yra teminis tinklas ir kaip jis sudaromas?

2.5.1 Teminis tinklas

Teminio tinklo analizei naudotasi literatūros šaltiniais [4], [5], [14] ir [16], kuriuose pateikiama teminio tinklų struktūra, paaiškinti teminių tinklų komponentai, jų reikšmės ir panaudojimas.

Teminiai tinklai yra duomenims koduoti skirta technologija, kuri nusako, kaip prijungti šiuos užkoduotus duomenis prie atitinkamų informacijos šaltinių. Teminiai tinklai rengiami apie:

1. temas, kurios atskleidžia aptariamus objektus;
2. sąsajas, kurios parodo ryšius tarp objektų;
3. pasirodymus, kurie sujungia objektus į atitinkamus informacijos šaltinius.

Teminiai tinklai gali būti pavaizduoti įvairiais būdais. Vienas iš būdų yra naudoti teminių tinklų sintaksę duomenų bazių dokumentuose taip pat, kaip naudojamos duomenų struktūros programose ar net mentališkai žmogaus mintyse. Šie skirtingi būdai skirti atvaizduoti tą pačią struktūrą. Tai yra tokia struktūra, kuri yra apibrėžiama kaip standarto *ISO/IEC13250* dalis duomenų formoje.

Teminių tinklą sudarantys komponentai ir jų apibūdinantys terminai surašyti 1 lentelėje.

Eil.Nr.	Teminio tinklo terminai	Terminų paaiškinimai
1.	Tema	Tema yra tarsi objekto įgaliojimas (<i>proxy</i>). Ji apibūdina objekto teminio tinklo sistemą.
2.	Objektas	Tai bet kuris objektas iš aplinkos, kurį galima aptarti, apibūdinti ir kuris kažkoku būdu yra susijęs su kitais objektais. Objektai būna kompiuterizuojami ir nekompiuterizuojami. Objektai esantys už kompiuterinės sistemos ribų negali būti pateikti tiesiogiai, todėl jiems sudaromas kompiuterinis pavidalas.
3.	<i>Reification</i>	Tai teminio tinklo kūrimo procesas yra vadinamas <i>reification</i> .
4.	Objekto tapatybė	Objekto tapatybė nurodo kur kuris objektas kurios temos yra sukurtas. Jei objektų tapatybės yra vienodos, tai objektai yra apjungiami. Tapatybės gali būti dviejų rūšių: <ul style="list-style-type: none"> • Nurodančios tiesiai į objektą. Tai įmanoma tik tada, kai objektas yra adresuojamas informacijos šaltinis. • Nurodančios objektą per objekto rodiklį.
5.	Objekto rodiklis	Objekto rodiklis vienareikšmiškai nurodo objekto tapatybę. Jei tapatybės yra vienodos, objektai sujungiami, bet jei rodikliai skiriasi, tai objektai neapjungiami. Yokios situacijos išvengiama panaudojus trečią objektą, kurio tapatybę nurodo anų dviejų objektų rodikliai. Tai naudinga

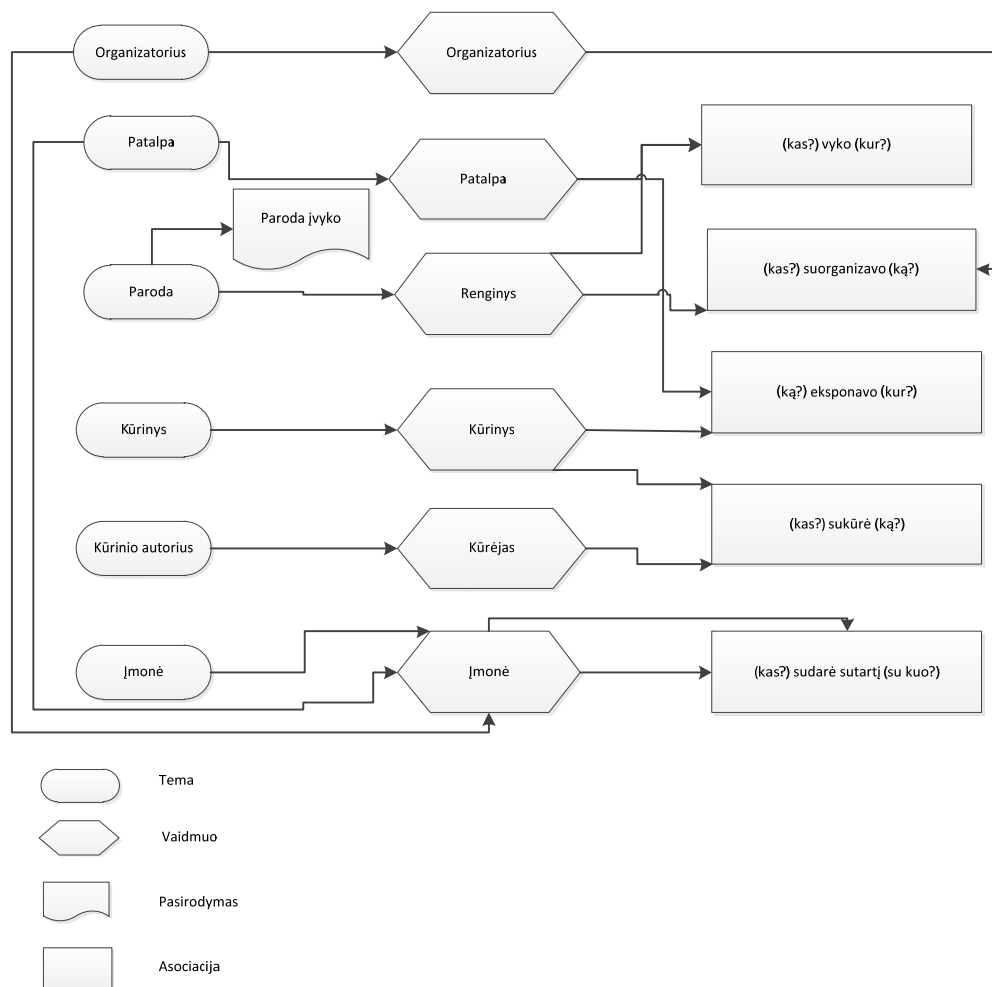
		tarpininkaujant dviems ontologijoms.
6.	Temos charakteristika	<p>Viskas, kas apibūdina teminį tinklą yra įvardinama kaip jo charakteristika. Tai yra:</p> <ul style="list-style-type: none"> • Temos vardas • Temos pasirodymas • Temos vaidmuo kaip sąsajos nario <p>Tokų savybių priskyrimas laikomas galiojančiu tam tikroje srityje arba kontekste.</p>
7.	Sritis (<i>Scope</i>)	Sritis nurodo temos charakteristikų galiojimo aplinkos ribas. Taip pat ji nusako charakteristikų priskyrimo aplinkybes. Kiekviena objekto charakteristika turi savo sritį.
8.	Vardas	Tema gali neturėti nei vieno ir turėti daug vardų. Vardas galioja tam tikroje srityje. Tema turi viena pagrindinį vardą ir keletą alternatyvių, kurie naudojami specifiniuose kontekstuose.
9.	Pagrindinis vardas	Pagrindinis temos vardas visada yra <i>string</i> tipo. Kai reikia naudoti temos vardą, tai naudojamas pagrindinis, nebent pagal kontekstą labiau tinka alternatyvus vardas. Toje pačioje galiojimo srityje negalima naudoti kelių temų su tokiais pačiais pagrindiniais vardais.
10.	Alternatyvus vardas	Alternatyvus vardas yra pagrindinio vardo variantas, kuris naudojamas pagal specifinį kontekstą.
11.	Parametrai	Parametrai yra informacija, kuri yra tam tikra forma įrašyta į tam tikras temas, kurios nurodo specifinį alternatyvaus vardo kontekstą. Renkantis vardą yra tyrinėjami parametrai, kad galima būtų pasirinkti tinkamiausią.
12.	Pasirodymas	Pasirodymas yra bet kokia informacija, kuri yra reikalinga analizuojamam objektui. Pasirodymas nurodo vieną iš tryjų charakteristikų (<i>Creator, Description, Version</i>). Jos gali būti priskirtos temai ir tema įgauna galiojimo sritį.

13.	Sąsaja	Tai ryšys tarp dviejų ir daugiau temų, kurios atlieka nario vaidmenį tame ryšyje. Kiekviena sąsaja yra susijusi su temos charakteristika ir priklauso tam tikrai sąsajos klasei. Sąsaja nustatoma pagal sąsajos skelbiamą objektą.
14.	Narys	Nario sąvoka nusako temos vaidmenį tam tikroje sąsajoje.
15.	Vaidmuo	Temos vaidmuo yra nario dalyvavimas tam tikroje sąsajoje.
16.	Klasė-instancija(<i>Class-Instance</i>)	Klasė-instancija yra sąsajos klasė, kuri reiškia klasės-instancijos ryšius tarp temų, kurios turi klasių ir atitinkamų instancijų vaidmenis.
17.	Pagrindinė klasė-poklasis(<i>Superclass-subclass</i>)	Pagrindinė klasė-poklasis yra sąsajos klasė, kuri reiškia ryšius tarp temų, kurios turi pagrindinių klasių ir atitinkamų poklasių vaidmenis.
18.	Teminis tinklas	Teminis tinklas yra temų, sąsajų ir sričių (bendrai vadinamų teminio tinklo mazgais) rinkinys, kuris gali egzistuoti vienoje iš dviejų formų: <p style="text-align: center;">Serijinį numerį turintis perdavimo formatas(pvz. Teminis tinklas išreikštas <i>XTM</i> ar kokia kita kalba);</p> <p style="text-align: center;">Aplikacijos-vidaus forma, kuri sudaryta pagal <i>XTM</i> kalbos naudojimo reikalavimus.</p> Teminio tinklo paskirtis yra perduoti žinias apie kokį tai objektą per kitų objektų informacijos tinklą.
19.	Teminio tinklo mazgas(<i>Topic Map Node</i>)	Teminio tinklo mazgas yra sistemos vidaus teminio tinklo objektai, kurie nusako temas, sąsajas ir sritis.
20.	Nuoseklus teminis tinklas(<i>Consistent Topic Map</i>)	Nuoseklus teminis tinklas, tai toks tinklas, kurio viduje kiekvienas objektas turi savo temą ir nėra galimybės jų apjungti ar dubliuoti.
21.	Teminio tinklo dokumentas	Tai toks dokumentas, kuriame yra vienas ar daugiau teminių tinklų, kurie atitinka <i>XTM</i> specifikaciją. Šie dokumentai gali turėti serijinį numerį dėl jų

		saugojimo ar perdavimo pagal <i>XTM</i> ar kokią kitokią specifikaciją.
22.	<i>XTM</i> dokumentas	Tai toks teminio tinklo dokumentas, kuris atitinka <i>XTM</i> specifikaciją.
23.	Apjungimas	Apjungimo terminas apima du skirtingus procesus: Dviejų teminių tinklų apjungimas; Dviejų temų apjungimas.

1 lentelė. Teminio tinklo terminai ir apibrėžimai.

Toliau pateiktas teminio tinklo pavyzdys sukurtas su įrankiu Ontopia, kad būtų aiškesnė teminio tinklo vizuali sandara.



1 pav. Demonstracinis teminis tinklas „Parodos“.

Teminio tinklo sandaros paaiškinimas yra 2-toje lentelėje. Taigi teminį tinklą sudaro temos, asociacijos, vaidmenys ir pasirodymai. Tema „Kūrinytis“ turi vaikinias temas „Paveikslas“ ir „Skulptūra“. Vadinasi „Paveikslo“ ir „Skulptūros“ temų tipas yra „Kūrinytis“. Kiekviena asociacija turi priklausančius vaidmenis, kuriuos naudoja temos. Pavyzdžiui

asociacijoje „(ką) eksponavo (kur?)“ yra du vaidmenys – „Kūrinys“ ir „Renginys“. Juos naudoja temos „Kūrinys“ ir „Paroda“. Pasirodymas nurodo, kada ir koku pavidalu pasirodo, sudalyvauja tema. Tai gali būti laikas, data ar netgi atvaizdas. Šio teminio tinklo atveju pasirodymas „Paroda įvyko“ yra naudojamas parodos įvykimo datai nusakyti. Teminio tinklo „Parodos“ vaizdas yra 1 paveiksle.

Teminio tinklo dalis	Pavadinimas	Aprašymas
Temos	Kūrinio autorius	Asmuo, kuris kuria meno kūrinius.
	Kūrinys	Meno kūrinys.
	Paveikslas	Meno kūrinys.
	Skulptūra	Meno kūrinys.
	Organizatorius	Įmonė ar asmuo, kuris organizuoja renginius (parodas).
	Paroda	Renginys, kuriame eksponuojami meno kūriniai (paveikslai, skulptūros).
	Patalpa	Įmonės ar organizacijos patalpos, kurios nuomojamos renginiams rengti.
	Įmonė	Įmonė apima ir patalpas, ir organizatorius, ir dar daugiau įmonių.
Asociacijos	(ką?) eksponavo (kur?)	Asociacija, kurioje dalyvauja du vaidmenys: kūrinys, renginys.
	(kas?) sukūrė (ką?)	Asociacija, kurioje dalyvauja du vaidmenys: kūrėjas, kūrinys.
	(kas?) suorganizavo (ką?)	Asociacija, kurioje dalyvauja du vaidmenys: organizatorius, renginys.
	(kas?) sudarė sutartį (su kuo?)	Asociacija, kurioje dalyvauja simetriniai vaidmenys įmonė

		ir įmonė.
	(kas?) vyko (kur?)	Asociacija, kurioje dalyvauja du vaidmenys: renginys, vieta.
Vaidmenys	Kūriny	Vaidmuo, kuris naudojamas temų paveikslas ir skulptūra.
	Kūrėjas	Vaidmuo, kuris naudojamas temos Kūrinio autorius.
	Organizatorius	Vaidmuo, kuris naudojamas temų organizatorius ir įmonė.
	Patalpa	Vaidmuo, kuris naudojamas temos patalpa.
	Renginys	Vaidmuo, kuris naudojamas temos paroda.
	Įmonė	Vaidmuo, kuris naudojamas temų organizatorius, patalpa.
Pasirodymai	Paroda įvyko	Nurodo, kada kokia paroda įvyko.

2 lentelė. Teminio tinklo „Parodos“ sandara.

2 paveiksle matosi, kaip atrodo asociacija *Sukure* ir kokie vaidmenys (angl. *roles*) jai priklauso: *Sukure* ir *Sukurtas autoriaus*. Šiuo atveju minėti vaidmenys rodo, kaip *Kuriny* ir *Kurinio autorius* dalyvauja sąsajoje *Sukure*.

2 pav. Asociacija *Sukure* ir vaidmenys joje: *Sukure* ir *Sukurtas autoriaus*.

Iš esmės teminiai tinklai gali būti sudaryti kelių kalbų pagrindu. Tačiau daugelyje straipsnių pabrėžiami *XTM* kalbos privalumai. Taigi, kokia yra kalba *XTM* ir kokie jos privalumai?

2.5.2 XTM kalba

XTM kalba yra sukurta *XML* kabos pagrindu, tačiau ji turi savo išskirtinų savybių. *XTM* yra daug glaustesnė ir pritaikyta konkrečiai sričiai. Tuo atžvilgiu ji yra labai patogi, nes yra greitai įsisavinama, bet tuo pačiu leidžia atlikti viską, ko reikia. *XTM* kalboje kaip ir *XML* galioja tos pačios elementų, atributų bei jų parametrų sudarymo taisyklės. *XTM* kalbos pagrindu galima sukurti dinaminę sistemą. Tai yra šabloninė sistema, kuri supaprastina vartotojo sąsają tarp pačio vartotojo ir sistemos. *XTM* pagrindu sukurta sistema leidžia užrašyti užklaudas panaudojant tuos pačius elementus, atributus ir parametrus. Tam nėra svarbios jų reikšmės.

Teminių tinklų dokumentų, kurie atitinka *XTM* specifikaciją, sudarymo ir kaitaliojimo sintaksė, tai pilnas sąrašas *XTM* elementų tipų (3 lentelė).

Eil. Nr.	Elementų tipai	Paaiškinimai
1.	<topicRef>	Nuoroda į Temos elementą
2.	<subjectIndicatorRef>	Nuoroda į Objekto rodiklį
3.	<scope>	Nuoroda į Temą(-as), kurios sudaro Taikymo sritis
4.	<instanceOf>	Nurodo į Temą, kuri apibūdina klasę
5.	<topicMap>	Temų tinklo dokumento elementas
6.	<topic>	Temų elementas
7.	<subjectIdentity>	Temos sukurtas Objektas
8.	<baseName>	Temos Pagrindinis vardas
9.	<baseNameString>	Pagrindinio vardo Eilučių konteineris
10.	<variant>	Alternatyvi Pagrindinio vardo forma
11.	<variantName>	Alternatyvių vardų konteineris
12.	<parameters>	Apdorotas kontekstas Alternatyviems vardams
13.	<association>	Temų sąsaja
14.	<member>	Temą sąsajos narys
15.	<roleSpec>	Nurodo temą, kuri veikia kaip sąsajos Vaidmuo
16.	<occurrence>	Ištekliai laikomi Pasirodymui
17.	<resourceRef>	Nuoroda į šaltinius
18.	<resourceData>	Išteklų duomenų konteineris
19.	<mergeMap>	Apjungimas su kitu teminiu tinklu

3 lentelė. XTM elementų tipai [4].

Skirtingiems elementų tipams naudojami tie patys atributai. 4 lentelėje paminėti keletas iš jų, bei pateikti jų paaiškinimai.

Eil. Nr.	Atributai	Paaiškinimas
1.	<i>id</i>	Unikalus elemento numeris
2.	<i>xlink:type</i>	Nuorodos tipas
3.	<i>xlink:href</i>	Suteikia URI nuorodai
4.	<i>xmlns</i>	Vardo aplinkos identifikatorius Xlink vardo aplinkai pagal nutylėjimą
5.	<i>xmlns:xlink</i>	Vardo aplinkos identifikatorius Xlink vardo aplinkai
6.	<i>xml:base</i>	Nuoroda į dokumento pagrindinį URI

4 lentelė. Elementų atributai [4].

XMT sintaksės panaudojimo pavyzdys (3 pav.). Sujungiamas teminis tinklas „vaidina.xtm“ su esamu teminiu tinklu. Pridedamos temos „Šekspyras“ ir „drama“ į visų charakteristikų, esančių „vaidina.xtm“ teminiame tinkle, sritį:

```
<mergeMap xlink:href="http://www.sekspyras.org/vaidina.xtm">
  <topicRef xlink:href="#šekspyras"/>
  <topicRef xlink:href="#drama"/>
</mergeMap>

<topic id="šekspyras">
  ...
</topic>

<topic id="drama">
  ...
</topic>
```

3 pav. *XTM* kalba parašytas temų apjungimas [4].

XTM prieš *RDF/XML*, *CXTM*, *LTM*, *TM/XML* turi privalumą - ji yra labai paprasta ir greit įsisavinama. Tačiau ji vis tiek yra pakankamai išbaigta ir pilna, kad turėtų efektyvų funkcionalumą.

Kaip palyginimą paimkime Linijinį Teminio Tinklo (*LTM*) žymėjimą. Kaip ir *XTM* bei *XML* jis yra tekstinio pavidalo teminio tinklo standartas, tačiau skirtingai nei *XTM* jis yra dar kompaktiškesnis.

LTM gali būti parašytas bet kuriuo teksto redaktoriumi ir tvarkomas bet kuria teminio tinklo programine įranga, kuri jį palaiko. Taip pat *LTM* gali būti konvertuojamas į *XML* formatą, kuris yra palaikomas šios programinės įrangos.

XML teminio tinklo duomenų formatas yra pateiktas taip, kad jis būtų lengvai suprantamas ir vartotojui būtų lengva sukurti programinę įrangą. Tačiau šis formatas turi minusų - jis eiliniam vartotojui nepatogus skaityti ir rašyti. Tam yra sukurti specializuoti teminių tinklų redaktoriai, kurie suteikia vartotojui draugiškesnę vartotojo sąsają.

Nepaisant to, kad egzistuoja redaktoriai, vis tiek yra sričių, kur reikia trumpai ir aiškiai reikšti teminių tinklų konstrukcijas. Tai yra elektroninis paštas, e-diskusijos ir panašios sritys. Toks formatas taip pat leidžia lengvai ir greitai kurti ir palaikyti mažus teminius tinklus demonstravimui ir asmeniniams tikslams. Dėl patogumo, kita kalba parašyti standartinės formos teminiai tinklai keičiami į *XTM 1.0* sintaksę.

2.5.3 Įrankis Ontopia

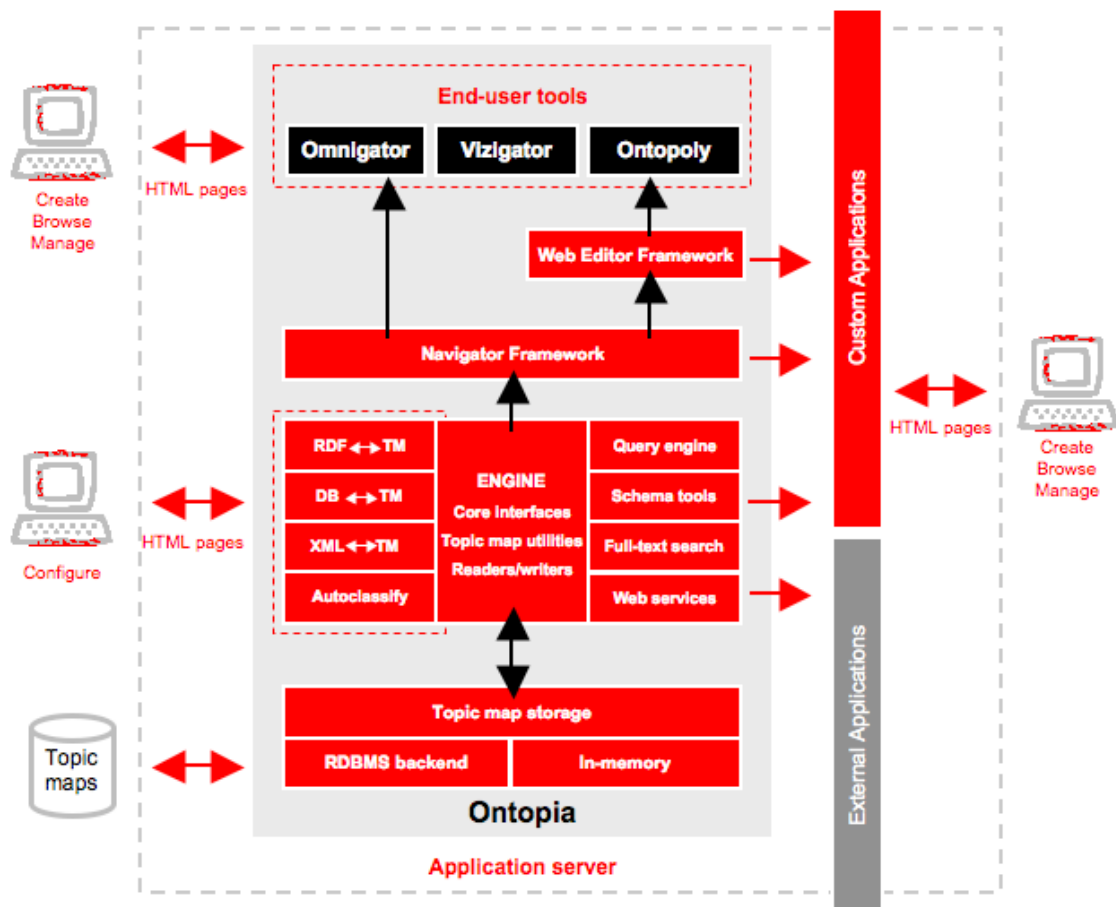
Teminiams tinklams kurti yra daug įrankių. Čia remiantis literatūros šaltiniu [17] plačiau paanalizuotas įrankis *Ontopia*, kuris yra patogus ir labai tinkamas kurti teminį tinklą natūralios kalbos užklausoms.

Ontopia yra įrankių rinkinys, kuriame yra viskas, ko reikia sukurti visą teminiu tinklu pagrįstą aplikaciją. Naudojant *Ontopia* galima:

- sukurti savo ontologiją ar globalų teminį tinklą rankiniu būdu ir/arba automatiškai,
- kurti vartotojo sąsają,
- rodyti grafinį teminio tinklo pavaizdavimą ir daug daugiau.

Iš esmės *Ontopia* yra variklis, kuris saugo ir palaiko teminius tinklus ir turi didelį *Java API* rinkinį. Jo pagrindu yra sukurta daugelis papildomų komponentų, kurie parodyti žemiau pateiktoje schemeje (4 pav.).

Ontopia pilnai naudoja tik *Java* ir veikia bet kokioje operacinėje sistemoje, kuri palaiko *Java 1.5*. Ji yra visiškai atviro kodo ir gali būti naudojama be jokių apribojimų. Norint ją naudoti reikia web serverio (rekomenduojama naudoti *Apache 2.0*, nes jis yra nemokamas).



4 pav. *Ontopia* veikimo schema.

4 pav. schemeje matome, kaip sudarytas *Ontopia* įrankis. Pagrindinė teminio tinklo dalis – širdis yra variklis, nes nuo jo priklauso saugojimo ir prieigos prie teminių tinklų sistema. Iš esmės, variklio *Java API* rinkinys skirtas:

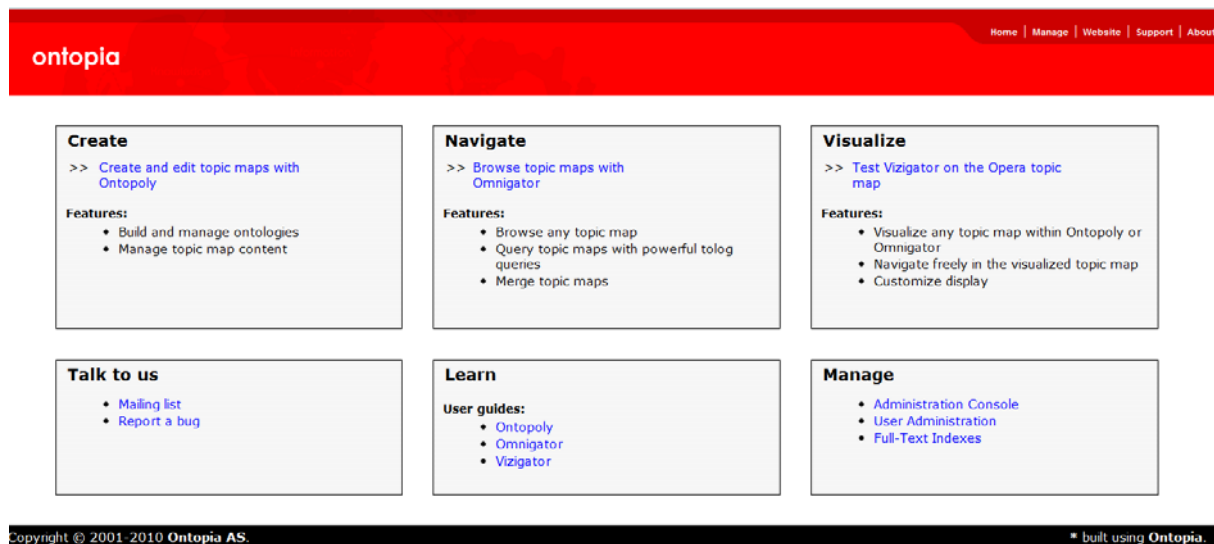
- Prieigai ir keitimui bet kuriai teminio tinklo daliai;

- Teminių tinklų įkėlimui (tinka XTM 1,0, 2,0, CTM, TM / XML, LTM formatai);
- Teminių tinklų eksportavimui į dokumentą (naudojami XTM 1,0, 2,0, TM / XML, LTM formatai);
- Konvertavimui iš RDF į teminius tinklus (ir atvirkščiai);
- Užklausų vykdymui, naudojant TOLOG užklausų kalbą;
- Scenos pilno teksto paieškai.

Variklis gali turėti teminius tinklus atmintyje arba laikyti juos reliacinėse duomenų bazėse. *API* variklis abiem atvejais tas pats, taigi administruojant *Ontopia* reikia tik pakeisti nustatymus administravimo aplinkoje. Reliacinės duomenų bazės palaiko:

- "Oracle" 8 arba naujesnę versiją;
- "Microsoft SQL Server 2005 ar naujesnį;
- PostgreSQL 7.4 ar naujesnį;
- MySQL 5.0 arba naujesnę versiją;
- h2 1.1 arba naujesnę versiją.

Galutinio vartotojo sąsaja ir įrankiai yra matomi pagrindiniame *Ontopia* valdymo lange (5 pav.).



5 pav. Pagrindinis *Ontopia* valdymo langas.

Pirmos trys funkcijos yra teminio tinklo kūrimui (*Ontopoly*), navigacijai (*Omnigator*) ir atvaizdavimui (*Vizigator*).

Ontopoly (6 pav.) yra teminių tinklų redaktorius, kuris leidžia laisvai kurti savo teminio tinklo ontologiją, naudojant patogią web sąsają. Vartotojas per vartotojo sąsają užpildo Teminį tinklą norima informacija. Taigi, *Ontopoly* sudaro galimybę sukurti teminį tinklą ir saugoti jį dokumente arba duomenų bazėje. Vartotojo redagavimo aplinkoje galimos konfigūracijos ir nustatymai. Tai labai paprasta ir lengvai suprantama sąsaja, todėl sumažėja

netvarkos ir painiavos tikimybė. Be to, *Ontopoly* yra integruojamas su kitomis turinio valdymo sistemomis. Integracijos metu *Ontopoly* įgauna tų sistemų galimybes redaguoti teminius tinklus.

The screenshot shows the 'Ontopoly' web application interface. At the top, there's a navigation bar with 'Signed in as', 'Sign out', 'Home', 'Manage', 'Website', 'Support', and 'About'. Below that, the 'ontopoly' logo is on the left, and 'Editing Jill's Music' with 'Open...', 'Save', and 'Find' buttons is on the right. The main content area is titled 'A Hard Day's Night' and has a 'Help' icon. It features several sections: 'Name' (A Hard Day's Night), 'Subject Identifier' (http://psi.example.org/AHardDaysNightAlbum), 'Description' (Classic stuff. Easily the best of the "early" Beatles. It's one of the most romantic albums by The Beatles. If you want to have a sweet, cozy night with your soulmate.), and 'Contains track' (listing songs like 'A Hard Day's Night', 'And I Love Her', etc.). On the right, there's a 'Topic types' section (Album), 'Views' (Default), and several action buttons like 'Create new Album', 'Delete this topic', and 'Browse this topic in Omnigator'. At the bottom, there's a footer with 'Version 4.1.0 Copyright © 2006-2009 Ontopia AS.' and 'Ontopoly, Ontopia's Topic Maps Editor, built using the OKS.'

6 pav. Vartotojo sąsajos temos redagavimo pavyzdys, kai šios temos tipas yra "Album" (kaip CD įrašymas).

Omnigator (7 pav.), tai technologija, skirta peržiūrėti teminius tinklus. Jis taip pat plačiai naudojamas kaip teminių tinklų klaidų taisymo ir prototipų gamybos priemonė. *Omnigator* turi RDF palaikymą, todėl jį galima vadinti įvairios paskirties semantiniu *Web agent*.

Omnigator yra gana universalus įrankis, nes juo galima apdoroti visus duomenis, kurie atitinka teminio tinklo formatą: *XTM 1.0*, *XTM 2.0*, *LTM* arba *RDF*.

Be peržiūros *Omnigator* leidžia su teminiais tinklais (*TM*) leidžia atlikti daug daugiau funkcijų. Keletas iš jų:

- gebėjimas juos sujungti on the fly;
- ieškoti juose informacijos - "Google" pakaitalas;
- eksportuoti skirtingo formato *TM*;
- pritaikyti teminius tinklus skirtingiems požiūriams;
- kurti *TM* poaibius priklausomai nuo to, kokia yra jų taikymo sritis;
- atlikti semantinius teminio tinklo patikrinimus.

Home | Manage | Website | Support | About

Browsing ItalianOpera.tsm
 Open... Reload Not Indexed

Italian Opera Topic Map | Customize | Filter | Export | Merge | Statistics | Query | Edit | No schema | Vizigate

Tosca

Type(s): Opera

Untyped Names (1)

- Tosca

Subject Identifiers (1)

- <http://psi.ontopedia.net/Tosca>

Associations (21)

- Based on**
 - La Tosca
- Composed by**
 - Puccini, Giacomo
- Contains**
 - E lucevan le stelle
 - O dolci mani
 - Recondita armonia
 - Va, Tosca!
 - Vissi d'arte
- Dramatis personae**
 - Baron Scarpia
 - Cesare Angelotti
 - Floria Tosca
 - Gaoler
 - Mario Cavaradossi
 - Sacristan
 - Sciarone
 - Shepherd-boy
 - Spoletta
- First performed at**
 - Teatro Costanzi
- Libretto by**
 - Giacosa, Giuseppe

Internal Occurrences (2)

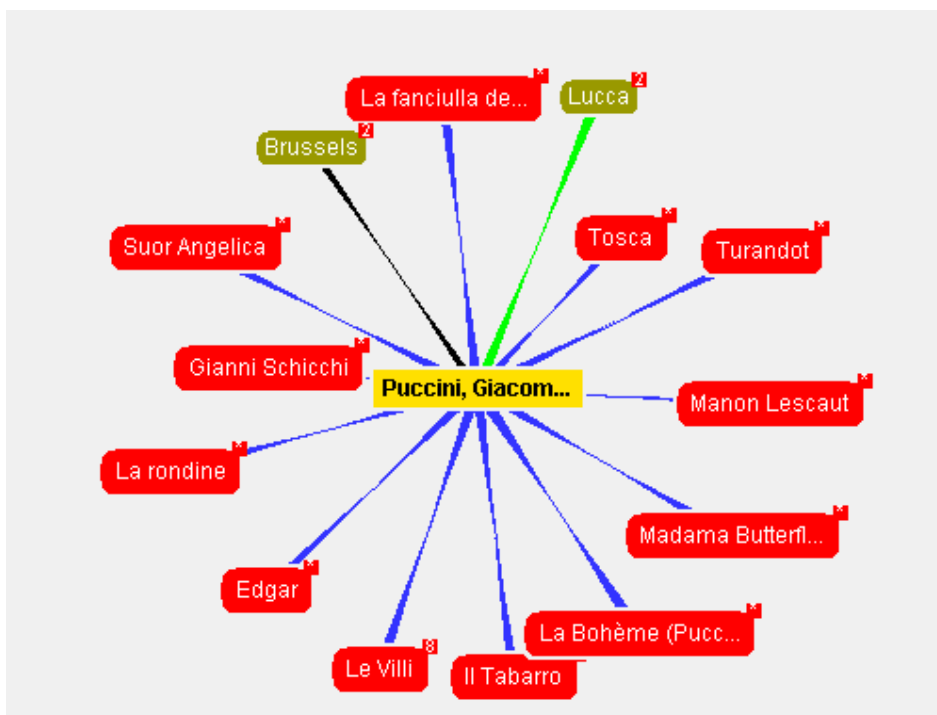
- Première date**
 - 1900-01-14
- Audio recording**
 - 7 47175 8

External Occurrences (16)

- Article**
 - <http://en.wikipedia.org/wiki/Tosca>
 - <http://www.ontopia.net/topicmaps/examples/opera/occurs/snl/tosca.htm> - Scope: *Store Norske Leksikon*; *Web*
- Illustration**
 - <http://localhost:8080/ItalianOpera/occurs/opera/puccini/tosca-poster1.jpg> - Scope: *Local*
- Libretto**
 - http://dante.di.unipi.it/ricerca/libretti/Tosca_1899.html - Scope: *Web*
 - http://dante.di.unipi.it/ricerca/libretti/Tosca_1900.html - Scope: *Web*
 - <http://localhost:8080/operamap/occurs/tosca-libretto.htm> - Scope: *Local*
 - <http://opera.stanford.edu/opera/Puccini/Tosca/libretto.html> - Scope: *Opera Glass*; *Web*
- Poster**
 - http://localhost:8080/operamap/occurs/tosca_poster.htm - Scope: *Local*
 - http://www.r-ds.com/opera/pucciniana/pictures/tosca_poster.htm - Scope: *OperaResource*; *Web*

7 pav. Omnigator aplinka.

Labiausiai vizuali nauja *Omnigator* funkcija yra *Vizigator* (8 pav.), kuri atvaizduoja teminius tinklus vizualiai. *Vizigator* yra pasiekiamas per *Vizigate plug-in* funkciją, paspaudus ant bet kurios temos puslapio. Dėka *RDF* palaikymo *Vizigator* buvo papildytas *RDF2TM plug-in* funkcija, kuri leidžia konfigūruoti teminio tinklo atvaizdavimą.



8 pav. Ontologijos atvaizdavimas su *Vizigator funkcija*.

Ontopia įrankis pasirinktas dėl to, kad yra lengvai integruojamas su daugeliu turinio valdymo sistemų. Tai suteikia galimybę teminiams tinklams panaudoti šių turinio valdymo sistemų funkcijas - organizuoti ir naviguoti turinį. Keletas turinio vadymo sistemų, kurios integruotos su *Ontopia*:

- Escenic;
- OfficeNet žinių portalas.

Taipogi *Ontopia* įrankyje yra viskas vienoje vietoje: teminių tinklų kūrimas, navigacija ir atvaizdavimas. Daugelis kitų įrankių neturi visų šių savybių, dažniausiai jie skirti kažkuriai tai vienai ar dviems paminėtoms funkcijoms atlikti. Taigi teminių tinklų programinės įrangos įrankiai, kurie yra nemokami ir prieinami galutiniams vartotojams, yra riboti: *Ontopia*, *Empolis*, *Mondeca* ir kiti. 5 lentelėje pavaizduota keletos įrankių ir kalbų suvestinė - jų savybės ir galimybės.

Įrankis/kalba	Web pagrindas	Kalba	Nemokamas prienamumas	Importavimas	Užklauso	Atvaizdavimas
Omnigator	X	Java	X	X		
Ontopia	X	Java	X	X	X	X
Empolis	X	Java		X		X
TM4J	X	Java	X	X	X	
tmproc	X	Python	X	X	X	
SemanText	X	Python				
Tm4Jscript	X	JavaScript	X	X	X	
tinyTIM	X	Java	X	X		
xSiteable	X	XSTL				X
V Topic Map Browser	X	Python	X			X
UNIVIT						X
Panckoucke	X	Java	X			X
TM4Web/Velocity	X	Java	X			X
TM3D		OpenGL XML				X
ThinkGraph		XML				X
Hypergraph	X	Java	X			X
TMNav	X	Java	X		X	X
Simple Topic Map		Perl	X	X		
TMTab	X	Java	X	X		
LTM		XML		X		
mapalizer		Perl	X	X		
XTM		XML		X		

5 lentelė. Teminių tinklų kūrimo įrankiai, kalbos [6].

Matome, kad pasirinktas įrankis *Ontopia* yra pranašesnis už kitus lyginamus. Jis turi visus minimus komponentus: web pagrindas, nemokamas prienamumas, importavimas, užklauso, atvaizdavimas.

2.5.4 TOLOG užklauso

Analizuojant TOLOG užklauso buvo remtasi literatūros šaltiniais [10] ir [15], kuriuose išsamiai paaiškinta TOLOG sintaksė ir galimybės.

Ontopia įrankyje viena iš naudotinų kalbų yra TOLOG, kuri labai patogi užklauso rašyti. Tolog kalba buvo sukurta pagal *Datalog* (*Prolog* pogrupis) ir SQL kalbas. Ši kalba skirta rašyti užklauso ieškant informacijos teminiuose tinkluose, panašiai kaip reliacinėse duomenų bazėse gali būti ieškoma tam tikrų duomenų naudojant SQL užklauso. Tolog kalba parašytomis užklausomis galima ieškoti įvairių dalykų. Keletas iš ieškomų dalykų:

- visų temų tipus;
- atitinkamo tipo visų temų vardus tam tikroje srityje;
- visus temų, naudojamų kaip asociacijos vaidmuo, tipus;
- visas asociacijas, turinčias daugiau nei du vaidmenis.

Užklauso naudoja operatorius *AND*, *OR*, *SELECT*, *NOT*, *OPTIONAL* bei *SQL* funkcijas, tokias kaip projektavimas, skaičiavimas, rūšiavimas ir informacijos paieška.

TOLOG užklauso sudaromos naudojant predikatus. Jie yra tryjų rūšių: *user-defined* (deklaracija), *built-in* (palyginimo predikatai) ir *dynamic* (sudaryti teminiame tinkle iš įvykių ir sąsajų tipų).

Paprasčiausias *TOLOG* užklauso sudarymo būdas:

instance-of(\$C, composer)?

Čia naudojamas *built-in* predikatas *instance-of*, kuris saugo šios užklauso objekto savybes. Sekantis argumentas *\$C* yra teminio tinklo kintamasis, kuris nurodo *composer*. Užklauso rezultatas yra visos kintamojo *\$C* reikšmės, kurios atitinka užklauso klausimą.

AND operatorius išreiškiamas kablelio ženklu:

instance-of(\$C, composer), born-in(\$C : person, \$P : place)

Čia *born-in* yra dinaminis predikatas.

Operatorius *OR* išreiškiamas tokiu simboliu „|“:

*instance-of(\$C, composer), born-in(\$C : person, \$P : place),
{ located-in(\$P : containee, norway : container) |*

located-in(\$P : containee, sweden : container) }

SELECT operatorius išrenka norimas reikšmes naudojant žodelį *select*:

select \$C from
instance-of(\$C, composer), born-in(\$C : person, \$P : place),
{ located-in(\$P : containee, norway : container) |
located-in(\$P : containee, sweden : container) }

Operatorius *NOT* išrenka tuos elementus, kurie neturi paminėtos savybės po žodelio *not*:

select \$C from
instance-of(\$C, composer), born-in(\$C : person, \$P : place),
not(located-in(\$P : containee, italy : container))

Operatorius *OPTIONAL* išreiškiamas per riestinius skliaustus *{}*. Jis naudojamas tada, kai pavyzdžiui reikia rasti įrašų *composer* predikatus *birthdate*, kuriuos turi ne visi įrašai:

instance-of(\$C, composer), { birthdate(\$C, \$D) }?

Rašant užklausas patartina naudoti predikatų indeksus – kintamuosius *\$kintamasis*:

instance-of(\$P, person) order by \$P limit 5?

Tai yra patogesnis ir trumpesnis užrašymas.

Dar vienas svarbus dalykas rašant TOLOG užklausas yra taisyklių sudarymas. Šios taisyklės būtinai turi būti paremtos teminio tinklo objektais. Šiomis taisyklėmis sukuriama virtualios asociacijos tarp teminio tinklo temų. Pavyzdys:

used-work-by(\$A, \$B) :- composed-by(\$OPERA, \$A),
based-on(\$OPERA, \$WORK),
written-by(\$WORK, \$B).

Iš šio pavyzdžio matome, kad kintamasis A naudojamas B darbe, jeigu kintamasis A parašė operą, kuri yra paremta raštišku kintamojo B darbu. Naudojant šią taisyklę galima užrašyti tokią užklausą: *used-work-by(puccini, \$A)*. Rezultatas būtų toks:

['A': 'dante'],
{'A': 'belasco'},
{'A': 'sardou'},
{'A': 'musset'},
{'A': 'murger'},
{'A': 'gold'},
{'A': 'prevost'},
{'A': 'gozzi'}]

Naujos taisyklės gali būti paremtos jau esamomis taisyklėmis.

Dažnai tenka susidurti su situacija, kad pagal užklausą atrenkama daug besikartojančių rezultatų. Kad to būtų išvengta, naudojamas simbolis \neq . Pavyzdys:

born-in(\$A, \$B), born-in(\$C, \$B), \$A \neq \$C.

Užrašas $\$A \neq \C nustato, kad atrinkti A ir C (gimę B vietovėj) nesikartotų.

Kaip matome, *TOLOG* užklausų kalba yra gana lanksti. Tai patogu, nes jos galimybės leidžia atlikti paiešką teminiame tinkle pagal didžiąją dalį natūralios kalbos užklausų.

2.5.1 Natūralios kalbos užklausos

Natūralios kalbos užklausos, jų poreikis ir panaudojimas išanalizuotas remiantis literatūros šaltiniais [1], [8], [9], [13].

Technologijos laikui bėgant tobulėja, o ypatingai greitai – informacinės technologijos. Taip pat neatsilieka ir kalba, kuri naudojama žinių perdavimui, komunikavimui. Kalba – tai lingvistinių ženklų grupė, svarbiausia žmonių bendravimo priemonė, atliekanti komunikacinę. XX a. mokslo pasiekimai suteikė galimybę naudoti kalbą ne tik tiesioginiam žmonių bendravimui, bet ir perkelti ją į kompiuterius.

Semantinė paieška iš pirmo žvilgsnio niekuo nesiskiria nuo įprastos „Google“ paieškos, tačiau ji yra kur kas sudėtingesnė. Semantinės technologijos yra painūs ir sunkiai suprantami dariniai.

Kai ko nors teminiame tinkle ieškoma, į paieškos laukelį įvedamas žodis ar frazė ir tikimasi, kad paieškos sistema teisingai supranta vartotojo informacijos poreikius. Tačiau

dažnai gaunami rezultatai, kuriuos reikia gebėti įvertinti ir pasirinkti tinkantį iš daugiau nei kelių ar keliasdešimt tūkstančių galimų.

Semantinėmis technologijomis paremta paieška patikslina, ko iš tikrųjų vartotojas ieško, ir randa tikslesnių rezultatų. Semantinės technologijos yra sumanesnės, nes analizuoja ieškomos informacijos prasmę, ryšius su kitais žodžiais ir bendresnėmis kategorijoms. Tačiau kol kas jos reikalauja, kad vartotojas rašytų užklausas specialia užklausų kalba arba kitaip prisidėtų formuodamas užklausą ir palaiapsniui pasirinkdamas rezultatus.

Galutiniam vartotojui dažnai tenka susidurti su informacijos paieška teminiame tinkle. Tačiau atlikti paiešką apsunkina tai, kad vartotojo sąsajos mažai pritaikytos rašyti užklausą natūralia kalba, arba yra iš viso labai primityvios. Taip yra todėl, kad susiduriama su nemaža problema – natūralios kalbos dviprasmybėmis. Be to, sistemos negali pilnai prisitaikyti prie naujų sričių. Jas reikia nuolat atnaujinti papildant žinių bazę.

2.5.2 Natūralios kalbos klausimų analizė TOLOG užklausų kalbos kontekste

Kadangi šiame darbe natūralios kalbos užklausos bus transformuojamos į TOLOG užklausas, reikia panagrinėti kuo įvairesnius natūralios kalbos klausimų ir jiems atitinkamų TOLOG užklausų variantus. Klausimai sudaromi pirmiausia atsižvelgiant į klausimą „Ko gali vartotojas ieškoti teminiame tinkle ir kokie galimi jo klausimai?“. Klausimai pagal įvairovę suskirstomi atsižvelgiant į TOLOG užklausų ypatumus bei teminio tinklo sandarą. Taigi užklausos suskirstytos pagal tokius aspektus:

1. Tema arba vaidmuo;
2. Konkretūs egzemplioriai;
3. Kelios asociacijos;
4. Abstraktūs egzemplioriai;
5. Laikas, laiko intervalas (naudojami pasirodymai);
6. Neiginiai.

Toliau bus paanalizuoti klausimų suskirstymo kriterijai ir pateikti pavyzdžiai: natūralios kalbos klausimas, atitinkama TOLOG užklausa bei užklausos rezultatas. Užklausos vykdomos pavyzdiniame anktesniame 2.6.1 skyriuje aprašytame teminiame tinkle „Parodos“ (1 pav.).

2.5.2.1 Temos ir vaidmens naudojimas užklausoje

Atsižvelgiant į teminio tinklo sandarą aišku, kad tema ir vaidmuo yra du skirtingi dalykai. Taigi, analizuojant ribotos lietuvių kalbos klausimą, reikia atsižvelgti į tai, ko klausiama – ar apie temą, ar apie temos vaidmenį. Teminiame tinkle „Parodos“ tema „Paroda“ atlieka

vaidmenį „Renginys“, bet juk šis vaidmuo gali būti naudojamas ne tik parodomis, bet ir pavyzdžiui dainų šventėms, nes renginys pagal reikšmę apima jas abi ir dar daugiau. Vadinasi sudarant TOLOG užklausą reikia tai atskirti. Pavyzdžiui, galimi klausimai: „Rasti renginius, kuriuos suorganizavo organizatoriai“ ir „Rasti parodas, kurias suorganizavo organizatoriai“. Jei rašoma „renginys“, tai reikia ieškoti visų temų, kurios naudoja šį vaidmenį, o jeigu rašoma „paroda“, tada tik apsiribojama konkrečiai parodomis, kurios naudoja vaidmenį „Renginys“. Beje iš pavyzdinio klausimo „Rasti renginius, kuriuos **suorganizavo** organizatoriai“ aišku, kad TOLOG užklausoje bus naudojama asociacija „(kas?) suorganizavo (ką?)“. Taipogi organizatorius gali būti ir tema, ir vaidmuo, nes „Parodos“ teminiame tinkle sutampa temos ir vaidmens pavadinimai. Taigi pateikti keturi pavyzdiniai klausimai, TOLOG užklausoje ir rezultatai. Kadangi organizatoriaus vaidmenį naudoja ir tema „Organizatorius“, ir tema „Įmonė“, tai antro klausimo atveju rezultatai apima tuos renginių egzempliorius (šiuo atveju tik parodos), kuriuos suorganizavo ir „Organizatorius“, ir „Įmonė“ egzemplioriai. Pirmo klausimo atveju randami tik organizatorių (tema „Organizatorius“) suorganizuoti renginiai (egzemplioriai).

1. Rasti renginius(vaidmuo), kuriuos suorganizavo organizatoriai(tema).

TOLOG užklausa	Rezultatas			
<pre>select \$RENGINYS from value(\$ASSOCNAME, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$TNAME2, "Organizatorius"), topic-name(\$TOPICTYPE2, \$TNAME2), instance-of(\$TOPIC2, \$TOPICTYPE2), value(\$ROLE11, "Renginys"), topic-name(\$ROLENAME1, \$ROLE11), type(\$ROLE1, \$ROLENAME1), role-player(\$ROLE1, \$RENGINYS), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2?</pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; padding: 5px;">RENGINYS</td> </tr> <tr> <td style="text-align: center; padding: 5px;">Smilgų šokis</td> </tr> <tr> <td style="text-align: center; padding: 5px;">Ruduo</td> </tr> </table>	RENGINYS	Smilgų šokis	Ruduo
RENGINYS				
Smilgų šokis				
Ruduo				

2. Rasti renginius(vaidmuo), kuriuos suorganizavo organizatoriai(vaidmuo).

TOLOG užklausa	Rezultatas
<pre>select \$RENGINYS from value(\$ASSOCNAME, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Renginys"), topic-name(\$ROLENAME1, \$NAME1), type(\$ROLE1, \$ROLENAME1), value(\$NAME2, "Organizatorius"), topic-name(\$ROLENAME2, \$NAME2), type(\$ROLE2, \$ROLENAME2), role-player(\$ROLE1, \$RENGINYS), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/=\$ROLE2?</pre>	<div data-bbox="1149 275 1349 548" style="border: 1px solid black; padding: 5px;"> <p>RENGINYS</p> <p><u>Smilgų šokis</u></p> <p><u>Kaukė</u></p> <p><u>Ruduo</u></p> </div>

3. Rasti parodas(tema), kurias suorganizavo organizatoriai(tema).

TOLOG užklausa	Rezultatas
<pre>select \$TOPIC1 from value(\$ASSOCNAME, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Paroda"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1), value(\$NAME2, "Organizatorius"), topic-name(\$TOPICNAME2, \$NAME2), instance-of(\$TOPIC2, \$TOPICNAME2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/=\$ROLE2?</pre>	<div data-bbox="1149 1169 1349 1373" style="border: 1px solid black; padding: 5px;"> <p>TOPIC1</p> <p><u>Smilgų šokis</u></p> <p><u>Ruduo</u></p> </div>

4. Rasti parodas(tema), kurias suorganizavo organizatoriai(vaidmuo).

TOLOG užklausa	Rezultatas				
<pre>select \$TOPIC1 from value(\$ASSOCNAME, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Paroda"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1), value(\$NAME2, "Organizatorius"), topic-name(\$ROLENAME2, \$NAME2), type(\$ROLE2, \$ROLENAME2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2?</pre>	<table border="1"> <tr><td>TOPIC1</td></tr> <tr><td><u>Smilgų šokis</u></td></tr> <tr><td><u>Kaukė</u></td></tr> <tr><td><u>Ruduo</u></td></tr> </table>	TOPIC1	<u>Smilgų šokis</u>	<u>Kaukė</u>	<u>Ruduo</u>
TOPIC1					
<u>Smilgų šokis</u>					
<u>Kaukė</u>					
<u>Ruduo</u>					

2.5.2.2 Konkretūs egzemplioriai

1-3 klausimai parodo atvejus, kai paieška vyksta pagal konkretų temos arba vaidmens pavadinimą. 5 klausime organizatorius yra tema, todėl nieko nerandama, nes temos „Kauno parodų organizatoriai“, kurios tipas yra „Organizatorius“ nesuorganizavo nei vieno parodos. Jei „Kauno parodų organizatoriai“ imami ne kaip tema, o kaip vaidmuo, tai randamas rezultatas paroda „Miegas“, nes vaidmuo „Organizatorius“ apima temas „Organizatorius“ ir „Įmonė“. Taigi randama paroda, kuria suorganizavo įmonė „Kauno parodų organizatoriai“. 7 klausime jau nesvarbu koks yra „Kauno parodų organizatoriai“ tipas, todėl randamos visos parodos, kurias suorganizavo tiesiog to pavadinimo objektas.

1. Rasti parodas(tema), kurias suorganizavo organizatorius(tema) „Kauno parodų organizatoriai“.

TOLOG užklausa	Rezultatas	
<pre>select \$TOPIC1 from value(\$ASSOCNAME, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Paroda"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1),</pre>	<table border="1"> <tr><td>TOPIC1</td></tr> </table>	TOPIC1
TOPIC1		

<pre> value(\$NAME2, "Organizatorius"), topic-name(\$TOPICNAME2, \$NAME2), instance-of(\$TOPIC2, \$TOPICNAME2), value(\$NAME3, "Kauno parodų organizatoriai"), topic-name(\$TOPICNAME3, \$NAME3), \$TOPIC2 = \$TOPICNAME3, role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2? </pre>	
--	--

2. Rasti parodas(tema), kurias suorganizavo organizatorius(vaidmuo) "Kauno parodų organizatoriai".

TOLOG užklausa	Rezultatas
<pre> select \$TOPIC1 from value(\$ASSOCNAME, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Paroda"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1), value(\$NAME2, "Organizatorius"), topic-name(\$ROLENAME2, \$NAME2), type(\$ROLE2, \$ROLENAME2), value(\$NAME3, "Kauno parodų organizatoriai"), topic-name(\$TOPICNAME3, \$NAME3), \$TOPIC2 = \$TOPICNAME3, role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2? </pre>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">TOPIC1</div> <div style="border: 1px solid black; padding: 5px;">Miegas</div>

3. Rasti parodas, kurias suorganizavo "Kauno parodų organizatoriai".

TOLOG užklausa	Rezultatas
<pre> select \$TOPIC1 from value(\$ASSOCNAME, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), </pre>	<div style="border: 1px solid black; padding: 5px;">TOPIC1</div>

<pre> type(\$ASSOC, \$ASSOCTYPE), value(\$NAME1, "Paroda"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1), value(\$NAME2, "Kauno parodų organizatoriai"), topic-name(\$TOPIC2, \$NAME2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2? </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Miegas</div>
---	---

2.5.2.3 Kelios asociacijos

Klausimuose galima klausti panaudojant ne tik vieną vaiksmažodį, todėl TOLOG užklausa sudaromos iš kelių asociacijų. 1-5 klausimuose esmė ta pati, kaip ir 2.5.2.2 skyriaus 1-7 klausimuose, tik čia daugiau apribojimų.

1. Rasti parodas, kurias suorganizavo organizatorius(vaidmuo), kuris sudarė sutartį su įmone(vaidmuo).

TOLOG užklausa	Rezultatas
<pre> select \$TOPIC1 from value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$TOPICNAME1, "Paroda"), topic-name(\$TOPICTYPE1, \$TOPICNAME1), instance-of(\$TOPIC1, \$TOPICTYPE1), value(\$TOPICNAME2, "Organizatorius"), topic-name(\$TOPICTYPE2, \$TOPICNAME2), type(\$ROLE2, \$TOPICTYPE2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), value(\$ASSOCNAME2, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$TOPICNAME3, "Įmonė"), </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">TOPIC1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">Ruduo</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Smilgų šokis</div>

<pre> topic-name(\$TOPICTYPE3, \$TOPICNAME3), type(\$ROLE3, \$TOPICTYPE3), role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), \$ROLE1/= \$ROLE2, \$ROLE3=\$ROLE4? </pre>	
---	--

2. Rasti parodas, kurias suorganizavo organizatorius(vaidmuo), kuris sudarė sutartį su įmone(tema) "Kauno paveikslų galerija".

TOLOG užklausa	Rezultatas
<pre> select \$TOPIC3 from value(\$ASSOCNAME1, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAM21, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$TOPICNAME1, "Įmonė"), topic-name(\$TOPICTYPE1, \$TOPICNAME1), type(\$ROLE1, \$TOPICTYPE1), value(\$TOPICNAME2, "Kauno paveikslų galerija"), topic-name(\$TOPIC1, \$TOPICNAME2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2, value(\$TOPICNAME3, "Paroda"), topic-name(\$TOPICTYPE3, \$TOPICNAME3), instance-of(\$TOPIC3, \$TOPICTYPE3), value(\$TOPICNAME4, "Organizatorius"), topic-name(\$TOPICTYPE4, \$TOPICNAME4), type(\$ROLE4, \$TOPICTYPE4), role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), </pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> TOPIC3 </div>

\$ROLE3/=\$ROLE4?	
-------------------	--

3. Rasti parodas, kurias suorganizavo organizatoriai(vaidmuo), kurie sudarė sutartį su patalpa(tema).

TOLOG užklausa	Rezultatas			
<pre>select \$TOPIC3 from value(\$ASSOCNAME1, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAM21, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$TOPICNAME1, "Patalpa"), topic-name(\$TOPICTYPE1, \$TOPICNAME1), instance-of(\$TOPIC1, \$TOPICTYPE1), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/=\$ROLE2, value(\$TOPICNAME3, "Paroda"), topic-name(\$TOPICTYPE3, \$TOPICNAME3), instance-of(\$TOPIC3, \$TOPICTYPE3), value(\$TOPICNAME4, "Organizatorius"), topic-name(\$TOPICTYPE4, \$TOPICNAME4), type(\$ROLE4, \$TOPICTYPE4), role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), \$ROLE3/=\$ROLE4?</pre>	<table border="1"> <tr> <td>TOPIC3</td> </tr> <tr> <td>Smilgų šokis</td> </tr> <tr> <td>Ruduo</td> </tr> </table>	TOPIC3	Smilgų šokis	Ruduo
TOPIC3				
Smilgų šokis				
Ruduo				

4. Rasti parodas, kurias suorganizavo organizatorius(tema), kuris sudarė sutartį su patalpa(tema) "Vilniaus paveikslų galerija".

TOLOG užklausa	Rezultatas		
<pre>select \$TOPIC3 from value(\$ASSOCNAME1, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1),</pre>	<table border="1"> <tr> <td>TOPIC3</td> </tr> <tr> <td>Smilgų</td> </tr> </table>	TOPIC3	Smilgų
TOPIC3			
Smilgų			

<pre> type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAM21, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$TOPICNAME1, "Patalpa"), topic-name(\$TOPICTYPE1, \$TOPICNAME1), instance-of(\$TOPIC1, \$TOPICTYPE1), value(\$TOPICNAME2, "Vilniaus paveikslų galerija"), topic-name(\$TOPIC11, \$TOPICNAME2), \$TOPIC1=\$TOPIC11, role-player(\$ROLE1, \$TOPIC11), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2, value(\$TOPICNAME3, "Paroda"), topic-name(\$TOPICTYPE3, \$TOPICNAME3), instance-of(\$TOPIC3, \$TOPICTYPE3), value(\$TOPICNAME4, "Organizatorius"), topic-name(\$TOPICTYPE4, \$TOPICNAME4), type(\$ROLE4, \$TOPICTYPE4), role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), \$ROLE3/= \$ROLE4? </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Šokis</div>
---	--

5. Rasti parodas, kurias suorganizavo organizatorius, kuris sudarė sutartį su “Klaipėdos paveikslų galerija”.

TOLOG užklausa	Rezultatas
<pre> select \$TOPIC3 from value(\$ASSOCNAME1, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAM21, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$TOPICNAME1, "Klaipėdos dailės galerija"), topic-name(\$TOPIC1, \$TOPICNAME1), </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">TOPIC3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ruduo</div>

<pre> role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2, value(\$TOPICNAME3, "Paroda"), topic-name(\$TOPICTYPE3, \$TOPICNAME3), instance-of(\$TOPIC3, \$TOPICTYPE3), value(\$TOPICNAME4, "Organizatorius"), topic-name(\$TOPICTYPE4, \$TOPICNAME4), type(\$ROLE4, \$TOPICTYPE4), role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), \$ROLE3/= \$ROLE4? </pre>	
--	--

2.5.2.4 Abstraktūs egzemplioriai

Čia parodytas atvejas, kai klausiami abstrakčių dalykų (13 klausimas). Pvz.: parodas suorganizavo bet kas arba įmonė sudarė sutartį su bet kuo. Tai fiksuojama TOLOG užklausoje kaip asociacija nenurodant vaidmenų (1 klausimas).

1. Rasti parodas(tema), kurias suorganizavo **bet kas**, kuris turi sutartį su **bet kuo**.

TOLOG užklausa	Rezultatas			
<pre> select \$TOPIC1 from value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAME2, "(kas?) sudarė sutartį (su kuo?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), value(\$NAME1, "Paroda"), topic-name(\$ROLENAME1, \$NAME1), instance-of(\$TOPIC1, \$ROLENAME1), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC1, \$ROLE1), association-role(\$ASSOC1, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), </pre>	<table border="1"> <tr> <td>TOPIC1</td> </tr> <tr> <td>Ruduo</td> </tr> <tr> <td>Smilgų šokis</td> </tr> </table>	TOPIC1	Ruduo	Smilgų šokis
TOPIC1				
Ruduo				
Smilgų šokis				

<pre> \$ROLE1/=\$ROLE2, role-player(\$ROLE3, \$TOPIC3), association-role(\$ASSOC2, \$ROLE3), association-role(\$ASSOC2, \$ROLE4), role-player(\$ROLE4, \$TOPIC2), \$ROLE3/=\$ROLE4? </pre>	
--	--

2.5.2.5 Laikas, laiko intervalas

Čia paanalizuoti atvejai, kai klausiama apie temos pasirodymą – data, valanda arba atvaizdas (1-2 klausimai).

1. Rasti parodas, kurios vyko "2011-07-14".

TOLOG užklausa	Rezultatas		
<pre> select \$TOPIC1 from occurrence(\$TOPIC1, \$OCC), type(\$OCC, \$OCCURENCE), value(\$OCC, "2011-07-14"), value(\$OCCNAME, "Paroda įvyko"), topic-name(\$OCCURENCE, \$OCCNAME)? </pre>	<table border="1"> <tr><td>TOPIC1</td></tr> <tr><td>Kaukė</td></tr> </table>	TOPIC1	Kaukė
TOPIC1			
Kaukė			

2. Rasti parodas, kurios vyko nuo 2011-07-14 iki 2011-09-01 (laikas-pasirodymas).

TOLOG užklausa	Rezultatas					
<pre> select \$TOPIC1 from occurrence(\$TOPIC1, \$OCC), type(\$OCC, \$OCCURENCE), value(\$OCCNAME, "Paroda įvyko"), topic-name(\$OCCURENCE, \$OCCNAME), \$OCC1 >= "2011-07-14", \$OCC1 <= "2011-09-01", value(\$OCC, \$OCC1)? </pre>	<table border="1"> <tr><td>TOPIC1</td></tr> <tr><td>Miegas</td></tr> <tr><td>Ruduo</td></tr> <tr><td>Smilgų šokis</td></tr> <tr><td>Kaukė</td></tr> </table>	TOPIC1	Miegas	Ruduo	Smilgų šokis	Kaukė
TOPIC1						
Miegas						
Ruduo						
Smilgų šokis						
Kaukė						

2.5.2.6 Neiginiai

Čia parodyti atvejai, kai reikia rasti informaciją nurodant temą ar vaidmenį, su kuriuo atsakymas kaip tik turi būti nesujęs (1-2 klausimai).

1. Rasti renginius, kurie vyko ne "Kauno paveikslų galerijoje".

TOLOG užklausa	Rezultatas			
<pre>select \$TOPIC2 from value(\$ASSOCNAME, "(kas?) vyko (kur?)"), topic-name(\$ASSOCTYPE, \$ASSOCNAME), type(\$ASSOC, \$ASSOCTYPE), not(value(\$TOPICNAME1, "Kauno paveikslų galerija"), topic-name(\$TOPIC1, \$TOPICNAME1)), value(\$TOPICNAME2, "Renginys"), topic-name(\$TOPICTYPE2, \$TOPICNAME2), type(\$ROLE2, \$TOPICTYPE2), role-player(\$ROLE1, \$TOPIC1), association-role(\$ASSOC, \$ROLE1), association-role(\$ASSOC, \$ROLE2), role-player(\$ROLE2, \$TOPIC2), \$ROLE1/= \$ROLE2?</pre>	<table border="1"> <tr><td>TOPIC2</td></tr> <tr><td>Ruduo</td></tr> <tr><td>Smilgų šokis</td></tr> </table>	TOPIC2	Ruduo	Smilgų šokis
TOPIC2				
Ruduo				
Smilgų šokis				

2. Rasti kūrinius(vaidmuo), kurie eksponuojami parodose(tema), kurias suorganizavo ne organizatoriai(tema).

TOLOG užklausa	Rezultatas										
<pre>select \$TOPIC3 from value(\$ASSOCNAME1, "(kas?) suorganizavo (ką?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$ASSOCNAME2, "(ką?) eksponavo (kur?)"), topic-name(\$ASSOCTYPE2, \$ASSOCNAME2), type(\$ASSOC2, \$ASSOCTYPE2), not(value(\$TOPICNAME1, "Organizatorius"), topic-name(\$TOPICTYPE1, \$TOPICNAME1), instance-of(\$TOPIC1, \$TOPICTYPE1)), value(\$TOPICNAME2, "Paroda"), topic-name(\$TOPICTYPE2, \$TOPICNAME2), instance-of(\$TOPIC2, \$TOPICTYPE2),</pre>	<table border="1"> <tr><td>TOPIC3</td></tr> <tr><td><u>Migla</u></td></tr> <tr><td><u>Vakarėja</u></td></tr> <tr><td><u>Žvaigždžių vartai</u></td></tr> <tr><td><u>Naujas</u></td></tr> <tr><td><u>Muilo burbulai</u></td></tr> <tr><td><u>Panašumas</u></td></tr> <tr><td><u>Kaspiniai</u></td></tr> <tr><td><u>Kalnai</u></td></tr> <tr><td><u>Kaminkrėtys</u></td></tr> </table>	TOPIC3	<u>Migla</u>	<u>Vakarėja</u>	<u>Žvaigždžių vartai</u>	<u>Naujas</u>	<u>Muilo burbulai</u>	<u>Panašumas</u>	<u>Kaspiniai</u>	<u>Kalnai</u>	<u>Kaminkrėtys</u>
TOPIC3											
<u>Migla</u>											
<u>Vakarėja</u>											
<u>Žvaigždžių vartai</u>											
<u>Naujas</u>											
<u>Muilo burbulai</u>											
<u>Panašumas</u>											
<u>Kaspiniai</u>											
<u>Kalnai</u>											
<u>Kaminkrėtys</u>											

```

value($TOPICNAME3, "Kūrinys"),
topic-name($TOPICTYPE3, $TOPICNAME3),
type($ROLE3, $TOPICTYPE3),

role-player($ROLE1, $TOPIC1),
association-role($ASSOC1, $ROLE1),
association-role($ASSOC1, $ROLE2),
role-player($ROLE2, $TOPIC2),
$ROLE1/=$ROLE2,
role-player($ROLE3, $TOPIC3),
association-role($ASSOC2, $ROLE3),
association-role($ASSOC2, $ROLE4),
role-player($ROLE4, $TOPIC2),
$ROLE3/=$ROLE4?

```

2.5.3 Įrankis LifeRay Portal

Įrankio Liferay trumpa analizė pateikta remiantis literatūros šaltiniu [11], nes šį įrankį galima integruoti su Ontopia ir panaudoti kurti vartotojo sąsajai. Tuo galima pasinaudoti praplečiant ir patobulinant šio tiriamojo darbo realizaciją.

Liferay Enterprice Portal yra sukurtas 2000-taisiais metais, bet savo dabartinį pavadinimą gavo tik 2004-taisiais. Tai nemokama atviro kodo programa, kuriai tinka šūkis – „Neužtenka, kad sprendimas tiesiog veiktų – jis turi veikti būtent tau.“ („A solution shouldn't just work—it should work for you“). Tai daug ką pasako apie šio įrankio funkcionalumą ir galimybę jį universaliai pritaikyti kiekvienam vartotojui pagal poreikius. Kuriant šiuo įrankiu vartotojo sąsają galima individualizuoti portalo puslapius. Tai daroma pridėdant, pašalinant bei koreguojant turinį arba portalo išvaizdos atributus. Šie puslapiai gali būti vieši arba privatūs, priklausomai nuo vartotojo pageidavimo.

Taigi Liferay portalas leidžia vartotojams lengvai susikurti įprastą internetinę svetainę. Galima pridėti vartotojo registraciją, *captcha*, dokumentų biblioteką, lucene pagrįstą paiešką, *wiki* platformas, naujienų kanalus, viešą pokalbių forumą ir dar daug kitokių savybių. *LifeRay Portal* turi įskiepi, kurio dėka ir galima vartotojams pridėti naujas funkcijas ir pritaikyti Liferay elgesį bei išvaizdą. Dėl šio įskiepio, praplečiamumo ir modulinės konstrukcijos, *LifeRay* kartais apibūdinamas kaip turinio valdymo sistema arba interneto taikymo sistema. *LifeRay* galima išplėsti iki kelių programavimo kalbų, įskaitant *PHP* ir *Ruby*. Šis

portalas sukurtas Java pagrindu, todėl veikia bet kuriame kompiuteryje, kur instaliuota *Java* ir yra severio palaikymas (pavyzdžiui serveris *Apache Tomcat*).

2.6 Sistemos vartotojų analizė

2.6.1 Vartotojų aibė, tipai ir savybės

Natūralios kalbos užklausų vartotojas, tai bet kuris natūralią kalbą naudojantis žmogus, kuriam yra poreikis rasti duomenis teminiame tinkle. Tai gali būti eilinis kokios nors informacinės sistemos vartotojas, įmonės darbuotojas ar net koks nors mokslininkas, bet ne informatikos srities specialistas. Visi jie susiję tuo, kad:

- Naudoja natūralią kalbą;
- Ieško tam tikros informacijos teminiame tinkle;
- Neturi žinių apie TOLOG užklausų kalbą;
- Jiems svarbi pati ieškoma informacija ir jie nenori aiškintis užklausų kalbos;
- Nori greitos ir patogios informacijos paieškos.

2.6.2 Vartotojų tikslai ir problemos

Vartotojas jau įpratęs naudoti tokias paieškos sistemas kaip „Google” ar „Yahoo”. Bet jis nemąsto, kaip veikia tokios paieškos sistemos. Taip yra todėl, kad tai jam nerūpi, lygiai kaip nerūpi, kas yra kompiuterio viduje, kaip ir kodėl jis veikia. Šiuos klausimus sprendžia tos srities specialistai, o vartotojas tiesiog tai naudoja.

Galutiniams sistemos vartotojams svarbiausia yra patogumas ir galimybė rasti informaciją teminiame tinkle be papildomo specialaus išsilavinimo. Tam pasiekti naudojamos sudėtingos semantinės technologijos.

Semantinės technologijos vartotojui turi būti paprastos, kad jis neturėtų atlikti papildomų varginančių funkcijų, bet gautų pakankamai tikslius rezultatus. Semantinių technologijų infrastruktūra vartotojui turi būti nematoma. Tačiau tai įmanoma tik tada, kai semantinės technologijos ims apdoroti natūralios kalbos struktūras, kas kol kas yra sunkiai pasiekama [12].

Taigi pagrindinė problema ir yra ta, kad nėra aiškiai suformuluotų taisyklių kaip turi būti suformuotas teminis tinklas, kuriame būtų įmanoma paieška natūralia kalba, nėra įrankių, kurie turėtų sąsają, leidžiančią formuluoti užklausas teminiame tinkle tokia forma, kuri artima natūraliai kalbai, ir kuri nepriklauso nuo dalykinės srities. Vadinasi, vartotojams reikia specialiai pritaikytos natūraliai kalbai semantinės grafinės sąsajos.

2.7 Esamų sprendimų analizė

Esamų sprendimų analizė atlikta remiantis literatūros šaltiniais [2], [3], [6].

Natūralios kalbos užklausoms interpretuoti ir konvertuoti į atitinkamą užklausų kalbos sintaksę yra sukurta daug priemonių – natūralios kalbos vartotojo sąsajos. Tačiau esmė tame, kad jos pritaikytos tik anglų kalbai ir yra priklausomos nuo dalykinės srities.

NLI (natūralios kalbos sąsajos) yra tinkamas ir patogus sprendimas, kad vartotojas lengvai ir greitai ieškotų informacijos teminiame tinkle. Todėl *NLI* yra tikrai naudingos galutiniams vartotojams. Daugumos su įrankiais ir kalbomis atliekamų eksperimentų tiklas yra kuo tikslesni paieškos rezultatai, tačiau svarbiau yra galimybė naudoti užklausas kuo paprasčiau. Taigi būtent tam yra sukurta įvairių patogių sąsajų. Išsamesnei analizei pasirinktos keturios populiarios ir plačiai naudojamos natūralios anglų kalbos sąsajos (NLI): *NLP-Reduce*, *Querix*, *Ginseng* ir *Semantic Crystal*.

NLP-Reduce yra pakankamai paprasta ir intuityvi natūralios kalbos vartotojo sąsaja. Joje natūralios kalbos užklausa priimama kaip trumpesnis žodžių rinkinys. Ieškoma atitikmenų pagal morfologiją ir sinonimus. Ši sąsaja leidžia vartotojui įvesti raktažodžius, sakinių fragmentus ir net pilnus angliškus sakinius. Pirmiausia iš užklaunos pašalinami *stopwords*, o tada nagrinėjama likusių žodžių sandara. Sistema analizuoja tryjų žodžių struktūrą. Nagrinėjami struktūros žodžiai sulyginami su turimais sinonimais, kurie paimti iš *OWL* žinių bazės. Tada ši struktūra paverčiama į *SPARQL* užklausą. Atlikus paiešką ontologijoje, vartotojui parodomi rezultatai ir šiek tiek su rezultatu susijusios statistikos. *NLP-Reduce* turi didelį sinonimų bagažą, kuris paimtas iš *WordNet*. Taigi vartotojas gali naudoti kuo įvairiausius žodžius sudarinėdamas užklausą.

Querix yra tokia natūralios kalbos vartotojo sąsaja, kurioje galima naudoti pilnus angliškus klausimus. Tačiau ši sistema visai nesprendžia natūralios kalbos dviprasmybių. Ji tiesiog pateikia vartotojui vienareikšmiškus užklaunos analogus ir klausia, kuris iš jų atitinka užduotą klausimą. Taigi iš esmės vartotojas yra tiesioginis šios sistemos konsultantas. *Querix* naudojamas analizatorius, kurio sintasės medyje ir jo šakose ieškoma analzuojamos užklaunos sinonimų (sinonimai imami iš *WordNet*) [4]. Tada sudaroma *SPARQL* sinatkse atitinkanti užklausa.

Pavyzdžiui *Querix* pateikta užklausa – „Kokia yra didžiausia Amerikos valstija?“. Dviprasmybė tame, kad valstija gali būti didžiausia pagal gyventojų skaičių arba pagal plotą. Vartotojui iššoka langas, leidžiantis pasirinkti iš dviejų variantų, pagal ką reikia atrinkti valstiją kaip didžiausią. Nuo vartotojo pasirinkimo priklauso galutinis rezultatas [7].

Ginseng yra vedlio kontroliuojama natūralios kalbos vartotojo sąsaja. Ši kontroliuojama įvedimo kalba panaši į anglų kalbą savo sintakse ir sakinių sudarymu. Esmė tokia: vartotojui pradėjus rašyti užklausą iš karto iššoka langas, kuris pasiūlo rašomo žodžio pabaigą, o po to galimą sekantį žodį. Kitokios žodžių formos, kurios neįeina į pasiūlymą, nepriimamos. Taigi

tokiu būdu *Ginseng* apsisaugo nuo natūralios kalbos dviprasmybių. Naudojama *OWL* žinių bazė. Vartotojo užklausa transformuojama į *SPARQL* užklausa. Gautas rezultatas pateikiamas vartotojui sąsajos lange.

Ginseng vos ją paleidus, iš karto sudaro taisykles, kaip galima įvesti užklausa, panaudodama esamą žinių bazę. Taigi ši vartotojo sąsaja lengvai perkeliama ir prisitaikanti.

Semantic Crystal natūralios kalbos vartotojo sąsaja yra pagrįsta grafiniais vaizdais. Ji naudoja labai formalią ir griežtai apribotą įvedimo kalbą. Taigi tam, kad vartotojas suprastų kaip reikia sudaryti užklausa, pateikiama vizuali schema. Ši schema sudaryta ir apskritimų ir vektorių. Tai grafinis ontologijos atvaizdavimas. Vartotojas naudodamas pavaizduotą ontologiją, sudarinėja sistemai tinkamas užklausa. Jis paspaudžia ant pasirinktos klasės ir iš karto išsiskleidžia sąrašas, kuriame pateikti galimi užklausa variantai. Pasirinkus norimą klausimą, sistema generuoja *SPARQL* užklausa, kuri iš karto matoma vartotojo sąsajos lange. Jei viskas tinkamai pasirinkta, tai rezultatas vartotojui pateikiamas naujame sąsajos lange.

Taigi išanalizavus visas keturias natūralios kalbos vartotojo sąsajas, galima apibendrinti. Visų jų tikslas yra vartotojo patogumas, visos jos naudoja *SPARQL* užklausa kalbą, bet skiriasi pagal šiuos kriterijus: dviprasmybių sprendimo būdai, įvedimo kalba, rezultatų tikslumas. Pateikta 6 lentelė:

Vartotojo sąsaja/Vertinimo kriterijus	Užklausa pateikimo kalba	Dviprasmybių sprendimas	Rezultatų tikslumas
<i>NLP-Reduce</i>	Natūrali anglų kalba (raktažodžiai, žodžių junginiai, pilni sakiniai).	Analizuojami sakiniai, jų reikšmė.	Gali būti nelabai tikslūs dėl kalbos dviprasmybių.
<i>Querix</i>	Natūrali anglų kalba (raktažodžiai, žodžių junginiai, pilni sakiniai).	Pateikiami dialogo langai, kad vartotojas pasirinktų vieną iš galimų dviprasmybės interpretavimo variantų.	Pakankamai tikslūs, nes vartotojas pats pasirenka kaip išspręsti dviprasmybę.
<i>Ginseng</i>	Kontroliuojama anglų kalba (pateikiama pasirinkti leidžiama sintaksė).	Neleidžia įvesti netinkamų, dviprasmiškų klausimų, pasiūlant leidžiamas žodžių	Tikslūs, bet nebūtinai tokie, kokių norėjo vartotojas, nes klausimų

		formas ir sakinių sudarymo sintaksę.	sudarymas kontroliuojamas.
<i>Semantic Cristal</i>	Grafinė ontologija, pasirenkamos užklausos.	Dviprasmybės negalimos, nes užklausos sudaromos pagal grafinę ontologiją pasirenkant pateiktus galimus jau suformuotus klausimus.	Tiklūs, jei vartotojas iš pateiktų klausimų randa sau tinkamą.

6 lentelė. NK vartotojo sąsajų palyginimas.

Kiekviena sąsaja reikalauja kitokios įvedimo užklausų kalbos dėl savo lankstumo, natūralumo ir formalumo: nuo raktažodžių iki užbaigtų anglišku sakinių, nuo meniu punktų iki galimybių grafiškai atvaizduoti užklausų kalbą.

Patogių naudoti nuo dalykinės srities nepriklausomų sprendimų praktiškai nėra, o esami yra labai primityvūs – ieško visų temų, kuriose yra paminėti paieškos raktiniai žodžiai.

2.8 Siekiamas sprendimas

Kaip jau buvo minėta, problema yra ta, kad kol kas nėra patogios vartotojo sąsajos, kurioje būtų galima naudoti natūralios kalbos užklausas.

Šiame darbe siekiama:

1. suformuluoti taisyklės teminio tinklo elementų vardų formulavimui;
2. suformuluoti taisyklės ir sukurti algoritmą TOLOG užklausų sudarymui;
3. realizuoti įvardintas taisyklės ir TOLOG užklausų sudarymo iš natūralios kalbos algoritmą.

Užklausoms užrašyti bus naudojama struktūrizuota natūrali kalba be žodžių kaitymo ir derinimo. Algoritmo pagalba natūralios kalbos užklausos bus verčiamos į TOLOG užklausas ir pateikiamos paieškai teminame tinkle. Šis sprendimas bus nepriklausomas nuo dalykinės srities, kur paieška bus atliekama naudojant paieškos žodžių ryšius su kitais teminiame tinkle esančiais žodžiais. Natūralios kalbos dviprasmybėms spręsti bus panaudota Ginseng idėja – jos nebus sprendžiamos, o bus apeitos. Skirtumas, kad Ginseng atveju pateikiamos galimos pasirinkti žodžių formos ir sakinių struktūros, o šiame darbe bus pasinaudojama vartotojo sąsajos ypatumais – struktūrizuotas klausimo formavimo būdas. Tai leis kontroliuoti įvedamus vartotojo klausimus ir gauti juos norimomis formomis, vadinasi bus leidžiama įvesti tik teisingas, sukurtai sistemai priimtinas užklausas.

Užlausos pateikimo kalba: struktūrizuota natūrali kalba.

Dviprasmybių sprendimas: vartotojo sąsajos leidžiamas klausimo formavimo būdas ribos vartotoją užduoti dviprasmišką klausimą, ir tuo pačiu padės suformuoti korektiškus klausimus – bus galima klausti tik to, kas yra teminiame tinkle. O tai turėtų užtikrinti ir atsakymų tikslumą.

Rezultatų tikslumas: rezultatai tiksūs tiek, kiek vartotojui tiksliai pagal savo norus pavyksta pateikti klausimą vadovaujantis natūralios kalbos užklausų sudarymo taisyklėmis.

2.9 Nuo dalykinės srities nepriklausomo TOLOG užklausų sudarymo analizės išvados

Atlikus esamų sprendimų analizę nustatyta, kad:

1. Vartotojams paieška teminiame tinkle yra nepatogi, nes nepritaikyta natūraliai kalbai;
2. Išanalizuotos natūralios kalbos vartotojo sąsajos pritaikytos paieškai tik anglų kalba;
3. Ginseng sistemos naudojamą natūralios kalbos kontroliavimo idėją galima panaudoti išvengti natūralios kalbos dviprasmybėms spręsti.
4. Reikia kurti struktūrizuotos natūralios kalbos vartotojo sąsają ir sudaryti natūralios kalbos užklausų formavimo, teminio tinklo temų ir asociacijų įvardinimo taisykles. Tokiu principu veikianti vartotojo sąsaja ir iš dalies patenkins vartotojo poreikį natūraliai kalbai, ir bus išvengta natūralios kalbos sudėtingos sintaksės analizės.

Išanalizavus teminio tinklo ypatumus, formalių užklausų kalbas, teminio tinklo ir vartotojo sąsajos kūrimo įrankius, sprendimui realizuoti pasirinkta *XTM 1.0* kalba, įrankis *Ontopia*, nes:

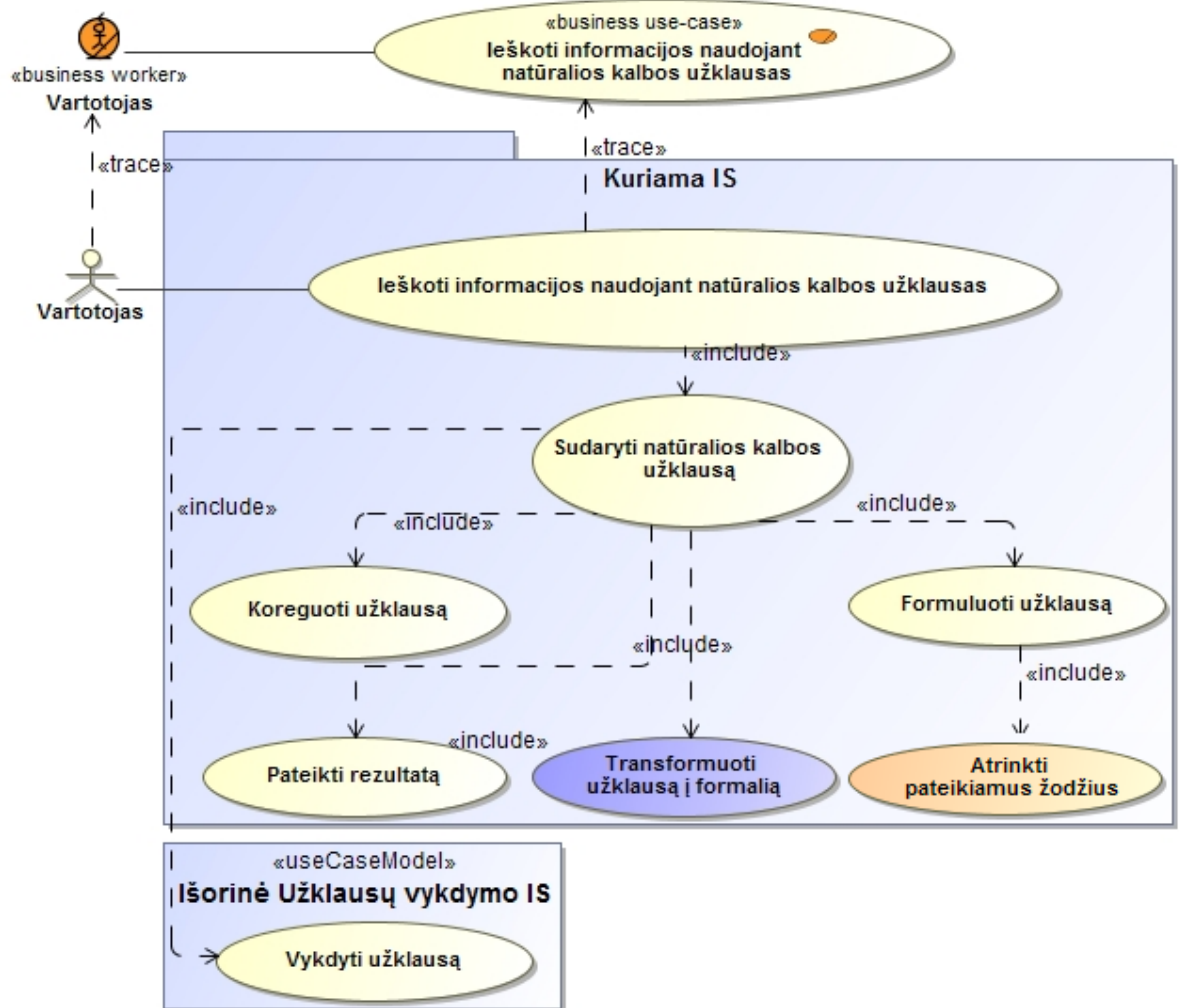
5. Teminio tinklo valdymo įrankis *Ontopia* yra visapusiškas, ko neturi kiti analizuoti įrankiai: teminio tinklo kūrimas, koregavimas ir grafinis atvaizdavimas.
6. Teminio tinklo užrašymo *XTM 1.0* kalba sukurta plačiai naudojamos XML kalbos pagrindu, yra glaustesnė ir tai yra standartas.
7. TOLOG užklausų kalba dėl savo lankstumo puikiai tiks natūralios kalbos užklausoms realizuoti.

3 Nuo dalykinės srities nepriklausomo TOLOG užklausių sudarymo sistemos reikalavimai

3.1 Sistemos reikalavimų specifikacija

3.1.1 Sistemos funkciniai reikalavimai

3.1.1.1 Kompiuterizuojamų panaudojimo atvejų diagrama



9 pav. Nuo dalykinės srities nepriklausomo TOLOG užklausių sudarymo sistemos panaudojimo atvejai.

3.1.1.2 Panaudojimo atvejų specifikacijos

Veiklos diagramos sudarytos tiems panaudojimo atvejams, kurie yra specifiniai ir esminiai kūrimos sistemos elementai: „Atrinkti pateikiamus žodžius“(10 pav.) ir „Transformuoti užklausą į formalią“(11 pav.).

Toliau 7-13 lentelėse pateiktos panaudojimo atvejų specifikacijos.

PA „Sudaryti natūralios kalbos užklausą“		
Tikslas. Greitai ir patogiai rasti informaciją teminiame tinkle		
Aprašymas. Šis PA yra tiesiogiai prieinamas vartotojui iš pagrindinio sistemos paieškos lango		
<i>Prieš sąlyga</i>	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori atlikti paiešką teminiame tinkle	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	Formuluoti užklausą Koreguoti užklausą Transformuoti užklausą į formalią Pateikti rezultatą
		Vykdyti užklausą
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas pasirinkinėja iš pateiktų iškrentančių sąrašų užklausos žodžius		Sistema po žodžio iš sąrašo pasirinkimo pateikia kitą logiškai galimų rinktis žodžių sąrašą
Vartotojas baigia PA		
<i>Po sąlyga</i>	-	
Alternatyvūs scenarijai		
1a. Jei vartotojas taiso užklausą		Sistema iš naujo pateikinėja galimų ieškoti žodžių sąrašus
Pastabos		

7 lentelė. PA „Sudaryti natūralios kalbos užklausą“ specifikacija.

PA „Koreguoti užklausą“		
Tikslas. Užklausos pakeitimas		
Aprašymas. Šis panaudojimo atvejis yra PA „Sudaryti natūralios kalbos užklausą“ dalis		
<i>Prieš sąlyga</i>	Pasirinktas bent vienas paieškos žodis	
Aktorius	Vartotojas	

Sužadinimo sąlyga	Vartotojas nori pakoreguoti užklausą	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas pasirenka kitus žodžius iš pateiktų sąrašų		Sistema iš naujo pateikinėja galimų ieškoti žodžių sąrašus
Vartotojas baigia PA		-
<i>Po sąlyga</i>		-
Alternatyvūs scenarijai		
Pastabos		
Vartotojas gali koreguoti užklausą kol ji nėra įvykdyta.		

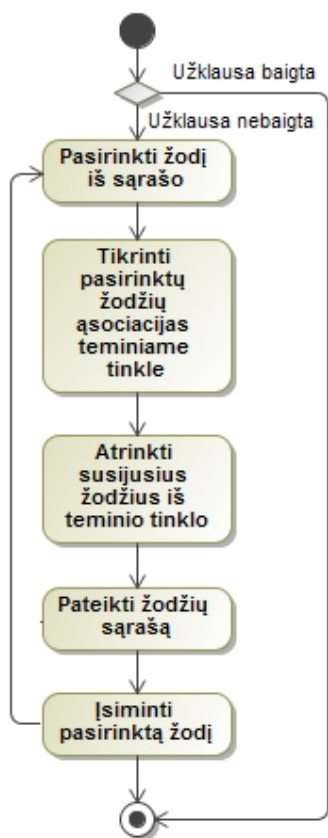
8 lentelė. PA „Koreguoti užklausą“ specifikacija.

PA „Formuluoti užklausą“		
Tikslas. Teisingai įvesti norimą užklausą paieškai		
Aprašymas. Šis PA yra tiesiogiai prieinamas vartotojui iš pagrindinio sistemos paieškos lango. Jis vykdomas kai sudaroma užklausa.		
<i>Prieš sąlyga</i>	Yra pasirinktas bent vienas užklauskos žodis	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori teisingai sudaryti užklausą	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	Sudaryti natūralios kalbos užklausą
	Apima PA	Atrinkti pateikiamus žodžius
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas pasirenka žodį iš pateikto sąrašo		Sistema pagal pasirinktą žodį atrenka iš teminio tinklo ir pateikia vartotojui loginėmis sąsajomis susijusių žodžių sąrašą
Vartotojas baigia PA		-
<i>Po sąlyga</i>		-
Pastabos		
Formulavimas vyksta tol, kol vartotojas sudarinėja užklausą		

9 lentelė. PA „Formuluoti užklausą“ specifikacija.

PA „Atrinkti pateikiamus žodžius“		
Tikslas. Patekti užklauso žodžių sąrašus		
Aprašymas. Šis panaudojimo atvejis yra PA „Formuluoti užklausa“ dalis		
Prieš sąlyga	Yra pasirinktas užklauso žodis iš pateikto sąrašo	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori pamatyti tolimesnių galimų pasirinkti užklauso žodžių sąrašą	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	Formuluoti užklausa
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas nori pamatyti tolimesnių galimų pasirinkti užklauso žodžių sąrašą		Sistema pagal pasirinktą žodį atrenka iš teminio tinklo loginėmis sąsajomis susijusius atitinkamus žodžius ir juos pateikia iškrentančio sąrašo pavidalu
1.	Sistema baigia PA	
Po sąlyga	Atrinkti žodžiai pateikiami iškrentančio sąrašo pavidalu paieškos lange	
Pastabos		
Atrinkinėjimas vykdomas po kiekvieno žodžio pasirinkimo		

10 lentelė. PA „Atrinkti pateikiamus žodžius“ specifikacija.



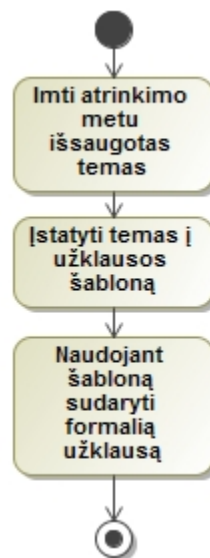
10 pav. Panaudojimo atvejo „Atrinkti pateikiamus žodžius“ veiklos diagrama.

10-tame paveiksle pateikta panaudojimo atvejo „Atrinkti pateikiamus žodžius“ veiklos inkle. Joje išsamiau ir tiksliau pavaizduota, kaip sistema vykdo temų atrinkimą iš teminio tinklo pagal vartotojo pasirinktus lietuvių kalbos užklauso žodžius. Pasirinkus žodį iš sąrašo vykdoma susijusių temų paieška teminiame tinkle. Ši paieška vykdoma atsižvelgiant į prieš tai pasirinktų įsimintų žodžių logines asociacijas. Atrinkti žodžiai pateikiami iškrentančio sąrašo pavidalu. Pasirinktas žodis įsimenamas. Visi įsiminti žodžiai po to naudojami sudaryti TOLOG užklausą.

PA „Transformuoti užklausą į formalią“		
Tikslas. Suformuoti TOLOG užklausą iš lietuvių kalbos užklauso		
Aprašymas. Šis panaudojimo atvejis yra PA „Sudaryti natūralios kalbos užklausą“ dalis		
Prieš sąlyga	Yra sudaryta teisinga natūralios kalbos užklausa ir paspaustas mygtukas „leškoti“	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori, kad sistema pagal jo klausimą rastų tinkamus atsakymus	
Veiklos taisyklės		
Susiję panaudojimo	Išplečia PA	Sudaryti natūralios kalbos užklausą

atvejai	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas paspaudžia mygtuką „Ieškoti“		Sistema paima užklauso formulavimo metu įsimintus žodžius(temas) ir įdeda į transformavimo šabloną. Tada suformuoja TOLOG užklausą ir pateikia išorinei užklausų vykdymo sistemai
Sistema baigia PA		
Po sąlyga	Natūralios kalbos užklausa transformuota į TOLOG užklausą	
Pastabos		
-		

11 lentelė. PA „Transformuoti užklausą į formalią“ specifikacija.



11 pav. Panaudojimo atvejo „Transformuoti užklausą į formalią“ veiklos diagrama.

11-tame paveiksle PA „Transformuoti užklausą į formalią“ pateiktoje veiklos diagramoje pavaizduota, kaip vykdoma lietuvių kalbos užklauso transformacija į formalią TOLOG užklausą. Iš atrinktų žodžių suformuotų sąrašų pasirinkami ir įsimenami žodžiai, o paspaudus „Ieškoti“ jie sudedami į paruoštą transformacijos šabloną. Šio šablono pagalba sugeneruojama TOLOG užklausa.

PA „Vykdėti užklausą“	
Tikslas. Įvykdyti paiešką teminiame tinkle	
Aprašymas. Šis panaudojimo atvejis yra PA „Sudaryti natūralios kalbos užklausą“ dalis	
Prieš sąlyga	Natūralios kalbos užklausa transformuota į TOLOG užklausą
Aktorius	Vartotojas
Sužadinimo sąlyga	Vartotojas nori rasti informaciją teminiame tinkle

Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	Sudaryti lietuvių kalbos užklausą
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Vartotojas laukia atsakymo		Sistema paduoda išorinei vykdymo sistemai TOLOG užklausą.
Sistema vykdo užklausą		Vykdymo sistema įvykdo paiešką teminiame tinkle ir grąžina rezultatus
Sistema baigia PA		
Po sąlyga	Užklauso atsakymas pateiktas rezultatų lange	
Pastabos		
-		

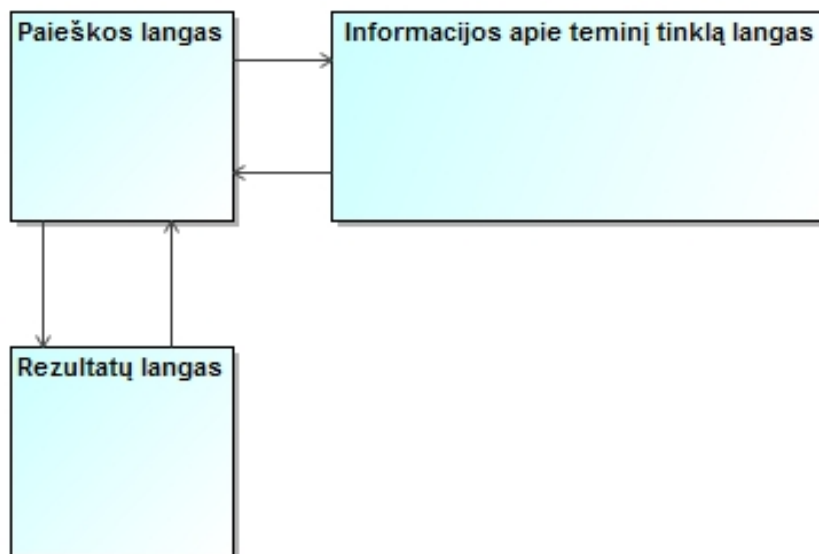
12 lentelė. PA „Vykdėti užklausą“ specifikacija.

PA „Pateikti rezultatą“		
Tikslas. Peržiūrėti paieškos rezultatus		
Aprašymas. Šis panaudojimo atvejis yra PA „Sudaryti natūralios kalbos užklausą“ dalis		
Prieš sąlyga	Įvykdyta paieška ir rezultatai pateikti rezultatų lange	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori pamatyti ieškomą informaciją	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	Sudaryti natūralios kalbos užklausą
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
Sistema pateikia rezultatą		Sistema gautus rezultato žodžius kaip sąrašą pateikia rezultatų lange
Vartotojas peržiūri rezultatų lange esančius rezultatus		
Vartotojas baigia PA		
Po sąlyga		
Alternatyvūs scenarijai		
Pastabos		

13 lentelė. PA „Pateikti rezultatą“ specifikacija.

3.1.1.3 Sistemos vartotojo sąsajos modeliai

12 paveiksle pateiktas vartotojo sąsajos navigacijos planas. Vartotojas atidaręs paieškos langą gali pereiti į rezultatų langą (įvykdžius užklausą jis atidaromas automatiškai), o iš jo – atgal. Taip pat iš paieškos lango vartotojas gali patekti į teminio tinklo atvaizdavimo langą ir grįžti atgal.



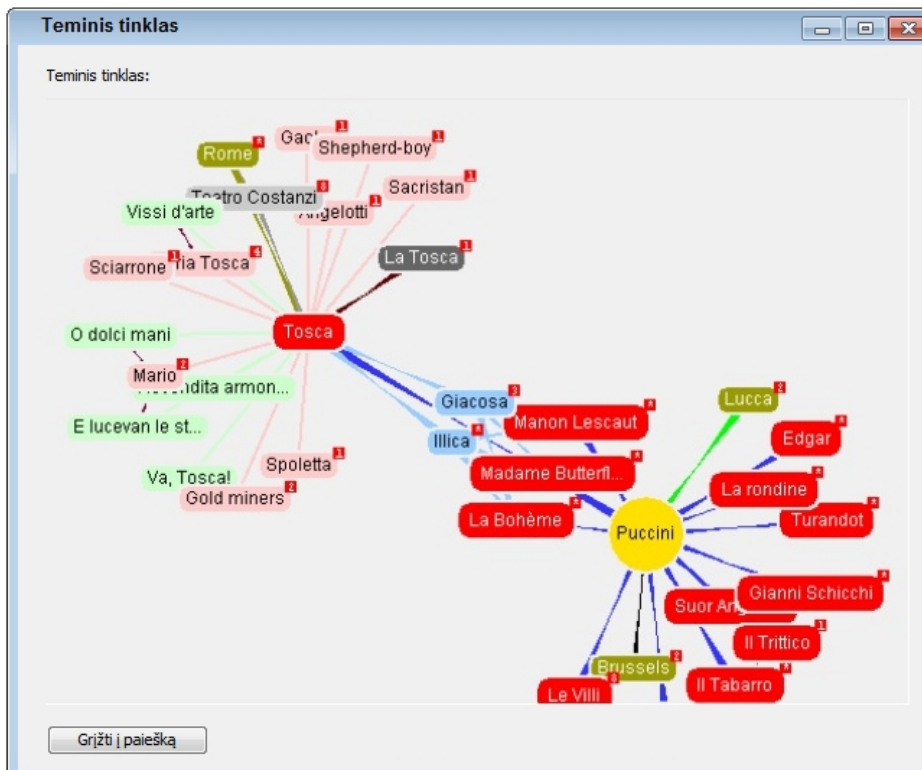
12 pav. Vartotojo sąsajos navigacijos planas.

Toliau 13-15 paveikslėliuose pateikiami vartotojo sąsajos langai: paieškos, teminio tinklo atvaizdavimo ir rezultatų.



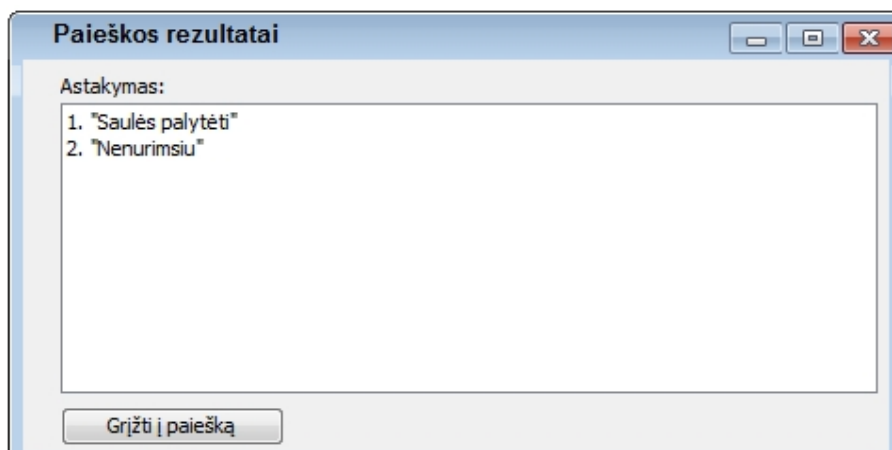
13 pav. Paieškos langas.

13 paveiksle pateiktas paieškos langas, kai yra pradėta vesti natūralios kalbos užklausa. Pradėtas vesti žodis iš p raidės ir suflerių lange pateikti galimi ieškoti žodžiai iš p raidės. Paspaudus mygtuką „Informacija apie teminį tinklą“ atidaromas teminio tinklo atvaizdavimo langas (14 pav.).



14 pav. Teminio tinklo vaizdavimo langas.

Peržiūrejus teminio tinklo informaciją, galima paspausti „Grįžti į paiešką“ ir atsidurti vėl paieškos lange.



15 pav. Rezultatų išvedimo langas.

Paieškos lange įvedus natūralios kalbos užklausą ir paspaudus mygtuką „Ieškoti“ įvykdoma paieška ir rezultatai pateikiami rezultatų lange (15 pav.). Iš rezultatų lango į paiešką grįžtama paspaudus mygtuką „Grįžti į paiešką“.

3.1.2 Sistemos nefunkciniai reikalavimai

Sistemos nefunkciniai reikalavimai pateikiami tolimesniuose skyreliuose (3.1.2.1 – 3.1.2.5) 14-18 lentelėse.

3.1.2.1 Sistemos išvaizda

Reikalavimas #1					
Aprašymas:	Vartotojo sąsaja turi būti intuityvi, funkcijos pasiekiamos iš pradinio sistemos paieškos lango				
Pagrindimas:	Vartotojui turi būti paprasta naudotis sistema				
Tinkamumo kriterijus:	Vartotojo sąsajos meniu punktų, laukų pavadinimai yra lengvai suprantami, naudojamos aiškios piktogramos, informacijos įvedimo laukai patogiai išdėstyti				

14 lentelė. Sistemos išvaizda.

3.1.2.2 Stilius

Reikalavimas #2					
Aprašymas:	Vartotojo sąsaja neturi naudoti ryškių, kontrastuojančių spalvų				
Pagrindimas:	Vartotojo neturi erzinti sistemos aplinka				
Tinkamumo kriterijus:	Vartotojo neerzina sistemos naudojamas spalvų derinys				

15 lentelė. Stilius.

3.1.2.3 Naudojimosi paprastumas

Reikalavimas #3					
Aprašymas:	Vartotojo sąsaja turi būti lengvai naudojama				
Pagrindimas:	Vartotojui turi būti paprasta naudotis sistema				
Tinkamumo kriterijus:	99% neapmokytų vartotojų pirmą kartą sėkmingai dirba su sistema				

16 lentelė. Naudojimosi paprastumas.

3.1.2.4 Vartotojui skirtos savybių ir kalbos konfigūravimo priemonės

Reikalavimas #4	
Aprašymas:	Privalomas teisingas lietuviškų simbolių apdorojimas
Pagrindimas:	Vartotojas gali naudoti bendrinę lietuvių kalbą
Tinkamumo kriterijus:	Simbolinės informacijos įvedimo laukuose galima įvesti lietuviškus simbolius. Atvaizduojant informaciją, vietoj lietuviškų simbolių nėra rodomi įvairūs netinkami simboliai.

17 lentelė. Vartotojui skirtos savybių ir kalbos konfigūravimo priemonės

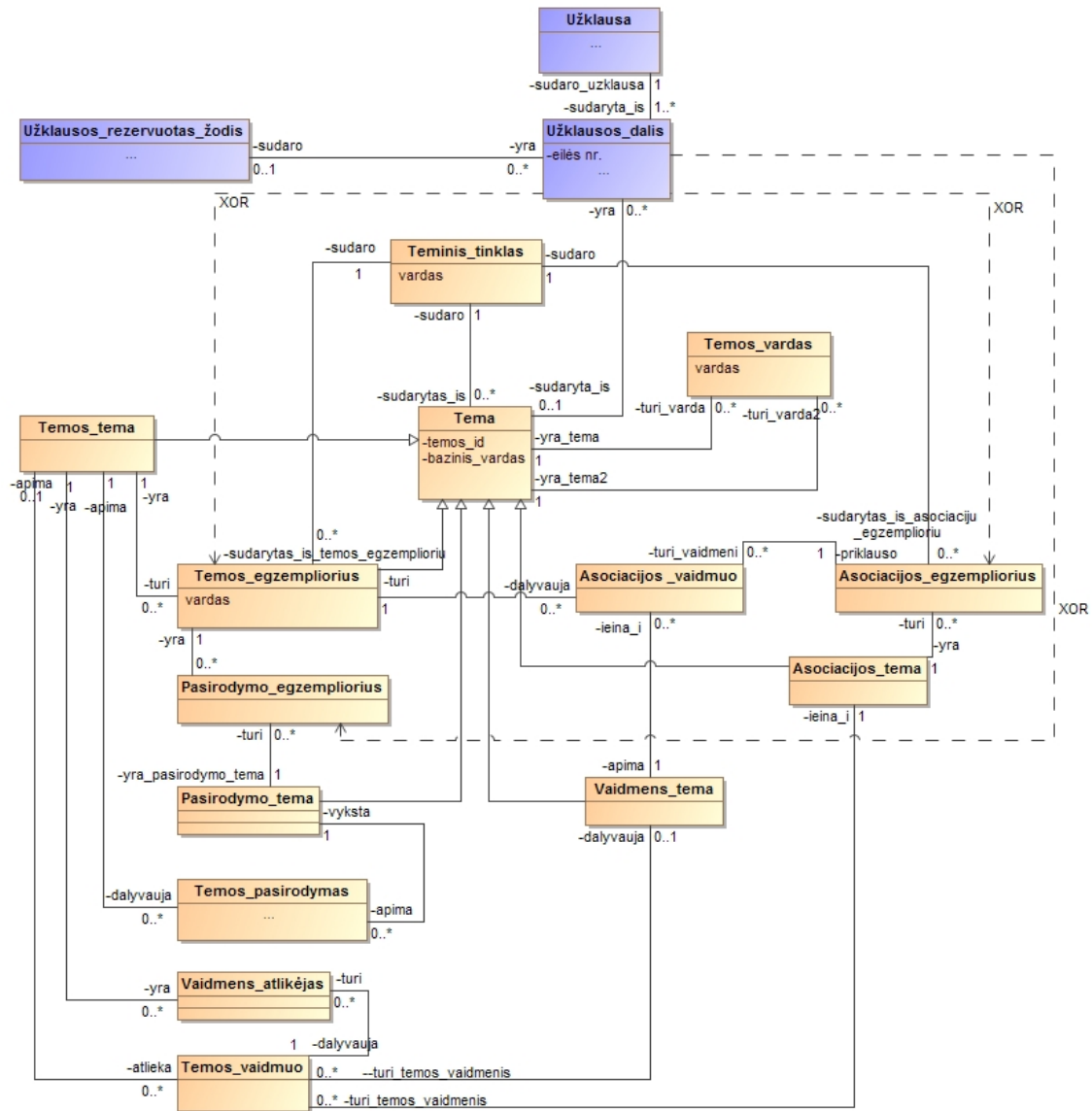
3.1.2.5 Mokymosi reikalavimai

Reikalavimas #5	
Aprašymas:	Sistema turi būti galima išmokti pakankamai greitai
Pagrindimas:	Neturi reikėti daugiau, nei standartinių žinių kaip naudotis kompiuteriu
Tinkamumo kriterijus:	Neapmokyti vartotojai išmoksta naudotis sistema per mažiau nei 10 minučių

18 lentelė. Mokymosi reikalavimai.

3.2 Dalykinės srities modelis

Dalykinė sritis, tai yra teminis tinklas, paieška naudojant natūralios lietuvių kalbos užklausas, užklausių transformavimas ir rezultato atvaizdavimas.



16 pav. Nuo dalykinės srities nepriklausomų TOLOG užklausių sudarymo ir vykdymo tyrimo dalykinės srities modelis.

16 paveikslėlyje matome, kad dalykinę sritį sudaro dvi dalys: užklausa, užklauso transformavimo apibūdinimas ir teminio tinklo sandara. Užklauso dalis susideda iš klasių: užklausa, užklauso_dalis, rezervuotas_zodis. Teminio tinklo dalis susideda iš klasių: teminis_tinklas, tema, temos_tema, asociacijos_tema, asociacijos_vaidmuo, vaidmens_tema, pasirodymo_tema, temos_egzempliorius, asociacijos_egzempliorius, pasirodymo_egzempliorius, temos_pasirodymas, temos_vaidmuo, vaidmens_atlikejas, temos_vaidmuo, vaidmens_atlikejas, temos_vardas.

Pagal dalykinės srities diagramą (16 pav.) aišku, kad struktūrizuota natūralios kalbos užklausa sudaryta iš užklauskos dalių, kas gali būti ne tik įvedami žodžiai, kurie bus panaudoti TOLOG užklauskos sudarymui, bet ir pagalbiniai – Užklauskos_rezervuotas_žodis. Tai įvardžiai (kuris, kuri, kurie, kurios..), kurie reikalingi įvedamo klausimo aiškumui.

Dalykinės srities dalis, kuri apibūdina teminio sandarą, nusako, kad teminį tinką sudaro temos, temų vaidmenys ir temų asociacijos su kitomis temomis. Asociacijos susieja temas pagal asociacijų vaidmenis. Be to temos turi vardus, egzempliorius ir pasirodymus.

19-toje lentelėje apibūdintos dalykinės srities klasės ir jų atributai.

Eil. Nr.	Klasės pavadinimas	Atributas	Atributo paaiškinimas
1.	Užklauskos_rezervuotas_žodis		Formuluojamos natūralios kalbos užklauskos žodeliai, kurie reikalingi dėl klausimo aiškumo.
2.	Užklauskos_dalis		Įvestas užklauskos žodis.
3.	Užklausa	eilės nr.	Struktūrizuotos natūralios kalbos sufomuota užklausa. Šios klasės atributai naudojami formalios užklauskos sudarymui.
4.	Teminis_tinklas	pavadinimas	Teminio tinklo pavadinimas.
5.	Tema	Temos_id	Teminio tinklo temos
		Bazinis_vardas	identifikacinis numeris ir bazinis unikalus vardas.
6.	Asociacijos_vaidmuo		Temos susijusios asociacijos rolėmis. Rolės turi pavadinimus.

7.	Temos_tema		Temos tema.
8.	Asociacijos_tema		Teminiame tinkle asociacija yra tema.
9.	Vaidmens_tema		Teminiame tinkle vaidmuo yra tema.
10.	Pasirodymo_tema		Teminiame tinkle pasirodymas yra tema.
11.	Temos_egzempliorius		Tema turi savo egzempliorius.
12.	Asociacijos_egzempliorius		Asociacijos turi savo egzempliorius.
13.	Pasirodymo_egzempliorius		Temų pasirodymai turi savo egzempliorius.
14.	Temos_pasirodymas		Tema turi pasirodymus, pvz., laikas arba vieta.
15.	Temos_vaidmuo		Tema atlieka vaidmenis asociacijose.
16.	Vaidmens_atlikėjas		Asociacijoje vaidmenį atlieka vaidmens atlikėjas.
17.	Temos_vardas		Tema turi vardą.

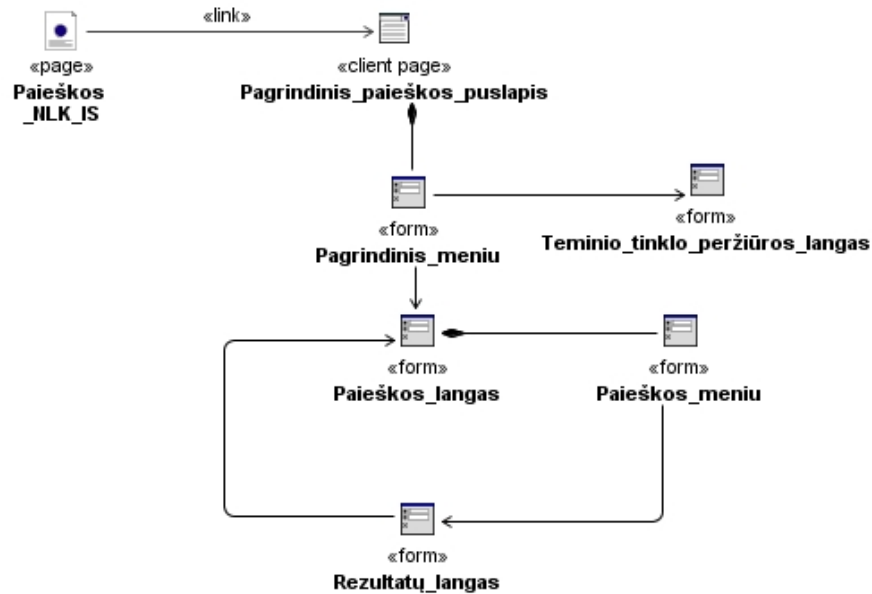
19 lentelė. Dalykinės srities klasių diagramos paaiškinimas.

3.3 Sistemos reikalavimų analizės apibendrinimas

1. Aprašyti sistemos nefunkciniai reikalavimai, pateikiant tinkamumo kriterijus reikalavimų išpildymui patikrinti.
2. Sudarant funkcinis reikalavimus pateikta panaudojimo atvejų diagrama su 9 panaudojimo atvejais.
3. Dviems panaudojimo atvejams („Atrinkti pateikiamus žodžius“ ir „Transformuoti užklausa į formalią“) pateiktos veiklos diagramos. Sudarytos specifikacijos kiekvienam panaudojimo atvejui.
4. Sudarytas pirminis vartotojo sąsajos modelis, kuris gali keistis projekto eigoje.
5. Dalykinė sritis atvaizduota per srities esybių klasių diagramą, apimančią visas teminio tinklo esybes, ryšius tarp jų ir savybes.
6. Per esybių klasių diagramą atvaizduotas konkretus teminis tinklas „Parodos“.

transformavimas ir pardedama paieška teminiame tinkle, kurią atlieka „Išorinis vykdymo posistemis“. Projektuojamas ir realizuojamas bus tik „Užklauso transformavimo posistemis“.

4.1.2 Sistemos vartotojo sąsajos navigavimas

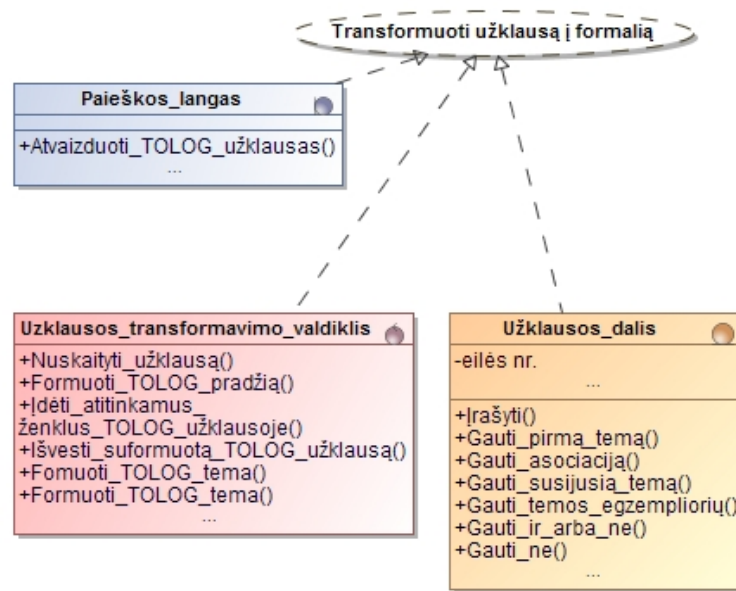


18 pav. Vartotojo sąsajos navigavimo planas.

18 paveiksle pateiktas visos sistemos vartotojo sąsajos navigavimo planas. Čia parodyti visi langai ir meniu. Vartotojas iš pagrindinio paieškos puslapio per pagrindinį meniu gali rinktis norimus veiksmus – paiešką arba teminio peržiūrą. Paieškos lange vartotojas naudodamasis paieškos meniu gali užrašyti ir paleisti paiešką. Rezultatai automatiškai pateikiami naujai atsidariusiame rezultatų lange. Iš šio lango vėl galima patekti į paieškos langą.

4.2 Detalus sistemos projektas

Tam, kad tiksliai nustatyti reikalingus projekto elementus, yra sudaromos panaudojimo atvejų realizacijos analizės klasių diagramos. Čia aptarsime detaliai panaudojimo atvejį, kuris atskleidžia sistemos naujumą. Tai būtų „Transformuoti užklausą į formalią“.



19 pav. PA “Transformuoti užklauską į formalią” realizuojančios klasės.

19 paveiksle pavaizduotos „Transformuoti užklauską į formalią“ realizuojančios klasės: *Paieškos_langas*, *Užklauskos_transformavimo_valdiklis* ir *Užklauskos_dalis*. Pagal klasių turimas funkcijas matome, kokius veiksmai atliekami jose.

Šiame tiriamajame darbe esmė yra natūralios kalbos užklauskos formavimas ir jos transformavimas į formalią TOLOG užklauską. Užklauskai užrašyti bus naudojama struktūrizuota natūrali kalba. Kad būtų aišku, kiek ir kaip kalba bus apribota, būtina sudaryti jos užrašymo taisykles. Bus sudaromos dviejų rūšių taisyklės:

1. Struktūrizuotos natūralios kalbos užklauskų formavimo;
2. Temų bei jų asociacijų įvardinimo.

Užklauskos formavimo metu pasirenkami paieškos žodžiai iš pateiktų žodžių sąrašų, kurie sudaromi pagal logines asociacijas ir teminio tinklo ryšius. Taip yra atrenkamos ne visos teminio tinklo temos, kurios atitinka pasirinktą užklauskos žodį, bet tikrinamos asociacijos tarp temų, kas susiaurina paiešką.

Taisyklių sudarymui bus naudojamas demonstracinis teminis tinklas „Parodos“ (1 pav.)

4.2.1 Teminio tinklo temų ir asociacijų įvardinimo taisyklės

Išanalizavus daugelį natūralios lietuvių kalbos klausimų atvejų, apibendrintos ir sudarytos teminio tinklo temų ir asociacijų įvardinimo taisyklės:

1. Teminiame tinkle temos turi būti daiktavardžiai vardininko linksnyje;
2. Temos pavadinimas gali būti iš kelių žodžių;
3. Jeigu temos pavadinimas yra iš kelių žodžių, tai pirmas žodis turi prasidėti iš didžiosios raidės, o sekantys žodžiai – iš mažosios;
4. Asociacija turi būti būtojo kartinio laiko parastas arba sudėtinis veiksmažodis;
5. Asociacijos pavadinimo pradžioje ir gale yra skliaustuose įrašyti klausiamieji žodeliai, kurie patikslina veiksmažodžio panaudojimo galimybes. Pvz.: (kas?) suorganizavo (ką?).

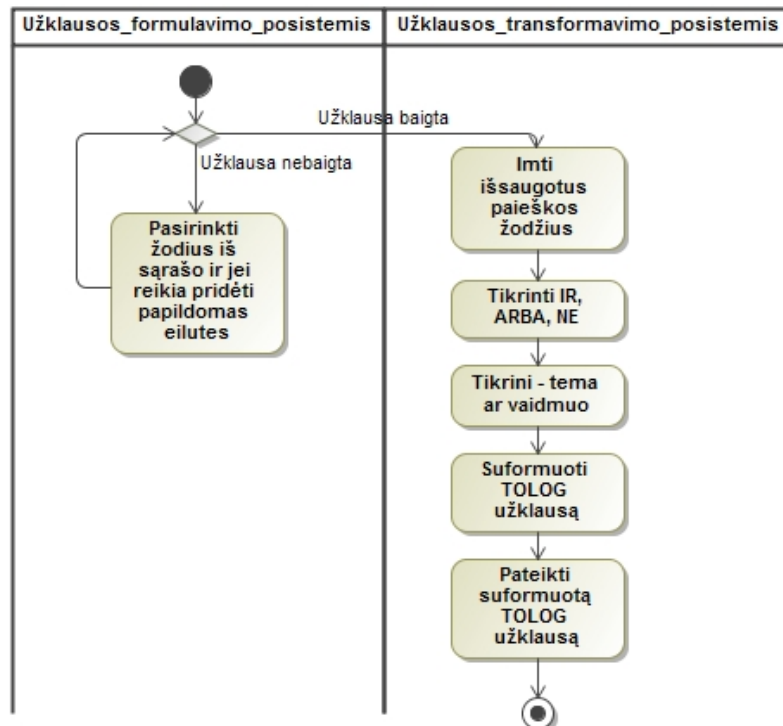
4.2.2 Struktūrizuotos natūralios kalbos užklauso sudarymo taisyklės

1. Užklauso formavimas pradedamas nuo pirmos eilutės. Kad būtų teisingai suformuota TOLOG užklausa, galimos tokios variacijos:
 - 1) Galima pasirinkti tik pirmą temą;
 - 2) Galima pasirinkti pirmą temą ir asociaciją;
 - 3) Galima pasirinkti pirmą temą, asociaciją ir antrą temą;
 - 4) Galima pasirinkti pirmą temą, asociaciją ir temos egzempliorių(egzemplorius bus bet kokios temos, nesusijęs su pirma pasirinkta tema);
 - 5) Galima pasirinkti pirmą temą, asociaciją, antrą temą ir antros temos egzempliorių.
2. Užklausa neapsiriboja tik pirma eilute – galima pridėti papildomų eilučių, kurių forma: „Asociacija-Tema-Temos egzemplorius“. Kad būtų teisingai suformuota TOLOG užklausa, galimos tokios variacijos:
 - 1) Asociaciją pasirinkti privaloma;
 - 2) Galima pasirinkti tik asociaciją;
 - 3) Galima pasirinkti asociaciją ir temą;
 - 4) Galima pasirinkti asociaciją ir temos egzempliorių;
 - 5) Galima pasirinkti asociaciją, temą ir temos egzempliorių.

3. Užklausa galią papildyti ir eilute „Tema-Temos egzempliorius“. Kad būtų teisingai suformuota TOLOG užklausa, galimos tokios variacijos:
 - 1) Galima įvesti arba temą, arba asociaciją;
 - 2) Galima įvesti ir temą ir jos asociaciją.
4. Užklauskos formuluojamos naudojant jungtukus IR, ARBA ir neiginį NE.
5. Pridedant eilutę naudojamas jungtukas IR. Kad būtų teisingai suformuota TOLOG užklausa, galimos tokios variacijos:
 - 1) Galima prie pirmos užklauskos eilutės pridėti kiek norima ir kokiomis tik norima kombinacijomis eilutes „Asociacija-Tema-Temos egzempliorius“ ir „Tema-Temos egzempliorius“.
6. Pridedant eilutę naudojamas jungtukas ARBA. Kad būtų teisingai suformuota TOLOG užklausa, galimos tokios variacijos:
 - 1) Galima prie pirmos užklauskos eilutės pridėti kiek norima ir kokiomis tik norima kombinacijomis eilutes „Asociacija-Tema-Temos egzempliorius“ ir „Tema-Temos egzempliorius“.
7. Norint baigti užklauską ir paleisti TOLOG užklauskos sudarymą spaudžiamas mygtukas „Generuoti“.

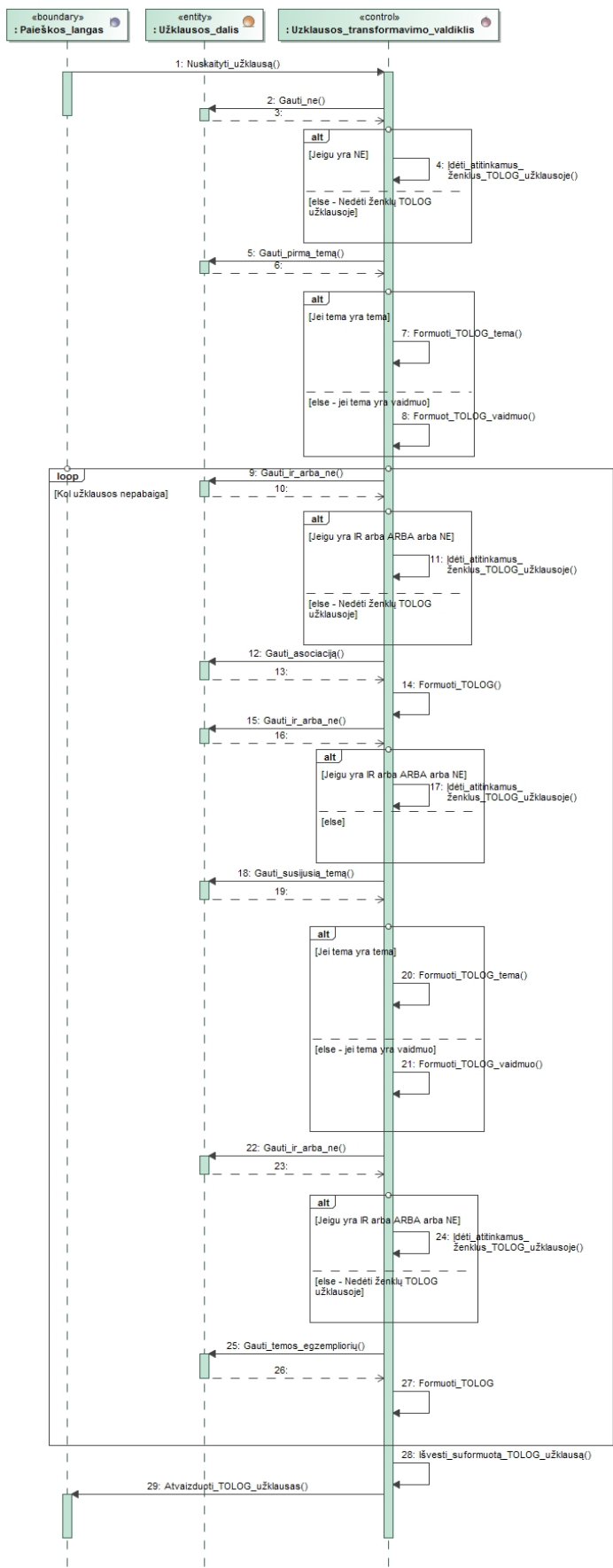
4.3 Sistemos elgsenos modelis

Visas nuoseklus sistemos veikimo procesas formaliai pavaizduotas veiklos diagramoje 20 paveiksle.



20 pav. Užklauskos formulavimo ir transformavimo veiklos diagrama.

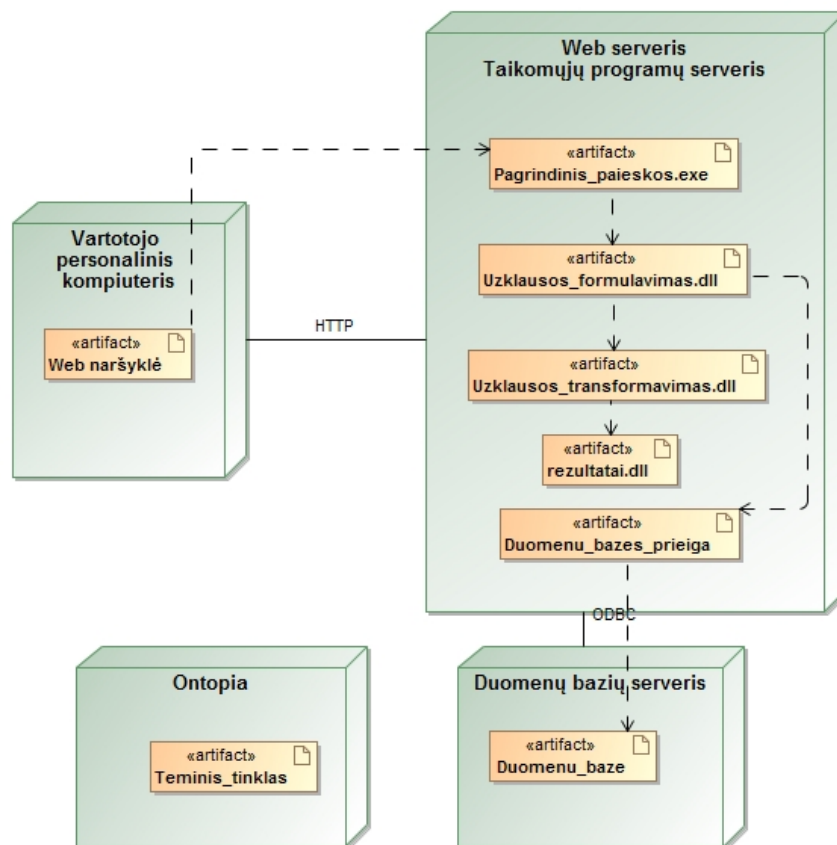
Pirmiausia pradedama formuoti užklausa ir tikrinama, ar ji dar nebaigta. Jeigu ne, tai užklauskos formulavimas tesiasi. Žodžiai pateikiami sąrašo pavidalu. Sistemos vartotojas pasirenka vieną iš žodžių. Jei reikia vartotojas prideda papildomų užklauskos eilučių ir formuoja sudėtingesnę užklauską. Kai užklausa baigta, iš jos suformuojama TOLOG formali užklausa paduodant pasirinktus žodžius į užklauskos transformavimo algoritmą. TOLOG užklauskos formavimo metu tikrinama, kokius vartotojas pasirinkęs jungtukus ar neiginius – ir, arba, ne. Tai irgi turi įtakos TOLOG užklauskos formavimui. Suformuota formali užklausa išvedama ir pavaizduojama vartotojui.



21 pav. Sekų diagrama panaudojimo atvejui „Transformuoti užklausa į formalią“.

Užklauso transformavimas detaliau pavaizduotas sekų diagramoje 21 paveiksle. Procesas prasideda nuo įvestos užklauso nuskaitymo ir įrašymo į kintamuosius – Užklauso_dalis. Prieš kiekvieną žodį tikrinama, ar nėra jungtukų arba neiginio, nes pagal juos TOLOG užklausoje įdedami atitinkami ženklai. Įmama pirma įvesta tema ir iš jos formuojama TOLOG užklauso pradžia. Toliau prasideda ciklas – kol užklauso nepabaiga. Cikle vykdomi vienas po kito veiksmai – tikrinami temos, asociacijos ir egzemplioriai ir iš jų formuojamos TOLOG užklauso dalys - tol, kol yra nuskaitytų temų, asociacijų ar temų egzempliorių. Prieš formuojant TOLOG užklauso dalį su tema, tikrinama, ar tema yra tema, ar vaidmuo. Taip suformuojama visa TOLOG užklausa. Pagal šią sekų diagramą konstruojamas programinis algoritmas – struktūrizuotos natūralios kalbos užklauso transformavimas į TOLOG užklauso.

4.4 Sistemos realizacijos modelis



22 pav. Kūriamos sistemos realizacijos modelis.

22 paveiksle pateiktoje diagramoje matome, kokie artefaktai reikalingi kūriamos sistemos realizacijai ir veikimui bei kaip jie tarpusavyje sąveikauja diegiant sistemą. Kaip matome, Ontopia įrankis yra nesujungtas su visa sistema, todėl, kad užklausų vykdymo sistema automatiškai nepasiekiamas, užklausos vykdymui paduodamos rankiniu būdu.

5 Nuo dalykinės srities nepriklausomo TOLOG užklausų sudarymo sistemos realizacija

Sistemos realizaciją galima pamatyti ir išmėginti internete:

<http://digitalfoto.lt/jovita/ktu/?page=tolog>

5.1 Sistemos realizacijos ir veikimo aprašymas

Pagal atliktą analizę ir projektavimą buvo daroma realizacija. Realizaciją sudaro tokie etapai:

1. Demonstracinio teminio tinklo sukūrimas.
2. Laikinos duomenų bazės sukūrimas teminio tinklo duomenims saugoti;
3. Vartotojo sąsajos sukūrimas;
4. Struktūrizuotos natūralios kalbos užklausų transformavimo į TOLOG užklausas algoritmo sudarymas;

5.1.1 Demonstracinio teminio tinklo sukūrimas

Sukurtos sistemos testavimui, eksperimentui ir demonstravimui sukurtas teminis tinklas „Parodos“. Jis pritaikytas esamai realizacijai:

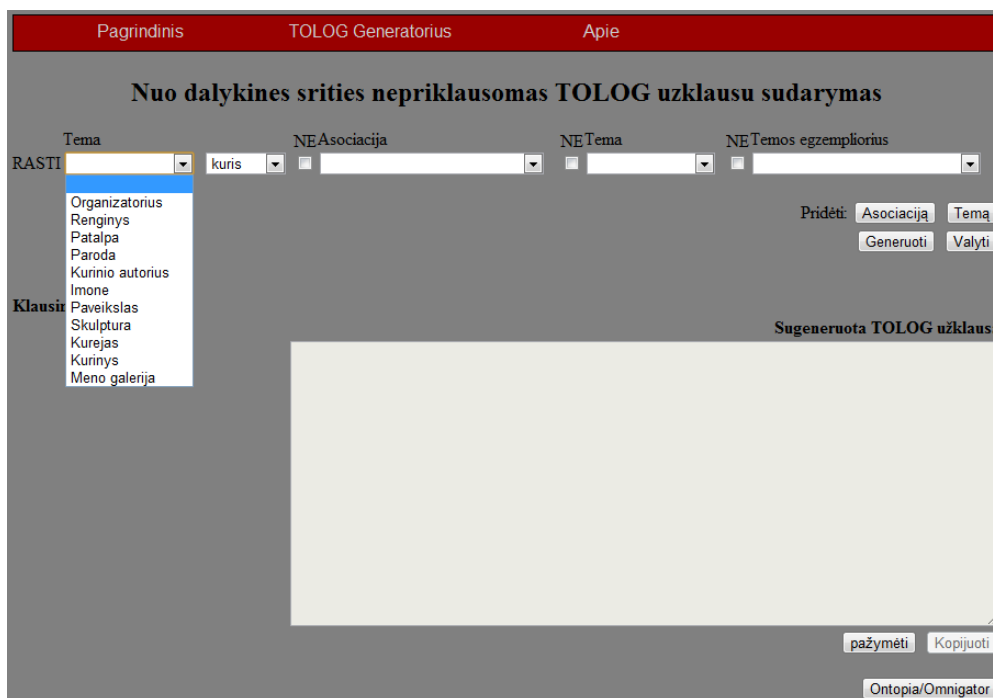
- 1) Remiasi teminio tinklo temų ir asociacijų įvardinimo taisyklėmis (4.2.1 skyrius);
- 2) Visi temų egzemplioriai galėtų turėti pradžioje parašytą savo temos pavadinimą, pvz: „Paroda Smilgų šokis“ arba „Paveikslas Uraganas“. Tai būtų naudinga tam, kad renkantis egzempliorius iš sąrašo būtų aišku, kokiai temai jis priklauso ir galima būtų lengviau formuoti užklausą.

5.1.2 Laikinos duomenų bazės sukūrimas teminio tinklo duomenims saugoti

Dėl vartotojo sąsajos demonstravimo patogumo buvo sukurta duomenų bazė, kurioje suvesti demonstracinio teminio tinklo „Parodos“ duomenys: temos, vaidmenys, asociacijos, temų egzemplioriai. Vartotojas gali nevesti ranka paieškos žodžių, o juos tiesiog pasirinkti iš iškrentančių sąrašų.

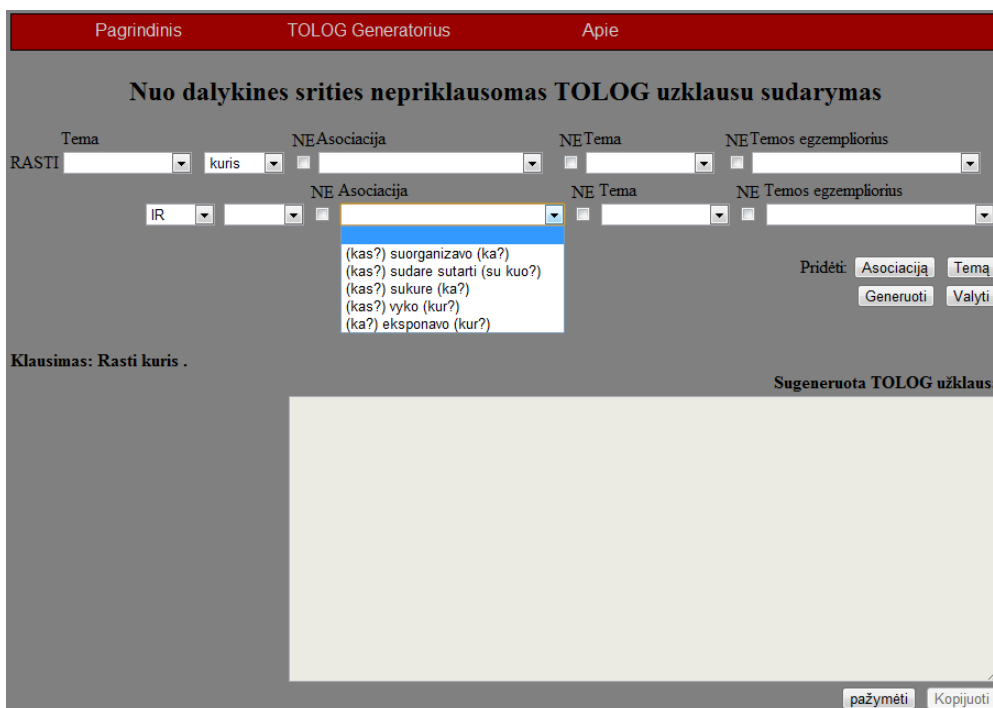
5.1.3 Vartotojo sąsajos sukūrimas

Vartotojo sąsaja sukurta WEB, o jos funkcionalumas suprogramuotas panaudojus PHP ir JAVA programavimo kalbas.



23 pav. Kūriamos sistemos puslapis „TOLOG Generatorius“.

23 paveiksle pavaizduotas vartotojo sąsajos langą „TOLOG Generatorius“. Kiekvieną paieškos žodį galima pasirinkti iš iškrentančių sąrašų. Pirmoje užklauso eilutėje galimi pasirinkti neiginiai „NE“. Jeigu norima užklausą praplėsti dar viena asociacija – spaudžiamas mygtukas „Asociaciją“. Prisideda dar viena eilutė (24 pav.).



24 pav. Pridėta papildoma užklauso eilutė „Asociacija-Tema-Temos egzempliorius“.

Paspaudus mygtuką „Tema“, pridama dar viena eilutė „Tema-Temos egzempliorius“ (25 pav.).

25 pav. Pridėta užklauso eilutė „Tema-Temos egzempliorius“.

Taip pridant vieną arba kitą eilutę formululuojama užklausa. Prieš kiekvieną pridėtą eilutę galima pasirinkti pagal poreikį jungtukus IR arba ARBA, o prieš kiekvieną pasirinktą žodį (išskyrus pirmą temą), galima pasirinkti neiginį NE. Teisingos užklauso formulavimo taisyklės aprašytos 4.2.2 syriuje.

Pabaigus formuluoti užklausą spaudžiamas mygtukas „Generuoti“ ir rezultate išvedama TOLOG užklausa (26 pav.). Be to išvedamas suformuotas klausimas - artimas natūraliai kalbai, kuris pavaizduotas 26 paveiksle viršuje („Rasti Organizatorius kuris (kas?) suorganizavo (ka?) Paroda Smilgu sokis.“).

26 pav. Išvesta sugeneruota TOLOG užklausa.

Dar yra galimybė patekti tiesiai į užklausų vykdymo įrankį Ontopia/Omnigator, kur galima įvykdyti sudarytą TOLOG užklausą teminiame tinkle ir gauti rezultatą. Tam naudojamas mygtukas „Ontopia/Omnigator“ (26 pav.). Mygtukai „pažymėti 1“ ir „pažymėti 2“ naudojami TOLOG užklausoms pažymėti, kad būtų patogų jas nukopijuoti.

5.1.4 Struktūrizuotos natūralios kalbos užklausų transformavimo į TOLOG užklausas algoritmo sudarymas

Remiantis išanalizuotomis natūralios kalbos užklausų ir TOLOG užklausų variacijomis bei sekų diagrama 4.3 skyriuje (22 pav.), sudarytas algoritmas, kuris transformuoja struktūrizuotos natūralios kalbos užklausą į TOLOG užklausą. Remiantis struktūrizuotos natūralios kalbos klausimo užrašymo taisyklėmis, teisingas klausimo pavyzdys būtų 27 paveiksle.

27 pav. Pateikiamos užklausos pavyzdys.

Nuskaitomi pasirinkti žodžiai į kintamuosius ir po gabaliuką sudaroma TOLOG užklausa. Vyksta tikrinimai, ar yra NE, IR, ARBA, nuo kurių priklauso TOLOG užklausos atitinkamų vietų pakeitimai arba papildymai. Be to prie kiekvienos temos tikrinama, ar ji yra tema, ar vaidmuo, kas nulemia formuojamos TOLOG užklausos atitinkamos dalies struktūrą.

Po 28 paveiksle pateiktos užklausos transformavimo gaunama suformuota TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
```

```

$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1,

value($ASSOCNAME2, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE2, $ASSOCNAME2),
type($ASSOC2, $ASSOCTYPE2),

value($NAME2, "Patalpa"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Klaipėdos paveikslu galerija"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC2, $FIRST_ROLE2),
association-role($ASSOC2, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2,

value($NAME3, "Patalpa"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Klaipėdos meno galerija"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC2, $FIRST_ROLE3),
association-role($ASSOC2, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3?

```

TOLOG užklausa randa ieškomas reikšmes. Pagal užduotą klausimą matoma, kad tema Paroda yra tik tema, o ne vaidmuo, todėl TOLOG užklauso dalis atrodo taip:

```

value($NAME0, „Paroda“),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

```

Jeigu tema „Paroda“ būtų vaidmuo, tai ši TOLOG užklauso dalis atrodytų taip:

```

value($NAME0, "Paroda"),
topic-name($ROLENAME0, $NAME0),
type($FIRST_ROLE0, $ROLENAME0),

```

Atitinkamai formuojamos visų variacijų ir kitos TOLOG užklauso. Tai panagrinėta 5.2 skyriuje, sistemos testavime.

Esamo sprendimo atveju duomenys nuskaityti iš duomenų bazės (sukurta sistemos demonstravimui), o gautos TOLOG užklausos vykdomos rankiniu būdu jas nukopijuojant į TOLOG užklausų vykdymo įrankį Ontopia/Omnigator.

Algoritmo veikimo logika vaizdžiai parodyta sekų diagramoje (21 pav.) ir paaiškinta 4.3 skyriuje, o programinis kodas pateiktas pirmame priede (Priedas 1).

5.2 Sistemos testavimo modelis

Nuo dalykinės srities nepriklausomo TOLOG užklausų sudarymo sistemos testavimas pateiktas 20 lentelėje. Testavimo metu buvo tikrinama, kaip reaguoja sistema į pateiktas įvairias ir kelių sudėtingumo lygių struktūrizuotos natūralios kalbos užklausas – ar gaunami teisingi ir tikslūs rezultatai. Teisingas rezultatas taipogi reiškia, kad natūralios kalbos užklauso transformacija į TOLOG užklausa irgi vykdoma teisingai.

Nr.	Atliekamas veiksmas	Sistemos reakcija į atliekamą veiksmą
TOLOG užklausų vykdymas		
1	Tikrinti ar teisingas gaunamas rezultatas, suformavus vieno lygio natūralios kalbos užklausa.	Sistema pateikia rezultatus – sąrašą rastų žodžių, kurie atitinka pateiktos užklauso atsakymą. Atsakymo teisingumą patikriname peržiūrėję teminį tinklą.
2	Tikrinti ar teisingas gaunamas rezultatas, suformavus natūralios kalbos užklausa su jungtuku IR.	TOLOG užklauso suformuotos teisingai, rezultatas gautas teisingas. 21 lentelėje pateiktas pavyzdys.
3	Tikrinti ar teisingas gaunamas rezultatas, suformavus natūralios kalbos užklausa su jungtuku ARBA.	TOLOG užklauso suformuotos teisingai, rezultatas gautas teisingas. 21 lentelėje pateiktas pavyzdys.
4	Tikrinti ar teisingas gaunamas rezultatas, suformavus natūralios kalbos užklausa su neiginiu NE.	TOLOG užklauso suformuotos teisingai, rezultatas gautas teisingas. 21 lentelėje pateiktas pavyzdys.
2	Tikrinti ar teisingas gaunamas rezultatas, suformavus kelių lygių natūralios kalbos užklausa.	2-3 punktai atitinka šį testą. Sistema geba apjungti kelių lygių klausimus į vieną TOLOG užklausa ir pateikia rezultatus – sąrašą rastų žodžių, kurie atitinka pateiktos užklauso atsakymą. Atsakymo teisingumą patikriname peržiūrėję

		teminį tinklą.
3	Tikrinti ar teisingai transformuota natūralios kalbos užklausa į TOLOG užklausa.	Transformacija teisinga, jeigu teisingi 2 ir 3 punktų rezultatai.
4	Patikrinti vartotojo sąsajos veikimą: mygtukų veikimas, pasirinkimo sąrašų pateikimas, reikšmių nuskaitymas, rezultatų išvedimas, pateikto klausimo išvedimas artina natūraliai kalbai forma.	Visos funkcijos veikia tinkamai.

20 lentelė. TOLOG užklausių vykdymo testavimo modelis.

5.3 Sistemos testavimo duomenys ir rezultatai

Testavimas buvo atliekamas sistemos kūrimo metu ir tuo pačiu buvo atliktas eksperimentas – išmėgintos įvairios kelių lygių užklauskos. Ištestavus sukurtą TOLOG užklausių formavimo ir vykdymo sistemą, buvo nustatyta, kad sistema tenkina visus jai keliamus reikalavimus ir gaunami numatyti rezultatai. Testavimo duomenys ir rezultatai pateikti 21 lentelėje.

Testavimo duomenys	Rezultatai
1) Užklausoje pirma tema yra tema, o antra - vaidmuo.	
Rasti Paroda kuria (kas?) suorganizavo (ka?) Organizatorius Vilniaus parodu organizatoriai .	<p>TOLOG užklausa:</p> <pre> select \$rasti from value(\$NAME0, "Paroda"), topic-name(\$TOPICNAME0, \$NAME0), instance-of(\$rasti, \$TOPICNAME0), value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$NAME1, "Organizatorius"), topic-name(\$TOPICNAME1, \$NAME1), type(\$SECOND_ROLE1, \$TOPICNAME1), </pre>

	<pre>value(\$INSTANCENAME1, "Vilniaus parodu organizatoriai"), topic-name(\$INSTANCETOPICNAME1, \$INSTANCENAME1), \$TOPIC1 = \$INSTANCETOPICNAME1, role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1?</pre>
	<p>Rezultatas:</p> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>rasti</p> <p><u>Smilgu sokis</u></p> </div>
<p>2) Užklausoje pirma tema yra vaidmuo, o antra - tema.</p>	
<p>Rasti Renginys kuri (kas?) vyko (kur?) Meno galerija Vilniaus meno galerija .</p>	<p>TOLOG užklausa:</p> <pre>select \$rasti from value(\$NAME0, "Renginys"), topic-name(\$ROLENAME0, \$NAME0), type(\$FIRST_ROLE0, \$ROLENAME0), value(\$ASSOCNAME1, "(kas?) vyko (kur?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$NAME1, "Meno galerija"), topic-name(\$TOPICNAME1, \$NAME1), instance-of(\$TOPIC1, \$TOPICNAME1), value(\$INSTANCENAME1, "Vilniaus meno galerija"), topic-name(\$INSTANCETOPICNAME1, \$INSTANCENAME1), \$TOPIC1 = \$INSTANCETOPICNAME1, role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1?</pre>
	<p>Rezultatas:</p> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>rasti</p> <p><u>Miegas</u></p> <p><u>Smilgu sokis</u></p> </div>
<p>3) Užklausa be antros temos egzemplioriaus.</p>	

<p>Rasti Paroda kuria (kas?) suorganizavo (ka?) Organizatorius .</p>	<p>TOLOG užklausa:</p> <pre> select \$rasti from value(\$NAME0, "Paroda"), topic-name(\$TOPICNAME0, \$NAME0), instance-of(\$rasti, \$TOPICNAME0), value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$NAME1, "Organizatorius"), topic-name(\$TOPICNAME1, \$NAME1), type(\$SECOND_ROLE1, \$TOPICNAME1), role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1? </pre>					
	<p>Rezultatas:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="text-align: center;">rasti</td></tr> <tr><td style="text-align: center;"><u>Smilgu sokis</u></td></tr> <tr><td style="text-align: center;"><u>Ruduo</u></td></tr> <tr><td style="text-align: center;">Kauke</td></tr> <tr><td style="text-align: center;"><u>Miegas</u></td></tr> </table>	rasti	<u>Smilgu sokis</u>	<u>Ruduo</u>	Kauke	<u>Miegas</u>
rasti						
<u>Smilgu sokis</u>						
<u>Ruduo</u>						
Kauke						
<u>Miegas</u>						
<p>4) Užklausoje nėra antra tema, yra tik jos egzempliorius.</p>						
<p>Rasti Paroda kuria (kas?) suorganizavo (ka?) Kauno parodu organizatoriai .</p>	<p>TOLOG užklausa:</p> <pre> select \$rasti from value(\$NAME0, "Paroda"), topic-name(\$TOPICNAME0, \$NAME0), instance-of(\$rasti, \$TOPICNAME0), value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), value(\$INSTANCENAME1, "Kauno parodu organizatoriai"), topic-name(\$INSTANCETOPICNAME1, \$INSTANCENAME1), \$TOPIC1 = \$INSTANCETOPICNAME1, role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), </pre>					

	<pre>role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1?</pre> <p>Rezultatas:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">rasti</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"><u>Miegas</u></div>
5) Užklausoje yra tik pirma tema ir asociacija.	
<p>Rasti Paroda kuria (kas?) suorganizavo (ka?) .</p>	<p>TOLOG užklausa:</p> <pre>select \$rasti from value(\$NAME0, "Paroda"), topic-name(\$TOPICNAME0, \$NAME0), instance-of(\$rasti, \$TOPICNAME0), value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1), role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1?</pre> <p>Rezultatas:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">rasti</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"><u>Smilgu sokis</u></div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"><u>Ruduo</u></div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">Kauke</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"><u>Miegas</u></div>
6) Užklausoje naudojamas neiginys.	
<p>Rasti Paroda kuria (kas?) suorganizavo (ka?)</p> <p>Organizatorius ne Klaipedos parodu organizatoriai .</p>	<p>TOLOG užklausa:</p> <pre>select \$rasti from value(\$NAME0, "Paroda"), topic-name(\$TOPICNAME0, \$NAME0), instance-of(\$rasti, \$TOPICNAME0), value(\$ASSOCNAME1, "(kas?) suorganizavo (ka?)"), topic-name(\$ASSOCTYPE1, \$ASSOCNAME1), type(\$ASSOC1, \$ASSOCTYPE1),</pre>

	<pre>value(\$NAME1, "Organizatorius"), topic-name(\$TOPICNAME1, \$NAME1), type(\$SECOND_ROLE1, \$TOPICNAME1), not(value(\$INSTANCENAME1, "Klaipėdos parodu organizatoriai"), topic-name(\$INSTANCETOPICNAME1, \$INSTANCENAME1), \$TOPIC1 = \$INSTANCETOPICNAME1), role-player(\$FIRST_ROLE1, \$rasti), association-role(\$ASSOC1, \$FIRST_ROLE1), association-role(\$ASSOC1, \$SECOND_ROLE1), role-player(\$SECOND_ROLE1, \$TOPIC1), \$FIRST_ROLE1/= \$SECOND_ROLE1?</pre> <p>Rezultatas – Klaipėdos parodų organizatoriaus suorganizuotos parodos nėra:</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;">rasti</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;"><u>Smilgu sokis</u></div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;">Kauke</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;"><u>Miegas</u></div>
7) Užklausoje naudojami jungtukai IR ir ARBA.	
<p>Rasti Paroda kuria (kas?) suorganizavo (ka?) Organizatorius Kauno parodu organizatoriai IR kuri (kas?) vyko (kur?) Patalpa Kauno paveikslu galerija ARBA kuri (kas?) vyko (kur?) Patalpa Vilniaus paveikslu galerija .</p>	<p>TOLOG užklausa:</p> <p>Rezultatas – Kauno parodų organizatoriaus suorganizuotos parodos, kurios vyko arba Kauno, arba Vilniaus paveikslų galerijoje:</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;">rasti</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px; text-align: center;"><u>Miegas</u></div>

21 lentelė. Sistemos testavimo duomenys ir rezultatai.

5.4 Realizacijos rezultatai

1. Sukurtas automatinę transformaciją realizuojantis algoritmas – iš struktūrizuotos natūralios kalbos užklauso į formalią TOLOG užklauso.
2. Sukurta patogi vartotojo sąsaja, kurioje galima įvesti struktūrizuotos natūralios kalbos klausimus ir gauti TOLOG užklauso.
3. Vartotojo sąsajoje realizuotas užklauso žodžių pasirinkimas iš ikrentančių sąrašų – galima rinktis tik tas temas ar asociacijas, kurios yra demonstraciniame teminiame tinkle.

Tolimesni galimi darbai:

4. Suintegruoti įrankius Ontopia su LifeRay Portal patogesnės vartotojo sąsajos kūrimui.
5. Realizuoti temų atrinkimą pagal logines asociacijas ir pateikimą vartotojo sąsajoje natūralios kalbos užklauso formulavimo metu.
6. Sudaryti galimybę pateiktas natūralios kalbos užklauso automatiškai paleisti vykdyti teminiame tinkle ir gauti rezultatą vartotojo sąsajoje.
7. Realizuoti lietuvių kalbos užklauso transformavimą į TOLOG užklauso panaudojant išorinę žodžių linksniavimo-lemavimo sistemą.

6 Nuo dalykinės srities nepriklausomo TOLOG užklausių sudarymo sistemos eksperimentinis tyrimas

Šiame skyriuje pateikiama informacija apie atliktą eksperimentą. Aprašomas sukurtos sistemos eksperimentinis tyrimas – struktūrizuotos natūralios kalbos užklausių tranfosmavimas į TOLOG užklausas ir jų vykdymas bei rezultatų gavimas. Antrame priede (Priedas 2) pateikta demonstracinio teminio tinklo „Parodos“ turinys ir sąsajos, kas bus reikalinga užklausoms sudaryti ir gautų rezultatų teisingumui nustatyti.

6.1 Eksperimento planas

Eksperimento tikslas – išsiaiškinti, kiek skirtingų tipų ir sudėtingumo lygių TOLOG užklausių gali suformuoti sukurta sistema ir ar užklausa įvykdoma teisingai – gaunami klausimus atitinkantys rezultatai. Eksperimentas vykdomas ieškant informacijos demonstraciniame teminiame tinkle „Parodos“ ir naudojant natūralios kalbos užklausas. Eksperimentas vykdomas atsižvelgiant į punktus:

1. Jungtukai IR;
2. Jungtukai ARBA;
3. Jungtukai NE;
4. IR, ARBA, NE junginiai.
5. Abstrakcijos;

6.1.1 Jungtukai IR

IR jungtukų galima daryti įvairias kombinacijas.

1. IR jungia keletą asociacijų - klausimas 28 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)	Organizatorius	Kauno parodu organizatoriai
IR	kuria (kas?) suorganizavo (ka?)	Organizatorius	Vilniaus parodu organizatoriai
IR	kuri (kas?) vyko (kur?)	Patalpa	Kauno paveikslu galerija

28 pav. Kelios asociacijos sujungtos jungtukais IR.

TOLOG užklausa:

```
select $rasti from  
  
value($NAME0, "Paroda"),  
topic-name($TOPICNAME0, $NAME0),  
instance-of($rasti, $TOPICNAME0),  
  
value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),  
topic-name($ASSOCTYPE1, $ASSOCNAME1),  
type($ASSOC1, $ASSOCTYPE1),  
  
value($NAME1, "Organizatorius"),
```

```

topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1,

value($ASSOCNAME2, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE2, $ASSOCNAME2),
type($ASSOC2, $ASSOCTYPE2),

value($NAME2, "Organizatorius"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

not(value($INSTANCENAME2, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2),

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC2, $FIRST_ROLE2),
association-role($ASSOC2, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2,

value($ASSOCNAME3, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE3, $ASSOCNAME3),
type($ASSOC3, $ASSOCTYPE3),

value($NAME3, "Patalpa"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Kauno paveikslu galerija"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC3, $FIRST_ROLE3),
association-role($ASSOC3, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3?

```

Rezultatas:



Rastos visos parodos, kurias suorganizavo organizatorius „Kauno parodų organizatoriai“, bet ne organizatorius „Vilniaus parodų organizatoriai“, ir kuri vyko patalpoje „Kauno paveikslų galerija“.

2. IR jungia keletą temų ir jų egzempliorių - klausimas 29 paveiksle.

29 pav. Kelios temos sujungtos jungtukais IR.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Kurinio autorius"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) sukure (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Kurinys"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Naujas"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1,

value($NAME2, "Kurinys"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Kaspiniai"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC1, $FIRST_ROLE2),
association-role($ASSOC1, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
```

```

$FIRST_ROLE2/= $SECOND_ROLE2,

value($NAME3, "Kuriny"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Muilo burbulai"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC1, $FIRST_ROLE3),
association-role($ASSOC1, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3?

```

Rezultatas:

Antanas Obcarskas

Rastas kūrinio autorius, kuris sukūrė kūrinį „Naujas“, ir kūrinį „Kaspinaĩ“, ir kūrinį „Muilo burbulai“.

3. IR jungia keletą asociacijų ir temų - klausimas 30 paveiksle.

30 pav. Kelios asociacijos, kelios temos sujungtos jungtukais IR.

TOLOG užklausa:

```

select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

```

```

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1,

value($NAME2, "Organizatorius"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

not(value($INSTANCENAME2, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2),

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC1, $FIRST_ROLE2),
association-role($ASSOC1, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2,

value($ASSOCNAME3, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE3, $ASSOCNAME3),
type($ASSOC3, $ASSOCTYPE3),

value($NAME3, "Patalpa"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Kauno paveikslu galerija"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC3, $FIRST_ROLE3),
association-role($ASSOC3, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3,

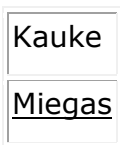
value($NAME4, "Patalpa"),
topic-name($TOPICNAME4, $NAME4),
type($SECOND_ROLE4, $TOPICNAME4),

not(value($INSTANCENAME4, "Klaipėdos paveikslu galerija"),
topic-name($INSTANCETOPICNAME4, $INSTANCENAME4),
$TOPIC4 = $INSTANCETOPICNAME4),

role-player($FIRST_ROLE4, $rasti),
association-role($ASSOC3, $FIRST_ROLE4),
association-role($ASSOC3, $SECOND_ROLE4),
role-player($SECOND_ROLE4, $TOPIC4),
$FIRST_ROLE4/= $SECOND_ROLE4?

```

Rezultatas:



Rastos visos parodos, kurias suorganizavo organizatorius „Kauno parodų organizatoriai“, bet ne organizatorius „Vilniaus parodų organizatoriai“, ir kuri vyko patalpoje „Kauno paveikslų galerija“, bet ne patalpoje „Klaipėdos paveikslų galerija“.

6.1.2 Jungtukai ARBA

Čia panagrinėtos užklausų variacijos su jungtuku ARBA.

1. ARBA jungia keletą asociacijų - klausimas 31 paveiksle.

31 pav. Kelios asociacijos sujungtos jungtukais ARBA.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

{value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodų organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1 |

value($ASSOCNAME2, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE2, $ASSOCNAME2),
type($ASSOC2, $ASSOCTYPE2),

value($NAME2, "Patalpa"),
topic-name($TOPICNAME2, $NAME2),
```

```

type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Kauno meno galerija"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC2, $FIRST_ROLE2),
association-role($ASSOC2, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2 |

value($ASSOCNAME3, "(ka?) eksponavo (kur?)"),
topic-name($ASSOCTYPE3, $ASSOCNAME3),
type($ASSOC3, $ASSOCTYPE3),

value($NAME3, "Kurinys"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Zvaigzdzio vartai"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC3, $FIRST_ROLE3),
association-role($ASSOC3, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3}?

```

Rezultatas:

Miegas
Kauke
Ruduo

Rastos visos parodos, kurias suorganizavo organizatorius „Kauno parodų organizatoriai“, arba kuri vyko patalpoje „Kauno meno galerija“, arba kurioje eksponavo kūrinį „Žvaigždžių vartai“.

2. ARBA jungia keletą temų su egzemplioriais - klausimas 32 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)	Organizatorius	Kauno parodų organizatoriai
ARBA	Organizatorius	Klaipėdos parodų organizatoriai	
ARBA	Organizatorius	Vilniaus parodų organizatoriai	

32 pav. Kelios temos su egzemplioriais sujungtos jungtukais ARBA.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

{value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1 |

value($NAME2, "Organizatorius"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Klaipėdos parodu organizatoriai"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC1, $FIRST_ROLE2),
association-role($ASSOC1, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2 |

value($NAME3, "Organizatorius"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC1, $FIRST_ROLE3),
association-role($ASSOC1, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3}?
```

Rezultatas:

Miegas
Kauke
Ruduo
Smilgu sokis

Rastos visos parodos, kurias suorganizavo organizatorius „Kauno parodų organizatoriai“, arba organizatorius „Klaipėdos parodų organizatoriai“, arba organizatorius „Vilniaus parodų organizatoriai“.

3. ARBA jungia keletą asociacijų ir temų su egzemplioriais - klausimas 33 paveiksle.

33 pav. Kelios asociacijos, kelios temos sujungtos jungtukais IR.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

{value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno parodų organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1 |

value($NAME2, "Organizatorius"),
topic-name($TOPICNAME2, $NAME2),
```

```

type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC1, $FIRST_ROLE2),
association-role($ASSOC1, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2 |

value($ASSOCNAME3, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE3, $ASSOCNAME3),
type($ASSOC3, $ASSOCTYPE3),

value($NAME3, "Patalpa"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Kauno paveikslu galerija"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC3, $FIRST_ROLE3),
association-role($ASSOC3, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3 |

value($NAME4, "Patalpa"),
topic-name($TOPICNAME4, $NAME4),
type($SECOND_ROLE4, $TOPICNAME4),

value($INSTANCENAME4, "Vilniaus paveikslu galerija"),
topic-name($INSTANCETOPICNAME4, $INSTANCENAME4),
$TOPIC4 = $INSTANCETOPICNAME4,

role-player($FIRST_ROLE4, $rasti),
association-role($ASSOC3, $FIRST_ROLE4),
association-role($ASSOC3, $SECOND_ROLE4),
role-player($SECOND_ROLE4, $TOPIC4),
$FIRST_ROLE4/= $SECOND_ROLE4}?

```

Rezultatas:

Miegas
Kauke
Ruduo
Smigu sokis

Rastos visos parodos, kurias suorganizavo organizatorius „Kauno parodų organizatoriai“ arba organizatorius „Vilniaus parodų organizatoriai“, arba kuri vyko patalpoje „Kauno paveikslų galerija“, arba patalpoje „Vilniaus paveikslų galerija“.

6.1.3 Neiginiai NE

Neiginys NE gali būti prieš temą, prieš asociaciją, prieš temos egzempliorių.

1. NE yra prieš asociaciją - klausimas 34 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	<input checked="" type="checkbox"/> kurios (kas?) suorganizavo (ka?)	<input type="checkbox"/> Patalpa	<input type="checkbox"/> Kauno paveikslu galerija

34 pav. Neiginys NE yra prieš asociaciją.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

not(value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1)),

value($NAME1, "Patalpa"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

value($INSTANCENAME1, "Kauno paveikslu galerija"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1?
```

Rezultatas:

Kauke
Miegas

Rastos visos parodos, kurių nesuorganizavo patalpa „Kauno paveikslų galerija“ – tai yra, rado visas parodas, kurios dalyvauja asociacijoje su patalpa „Kauno paveikslų galerija“, bet ta asociacija yra ne „(kas?) suorganizavo (ka?)“.

2. NE yra prieš temą - klausimas 35 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)	<input checked="" type="checkbox"/> Kurejas	<input type="checkbox"/> Vilniaus parodu organizatoriai

35 pav. Neiginys NE yra prieš temą.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

not(value($NAME1, "Kurejas"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1)),

value($INSTANCENAME1, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1?
```

Rezultatas:

Smilgu sokis

Rastos visos parodos, kurias suorganizavo ne patalpa „Kauno parodų organizatoriai“ – tai yra, rado visas parodas, kurias suorganizavo „Kauno parodų organizatoriai“, bet jie tikrai nėra patalpa.

3. NE yra prieš temos egzempliorių - klausimas 36 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)	<input type="checkbox"/> Organizatorius	<input checked="" type="checkbox"/> Kauno parodu organizatoriai

36 pav. Neiginys NE yra prieš temos egzempliorių.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
```

```

type($ASSOC1, $ASSOCTYPE1),

not(value($NAME1, "Kurejas"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1)),

value($INSTANCENAME1, "Vilniaus parodu organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1?

```

Rezultatas:

Smilgu sokis
Kauke
Miegas

Rastos visos parodos, kurias suorganizavo bet kuris organizatorius, bet ne „Kauno parodų organizatoriai“.

6.1.4 IR, ARBA, NE junginiai

Kiekvienas jungtukas ar neiginys yra tikrinamas atskirai, todėl jie vienas nuo kito nėra priklausomi. Pateikiamas vienas sudėtingesni bandymas.

1. NE yra prieš temos egzempliorių, IR ir ARBA jungia ir asociacijas ir temas su egzemplioriais - klausimas 37 paveiksle.

The screenshot shows a query builder interface with several rows of conditions. Each row contains a logical connector (IR, ARBA, NE) and a condition. The conditions are:

- Row 1: Tema: Paroda; NE Asociacija: kuria; (kas?) suorganizavo (ka?); NE Tema: Organizatorius; NE Temos egzempliorius: Kauno parodu organizatoriai
- Row 2: IR; NE Asociacija: kuri; (kas?) vyko (kur?); NE Tema: Patalpa; NE Temos egzempliorius: Vilniaus paveikslu galerija
- Row 3: ARBA; NE Asociacija: kurioje; (ka?) eksponavo (kur?); NE Tema: Patalpa; NE Temos egzempliorius: Klaipedos paveikslu galerija
- Row 4: ARBA; NE Asociacija: kurioje; (ka?) eksponavo (kur?); NE Tema: Kurinys; NE Temos egzempliorius: Papugos
- Row 5: IR; NE Tema: Kurinys; NE Temos egzempliorius: Saule

37 pav. IR, ARBA, NE kombinacija klausime.

TOLOG užklausa:

```

select $rasti from

value($NAME0, "Paroda"),

```



```

topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

not(value($INSTANCENAME1, "Kauno parodų organizatoriai"),
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),
$TOPIC1 = $INSTANCETOPICNAME1),

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1,

value($ASSOCNAME2, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE2, $ASSOCNAME2),
type($ASSOC2, $ASSOCTYPE2),

{value($NAME2, "Patalpa"),
topic-name($TOPICNAME2, $NAME2),
type($SECOND_ROLE2, $TOPICNAME2),

value($INSTANCENAME2, "Vilniaus paveikslų galerija"),
topic-name($INSTANCETOPICNAME2, $INSTANCENAME2),
$TOPIC2 = $INSTANCETOPICNAME2,

role-player($FIRST_ROLE2, $rasti),
association-role($ASSOC2, $FIRST_ROLE2),
association-role($ASSOC2, $SECOND_ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE2/= $SECOND_ROLE2 |

value($NAME3, "Patalpa"),
topic-name($TOPICNAME3, $NAME3),
type($SECOND_ROLE3, $TOPICNAME3),

value($INSTANCENAME3, "Klaipėdos paveikslų galerija"),
topic-name($INSTANCETOPICNAME3, $INSTANCENAME3),
$TOPIC3 = $INSTANCETOPICNAME3,

role-player($FIRST_ROLE3, $rasti),
association-role($ASSOC2, $FIRST_ROLE3),
association-role($ASSOC2, $SECOND_ROLE3),
role-player($SECOND_ROLE3, $TOPIC3),
$FIRST_ROLE3/= $SECOND_ROLE3 |

value($ASSOCNAME4, "(ka?) eksponavo (kur?)"),
topic-name($ASSOCTYPE4, $ASSOCNAME4),
type($ASSOC4, $ASSOCTYPE4),

```

```

value($NAME4, "Kuriny"),
topic-name($TOPICNAME4, $NAME4),
type($SECOND_ROLE4, $TOPICNAME4),

not(value($INSTANCENAME4, "Papūgos"),
topic-name($INSTANCETOPICNAME4, $INSTANCENAME4),
$TOPIC4 = $INSTANCETOPICNAME4),

role-player($FIRST_ROLE4, $rasti),
association-role($ASSOC4, $FIRST_ROLE4),
association-role($ASSOC4, $SECOND_ROLE4),
role-player($SECOND_ROLE4, $TOPIC4),
$FIRST_ROLE4/= $SECOND_ROLE4},

value($NAME5, "Kuriny"),
topic-name($TOPICNAME5, $NAME5),
type($SECOND_ROLE5, $TOPICNAME5),

not(value($INSTANCENAME5, "Saulė"),
topic-name($INSTANCETOPICNAME5, $INSTANCENAME5),
$TOPIC5 = $INSTANCETOPICNAME5),

role-player($FIRST_ROLE5, $rasti),
association-role($ASSOC4, $FIRST_ROLE5),
association-role($ASSOC4, $SECOND_ROLE5),
role-player($SECOND_ROLE5, $TOPIC5),
$FIRST_ROLE5/= $SECOND_ROLE5?

```

Rezultatas:

Smilgu sokis

Rastos visos parodos, kurias suorganizavo organizatorius, bet ne „Kauno parodų organizatoriai“, ir kuri vyko patalpoje „Vilniaus paveikslų galerija“ arba patalpoje „Klaipėdos paveikslų galerija“ ir kurioje eksponavo kurinį, bet ne „Papūgos“ ir kurinį, bet ne „Saulė“.

6.1.5 Abstrakcijos

Abstrakcijos gaunamos, kai neįvardinama asociacija, arba tema, arba temos egzempliorius.

1. Neįvardinta asociacija, antra tema ir temos egzempliorius po pirmos temos - klausimas 38 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda <input type="text" value=""/>	kuris <input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

38 pav. Tik pirma tema klausime.

TOLOG užklausa:

```
select $rasti from
```

```

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0)?

```

Rezultatas:

Smilgu sokis
Ruduo
Kauke
Miegas

Rastos visos parodos.

2. Neįvardinta tema ir temos egzempliorius po asociacijos - klausimas 39 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)		

39 pav. Klausime tik pirma tema ir jos asociacija.

TOLOG užklausa:

```

select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1?

```

Rezultatas:

Smilgu sokis
Ruduo
Kauke
Miegas

Rastos parodos, kurias suorganizavo bet kas.

3. Neįvardintas temos egzempliorius po temos - klausimas 41 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuria (kas?) suorganizavo (ka?)		

40 pav. Klausime yra tema, asociacija ir antra tema.

TOLOG užklausa:

```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) suorganizavo (ka?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
type($ASSOC1, $ASSOCTYPE1),

value($NAME1, "Organizatorius"),
topic-name($TOPICNAME1, $NAME1),
type($SECOND_ROLE1, $TOPICNAME1),

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC1, $FIRST_ROLE1),
association-role($ASSOC1, $SECOND_ROLE1),
role-player($SECOND_ROLE1, $TOPIC1),
$FIRST_ROLE1/= $SECOND_ROLE1?
```

Rezultatas:

Smilgu sokis
Ruduo
Kauke
Miegas

Rastos parodos, kurias suorganizavo bet kokie organizatoriai.

4. Neįvardinta egzemplioriaus tema - klausimas 41 paveiksle.

Tema	NE Asociacija	NE Tema	NE Temos egzempliorius
RASTI Paroda	kuri (kas?) vyko (kur?)		Kauno meno galerija

41 pav. Klausime neįvardinta egzemplioriaus tema.

TOLOG užklausa:

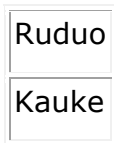
```
select $rasti from

value($NAME0, "Paroda"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0),

value($ASSOCNAME1, "(kas?) vyko (kur?)"),
topic-name($ASSOCTYPE1, $ASSOCNAME1),
```

```
type($ASSOC1, $ASSOCTYPE1),  
  
value($INSTANCENAME1, "Kauno meno galerija"),  
topic-name($INSTANCETOPICNAME1, $INSTANCENAME1),  
$TOPIC1 = $INSTANCETOPICNAME1,  
  
role-player($FIRST_ROLE1, $rasti),  
association-role($ASSOC1, $FIRST_ROLE1),  
association-role($ASSOC1, $SECOND_ROLE1),  
role-player($SECOND_ROLE1, $TOPIC1),  
$FIRST_ROLE1/= $SECOND_ROLE1?
```

Rezultatas:



Rastos parodos, kurios vyko bet kur, kieno pavadinimas-egzempliorius yra „Kauno meno galerija“.

6.2 Eksperimento rezultatai

Atlikus eksperimentą gauti tokie rezultatai:

1. Sudarytos ir įvydytos įvairaus sudėtingumo TOLOG užklauso: vieno lygio, daug lygių, su jungtukais IR, ARBA, su neiginiu NE, su šių jungtukų ir neiginio variacijomis, su abstrakcijomis;
2. Rezultatų tikslumas – gauti visi suformuotą struktūrizuotos natūralios kalbos klausimą atitinkantys rezultatai;
3. TOLOG užklauso iš struktūrizuotos natūralios kalbos užklauso suformuotos teisingai – sistemos algoritmas veikia teisingai;
4. TOLOG užklauso šablonai padengia daugelį natūralios kalbos klausimų ir jų variacijų.

7 Nuo dalykinės srities nepriklausomo TOLOG užklausų sudarymo išvados

1. Teminio tinklo užrašymo kalbos XTM ir įrankio Ontopia analizė parodė, kad teminius tinklus galima kurti ir juose ieškoti informacijos naudojantis įrankiu Ontopia.
2. Išanalizavus TOLOG užklausų kalbą nustatyta, kad jos struktūra leidžia sudaryti universalią parametrizuotą užklausą, pritaikomą informacijos paieškai bet kokios dalykinės srities teminiame tinkle.
3. Sukurtasis demonstracinis tinklas parodė, kad teminį tinklą galima sudaryti laikantis darbe pateiktų teminio tinklo elementų įvardinimo taisyklių ir kad taip sukurtas teminis tinklas sudaro sąlygas formuluoti užklausas natūraliai kalbai artima forma.
4. Pateikti užklausų pavyzdžiai parodė, kad įmanoma suformuluoti užklausas artima natūraliai kalbai forma, jei laikomasi teminio tinklo elementų įvardinimo ir užklausų formulavimo taisyklių.
5. Atlikus eksperimentą, nustatyta, kad realizuotas sistemos prototipas suteikia vartotojams galimybę pateikti užklausą bet kokios dalykinės srities teminiam tinklui natūraliai kalbai artima forma, ją transformuoti į TOLOG užklausą ir sėkmingai įvykdyti.
6. Tačiau realizuotos sistemos grafinė sąsaja neleidžia suformuluoti užklausos bet kokia forma. Šis trūkumą būtų galima pašalinti apibrėžus daugiau užklausos formulavimo ir jos transformavimo į TOLOG užklausą taisyklių. Taip pat sistema turi būti papildyta žodžių derinimo funkcija.

8 Literatūros saraksts

1. Hart, G., Johnson, M., Dolbear, C. Rabbit: Developing a Control Natural Language for Authoring Ontologies. S. Bechhofer et al.(Eds.): ESWC 2008, Springer-Verlag Berlin Heidelberg, LNCS 5021, 2008, 348–360.
2. Kaljurand, K. Attempto **Controlled English as a Semantic Web Language**. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.
3. Kaufmann, E., Bernstein, A., Zumstein, R. Querix: **A Natural Language Interface to Query Ontologies Based on Clarification Dialogs**. Proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, GA, 2006, 980-981.
4. Pepper, S., Moore, G. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification [žiūrēta 2010-11-16]. Prieiga per internetu: <http://www.topicmaps.org/xtm/1.0/>, 2001.
5. Garshol, L. M., Moore, G. Topic Maps – Data Model. ISO/IEC JTC1/SC34. Information Technology – Document Description and Processing Languages, 2008 [žiūrēta 2010-11-16]. Prieiga per internetu: <http://www.isotopicmaps.org/sam/sam-model/>
6. Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004, Thessaloniki, Greece. TOPIC MAP EXISTING TOOLS: A BRIEF REVIEW by Athanasios Hatzigaidas, Anastasia Papastergiou, George Tryfon, Despoina Maritsa [žiūrēta 2010-11-17]. Prieiga per internetu:
<http://www.emis.de/journals/AUA/acta8/Hatzigaidas-Papastergiou-Tryfon-Maritsa.pdf>
7. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. University of Zurich. Department of Informatics, Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland.
8. Kaufmann, E., Bernstein, A. How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users? Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, 2007, 281-294.
9. Talking to the Semantic Web [žiūrēta 2010-12-12]. Prieiga per internetu:
<http://www.ifi.uzh.ch/ddis/research/talking-to-the-semantic-web/>

10. Tolog – a topic maps query language. Lars Marius Garshol Ontopia AS, Oslo, Norway [žiūrėta 2010-12-13]. Prieiga per internetą:
<http://www.garshol.priv.no/download/text/tolog.pdf>
11. Liferay Portal Overview, Features, Benefits [žiūrėta 2010-12-13]. Prieiga per internetą:
<http://www.liferay.com/products/liferay-portal/overview>
12. Kas bendro tarp verslo inovacijų ir semantinių technologijų? Tomas Pranckevičius, Žinių ekonomikos forumas, 2010 gruodžio mėn. 13 d. 15:46 [žiūrėta 2010-11-17]. Prieiga per internetą: <http://verslas.delfi.lt/archive/print.php?id=39654187>
13. Verb-ish Models for Verbalization: Give Us Back Our Verbs! Ronald G. Ross [žiūrėta 2011-06-18]. Prieiga per internetą:
http://www.brcommunity.com/b506.php?zoom_highlight=statement
14. Lars Marius Garshol. What Are Topic Maps. September 11, 2002 [žiūrėta 2011-09-02]. Prieiga per internetą:
<http://www.xml.com/pub/a/2002/09/11/topicmaps.html>
15. The built-in tolog predicates. Published by Ontopia 2005/07/13 [žiūrėta 2011-09-10]. Prieiga per internetą:
<http://www.ontopia.net/topicmaps/materials/tolog-predicate-reference.html#d0e49>
16. Garshol, L. M. **Towards a Methodology for Developing Topic Maps Ontologies**. *TMRA 2006, LNAI, Vol. 4438*, 2007, 20–31.
17. Ontopia - the product [žiūrėta 2011-11-17]. Prieiga per internetą:
<http://www.ontopia.net/section.jsp?id=ontopia-the-product>

Priedas 1

Struktūrizuotos natūralios kalbos užklauso transformavimo į TOLOG užlausa algoritmo

programinis kodas:

```
<table width="900" border="0" cellpadding="0" cellspacing="0">
<tr><td align="left">
<?php
# selecto boolean suradimas
require ('connect.php');
?>
<strong>Klausimas:
Rasti <?echo $_POST['temas1'];
    for($i=1; $i<10; $i++){
        ?> <?echo $_POST['ir'.$i.''];?><?echo $_POST['irr'.$i.''];?> <?echo
$_POST['pasirinkimas'.$i.''];?> <?echo $_POST['neA'.$i.''];?> <?echo $_POST['asociacija'.$i.''];?>
<?echo $_POST['neT'.$i.''];?> <?echo $_POST['tema'.$i.''];?> <?echo $_POST['neE'.$i.''];?> <?echo
$_POST['egzempliorius'.$i.''];?>.
</strong></td></tr>
<tr>
<td align="right" valign="middle">
<strong>Sugeneruota TOLOG užklausa</strong>
</td></tr></table>
<form id="area" name="area" action="" method="POST">
<table width="900" border="0" cellpadding="0" cellspacing="0">
<?//-----UZKLAUSA1 - TEMA = TEMA-----
-----?>
<tr><td align="right">
<form name="selectall" action="" method="post">
<textarea name="textareaname" cols="78" rows="16" disabled="disabled">
<?

if ($_POST['temas1']!=''){//-----pradzia - temas1 - pirma tema
uzklausoje-----
$tag=$_POST['temas1'];
$result = mysql_query("select * from tema where pavadinimas='$tag'");
$row = mysql_fetch_array($result) or die(mysql error());

?>select $rasti from

<?
$int=$intA=$intT=$intE=1;
if ($row['yra vaidmuo']==1)
{

?>
value($NAME0, "<?php echo $_POST['temas1']; ?>"),
topic-name($ROLENAME0, $NAME0),
type($FIRST_ROLE1, $ROLENAME0)<?if($_POST['asociacija'.$intA.'']!='){?>,

<?}else{?>,

role-player($FIRST_ROLE1, $rasti),
association-role($ASSOC, $FIRST_ROLE1),
association-role($ASSOC, $ROLE2),
role-player($SECOND_ROLE2, $TOPIC2),
$FIRST_ROLE1/=$SECOND_ROLE2?
<?}

}elseif ($row['yra_vaidmuo']==0)
{
?>
value($NAME0, "<?php echo $_POST['temas1']; ?>"),
topic-name($TOPICNAME0, $NAME0),
instance-of($rasti, $TOPICNAME0)<?if($_POST['asociacija'.$intA.'']!='){?>,

<?}else{?>?<?}
}

//-----kol yra bent viena sekanti asociacija ar tema ar
egzempliorius-----

while($_POST['asociacija'.$intA.'']!=' || $_POST['tema'.$intT.'']!=' ||
$_POST['egzempliorius'.$intE.'']!='){

    if($_POST['asociacija'.$intA.'']!='){// -----
asociacija -----
        if($_POST['ir'.$intA.'']!='ARBA' &&
$_POST['ir'.($intA+1).'] =='ARBA'){
            ?><?

        }

        $assoc=$intA;
```

```

// $assoc1 = $assoc;

if ($_POST['neA'.$sintA.'']=='ne'){?>not(<?)?>value($ASSOCNAME<?echo $sintA;?>, "<?php echo
$_POST['asociacija'.$sintA.''];?>"),
topic-name($ASSOCTYPE<?echo $sintA;?>, $ASSOCNAME<?echo $sintA;?>),
type($ASSOC<?echo $sintA;?>, $ASSOCTYPE<?echo $sintA;?>)<?if ($_POST['neA'.$sintA.'']=='ne'){?><?>?,
<? } //else $assoc=$assoc1; //-----nera asociacijos-----
-----
if($_POST['tema'.$sintT.'']!=''){ // ----- tema
-----
$tag2=$_POST['tema'.$sint.''];
$result2 = mysql_query("select * from tema where pavadinimas='$tag2'");
$row1 = mysql_fetch_array($result2) or die(mysql_error());

if($_POST['irr'.$sint.'']!='ARBA' && $_POST['irr'.$sint.'']!='ARBA' &&
$_POST['irr'.($sint+1).']']!='ARBA'){
    ?><?
}
if($row1['yra vaidmuo']==0){ //jeigu tema yra tema

if ($_POST['neT'.$sintT.'']=='ne'){?>not(<?)?>value($NAME<?echo $sintT;?>, "<?php echo
$_POST['tema'.$sintT.'']; ?>"),
topic-name($TOPICNAME<?echo $sintT;?>, $NAME<?echo $sintT;?>),
instance-of($TOPIC<?echo $sintT;?>, $TOPICNAME<?echo $sintT;?>)<?if
($_POST['neT'.$sintT.'']=='ne'){?><?>?,
<?>}elseif($row1['yra vaidmuo']==1){ //jeigu tema yra vaidmuo, arba - ir tema, ir vaidmuo

if ($_POST['neT'.$sintT.'']=='ne'){?>not(<?)?>value($NAME<?echo $sintT;?>, "<?php echo
$_POST['tema'.$sintT.'']; ?>"),
topic-name($TOPICNAME<?echo $sintT;?>, $NAME<?echo $sintT;?>),
type($SECOND_ROLE<?echo $sintT;?>, $TOPICNAME<?echo $sintT;?>)<?if
($_POST['neT'.$sintT.'']=='ne'){?><?>?,
<?
}
} //if tema
if($_POST['egzempliorius'.$sintE.'']!='){ // -----
egzempliorius -----
if($_POST['irr'.$sint.'']!='ARBA' && $_POST['irr'.($sint+1).']']!='ARBA' &&
$_POST['tema'.$sintT.'']!='){
    ?><?
}
if ($_POST['neE'.$sintE.'']=='ne'){?>not(<?)?>value($INSTANCENAME<?echo $sintE;?>, "<?php echo
$_POST['egzempliorius'.$sintE.'']; ?>"),
topic-name($INSTANCETOPICNAME<?echo $sintE;?>, $INSTANCENAME<?echo $sintE;?>),
$TOPIC<?echo $sintE;?> = $INSTANCETOPICNAME<?echo $sintE;?><?if
($_POST['neE'.$sintE.'']=='ne'){?><?>?,
<?
}

?>role-player($FIRST_ROLE<?echo $sintT;?>, $rasti),
association-role($ASSOC<?echo $assoc;?>, $FIRST_ROLE<?echo $sintT;?>),
association-role($ASSOC<?echo $assoc;?>, $SECOND_ROLE<?echo $sintT;?>),
role-player($SECOND_ROLE<?echo $sintT;?>, $TOPIC<?echo $sintT;?>),
$FIRST_ROLE<?echo $sintT;?> != $SECOND_ROLE<?echo $sintT;?>

if($_POST['irr'.($sint+1).']']!=' || $_POST['irr'.($sint+1).']']!='){
if(($_POST['irr'.$sint.'']!='ARBA' || $_POST['irr'.$sint.'']!='ARBA') &&
$_POST['irr'.($sint+1).']']!='ARBA' && $_POST['irr'.($sint+1).']']!='ARBA'){
    ?>,
<?>}elseif($_POST['irr'.($sint+1).']']!='ARBA' || $_POST['irr'.($sint+1).']']!='ARBA'){
    ?> |
<?>}else{?>,
<?>
} elseif($_POST['irr'.($sint+1).']']=='' && $_POST['irr'.($sint+1).']']=='){
if(($_POST['irr'.$sint.'']!='ARBA' || $_POST['irr'.$sint.'']!='ARBA') &&
$_POST['irr'.($sint+1).']']!='ARBA' && $_POST['irr'.($sint+1).']']!='ARBA'){
    ?><?
}
}

if($_POST['asociacija'.($sintA+1).']']=='' && $_POST['tema'.($sintT+1).']']=='' &&
$_POST['egzempliorius'.($sintE+1).']']==''){?><?> //uzklauso pabaiga

$sint++;

```

```

$intA++;
$intT++;
$intE++;
} //while
} //main if
mysql_close($con);

?>
</textarea>
</td></tr>
</table>
<?//-----ALGORITMO PABAIGA-----?>
<table width="900" border="0" cellpadding="0" cellspacing="0">
<tr>
<td align="right" valign="middle">
<input type="button" name="selectall" value="pažymėti"
onClick="javascript:this.form.textareaname.focus();this.form.textareaname.select();"/>
<input type="button" disabled=disabled name="copyall" value="Kopijuoti"
onClick='document.formname.textareaname.select()'/>
</td></tr></table></form>
<table width="900" border="0" cellpadding="0" cellspacing="0"><tr><td align="right">
<form
action="http://localhost:8080/omnigator/plugins/query/form.jsp?tm=parodos_mag.xtm&id=null&redirect=%
2Fomnigator%2Fmodels%2Ftopicmap_complete.jsp%3Ftm%3Dparodos_mag.xtm" method="POST">
<input type="submit" value="Ontopia/Omnigator" />
</td>
</tr>
</table>

```

Priedas 2

Demonstracinio teminio tinklo "Parodos" duomenys ir sąsajos.

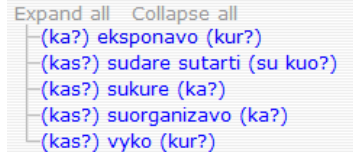
Temos:

Topic Types



Asociacijos:

Association Types



Vaidmenys:

Role Types



Temos, kurias turi egzempliorius:

Instances



Temos patalpa egzemplioriai:

Patalpa

- Kauno paveikslų galerija
- Klaipėdos dailės galerija
- Vilniaus paveikslų galerija

Temos įmonė egzemplioriai:

Imone

- Kauno parodų organizatoriai
- Kauno parodų organizatoriai
- Kauno paveikslų galerija
- Klaipėdos dailės galerija
- Klaipėdos parodų organizatoriai
- UAB "Organizuojame"
- Vilniaus parodų organizatoriai
- Vilniaus paveikslų galerija

Temos kūrinio autorius egzemplioriai:

Kurinio autorius

- Antanas Obcarskas
- Antanas Pukas
- Aušra Vaitkūnienė
- Gintautas Vaičys
- Romualdas Čarna
- Saulius Mikalauskis

Temos kūrinys egzemplioriai:

Kurinys

- Kalnai
- Kaminkrėtys
- Kaspiniai
- Migla
- Muilo burbulai
- Naujas
- Panašumas
- Papūgos
- Saulė
- Vakarėja
- Žvaigždžių vartai

Temos meno galerija egzemplioriai:

Meno galerija

- Kauno meno galerija
- Klaipėdos meno galerija
- Vilniaus meno galerija

Temos organizatorius egzempliorius:

Organizatorius

- Kauno parodų organizatoriai
- Klaipėdos parodų organizatoriai
- Vilniaus parodų organizatoriai

Temos paroda egzemplioriai:



Paroda


- Kaukė
- Miegas
- Ruduo
- Smilgų šokis

Egzempliorių sąsajos su kitais egzemplioriais:



Organizatoriai:


Kauno parodų organizatoriai


Name:  

suorganizavo renginį: [Miegas](#) 



Klaipėdos parodų organizatoriai


Name:  


sudarė sutartį su: [Klaipėdos dailės galerija](#) 

suorganizavo renginį: [Miegas](#) 

Vilniaus parodų organizatoriai




Name:  


sudarė sutartį su: [Vilniaus paveikslų galerija](#) 


suorganizavo renginį: [Smilgų šokis](#) 


Parodos:


Kaukė




renginyje buvo eksponuojamas kūrinys: [Kalnai](#)   


[Kaminkrėtys](#) 




[Kaspiniai](#) 


[Muilo burbulai](#) 


[Panašumas](#) 




renginį suorganizavo: [Kauno parodų organizatoriai](#)   

[UAB "Organizuojame"](#) 

Vyko patalpoje: [Kauno meno galerija](#)   


[Kauno paveikslų galerija](#) 

[Klaipėdos meno galerija](#) 

Paroda įvyko:   

Miegas

renginyje buvo eksponuojamas kūrinys: [Migla](#)   

[Naujas](#) 

[Vakarėja](#) 

[Žvaigždžių vartai](#) 

renginį suorganizavo: [Kauno parodų organizatoriai](#)   

[Klaipėdos parodų organizatoriai](#) 

Vyko patalpoje: [Kauno paveikslų galerija](#)   

[Klaipėdos meno galerija](#) 

[Vilniaus meno galerija](#) 

Paroda įvyko:   

Ruduo





renginyje buvo eksponuojamas kūrinys: [Žvaigždžių vartai](#)   




renginį suorganizavo: [Kauno parodų organizatoriai](#)   






Vyko patalpoje: [Kauno meno galerija](#)   
[Klaipėdos meno galerija](#) 
[Klaipėdos dailės galerija](#) 




Paroda įvyko:   

Smilgų šokis

renginyje buvo eksponuojamas kūrinys: [Papūgos](#)   
[Saulė](#) 

renginį suorganizavo: [Vilniaus parodų organizatoriai](#)   








Vyko patalpoje: [Klaipėdos meno galerija](#)   
[Vilniaus meno galerija](#) 
[Vilniaus paveikslų galerija](#) 

Paroda įvyko:   

Kūrinio autorius:

Antanas Obcarskas




Name:




sukūrė kūrinį: [Kaminkrėtys](#)   
[Kaspiniai](#) 
[Muilo burbulai](#) 
[Naujas](#) 
[Vakarėja](#) 

Kūrinys:

Kaspiniai




Name:




sukurtas kūrėjo: [Antanas Obcarskas](#)   

buvo eksponuojamas renginyje: [Kaukė](#)   

Muilo burbulai




Name:




sukurtas kūrėjo: [Antanas Obcarskas](#)   

buvo eksponuojamas renginyje: [Kaukė](#)   

Naujas




Name:




sukurtas kūrėjo: [Antanas Obcarskas](#)   

buvo eksponuojamas renginyje: [Miegas](#)   

Panašumas




Name:




sukurtas kūrėjo: [Antanas Pukas](#)   

buvo eksponuojamas renginyje: [Kaukė](#)   

Papūgos

Name:


sukurtas kūrėjo: [Saulius Mikalauskis](#)   

buvo eksponuojamas renginyje: [Smilgų šokis](#)   

Saulė




Name:




buvo eksponuojamas renginyje: [Smilgų šokis](#)   

sukurtas kūrėjo: [Aušra Vaitkūnienė](#)   

Vakarėja




Name:



buvo eksponuojamas renginyje: [Miegas](#)   

sukurtas kūrėjo: [Antanas Obcarskas](#)   

Žvaigždžių vartai


Name:



buvo eksponuojamas renginyje: [Miegas](#)   
[Ruduo](#) 

sukurtas kūrėjo: [Gintautas Vaičys](#)   

Migla

Name:




sukurtas kūrėjo: [Gintautas Vaičys](#)   





buvo eksponuojamas renginyje: [Miegas](#)   

Patalpa:

Kauno paveikslų galerija




Name:




sudarė sutartį su: [Kauno parodų organizatoriai](#)   

Vyko renginys: [Kaukė](#)   
[Miegas](#) 

Klaipėdos dailės galerija





Name:




sudarė sutartį su: [Klaipėdos parodų organizatoriai](#)   

Vyko renginys: [Ruduo](#)   

Vilniaus paveikslų galerija

Name:





sudarė sutartį su: [Kauno parodų organizatoriai](#)   
[Vilniaus parodų organizatoriai](#) 

Vyko renginys: [Smilgų šokis](#)   

Meno galerija:







Kauno meno galerija

Name:

Vyko renginys: [Kaukė](#)   
[Ruduo](#) 





Klaipėdos meno galerija

Name:

Vyko renginys: [Kaukė](#)   
[Miegas](#) 
[Ruduo](#) 
[Smilgų šokis](#) 

Vilniaus meno galerija





Name:





Vyko renginys: [Miegas](#)   
[Smilgų šokis](#) 

Įmonė:

Kauno parodų organizatoriai

Name:

sudarė sutartį su: [Kauno paveikslų galerija](#)   
[Vilniaus paveikslų galerija](#) 

suorganizavo renginį: [Kaukė](#)   
[Ruduo](#) 

Kauno paveikslų galerija

Name:

sudarė sutartį su: [Kauno parodų organizatoriai](#) ⊖ + *

Vyko renginys: [Kaukė](#) ⊖ + *
[Miegas](#) ⊖

Klaipėdos dailės galerija

Name:

sudarė sutartį su: [Klaipėdos parodų organizatoriai](#) ⊖ + *

Vyko renginys: [Ruduo](#) ⊖ + *

UAB "Organizuojame"

Name:

suorganizavo renginį: [Kaukė](#) ⊖ + *

Asociacijų vaidmenys ir jų atlikėjai:

(ka?) eksponavo (kur?)

Name:

Subject identifier: >> ⊖ + ⊖

Description: ⊖

Symmetric:

Association field:

Roles:

Name: >> ⊖

Role type: [Kurinys](#) ⊖ + *

Used by: [Paveikslas](#) ⊖ + *
[Skulptura](#) ⊖

Cardinality: [One or more](#) ⊖ + *

Interface control: [Drop-down list](#) ⊖ + *

Name: >> ⊖

Role type: [Renginys](#) ⊖ + *

Used by: [Paroda](#) ⊖ + *

Cardinality: [One or more](#) ⊖ + *

Interface control: [Drop-down list](#) ⊖ + *

(kas?) sudare sutarti (su kuo?)

Name: (kas?) sudare sutarti (su kuo?)

Subject identifier:

Description: Vienos įmonės turi sutartis su kitomis. Pvz.: Kauno parodų organizatoriai turi sutartį su Kauno paveikslų galerija.

Symmetric:

Association field:

Roles:

Name: sudarė sutartį su

Role type: Įmonė

Used by: Organizatorius

Cardinality: Pasirinkite

Interface control: Pasirinkite

Name: sudarė sutartį su

Role type: Įmonė

Used by: Patalpa

Cardinality: Zero or more

Interface control: Search dialog

(kas?) sukure (ka?)

Name: (kas?) sukure (ka?)

Subject identifier:

Description: Kūrinio autorius sukūrė vieną ar daugiau kūrinį.

Symmetric:

Association field:

Roles:

Name: sukurtas kūrėjo

Role type: Kurinys

Used by: Paveikslas
Skulptura

Cardinality: Exactly one

Interface control: Drop-down list

Name: sukūrė kūrinį

Role type: Kurėjas

Used by: Kurinio autorius

Cardinality: One or more

Interface control: Drop-down list

(kas?) suorganizavo (ka?)

Name:

Subject identifier: >> - +

Description: -

Symmetric:

Association field: *

Roles: *

Name:	<input type="text" value="renginj suorganizavo"/>	>> -
Role type:	Renginys -	+ *
Used by:	Paroda -	+ *
Cardinality:	One or more -	+ *
Interface control:	Drop-down list -	+ *

Name:	<input type="text" value="suorganizavo renginj"/>	>> -
Role type:	Organizatorius -	+ *
Used by:	Imone - Organizatorius -	+ *
Cardinality:	Zero or more -	+ *
Interface control:	Drop-down list -	+ *

(kas?) vyko (kur?)

Name:

Subject identifier:

Description:

Symmetric:

Association field:

Roles:

Name:

Role type: Renginys ⊖

Used by: Paroda ⊖

Cardinality: One or more ⊖

Interface control: Drop-down list ⊖

Name:

Role type: Patalpa ⊖

Used by: Meno galerija ⊖
Patalpa ⊖

Cardinality: One or more ⊖

Interface control: Drop-down list ⊖