

KAUNO TECHNOLOGIJOS UNIVERSITETA  
INFORMATIKOS FAKULTETAS  
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

ARVYDAS POCIUS

SBVR STRUKTŪRIZUOTOS NATŪRALIOS KALBOS  
KLAUSIMŲ REDAGAVIMO ĮRANKIS

Magistro darbas

Darbo vadovas  
prof. dr. Lina Nemuraitė

Konsultantas  
mag. A. Šukys

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

ARVYDAS POCIUS

SBVR STRUKTŪRIZUOTOS NATŪRALIOS KALBOS  
KLAUSIMŲ REDAGAVIMO ĮRANKIS

Magistro darbas

Darbo vadovas  
prof. dr. Lina Nemuraitė  
2013-05-22

Konsultantas  
mag. A. Šukys

Recenzentas  
Doc. dr. Vytautas Pilkauskas  
2013-05-24

Atliko  
IFM-1/4 gr. Studentas  
Arvydas Pocius  
2013-05-22

KAUNAS, 2013

## AUTENTIŠKUMO PATVIRTINIMAS

### AUTORIŲ GARANTINIS RAŠTAS

### DĖL PATEIKIAMO KŪRINIO

2013 - Gegužės - 22d.

Kaunas

**Autoriai,** \_\_\_\_\_ Arvydas Pocius \_\_\_\_\_  
(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis magistro darbas (toliau vadinama – Kūrinys) SBVR struktūrizuotos natūralios kalbos klausimų redagavimo įrankis  
(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektualios nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

**Autoriai**

\_\_\_\_\_ Arvydas Pocius \_\_\_\_\_  
(vardas, pavardė) (parašas)

## **SANTRAUKA**

### **SBVR struktūrizuotos natūralios kalbos klausimų redagavimo įrankis**

Ontologijos užklausų rašymas paprastiems vartotojams yra per daug sudėtingas. Vartotojai labiausiai mėgsta pateikti užklausas natūralia kalba. Tačiau dar nėra sukurtų programų, kurios suprastų tikrą natūralią kalbą. Šiuo metu galimas sprendimas yra naudoti ribotą (struktūrizuotą) natūralią kalbą, kuri sudaroma pagal tam tikrus šablonus. Sukūrus *SBVR* metamodeliu grindžiamą struktūrizuotos natūralios kalbos užklausų redaktorių, vartotojui nereikėtų specialistų pagalbos norint peržiūrėti ontologijų duomenis. Informaciją būtų galima peržiūrėti rašant klausimus struktūrizuota natūralia kalba. Šiame darbe nekeliamas tikslas iš struktūrizuotos kalbos klausimų gauti *SPARQL* užklausas, kadangi tai būtų per didelis uždavinys. Šio darbo tikslas yra atpažinti ir struktūrizuoti įvedamą tekstą, kad vėliau būtų galima generuoti *XMI* schemą ir transformuoti ją į *SPARQL* užklausas.

## **SUMMARY**

### **Tool for Editing Questions in SBVR Structured Natural Language**

Ontology query writing for ordinary users too complicated. People mostly like to submit queries in natural language. However, software systems that could understand natural language do not exist yet. A possible solution is to use a limited (structured) natural language, which is formed according to certain patterns. Users having a structured natural language question editor based on SBVR metamodel would be able to analyse ontology data without help of professionals. They could write questions in structured natural language which is understandable for humans and computers. This work is not intended to transform structured questions into SPARQL queries, since it would be too big task. The aim of this work is to identify and formalize textual questions, typed by users, for subsequent generation of XMI schema and transforming them into a SPARQL queries.

## PAVEIKSLELIŲ SĄRAŠAS

1.1 pav. <i>Nlp-Reduce</i> sąsajos langas .....	12
1.2 pav. <i>Querix</i> sąsajos langas .....	13
1.3 pav. <i>Ginseng</i> sąsajos langas .....	14
1.4 pav. <i>Semantic Crystal</i> sąsajos langas .....	15
1.5 pav. <i>Nlp-Reduce</i> klausimo apdorojimas.....	15
1.6 pav. <i>Querix</i> klausimo apdorojimas .....	16
1.7 pav. <i>Ginseng</i> klausimo apdorojimas .....	16
1.8 pav. <i>SBVR Visual Editor</i> sąsajos langas.....	17
1.9 pav. <i>VeTIS tool</i> sąsajos langas .....	18
1.10 pav. Užklausų kūrimas naudojantis programuotojo paslaugomis.....	20
1.11 pav. Paieška dalykinėje srityje sukūrus redaktorių .....	21
1.12 pav. <i>SBVR</i> metamodelis klausimui sudaryti .....	22
2.1 pav. <i>SBVR</i> struktūrizuotos natūralios kalbos užklausų redaktoriaus PA modelis.....	23
2.2 pav. Klausimų formavimo procesas .....	25
2.3 pav. Klausimo tipo išrinkimas .....	26
2.4 pav. Koncepto pasirinkimas .....	26
2.5 pav. Žodžio formos pasirinkimas.....	26
2.6 pav. Fakto tipo pasirinkimas .....	27
2.7 pav. Klausimų esybių klasių diagrama .....	27
3.1 pav. Sistemos loginė architektūra .....	28
3.2 pav. Analizės klasių diagrama .....	28
3.3 pav. Redaktoriaus klasių modelis.....	29
3.4 pav. <i>SBVR</i> struktūrizuotos natūralios kalbos užklausų redaktoriaus sekų diagrama .....	30
3.5 pav. Klausimų formavimo sekos diagrama.....	31
3.6 pav. Sistemos programinių komponentų diagrama.....	32
3.7 pav. Sistemos programinių komponentų diagrama.....	32
3.8 pav. Klausimų redaktoriaus diegimo diagrama.....	33
4.1 Sukurto redaktoriaus langas, termino įvedimas .....	35
4.2 Sukurto redaktoriaus langas, veiksmožodžio įvedimas.....	35
4.3 Sukurto redaktoriaus langas, termino įvedimas .....	35
4.4 Sukurto redaktoriaus langas, vardo įvedimas .....	36

## LENTELIŲ SĄRAŠAS

1.1 lentelė. Natūralios kalbos užklausų kūrimo sistemų analizės rezultatai .....	17
1.2 lentelė. Paaiškinti simboliai .....	17
2.1 lentelė. Panaudojimo atvejo „Kurti klausimą“ aprašymas.....	24
2.2 lentelė. Panaudojimo atvejo „Padėti formuluoti klausimo dalį“ aprašymas.....	24
2.3 lentelė. Panaudojimo atvejo „Padėti formuoti užklausą“ aprašymas.....	25
5.1 lentelė. Pirmo žodyno sudarytų klausimų lentelė .....	39
5.2 lentelė. Antro žodyno sudarytų klausimų lentelė.....	40
5.3 lentelė. Natūralios kalbos užklausų apdorojimo sistemos .....	41
5.4 lentelė. Paaiškinti simboliai .....	42

# TURINYS

<b>IVADAS</b> .....	<b>9</b>
<b>1. SBVR STRUKTŪRIZUOTOS NATŪRALIOS KALBOS UŽKLAUSŲ FORMULAVIMO PROCESO ANALIZĖ</b> .....	<b>11</b>
1.1. TYRIMO PROBLEMA, SRITIS IR OBJEKTAS .....	11
1.2. TYRIMO TIKSLAS IR UŽDAVINIAI.....	11
1.3. ESAMŲ SPRENDIMŲ ANALIZĖ.....	12
1.3.1. <i>Esami natūralios kalbos užklausų formavimo įrankiai</i> .....	12
1.3.2. <i>Esami SBVR redagavimo įrankiai</i> .....	17
1.3.3. <i>Eclipse generatorių kūrimo ir gramatikų generavimo technologijos</i> .....	18
1.4. SBVR METAMODELIO IR KLAUSIMŲ FORMULAVIMO GALIMYBIŲ ANALIZĖ .....	19
1.4.1. <i>SBVR struktūrizuotos natūralios kalbos užklausų specifikavimo galimybių analizė</i> .....	19
1.4.2. <i>SBVR užklausų šablonai ir sudarymo principai</i> .....	20
1.5. SIEKIAMAS SPRENDIMAS .....	20
<b>2. REIKALAVIMŲ SPECIFIKACIJA, FUNKCINIAI REIKALAVIMAI</b> .....	<b>23</b>
2.1. PANAUDOJIMO ATVEJU MODELIS .....	23
2.1.1. <i>Veiklos procesų modelis</i> .....	25
2.1.2. <i>Vartotojo interfeiso modelis</i> .....	26
2.2. DALYKINĖS SRITIES MODELIS .....	27
2.3. REIKALAVIMŲ ANALIZĖS APIBENDRINIMAS .....	27
<b>3. SBVR KLAUSIMŲ REDAKTORIAUS PROJEKTAS</b> .....	<b>28</b>
3.1. SISTEMOS ARCHITEKTŪROS PROJEKTAS.....	28
3.1.1. <i>Sistemos loginė architektūra</i> .....	28
3.1.2. <i>Panaudojimo atvejų analizės klasių modelis</i> .....	28
3.1.3. <i>Bendras klasių modelis</i> .....	29
3.1.4. <i>Bendras redaktoriaus klasių modelis</i> .....	29
3.1. SISTEMOS ELGSENOS MODELIS .....	29
3.2. REALIZACIJOS MODELIS.....	32
3.2.1. <i>Programinių komponentų architektūra</i> .....	32
3.2.2. <i>Diegimo modelis</i> .....	33
<b>4. REDAKTORIAUS REALIZACIJA</b> .....	<b>34</b>
4.1. SISTEMOS VEIKIMAS .....	34
4.2. SISTEMOS VEIKIMO APRAŠYMAS .....	34
4.3. TESTAVIMO MODELIS IR DUOMENYS, KONTROLINIS PAVYZDYS.....	36
<b>5. EKSPERIMENTINIS SISTEMOS TYRIMAS</b> .....	<b>38</b>
5.1. EKSPERIMENTO PLANAS .....	38
5.2. EKSPERIMENTO REZULTATAI.....	41
<b>6. IŠVADOS</b> .....	<b>43</b>
<b>7. LITERATŪRA</b> .....	<b>44</b>



## IVADAS

Pastaruoju metu informacinių technologijų srityje didelis dėmesys skiriamas semantinei paieškai, kuriai vykdyti taikomos ontologijos ir ontologijų užklausų kalbos. Ontologijos užklausų rašymas paprastiems vartotojams yra per daug sudėtingas. Vartotojai labiausiai mėgsta pateikti užklausas natūralia kalba. Tačiau dar nėra sukurtų programų, kurios suprastų tikrą natūralią kalbą.

Šiame darbe sprendžiama problema – padėti vartotojams formuluoti semantines užklausas, kurias būtų galima vykdyti duomenyse, suformuotuose pagal atitinkamų sričių ontologijas. Ieškant šios problemos sprendimo, buvo atlikta ontologijų ir jų užklausų kalbų bei esamų užklausų formulavimo metodų ir įrankių analizė. Buvo išanalizuoti:

1. *Nlp-Reduce* – tekstinis klausimų kūrimo įrankis. *Nlp-Reduce* kiekvienam klausimo žodžiui ieško ekvivalentaus varianto *OWL* žinių bazėje. Rasti duomenys apjungiami ir pateikiami vartotojui kaip rezultatas. Vartotojas matydamas rezultatą gali arba patikslinti užklausa, arba ją pakeisti.
2. *Querix* – tekstinis klausimų kūrimo įrankis. *Querix* kiekviena klausimo žodį suranda *OWL* žinių bazėje. Kiekvienai neatpažintai klausimo daliai įrankis prašo papildomos informacijos. Pavyzdžiui yra klausimas: didžiausias miestas Lietuvoje? Įrankis paklaus ką reiškia didžiausias ir pateiks pasirinkimo variantus, ar tai populiacija, ar tai užimamas plotas?
3. *Ginseng* – tekstinis klausimo kūrimo įrankis. Įrankis naudoja *OWL* žinių bazę. *Ginseng* kiekvienai dalykinei sričiai naudoja papildoma informacija, kuri suvedama sistemos kūrėju. Dėlto įrankis reikalauja papildomo pasiruošimo, prieš sudarant klausimus iš *OWL* žinių bazės.
4. *Semantic Crystal* – grafinis formalių klausimų kurio įrankis. Įrankis naudoja *OWL* žinių bazę. Kiekvienas žinių bazės elementas atvaizduojamas, vartotojui reikia pasirinkti greta einančius elementus. Klausimai sudaromi tik iš elementų rastų žinių bazėje.

*OMG SBVR* metamodelio ir *ISK* sukurto veiklos taisyklių redaktoriaus analizė parodė, kad užklausas galima formuluoti *SBVR* struktūrizuotos kalbos klausimų pavidalu. *SBVR* yra formali kalba, panaši į natūralią. Klausimams pateikti bus sudaromi šablonai pagal kurios, bus galima kurti klausimą iš *SBVR* elementų.

Atlikta analizė leido daryti išvadą, kad šiuo metu galimas sprendimas yra naudoti ribotą (struktūrizuotą) natūralią kalbą, kuri sudaroma pagal tam tikrus šablonus. Sukūrus *SBVR* metamodeliu grindžiamą struktūrizuotos natūralios kalbos užklausų redaktorių, vartotojui nereikėtų specialistų pagalbos norint peržiūrėti ontologijų duomenis. Informaciją būtų galima peržiūrėti rašant klausimus struktūrizuota natūralia kalba □[ 1 ]. Šiame darbe nekeliamas tikslas iš struktūrizuotos kalbos užklausų gauti *SPARQL* užklausas, kadangi tai būtų per didelis uždavinys. Šio darbo tikslas yra atpažinti ir struktūrizuoti įvedamą tekstą, kad vėliau būtų galima generuoti *XMI* schemą ir transformuoti ją į *SPARQL* užklausas.

Šio darbo **tyrimo sritis** yra *SBVR* (angl. *Semantics of Business Vocabulary and Business Rules*) struktūrizuotos natūralios kalbos užklausų šablonai, jų sudarymo ir atpažinimo technologijos.

Šio darbo **tyrimo objektas** yra struktūrizuotos natūralia kalba specifikuotų klausimų analizės procesas.

**Tikslas** – suteikti galimybes semantinės paieškos naudotojams lengviau (nerašant *SPARQL* užklausų) formuluoti ontologiją atitinkančias semantines užklausas sukuriant tam skirtą *SBVR* metamodeliu grindžiamą, struktūrizuotos natūralios kalbos užklausų formulavimo redaktorių.

### Uždaviniai:

#### 1. Išanalizuoti

- 1.1. esamus užklausų kūrimo struktūrizuotomis kalbomis įrankius;
- 1.2. *SBVR* struktūrizuotos natūralios kalbos klausimų specifikuojamą galimybes;

- 1.3. *SBVR* klausimų pateikimo šablonus ir jų sudarymo principus;
- 1.4. esamus *SBVR* redagavimo įrankius;
- 1.5. *Eclipse* generatorių kūrimo ir gramatikų generavimo technologijas, pasirinkti iš jų tinkamiausią.
2. Sudaryti tipinius *SBVR* struktūrizuotos natūralios kalbos klausimų šablonus, aprašyti šių klausimų pateikimo šablonus.
3. Integruoti šablonus į tekstinį redaktorių, kuris galėtų analizuoti ir atpažinti įvedamus klausimus.
4. Aprašyti pasirinktos dalykinės srities žodyną ir atlikti eksperimentą, kuris leistų iširti sukurto redaktoriaus galimybes atpažinti įvairių tipų klausimus.
5. Įvertinti darbo rezultatus.

**Darbo struktūra.** Darbą sudaro penki skyriai. Pirmame skyriuje aprašoma tyrimo tikslas, sritis, objektas, uždaviniai, esami sprendimai ir analizės išvados. Antrame skyriuje aprašoma redaktoriaus reikalavimai, dalykinės srities modelis. Trečiame skyriuje aprašomi sistemos projektai: architektūros, elgsenos ir realizacijos modelis. Ketvirtame skyriuje aprašoma sistemos veikimas, testavimo modelis. Penktame skyriuje aprašomas eksperimentas ir jo rezultatas.

# 1. SBVR struktūrizuotos natūralios kalbos užklausų formulavimo proceso analizė

**Analizės tikslas** yra išanalizuoti esamus užklausų kūrimo struktūrizuotomis kalbomis įrankius; *SBVR* struktūrizuotos natūralios kalbos užklausų specifikavimo galimybes; *SBVR* užklausų pateikimo šablonus ir jų sudarymo principus; esamus *SBVR* redagavimo įrankius; *Eclipse* generatorių kūrimo ir gramatikų generavimo technologijas, pasirinkti iš jų tinkamiausią.

Norint išanalizuoti *SBVR* klausimų struktūrą, bus taikomas dalykinės srities tyrimo metodas. Analizės metu bus nagrinėjami pagrindiniai *SBVR* metamodelio konceptai, jų paskirtis, ryšiai su kitais konceptais ir t. t. Atlikus analizę, turėtų paaiškėti *SBVR* klausimų struktūra.

## 1.1. Tyrimo problema, sritis ir objektas

**Tyrimo problema.** Ontologijos užklausų rašymas paprastiems vartotojams yra per daug sudėtingas. Verslo darbuotojams svarbu, kad informacijos gavimo procesas būtų kuo paprastesnis. Vartotojai labiausiai mėgsta pateikti užklausas natūralia kalba. Tačiau dar nėra sukurtų programų, kurios suprastų tikrą natūralią kalbą. Programuotojams užklausų rašymas neturėtų sudaryti problemų.

Šiuo metu verslo darbuotojams duomenų gavimas be informatikų įsikišimo per daug sudėtingas, todėl jie turi kreiptis į programuotojus. Todėl siekiama palengvinti užklausų sudarymą, sukuriant priemones, leidžiančias rašyti užklausas struktūrizuota kalba. Užklausų rašymas šia kalba dalykinės srities specialistams neturėtų būti sudėtingas (kadangi kalba panaši į natūralią), jiems tereiktų išsiaiškinti, kokias struktūrizuotos natūralios kalbos struktūras jie gali naudoti.

Šio darbo **tyrimo sritis** yra *SBVR* (angl. *Semantics of Business Vocabulary and Business Rules*) struktūrizuotos natūralios kalbos užklausų šablonai, jų sudarymo ir atpažinimo technologijos. Veiklos žodyno ir taisyklių semantika *SBVR* aprašo veiklos konceptų, veiklos taisyklių ir klausimų metamodelį, kurį galima taikyti struktūrizuotos kalbos užklausų sudarymui [5][15]. Norint formuoti struktūrizuotos kalbos užklausas, pirmiausia reikia aprašyti atitinkamos dalykinės srities žodyną, pagal kurį galima formuluoti užklausas ir gauti iš jų atsakymus. Norint atpažinti šias užklausas ir paversti į programinės įrangos apdorojamą užklausų kalbą, pirmiausia reikia išanalizuoti įvestą tekstą ir išsaugoti jį struktūrizuotu formatu. Šiam tikslui šiame darbe bus kuriamas redaktorius, kuris leis įvesti struktūrizuotos kalbos užklausas, jas išanalizuos ir išsaugos reikiamu formatu. Redaktorius bus paremtas *Eclipse* redaktorių kūrimo, gramatikų generavimo (*AntLR*, *Xtext*) ir kitomis panašiomis technologijomis.

Šio darbo **tyrimo objektas** yra struktūrizuota natūralia kalba specifikuotų užklausų analizės procesas.

Vartotojas įveda klausimus struktūrizuotu tekstu, laikydamasis tam tikrų šablonų. Redaktoriaus programa analizuoja tekstą, randa jai pažįstamus šablono elementus, atskiria juos skirtukais ir pažymi jų tipus. Jei tekstas sudarytas ne pagal taisykles, redaktorius jo neatpažįsta. Iš redaktoriaus apdoroto teksto galima sugeneruoti standartinę *SBVR XMI* schemą, kuri gali būti transformuojama į ontologijų užklausų kalbos *SPARQL* užklausas.

## 1.2. Tyrimo tikslas ir uždaviniai

**Tyrimo tikslas** – suteikti galimybes semantinės paieškos naudotojams lengviau (nerašant *SPARQL* užklausų) formuluoti ontologiją atitinkančias semantines užklausas sukuriant tam skirtą *SBVR* metamodeliu grindžiamą struktūrizuotos natūralios kalbos užklausų formulavimo redaktorių, kuris leistų formuoti apibrėžtų tipų klausimus.

### Uždaviniai:

#### 1. Išanalizuoti

1.1. esamus užklausų kūrimo struktūrizuotomis kalbomis įrankius;

1.2. *SBVR* struktūrizuotos natūralios kalbos klausimų specifikavimo galimybes;

- 1.3. SBVR klausimų pateikimo šablonus ir jų sudarymo principus;
- 1.4. esamus SBVR redagavimo įrankius;
- 1.5. Eclipse generatorių kūrimo ir gramatikų generavimo technologijas, pasirinkti iš jų tinkamiausią.
2. Sudaryti tipinius SBVR struktūrizuotos natūralios kalbos klausimų šablonus, aprašyti šių klausimų pateikimo šablonų gramatiką.
3. Integruoti gramatiką į tekstinį redaktorių, kuris galėtų analizuoti ir atpažinti įvedamus klausimus.
4. Aprašyti pasirinktos dalykinės srities žodyną ir atlikti eksperimentą, kuris leistų ištirti sukurto redaktoriaus galimybes atpažinti įvairių tipų klausimus.
5. Įvertinti darbo rezultatus.

### 1.3. Esamų sprendimų analizė

#### 1.3.1. Esami natūralios kalbos užklausų formavimo įrankiai

Viena iš galimybių transformuoti SBVR į SQL užklausų kalbą buvo nagrinėjama [ 6 ], tačiau jis nesusijęs su semantine paieška. Toliau bus analizuojami įrankiai, kurie buvo naudojami [ 7 ], [ 14 ] patikrinti struktūrizuotų natūralios kalbos užklausų naudą vartotojams.

#### Sistemų apžvalga:

- *Nlp-Reduce* (1.1 pav.) – iš įvesto klausimo generuoja visas galimas užklausas. Iš visų galimų atsakymų parenka tik vieną [ 11 ].

**Query 0**

```
select distinct * WHERE {
  ?Restaurant <http://www.mooney.net/restaurant#isIn> ?City .
  ?Restaurant <http://www.mooney.net/restaurant#foodType> 'chinese' .
  ?City <http://www.mooney.net/restaurant#label> 'san francisco' .
  ?City <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.mooney.net/restaurant#City> .
  ?Restaurant <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.mooney.net/restaurant#Restaurant>
}
```

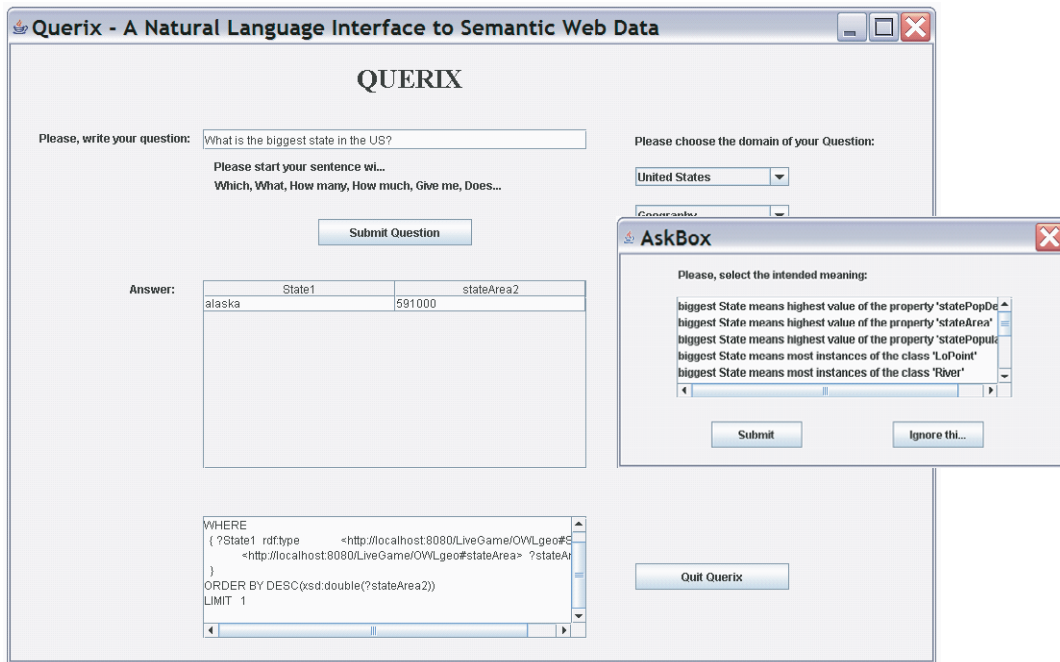
172 rows found

Restaurant	City
http://www.mooney.net/restaurant#ID_sunWahKue4686	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_luckyCreation6548	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_pekingRestaurant552	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_pandaRestaurant8236	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_camHungChineseRestaurant7591	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_hennyChungSOriginalHunanRestaurant2720	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_snowGarden9546	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_missionHunanRestaurant6989	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_hunanHomesRestaurant535	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_pearlCity6770	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_heungYuenRestaurant8480	http://www.mooney.net/restaurant#sanFrancisco
http://www.mooney.net/restaurant#ID_chinaVillageSeafoodAndVegeta3711	http://www.mooney.net/restaurant#sanFrancisco

Tue May 08 14:35:00 CEST 2007 Application started  
 Tue May 08 14:43:38 CEST 2007  
 Tue May 08 14:43:41 CEST 2007 Question: 'chinese restaurants in san francisco'  
 Tue May 08 14:43:41 CEST 2007 Parsed question: 'chinese restaurant in san\_francisco'  
 Tue May 08 14:43:41 CEST 2007 SPARQL-Queries:  
 Tue May 08 14:43:41 CEST 2007 select distinct \* WHERE { ?Restaurant <http://www.mooney.net/restaurant#isIn> ?City . ?Restaurant <http://www.mooney.net/restaurant#foodType> 'chinese' . ?City <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.mooney.net/restaurant#City> . ?Restaurant <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.mooney.net/restaurant#Restaurant> }

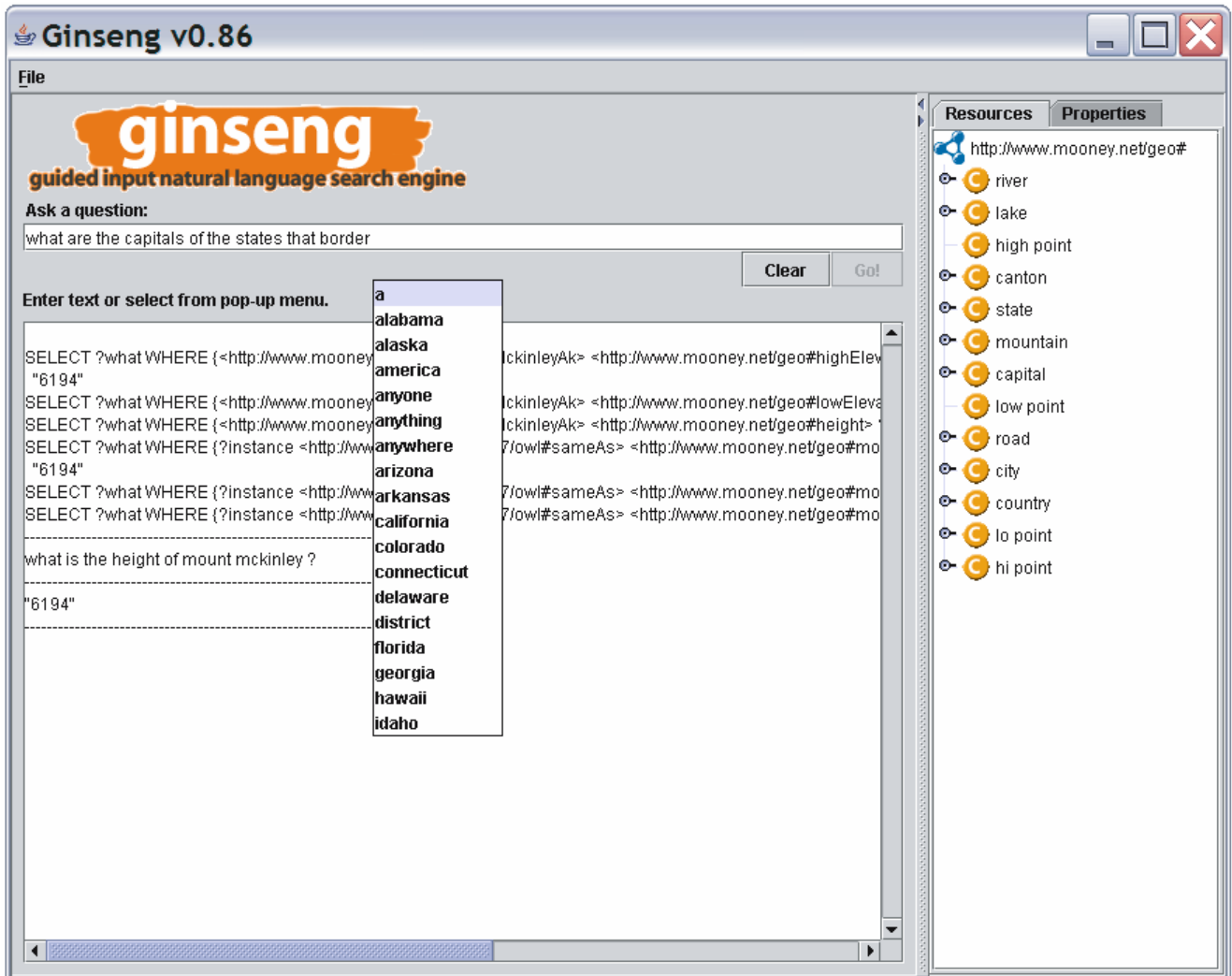
1.1 pav. *Nlp-Reduce* sąsajos langas

- *Querix* (1.2 pav.) – įvestą klausimą analizuoja dalimis. Nežinomiems žodžiams prašo nustatyti reikšmę [ 12 ].



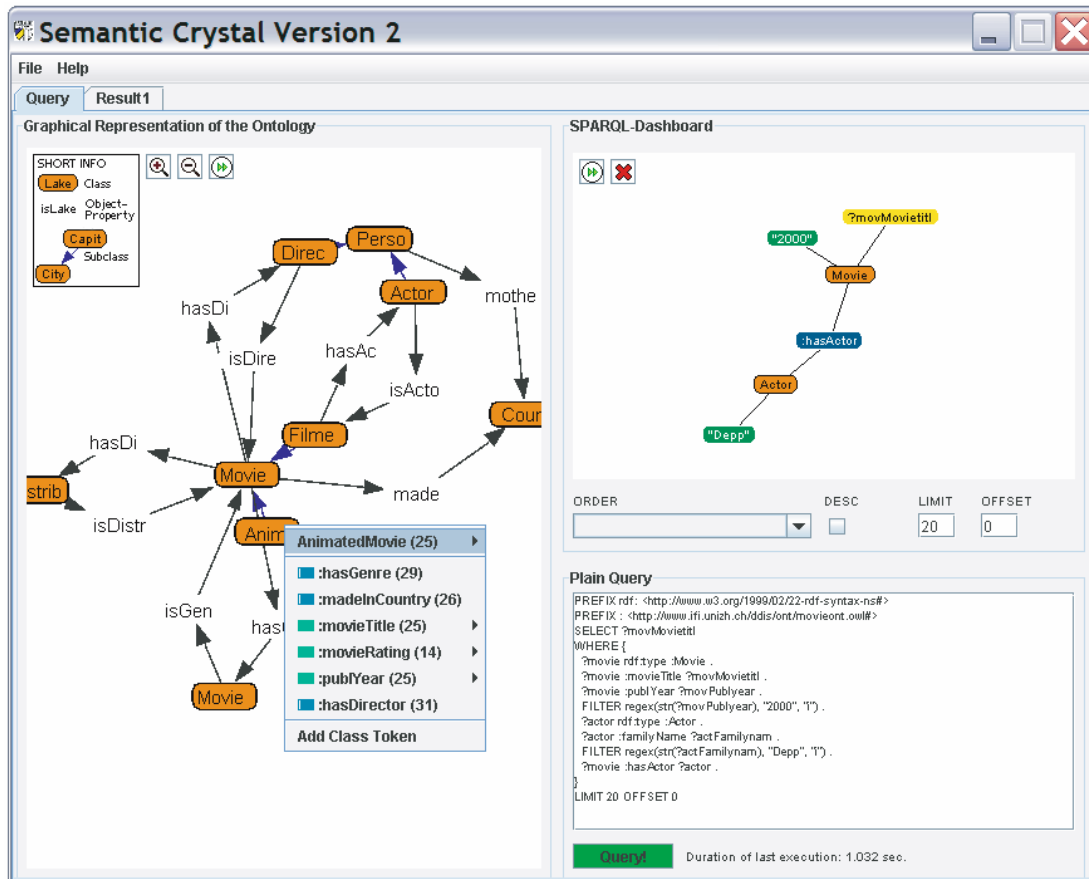
1.2 pav. *Querix* sąsajos langas

- *Ginseng* (1.3 pav.) – visi klausimo variantai aprašomi sistemos kūrėjų. Klausimą galima sudaryti iš žodžių pateiktų sistemos □ [ 10 ].



1.3 pav. *Ginseng* sąsajos langas

- *Semantic Crystal* (1.4 pav.) – klausimams sudaryti naudojama grafine vartotojo sąsaja. Klausimus sudarinėjami iš stačiakampių ir linijų □ [ 13 ].



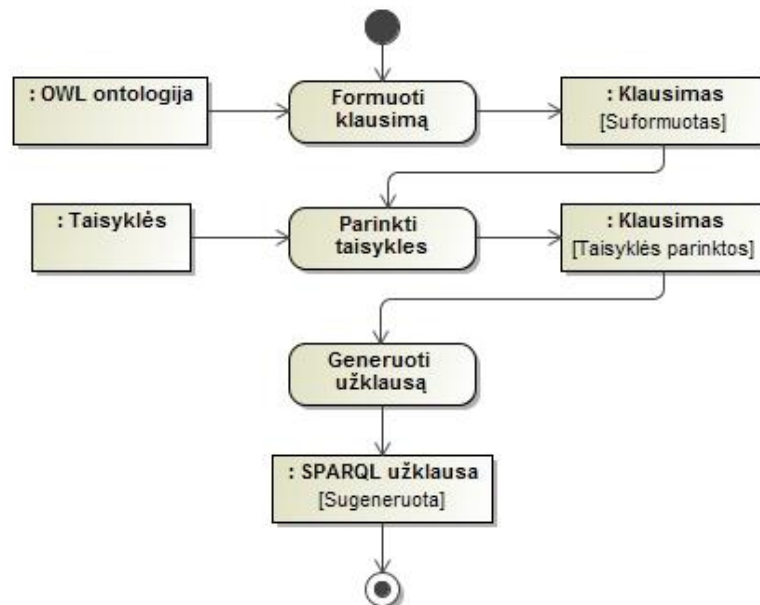
1.4 pav. *Semantic Crystal* sąsajos langas

### Sistemų klausimų apdorojimo procesai:

#### *Nlp-Reduce*

„Formuoti klausimą“ – klausime suranda visas dalis sutampančias su žinių baze. Žodžiai neturintys prasmės žinių bazėje, ignoruojami.

„Parinkimas“ – iš parinktų taisyklių parenka geriausią variantą.



1.5 pav. *Nlp-Reduce* klausimo apdorojimas

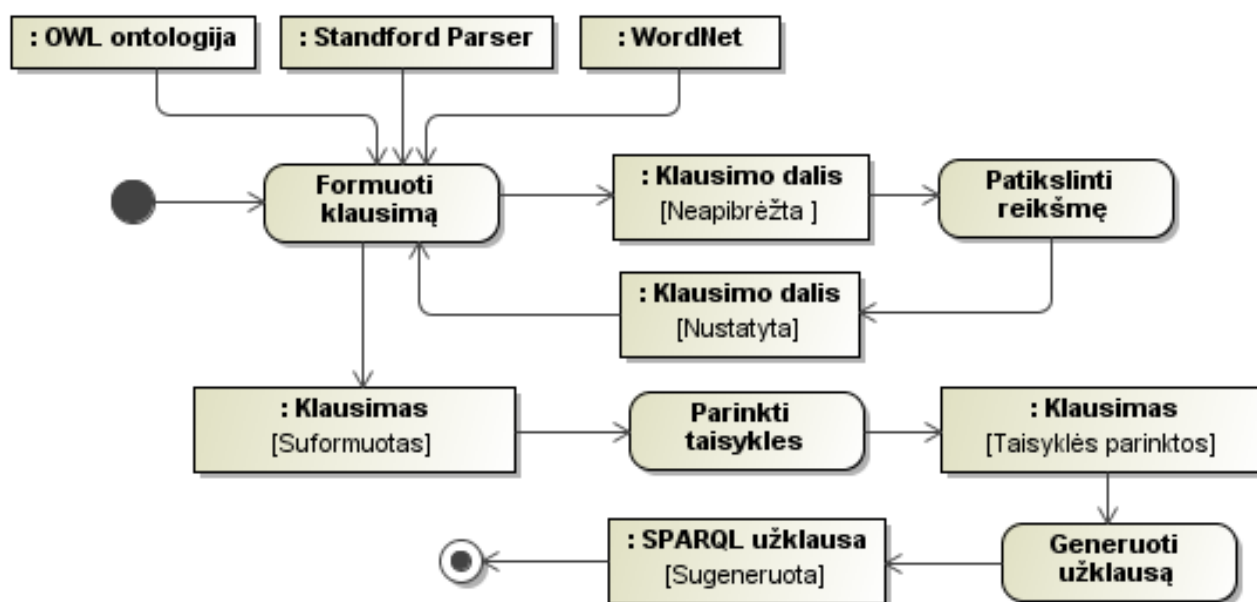
#### *Querix*

„*WordNet*“ – visiems klausimo žodžiams randa sinonimą.

„*Stanford Parser*“ – klausimus išskaido sakinio dalimis.

„Formuoti klausimą“ – klausime suranda visas dalis sutampančias su žinių baze. Visus žodžius kurių nerado prašo patikslinti vartotojo.

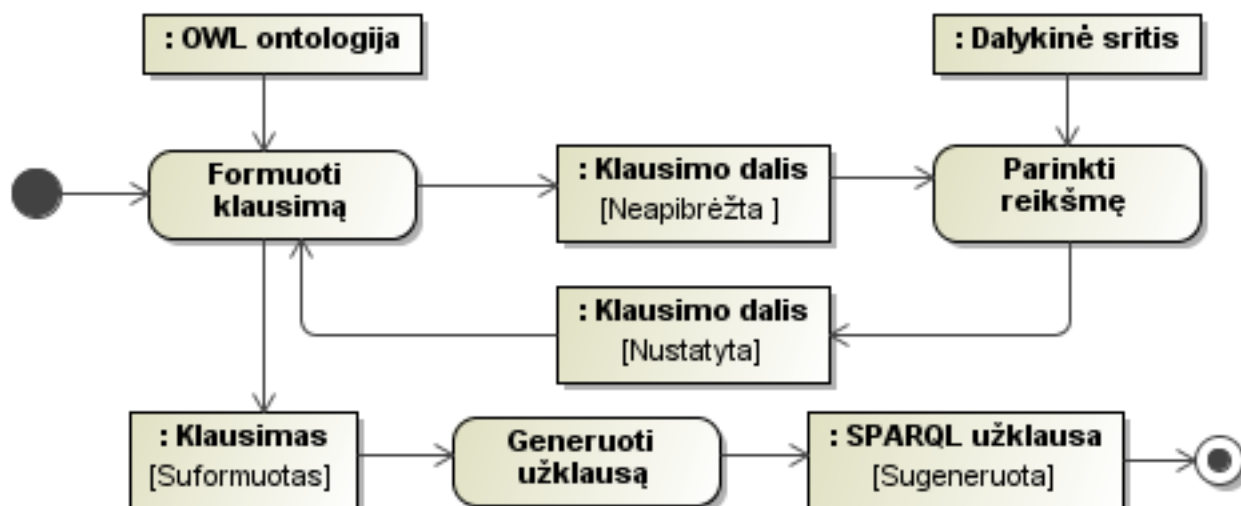
„Parinkti taisykles“ – pagal klausimo struktūrą ir taisykles parenka taisyklių tvarką.



1.6 pav. *Querix* klausimo apdorojimas

### *Ginseng*

„Formuoti klausimą“ – vartotojui iškarto siūlomi galimi žodžių variantai klausimui sudaryti.



1.7 pav. *Ginseng* klausimo apdorojimas

### *Semantic Crystal*

grafiškai formuluoja sakinius.

Visos nagrinėtos sistemos turi savų plusų ir minusų. Sistemos plusai ir minusai daugiausia priklausė nuo veikimo būdo.

- *Nlp-Reduce* – apdoroja visus klausimus, kuriems randa taisyklę žinių bazėje.
- *Querix* – apdorojamų klausimų kiekis mažesnis už *Nlp-Reduce*. Klausimą išskaido pagal gramatikos taisykles.
- *Ginseng* – klausimams sudaryti naudoja žodžius aprašytus sistemoje.



- *Semantic Crystal* – klausimas formuojamas grafiškai, klausimas sudaromas iš esybių ir ryšių tarp jų.  
Esminiai sistemų skirtumai palyginti 1.1 lentelėje.

1.1 lentelė. Natūralios kalbos užklausų kūrimo sistemų analizės rezultatai

Teiginys	Sistema	<i>Nlp-Reduce</i>	<i>Querix</i>	<i>Ginseng</i>	<i>Semantic Crystal</i>
Kalba panaši į natūralią		9	7	5	1
Klausimus priima tekstiniu formatu (ne grafiniu)		+	+	+	-
Neleidžia įvesti nekorektiškų klausimų		+	+	-	-
Sistema padeda formuluoti klausimus, pateikia raktinius žodžius		-	-	+	-
Klausimams atpažinti naudoja sinonimų žodyną		-	+	-	-
Klausimams nereikia įvesti papildomos informacijos		+	-	-	-

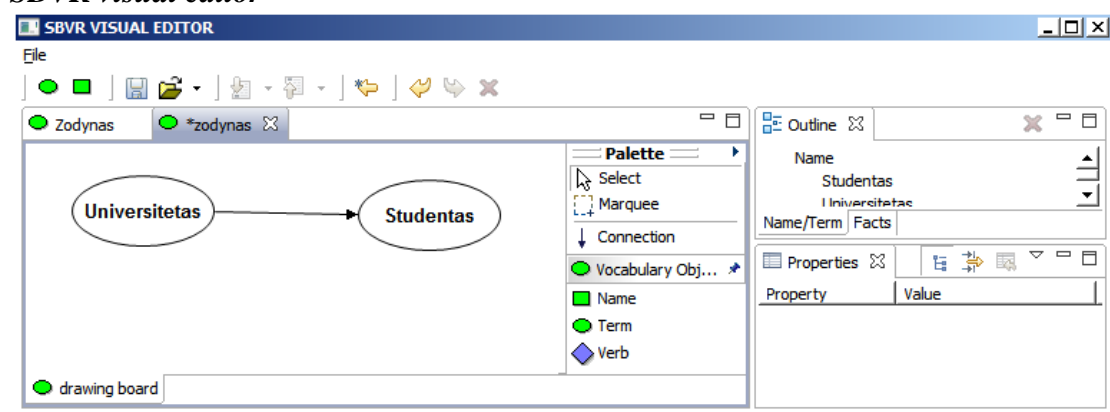
1.2 lentelė. Paašškinti simboliai

Simbolis	Paašškinitimas
+	Sistema tenkina teiginį.
-	Sistema netenkina teiginio.
0-10	Įvertinimas nuo 0 iki 10, kaip sistema atitinka teiginį. 0 – visiškai nesutampa; ... 10 – visiškai sutampa;

### 1.3.2. Esami SBVR redagavimo įrankiai

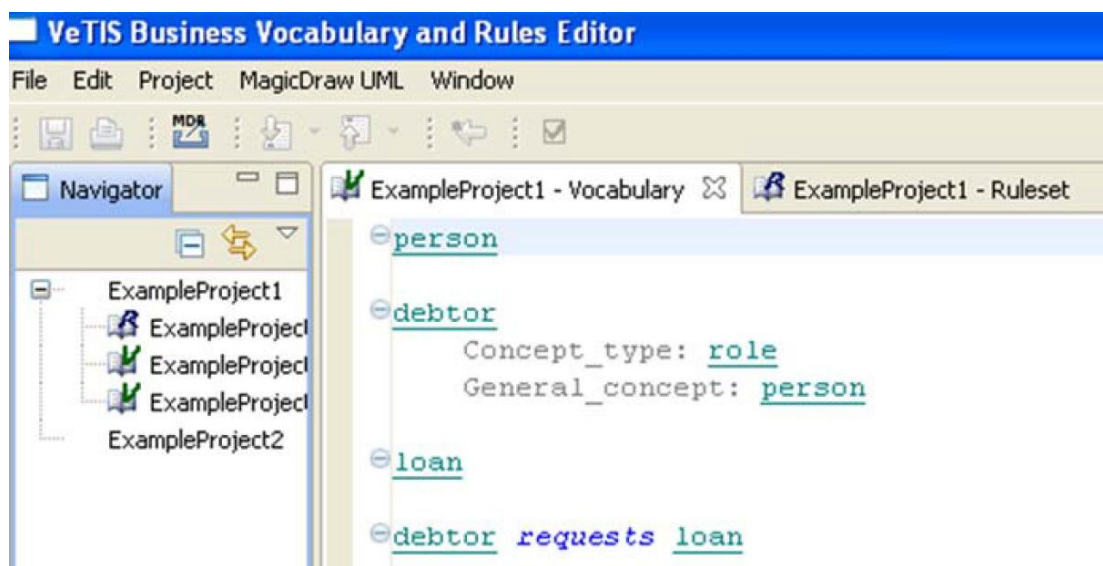
SBVR taisyklėms aprašyti pasirinkti du skirtingi įrankiai *SBVR visual editor* (1.8 pav.) – grafinis redaktorius ir *VeTIS tool* (1.9 pav.) – tekstinis redaktorius [3], .

#### *SBVR visual editor*



1.8 pav. *SBVR Visual Editor* sąsajos langas

*VeTIS tool* [ 3 ] , [ 4 ] .



1.9 pav. *VeTIS tool* sąsajos langas

#### ***SBVR visual editor:***

- Grafinis redaktorius.
- Vaizdingas taisyklių vaizdavimas.
- Ribotas taisyklių aprašymas.
- Nepatogi sąsaja.

#### ***VeTIS tool:***

- Tekstinis redaktorius
- Taisykles galima eksportuoti į *Magic draw* klasių diagramą.
- Paprastas taisyklių rašymas.
- Pilnai aprašomos taisyklės.

*VeTIS tool* palyginus su *SBVR visual editor* leidžia geriau aprašyti taisykles. Grafinis taisyklių aprašymas lėtas ir sudėtingas. *VeTIS tool* aplinką naudosome kuriant redaktorių.

### **1.3.3. Eclipse generatorių kūrimo ir gramatikų generavimo technologijos**

Vieni iš populiariausių kalbų kūrimo įrankių *Xtext* ir *Antlr* buvo pasirinkti kaip geriausi kalbų kūrimo atstovai.

#### ***Xtext:***

- Klaidų tikrinimas vyksta realiu laiku.
- Galimus elementų sąrašas pateikiamas realiu laiku
- Vartotojas sąsają gali prisitaikyti savo poreikiams
- Automatinis klaidų taisymas.
- Atviro kodo.
- Palaiko *Java*, *C*, *C++*, *C#*, *Objective C*, *Python*, ir *Ruby*

#### ***Antlr:***

- Lengviau suprantama sintaksė
- Rezultatus atvaizduoja medžio struktūrą
- Automatinis klaidų taisymas
- Atviro kodo
- Pradėtas kurti 1989m.
- Palaiko *Java*, *C#*, *C++* ir *Python*

*Xtext* palyginus su *Antr* yra modernesnis ir šiuolaikiškesnis. Savo darbe naudosime *Xtext* redaktorių.

#### 1.4. SBVR metamodelio ir klausimų formulavimo galimybių analizė

Pagrindiniai *SBVR* struktūrizuotos kalbos elementai □[ 3 ], □[ 5 ] yra terminai, vardai, veiksmožodžiai ir raktiniai žodžiai.

**Terminai** naudojami aprašyti esybėms arba rolėms, pavyzdžiui:

[university](#), [student](#), [city](#), ....

**Veiksmožodžiai** (angl. *verb*) arba prielinksniai apibūdina terminus arba aprašo ryšius tarp terminų, pavyzdžiui:

[are\\_in](#), [by](#), [is\\_damaged](#), ....

**Vardai** – tikriniai daiktavardžiai, reiškiantys asmenis, miestus ir kitus įvardytus objektus, skaičius:

[Kaunas](#), [Vilnius](#), [Petras](#), [200](#), ....

**Raktiniai žodžiai** – kalbos simboliai, naudojami taisyklėse ir klausimuose, pavyzdžiui:

[not](#), [at most 5](#), ...

Jie leidžia sujungti sakinius, pavyzdžiui:

[or](#), [and](#), ....

Panaudoti prieš taisyklę ar klausimą, raktiniai žodžiai nustato taisyklės ar klausimo tipą, pavyzdžiui:

[not](#), [what](#), [who](#)....

Žodyno elementai išreiškia *SBVR* metamodelio elementus – konceptus, veiklos taisykles ir klausimus. Konceptai gali būti bendriniai konceptai (objektų tipai), rolės, individualūs konceptai, faktų tipai ir faktai.

**Fakto tipas** – derinys vieno fakto, ir kelių terminų, pavyzdžiui:

[students study\\_in university](#), [bank gives loan](#), ...

**Faktai** sieja individualius konceptus arba individualius konceptus ir objektų tipus, pavyzdžiui:

[John Smith studies\\_in KTU](#)

[John Smith studies\\_in university](#)

**Taisyklė** – konceptų (objektų tipų, rolių, individų, faktų tipų, faktų ir raktinių žodžių derinys, reiškiantis veiklos struktūros ar veiklos veiksmų apribojimą. Taisyklės skirstomos į struktūrinės ir operacinės. Struktūrinės taisyklės yra nepažeidžiamos, pavyzdžiui, „[Privaloma, kad asmuo turi ne daugiau kaip vieną tėvą](#)“. Operacinės taisyklės priklauso nuo žmogaus veiksmų ir gali būti pažeidžiamos, pavyzdžiui, „[Įpareigojama, kad skolininkas grąžintų skolą ne vėliau kaip po 3 mėnesių](#)“.

**Klausimai** panašūs į taisykles, tačiau jų prasmė kitokia. Klausimų formuluotės reiškia užklausas, kurios turi turėti atsakymus analizuojamoje srityje, kurią aprašo tos srities žodynas.

##### 1.4.1. SBVR struktūrizuotos natūralios kalbos užklausų specifikavimo galimybių analizė

*SBVR* klausimo struktūrą apibrėžia taisyklės [3]:

- *SBVR* klausimai prasidės raktiniais žodžiais *wh\**, pavyzdžiui: [what](#), [wheather](#), [who](#),
- ....
- Klausime terminai turės būti susieti veiksmožodžiais, pavyzdžiui:

universities are in city, ....

- Prieš terminą galės būti raktinis žodis, pavyzdžiui: **not**, **at most**, **at most 5** accounts, ....
- Faktų tipus sujungia raktiniai žodžiai, pavyzdžiui: *and* are in city ‚Kaunas‘ *and* are in city ‚Vilnius‘, ....

#### 1.4.2. SBVR užklausų šablonai ir sudarymo principai

SBVR struktūrizuotos natūralios kalbos užklausai sudaryti reikalingi faktų tipai, susieti vieni su kitais per raktinius žodžius. Faktų tipai bus renkami iš žodyno.

SBVR struktūrizuotos natūralios kalbos užklausa pavyzdys □ [ 3 ]:

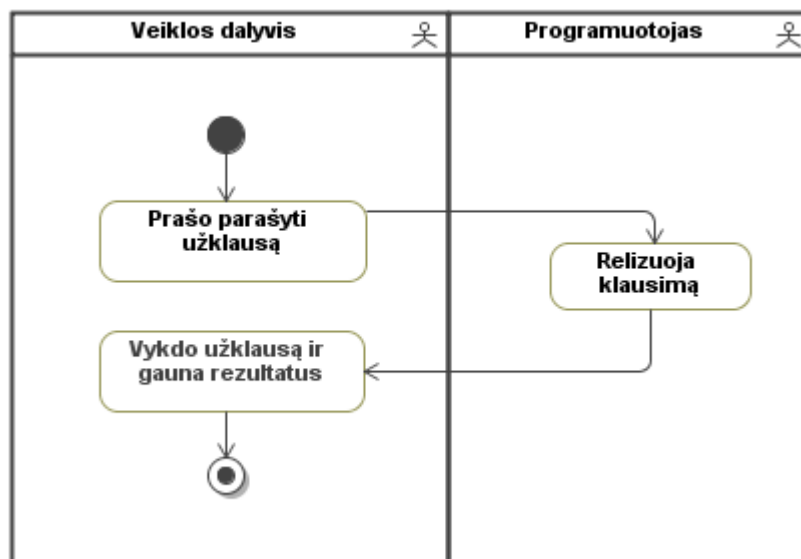
What journals are referred in the database ‚Web of Science‘ *and* have an impact factor, which is greater than 0, *and* are published by the university that is in city ‚Kaunas‘?

SBVR struktūrizuotą natūralios kalbos užklausa sudarantys faktų tipai, kurie bus naudojami užklausiai generuoti:

journals are referred in database ‚Web of Science‘  
journals have impact factor, *which is greater than 0*  
journals are published by the university  
university that is in city ‚Kaunas‘.

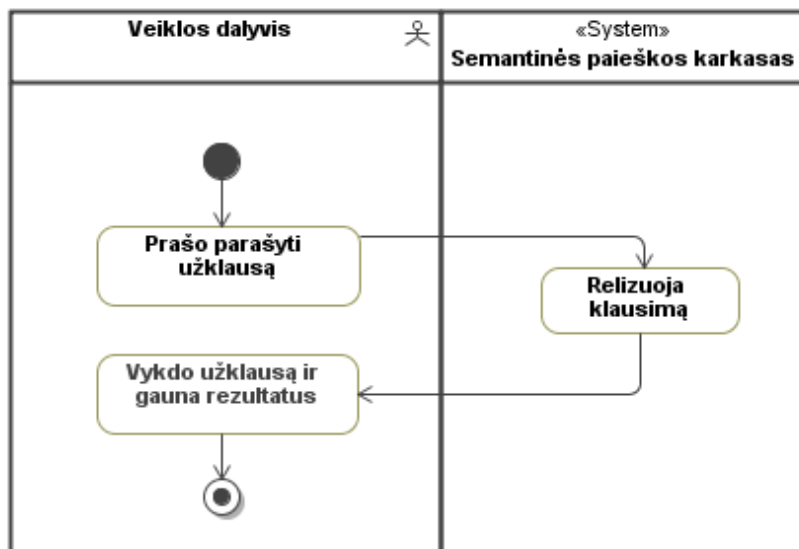
#### 1.5. Siekiamas sprendimas

Esamą užklausų kūrimo procesą būtų galima pavaizduoti taip (1.10 pav.). Jis trunka ilgai, veiklos dalyviai priklausomi nuo programuotojų.



1.10 pav. Užklausų kūrimas naudojantis programuotojo paslaugomis

Siekiamas sprendimas susijęs su semantinės paieškos karkaso sukūrimu □ [ 2 ], □ [ 8 ]. Semantinės paieškos karkasas leis veiklos dalyviams rašyti klausimus SBVR struktūrizuotos kalbos redaktoriuje, transformuos juos į SPARQL užklausas ir įvykdys SPARQL užklausų vykdymo įrankiu.



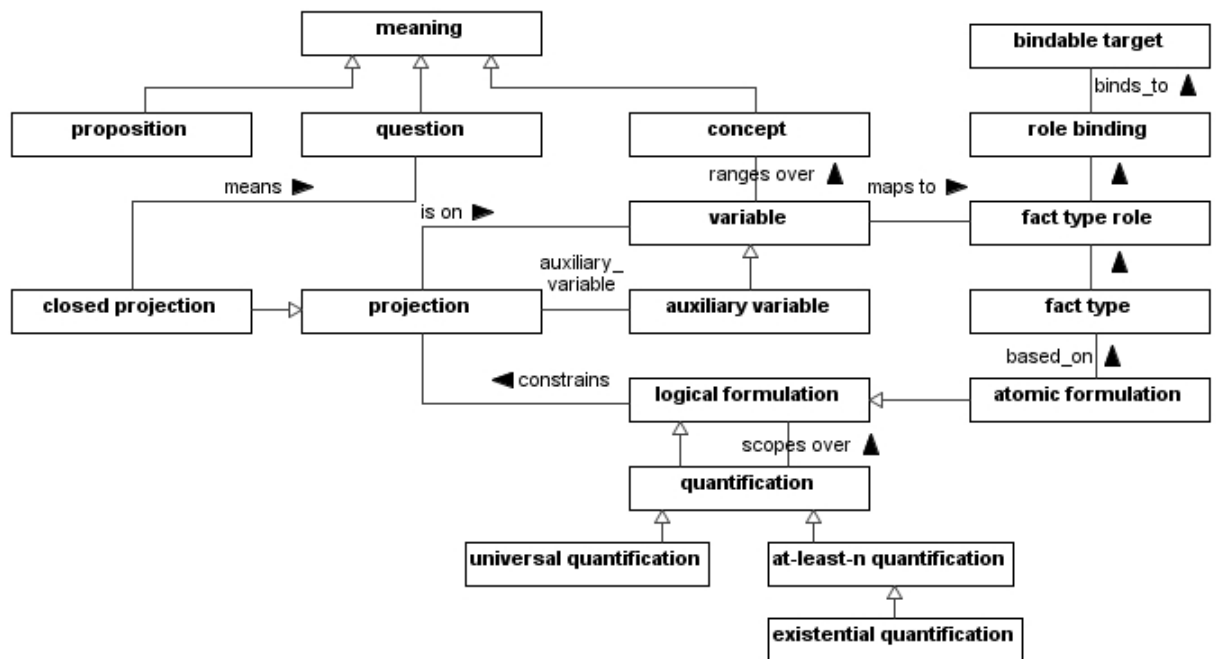
1.11 pav. Paieška dalykinėje srityje sukūrus redaktorių

Šiame darbe siekiama sukurti sistemą, kuri :

1. leistų rašyti klausimus tekstine kalba, panašia į natūralią;
2. klausimams formuluoti pateiktų raktinius žodžius (kaip *Ginseng* programa);
3. naudotų aprašytas sintaksines taisykles panašiai kaip *Querix* programa, ir skaidytų sakinius panašiai kaip „*Stanford parser*“ procese;
4. atpažintų kuo didesnę dalį natūralios kalbos klausimų.

Klausimų formavimo šablonai bus sudaromi remiantis *SBVR* klausimų metamodeliu [ 9 ] (1.12 pav.). Redaktoriui keliami kokybės kriterijai: *SBVR* struktūrizuotų užklausų redaktoriui turi atpažinti šiuos *SBVR* struktūrizuotos kalbos klausimų tipus:

1. Klausimai, paremti tik terminais;
2. Klausimai sudaryti iš vieno fakto tipo;
3. Klausimai, paremti keliais faktų tipais;
4. Klausimai su terminų disjunkcijomis ir konjunkcijomis;
5. Klausimai su faktų tipų disjunkcijomis ir konjunkcijomis.



1.12 pav. SBVR metamodelis klausimui sudaryti [ 9 ]

#### Analizės išvados

1. SBVR metamodelio analizė leido aprašyti pagrindinius SBVR konceptus, nustatyti SBVR struktūrizuotos natūralios kalbos klausimų struktūrą, specifikavimo galimybes, sudarymo principus..

2. Išanalizavus ir palyginus natūralios kalbos užklausų redaktorių, buvo nuspręsta perimti dalį siekiamos sistemos sprendimų: iš *Ginseng* – raktinių žodžių vaizdavimą, iš *Querix* – sintaksinių taisyklių taikymą.

3. Išanalizavus esamus SBVR redagavimo įrankius, buvo nuspręsta, kad geriausia klausimų redaktoriaus realizavimo aplinka yra *VeTIS* įrankis, kadangi klausimų redaktorių gali panaudoti dalį jo funkcionalumo.

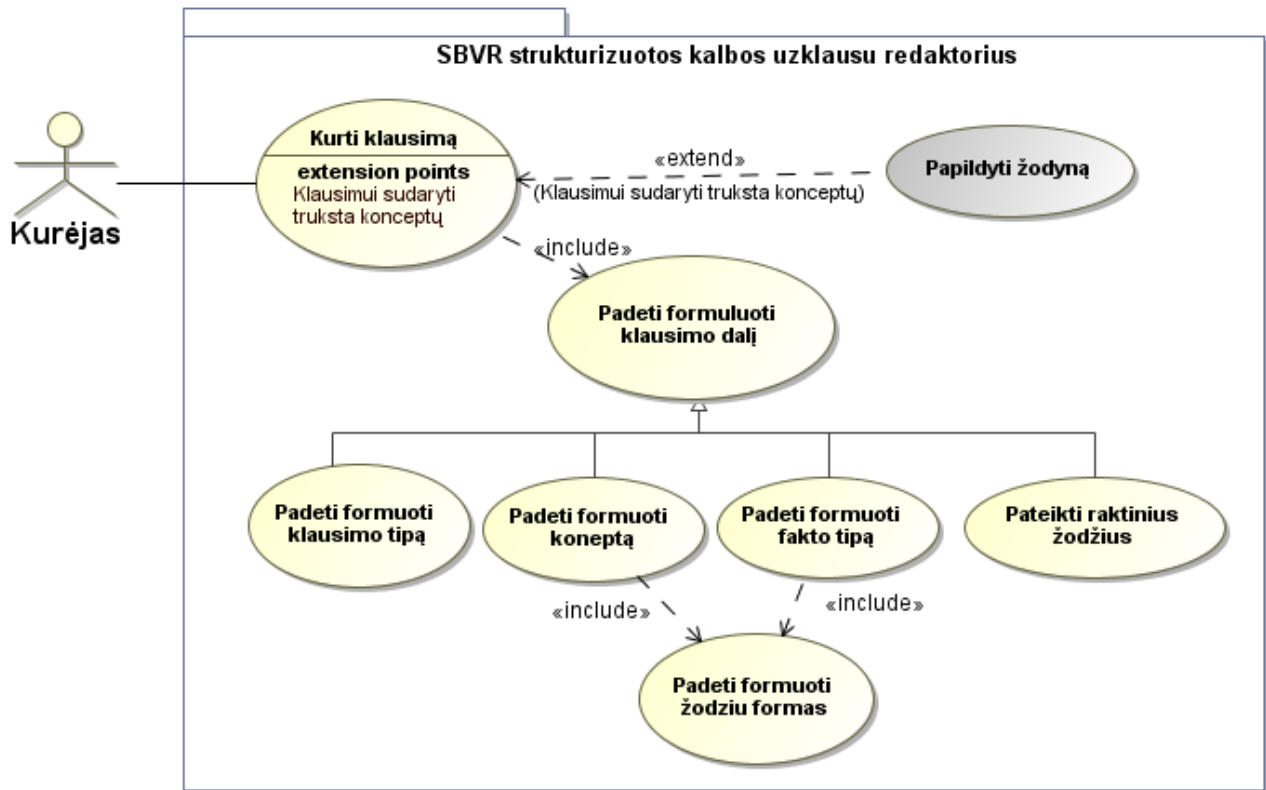
4. SBVR struktūrizuotos natūralios kalbos užklausų redaktoriui kurti pasirinktas *Xtext*, nes tai šiuolaikiškesnis kalbos kūrimo įrankis. *Xtext* palyginus su *Antlr* turi didesnę funkcionalumą. Išanalizavus *SPARQL* užklausų tipus, buvo pasirinkti pagrindiniai klausimų tipai, kuriuos klausimų redaktorių turi mokėti formuluoti ir kurie toliau bus taikomi kaip rezultato kokybės kriterijai.

# Klausimų redaktoriaus reikalavimai

## 2. Reikalavimų specifikacija, funkciniai reikalavimai

### 2.1. Panaudojimo atveju modelis

Panaudojimo atvejų modelyje (2.1 pav.) pateikti *SBVR* struktūrizuotos natūralios kalbos užklausų redaktoriaus panaudojimo atvejai. Papildyti žodyną panaudojimo atvejis nebus kompiuterizuojamas, bet jis reikalingas klausimų kūrimui. Klausimui sudaryti bus naudojami konceptai iš žodyno.



2.1 pav. *SBVR* struktūrizuotos natūralios kalbos užklausų redaktoriaus PA modelis

**Kompiuterizuojamų panaudojimo atvejų specifikacijos:**

2.1 lentelė. Panaudojimo atvejo „Kurti klausimą“ aprašymas

PA „Kurti klausimą“	
<b>Aprašymas.</b> Sistema gramatiškai išanalizuoja užklausą	
<b>Prieš sąlyga</b>	Sistema užkrauna žodyną
<b>Aktorius</b>	Kūrėjas
<b>Sužadinimo sąlyga</b>	Kūrėjas pasirenka klausimo formavimą
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Kūrėjas veda klausimo dalį	1.1. Sistema atpažįsta klausimo dalį ir vykdo PA „Padėti formuluoti“
2. Jei klausimas neužbaigtas, kūrėjas eina į 1 žingsnį.	
3. Kūrėjas išsaugo klausimą	
<b>Po sąlyga:</b>	Gaunama gramatiškai taisyklingas klausimas, tinkamas <i>XMI</i> schemai generuoti.
<b>Alternatyvūs scenarijai</b>	
<b>1 alternatyvus scenarijus. Kūrėjas papildo žodyną</b>	
	1.1a. Sistema neatpažįsta klausimo dalies, praneša apie tai kūrėjui ir saugo nepilną klausimo formuluotę laikinoje atmintyje
1.1a.1. Vartotojas vykdo PA „Papildyti žodyną“	
1.1a.2. Vartotojas grįžta į PA „Kurti klausimą“ 1 žingsnį ir tęsia klausimo formulavimą	
<b>2 alternatyvus scenarijus. Kūrėjas keičia klausimą</b>	
	1.1b. Sistema neatpažįsta klausimo dalies, praneša apie tai kūrėjui ir saugo nepilną klausimo formuluotę laikinoje atmintyje
1.1b.1. Kūrėjas keičia klausimą ar jo dalį, grįžta į PA „Kurti klausimą“ 1 žingsnį ir tęsia klausimo formulavimą	
<b>3 alternatyvus scenarijus. Kūrėjas naikina klausimą</b>	
	1.1c. Sistema neatpažįsta klausimo dalies, praneša apie tai kūrėjui ir saugo nepilną klausimo formuluotę laikinoje atmintyje
1.1c. Kūrėjas naikina klausimą	

2.2 lentelė. Panaudojimo atvejo „Padėti formuluoti klausimo dalį“ aprašymas

PA „Atpažinti klausimo dalį“	
<b>Aprašymas.</b> Padeda formuoti klausimą	
<b>Prieš sąlyga</b>	Sistema užkrauna žodyną
<b>Aktorius</b>	Kūrėjas
<b>Sužadinimo sąlyga</b>	Vedama užklausa
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1.1, Kūrėjas veda žodžius	1.1, Sistema randa žodį žodyne ir pateikia rastų konceptų galimus variantus
1.2, Keičia koncepto forma	1.2, pateikia galimas žodžio formas
<b>Po sąlyga:</b>	Vartotojui padedama formuoti klausimą
<b>Alternatyvūs scenarijai</b>	

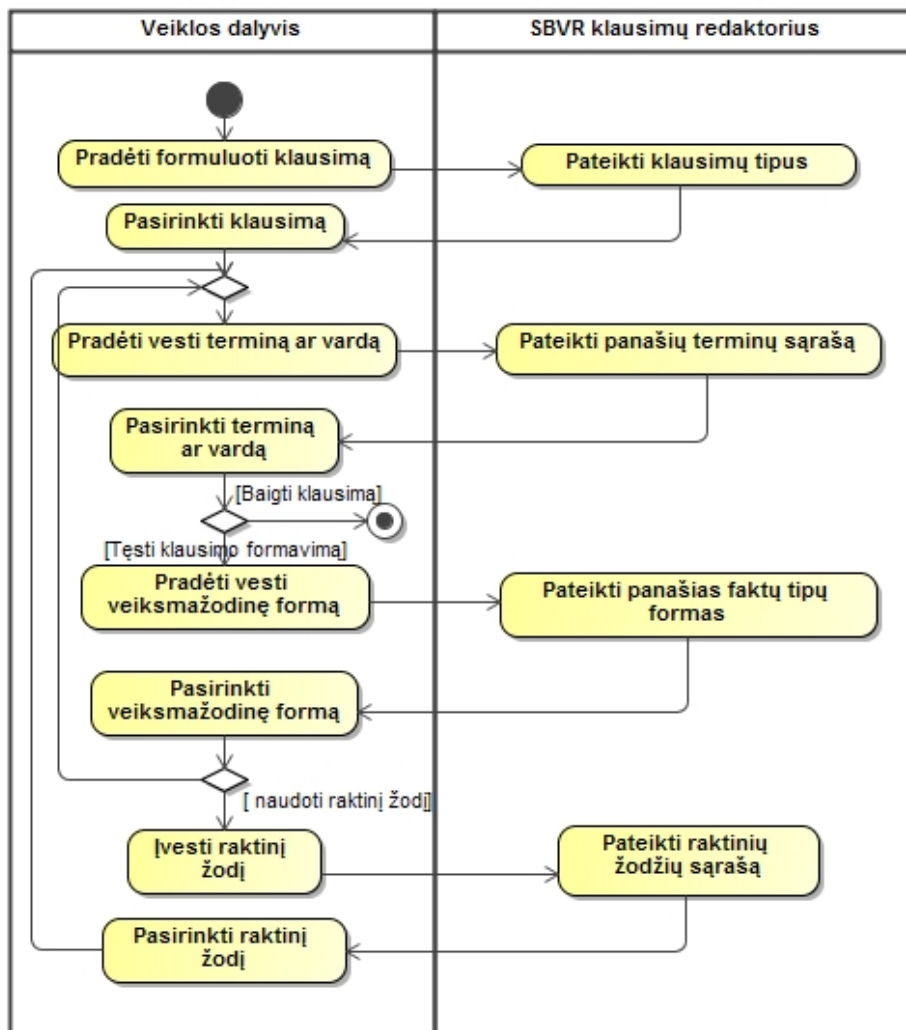


2.3 lentelė. Panaudojimo atvejo „Padėti formuoti užklausa“ aprašymas

PA „Padėti formuoti“	
<b>Aprašymas.</b> Padeda formuoti klausimą	
<b>Prieš sąlyga</b>	Sistema užkrauna žodyną
<b>Aktorius</b>	Kūrėjas
<b>Sužadinimo sąlyga</b>	Vedama užklausa
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1.1, Kūrėjas veda konceptą	1.1, pateikia konceptą pirmine forma
1.2, Kūrėjas veda pradžia	1.2, pateikia klausimo tipą
1.3, Kūrėjas veda koncepta, kuris yra fakto tipa	1.3, pateikia to fakto tipo konceptų veiksmazodžius
1.4, Kūrėjas įvedė fakto tipo veiksmazodį	1.4, pateikia fakto tipo paskutinį konceptą
<b>Po sąlyga:</b>	Vartotojui parodomas galimų konceptų sąrašas
<b>Alternatyvūs scenarijai</b>	

2.1.1. Veiklos procesų modelis

2.2 paveikslėlyje pateiktas klausimo formavimo procesas. Veiklos dalyvis, naudodamas redaktorių, veda klausimo dalis (veiksmažodines formas, terminus, vardus). Sistema pateikia galimus pasirinkimo variantus.



2.2 pav. Klausimų formavimo procesas

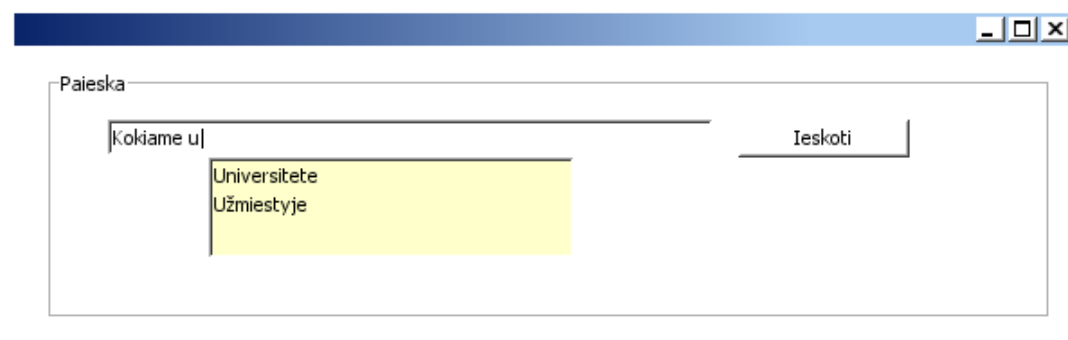
## 2.1.2. Vartotojo interfeiso modelis

2.3 paveikslėlyje pateiktas galimas vartotojo interfeiso modelis. Vartotojas, pradėdamas rašyti užklausa, turi pasirinkti vieną iš klausimo tipų.



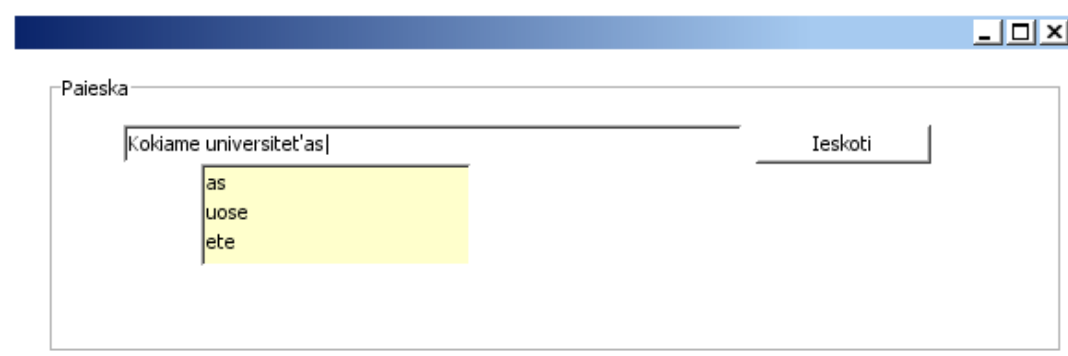
2.3 pav. Klausimo tipo išrinkimas

2.4 paveikslėlyje pateiktas koncepto išsirinkimo atvejis.



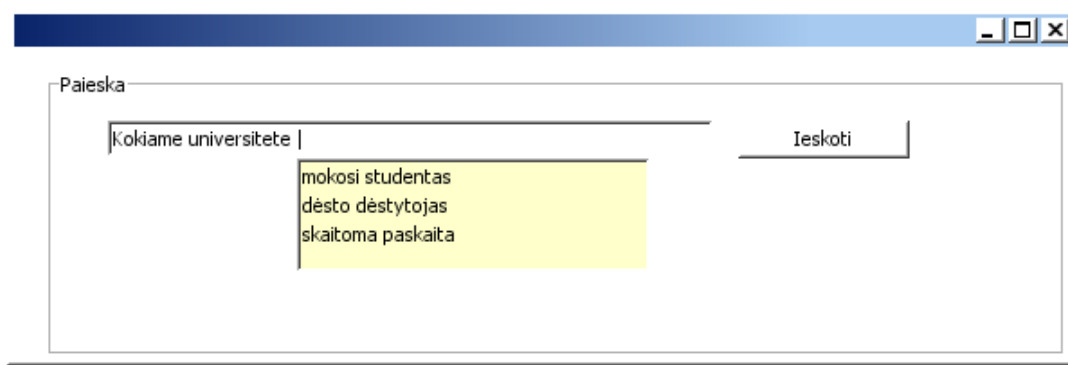
2.4 pav. Koncepto pasirinkimas

2.5 paveikslėlyje pateiktas koncepto tipo išsirinkimo atvejis.



2.5 pav. Žodžio formos pasirinkimas

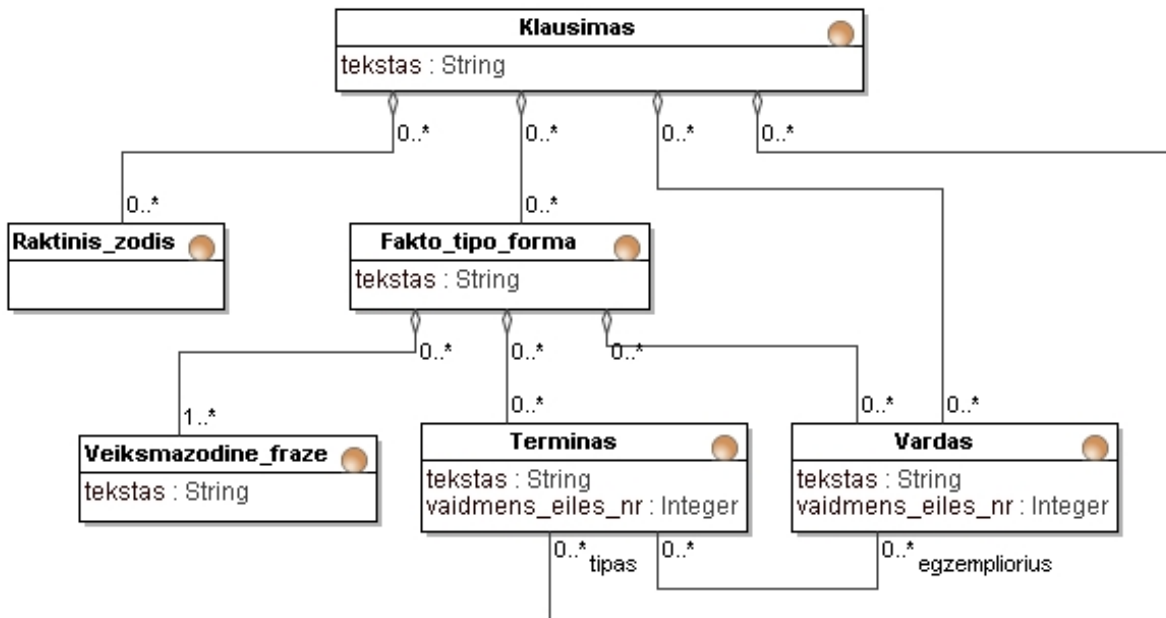
2.6 paveikslėlyje pateiktas fakto tipo pasirinkimo atvejis.



2.6 pav. Fakto tipo pasirinkimas

## 2.2. Dalykinės srities modelis

2.7 paveiksle pateikta esybių klasių diagrama, vaizduojanti klausimų sudarymo duomenų struktūrą. Klausimą sudaro terminai, vardai, raktiniai žodžiai ir faktų tipai. Fakto tipas susideda iš terminų ar vardų ir juos jungiančių veiksmažodžių.



2.7 pav. Klausimų esybių klasių diagrama

## 2.3. Reikalavimų analizės apibendrinimas

Atlikus sistemos reikalavimų analizę, buvo apibrėžtas *SBVR* struktūrizuotos kalbos klausimų redaktoriaus veikimo procesas, aprašyti redaktoriaus kūrimo reikalavimai, sudarytas dalykinės srities esybių klasių modelis. Šių rezultatų pagrindu bus sudaromas redaktoriaus projektas.

### 3. SBVR KLAUSIMŲ REDAKTORIAUS PROJEKTAS

#### 3.1. Sistemos architektūros projektas

##### 3.1.1. Sistemos loginė architektūra

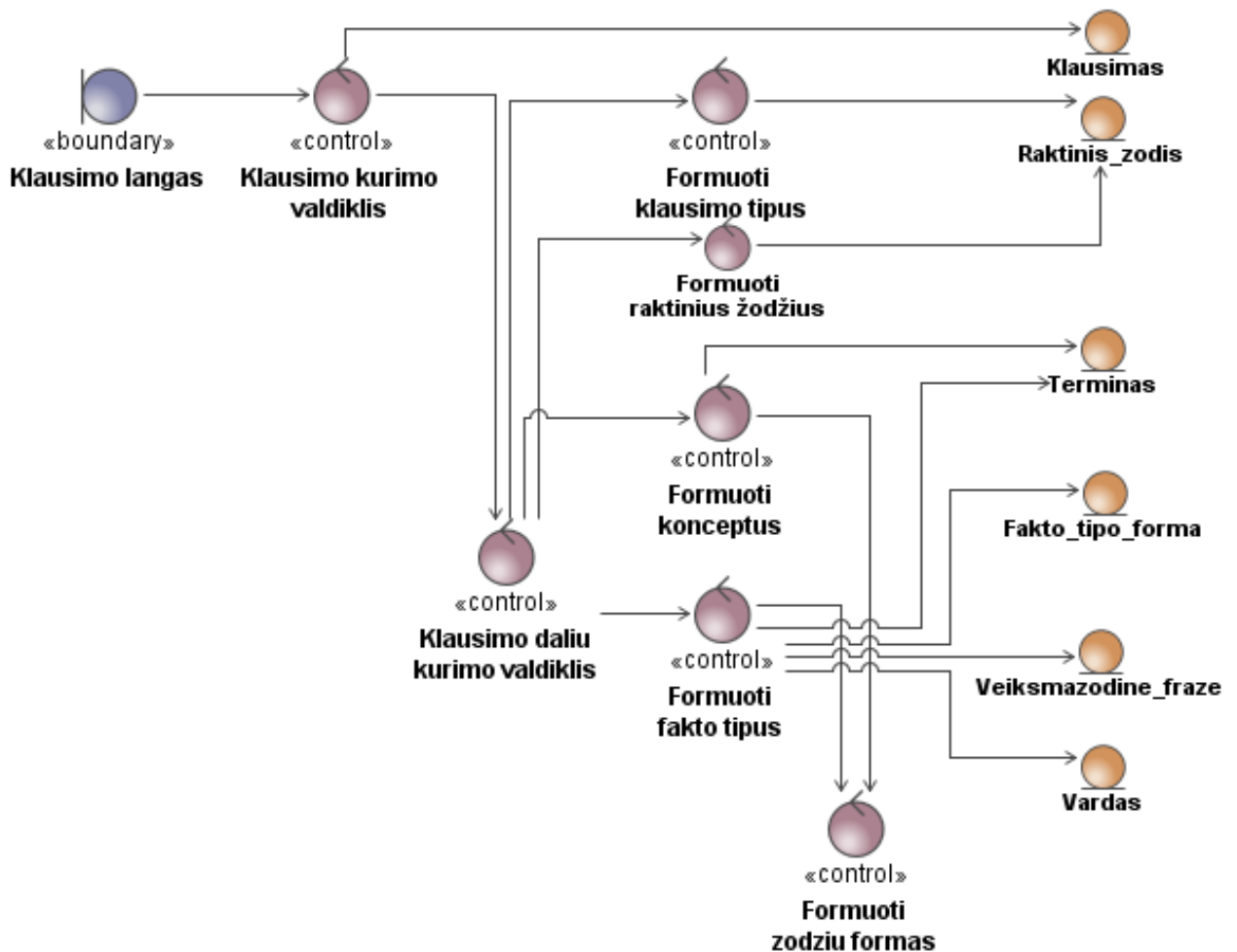
3.1 paveikslėlyje pavaizduota sistemos loginė architektūra. Sistemos loginėje architektūroje atvaizduotos sistemos posistemos ir jų tarpusavio sąryšį.



3.1 pav. Sistemos loginė architektūra

##### 3.1.2. Panaudojimo atvejų analizės klasių modelis

Panaudojimo atvejų analizės klasių modelis parodo visus elementus, kuriuos reikės sukurti: ribines, valdymo ir esybių klases (3.2 pav.).



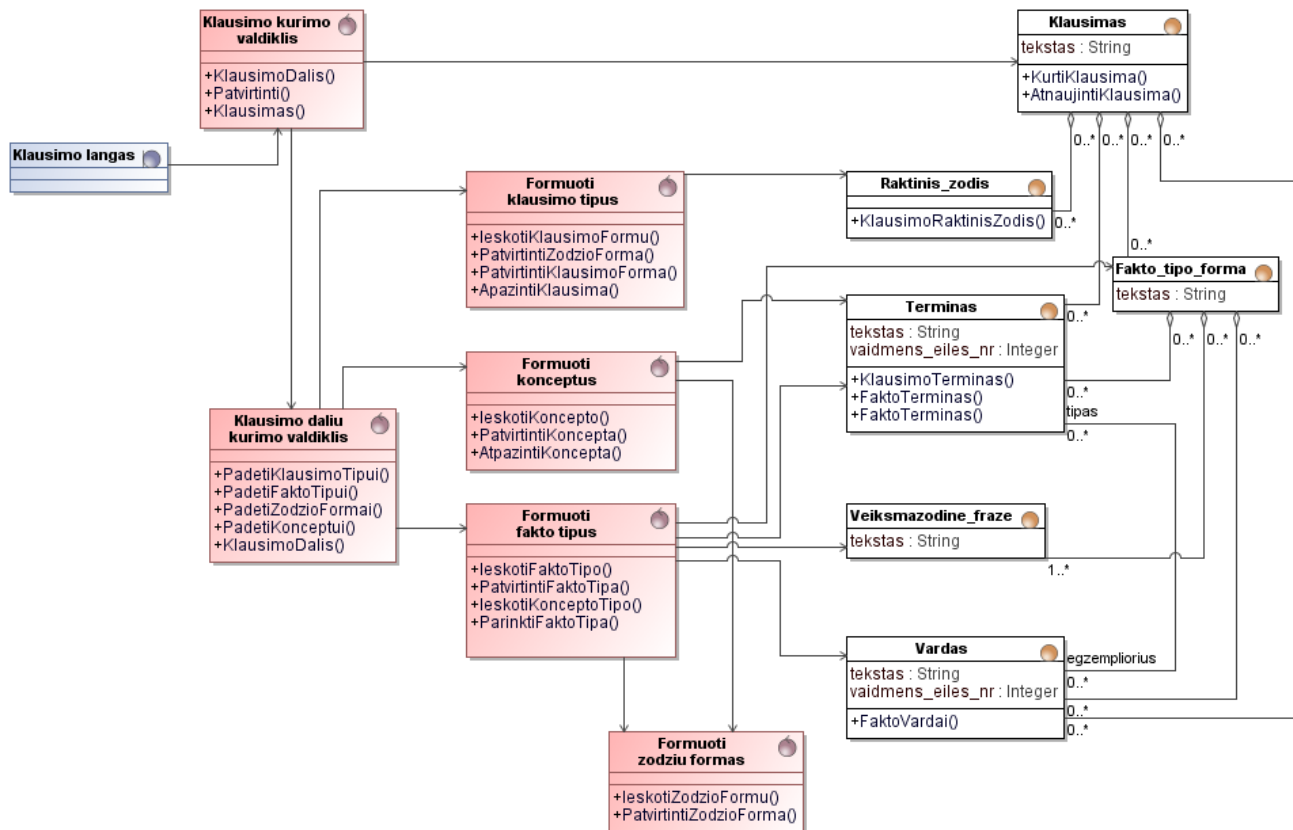
3.2 pav. Analizės klasių diagrama

### 3.1.3. Bendras klasių modelis

Bendrame klasių modelis parodo kokios operacijos vykdomos ribinėse, valdymo ir esybių klasėse.

### 3.1.4. Bendras redaktoriaus klasių modelis

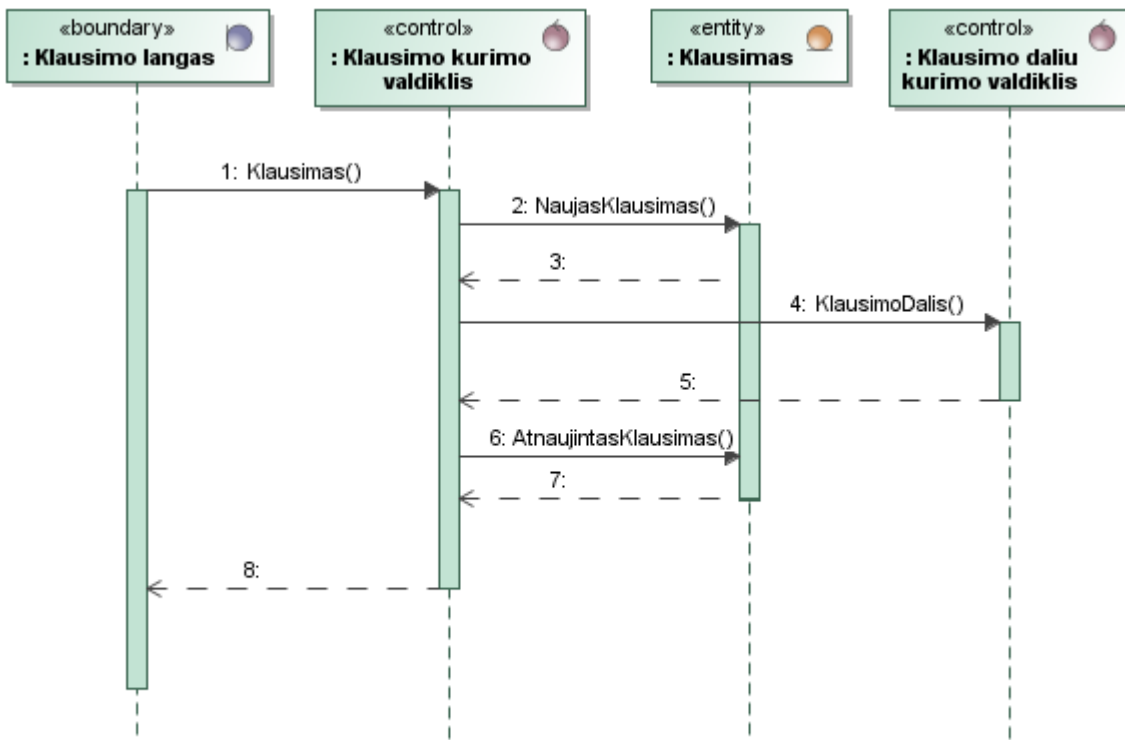
3.3 paveikslėlyje pavaizduotas bendras redaktoriaus klasių modelis.



3.3 pav. Redaktoriaus klasių modelis

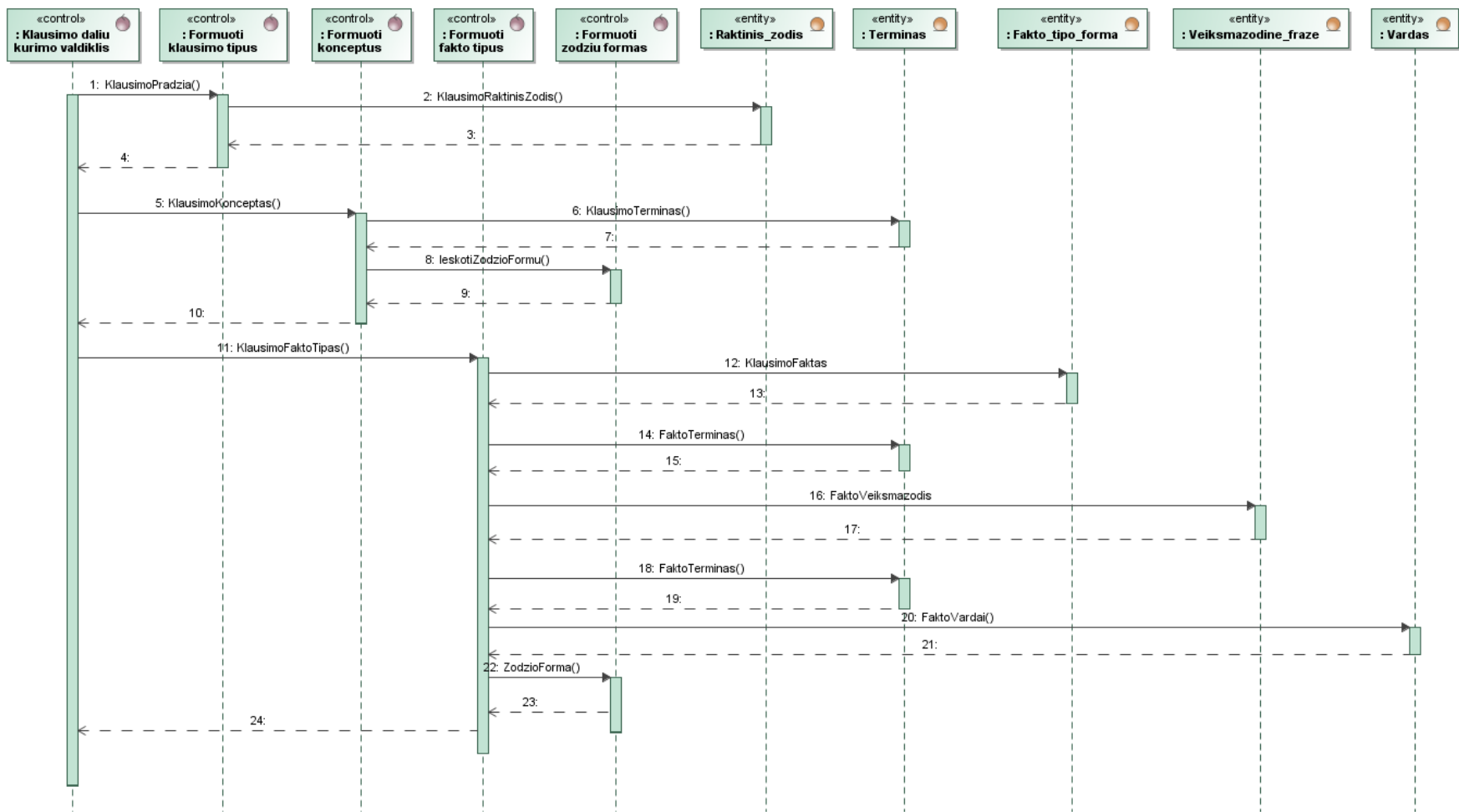
### 3.1. Sistemos elgsenos modelis

3.4 paveiksle pavaizduota struktūrizuotos natūralios kalbos užklausų redaktoriaus užklausoje apdorojimo sekų diagrama. Kiekvienas klausimas skaidomas į dalis iš kurių formuojamas klausimas.



3.4 pav. SBVR struktūrizuotos natūralios kalbos užklausų redaktoriaus sekų diagrama

3.5 paveiksle pavaizduota klausimo formavimo sekos.



3.5 pav. Klausimų formavimo sekos diagrama

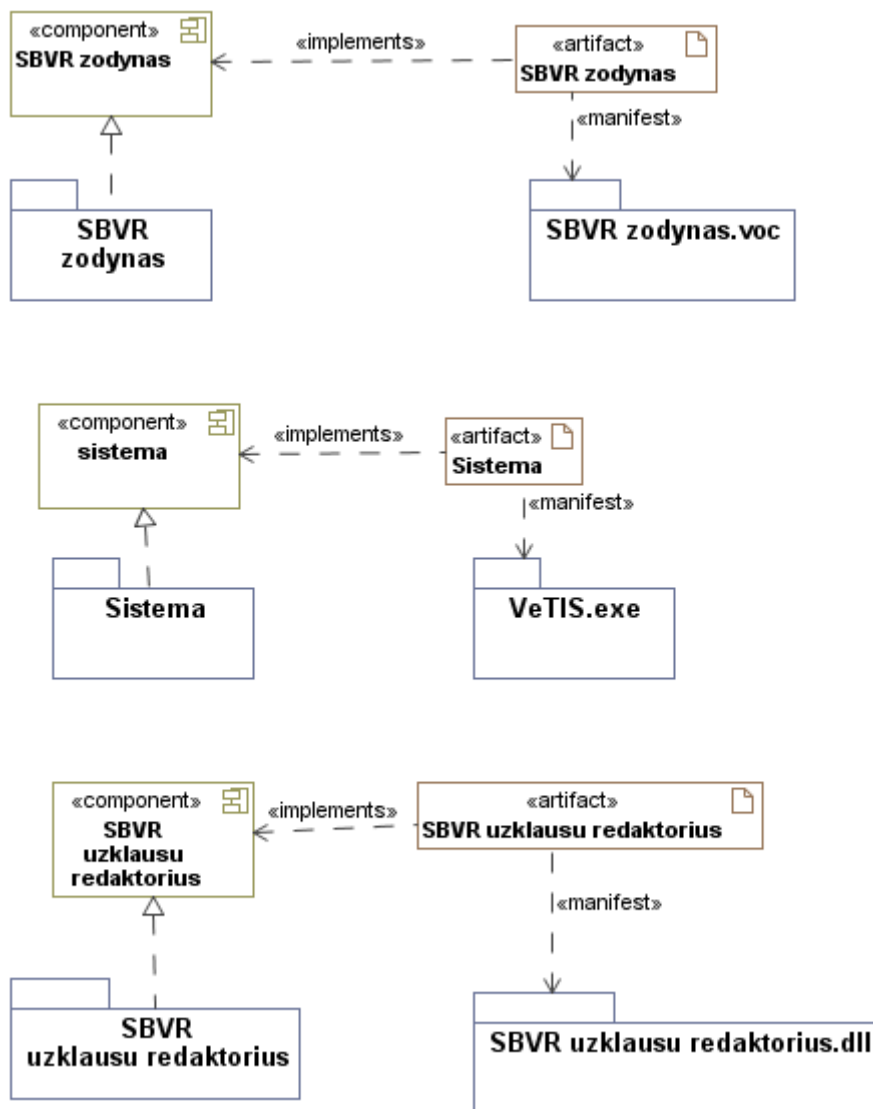
## 3.2. Realizacijos modelis

### 3.2.1. Programinių komponentų architektūra

3.6 paveikslėlyje pateikta programinių komponentų diagrama.



3.6 pav. Sistemos programinių komponentų diagrama

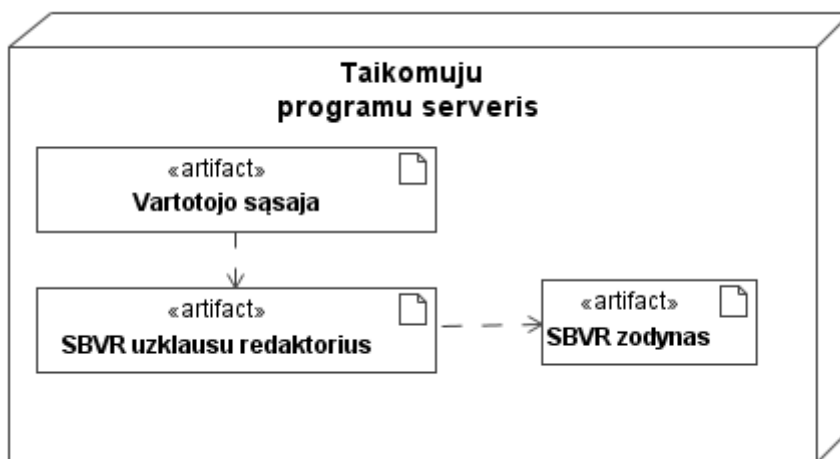


3.7 pav. Sistemos programinių komponentų diagrama



### 3.2.2. Diegimo modelis

3.8 paveiksle pateikta *SBVR* struktūrizuotos natūralios kalbos diegimo diagrama. Sistemos pilnam funkcionavimui reikalingas *SBVR* žodynas. *SBVR* užklausų redaktorius bus įdiegtas į *VeTIS* įskiepi.



3.8 pav. Klausimų redaktoriaus diegimo diagrama

## 4. REDAKTORIAUS REALIZACIJA

### 4.1. Sistemos veikimas

Klausimai formuojami pagal šablonus, pagal kuriuos jie atpažįstami. Kiekvienai grupei rašomas atskiras algoritmas. Šablonai ir klausimų pavyzdžiai:

1. Klausimai, paremti tik terminais

<Qs> <T> „?“

What is organization ?

2. Klausimai sudaryti iš vieno fakto tipo

<Qs> (<Ft> | <F>) „?“

What persons work\_in organization?

3. Klausimai, paremti keliais faktų tipais

<Qs> (<Ft> | <F>)\* „?“

What persons work\_in organization that is\_located\_in city Kaunas?

4. Klausimai su terminų disjunkcijomis ir konjunkcijomis

<Qs> (<T> (( „AND“ | „OR“) <T>)\* <V> (<T> (( „AND“ | „OR“) <T>)\* „?“

Find organizations and persons that work\_in those organizations

5. Klausimai su disjunkcijomis ir konjunkcijomis.

<Qs> (<Ft> | <F> | <T >) ((∅ | („AND“ | „OR“)) (<Ft> | <F> | <T >)), „?“

What person work\_in organization KTU or work\_in organization VDU?

<Qs> – Klausimo pirmas žodis pvz: what, where

<T> - Terminas

<V> - Veiksmažodis

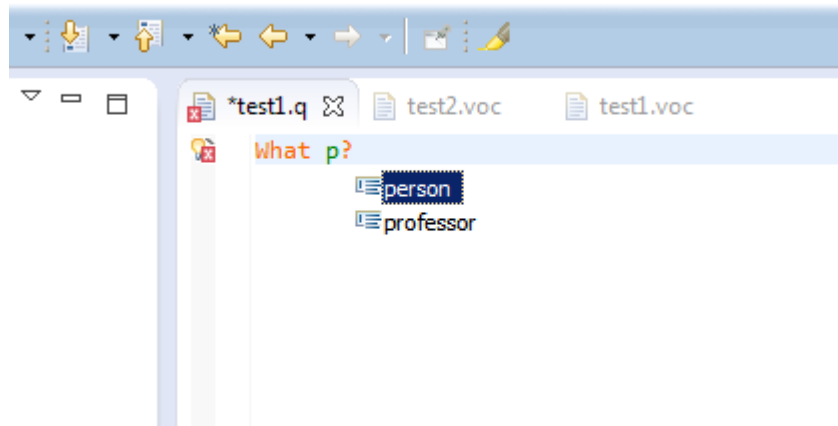
<N> - Vardas

<F> - Faktas

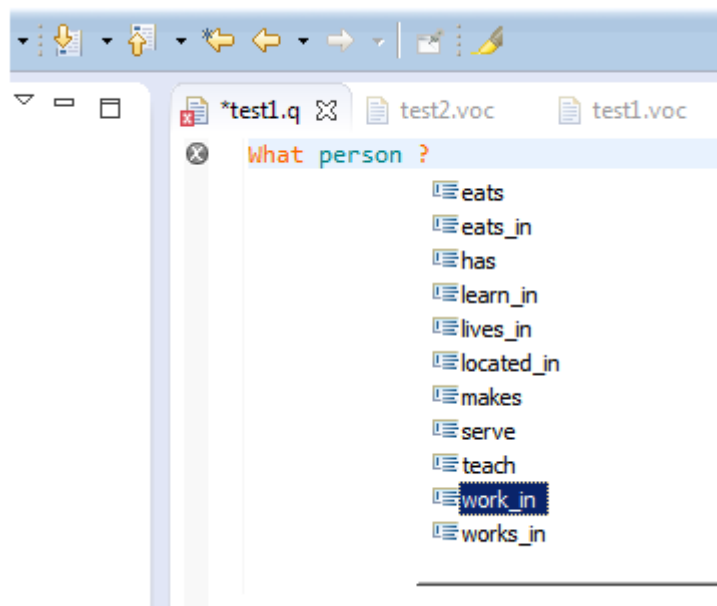
<Ft> - Fakto tipas

### 4.2. Sistemos veikimo aprašymas

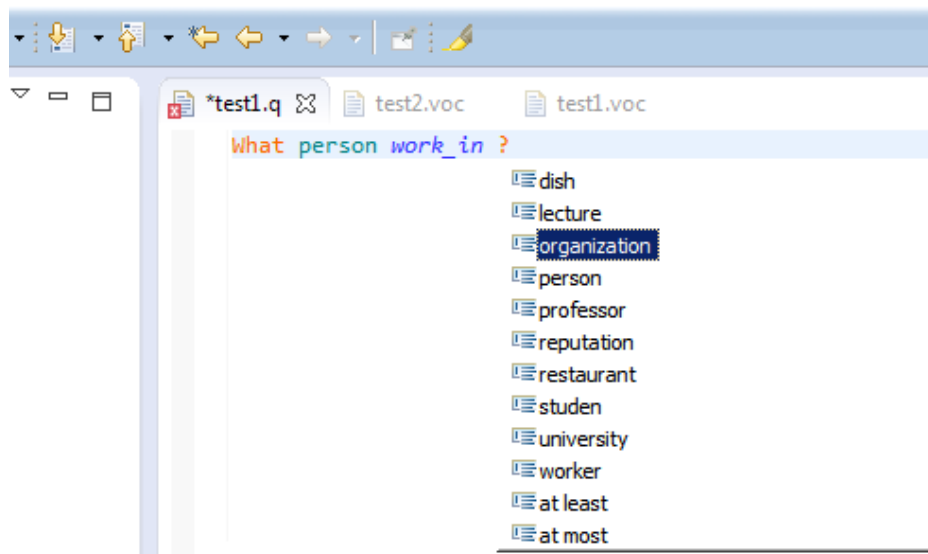
Sistema padeda vartotojui sukurti klausimą, kuris sudaromas iš dalykinės srities konceptų. Vartotojui pradėjus vesti klausimą, jam pateikiamas pasirinkimo meniu, kuriame jis gali pasirinkti klausimo tipą ir dalis iš dalykinės srities žodyno. Klausimo tipas nusako jo šabloną, pagal kurį aprašoma klausimo struktūra. Rašant klausimą, vartotojui bus pateikiami tik tie konceptai, kurie aprašyti žodyne ir leidžiami šablono. 4.1, 4.2, 4.3, 4.4 paviekslėlyje pateikta įvairių klausimo kūrimo dalių žingsniai.



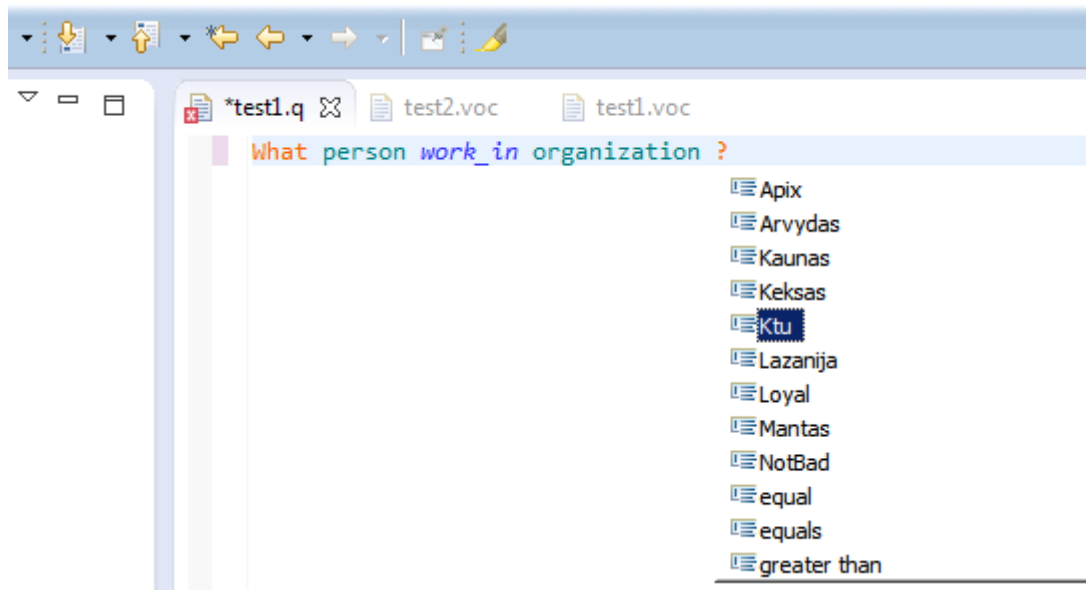
4.1 Sukurto redaktoriaus langas, termino įvedimas



4.2 Sukurto redaktoriaus langas, veiksmažodžio įvedimas



4.3 Sukurto redaktoriaus langas, termino įvedimas



4.4 Sukurto redaktoriaus langas, vardo įvedimas

### 4.3. Testavimo modelis ir duomenys, kontrolinis pavyzdys

Testuojama juodosios dėžės principu. Klausimą sudaro konceptai iš žodyno. Klausimas prasideda vienu iš klausimams būdingu identifikatoriumi.

Žodyno pavyzdys:

person

organization

university

student

worker

lecture

professor

city

person *lives\_in* city

student *learn\_in* university

worker *works\_in* organization

university *has* lecture

organization *is\_ocated\_in* city

professor *works\_in* university

professor *teach* lecture

student *has* lecture

Ktu

General\_concept: organization

Apix General\_concept: organization

Arvydas General\_concept: person

Kaunas General\_concept: city

Rezultatas – galimi *SBVR* klausimai:

Suprasti klausimai:

what is organization?

what are person?

where person Arvydas *works\_in*?

Who *works\_in* organization?

Who *works\_in* organization Ktu?

what persons *work\_in* organization?

what person *works\_in* organization KTU?

what persons *not works\_in* organization KTU?

what organizations *are\_located\_in* city Kaunas?

what person *works\_in* organization *that is\_located\_in* city Kaunas?

what person *works\_in* organization KTU *and lives\_in* city Vilnius?

what students *and professors* *work\_in* organization ?

what person *work\_in* organization KTU *or* *work\_in* organization VDU?

what person *work\_in* organization KTU *and live\_in* city Vilnius?

Nesuprasti klausimai:

how many organization persons Arvydas *work\_in*?

what person student *work\_in* organization?

## 5. EKSPERIMENTINIS SISTEMOS TYRIMAS

### 5.1. Eksperimento planas

Eksperimento tikslai buvo dvejopi:

1. Patikrinti, ar redaktorius atpažįsta apibrėžtus klausimų tipus.
2. Parodyti, kad redaktorius nepriklausomas nuo dalykinės srities

Šiam tikslui buvo paimti du žodynai ir sukurti klausimai visiems apibrėžtiems klausimų tipams.

Pirmas SBVR žodynas

person

organization

university

student

worker

lecture

professor

city

person *Lives\_in* city

student *Learn\_in* university

worker *works\_in* organization

university *has* lecture

organization *is\_ocated\_in* city

professor *works\_in* university

professor *teach* lecture

student *has* lecture

Ktu

General\_concept: organization

Apix General\_concept: organization

Arvydas General\_concept: person

Kaunas General\_concept: city

5.1 lentelėje pateikti iš pirmo žodyno sudarytų klausimų pavyzdžiai

5.1 lentelė. Pirmo žodyno sudarytų klausimų lentelė

Klausimo tipas	Klausimas	Atpažinta
Klausimai, paremti tik terminais	What is <u>organization</u> ?	+
	What are <u>person</u> ?	+
Klausimai, paremti vienu fakto tipu	Where <u>person Arvydas</u> <u>works_in</u> ?	+
	Who <u>works_in organization</u> ?	+
	Who <u>works_in organization Ktu</u> ?	+
	what <u>persons work_in organization</u> ?	+
	what <u>person works_in organization KTU</u> ?	+
	what <u>persons not works_in organization KTU</u> ?	+
	what <u>organizations are_located_in city Kaunas</u> ?	+
Klausimai, paremti keliais faktų tipais	what <u>person works_in organization that is_located_in city Kaunas</u> ?	+
	what <u>person works_in organization KTU and lives_in city Vilnius</u> ?	+
Klausimai su terminų konjunkcijomis arba disjunkcijomis	what <u>students and professors work_in organization</u> ?	+
Klausimai su konjunkcijomis arba disjunkcijomis	what <u>person work_in organization KTU or work_in organization VDU</u> ?	+
	what <u>person work_in organization KTU and live_in city Vilnius</u> ?	+

Antras SBVR žodynas

person

organization

restaurant

customer

client Synonym: customer

chef  
 dish  
 reputation  
 chef *makes* dish  
 chef *work\_in* organization  
 client *eats* dish  
 client *eats\_in* restaurant  
 chef *eats\_in* restaurant  
 chef *serve* client  
 restaurant *has* chef  
 restaurant *has* reputation  
 person *has* reputation  
 restaurant General\_concept: organization  
 person General\_concept: organization  
 Arvydas General\_concept: client  
 Mantas General\_concept: chef  
 chef General\_concept: person  
 PyragaiciaiIrKO General\_concept: restaurant  
 Keksas General\_concept: dish  
 Lazanija General\_concept: dish  
 NotBad General\_concept: reputation  
 Loyal General\_concept: reputation

5.2 lentelėje pateikta iš antro žodyno sudarytų klausimo variantus

5.2 lentelė. Antro žodyno sudarytų klausimų lentelė

Klausimo tipas	Klausimas	Atpažinta
Klausimai, paremti tik terminais	who is <u>chef</u> ?	+
	what is <u>dish</u> ?	+
	who is <u>client</u> ?	+



	what is <u>restaurant</u> ?	+
	what is <u>reputation</u> ?	+
Klausimai, paremti vienu faktų	what <u>chef</u> <i>makes</i> ?	+
	what <u>chef</u> <i>works_in</i> <u>organization</u> ?	+
	where <u>chef</u> <u>Mantas</u> <i>work_in</i> ?	+
	what <u>chef</u> <i>work_in</i> <u>organization</u> ?	+
	what <u>clients</u> <i>eats_in</i> <u>restaurant</u> <u>PyragaiciaiIrKO</u> ?	+
	what <u>chef</u> <i>not serve</i> <u>client</u> <u>Arvydas</u> ?	+
	what <u>chef</u> <i>makes</i> <u>dish</u> <u>Lazanija</u> ?	+
Klausimai, paremti keliais faktų tipais	what <u>client</u> <i>eats_in</i> <u>restaurant</u> <i>that has</i> <u>reputation</u> <u>NotBad</u> ?	+
	what <u>restourant</u> <i>serving</i> <u>client</u> <u>Arvydas</u> <i>has</i> <u>chef</u> <u>Mantas</u> ?	+
Klausimas su terminų konjunkcijomis arba disjunkcijomis	what <u>client</u> <i>and</i> <u>chef</u> <i>eats_in</i> <u>restourant</u> <u>PyragaiciaiIrKO</u> ?	+
Klausimai su faktų tipų konjunkcijomis arba disjunkcijomis	what <u>client</u> <i>eats</i> <u>dish</u> <u>Lazanija</u> <i>or</i> <i>eats</i> <u>dish</u> <u>keksas</u> ?	+
	what <u>client</u> <i>eats</i> <u>dish</u> <u>Lazanija</u> <i>and</i> <i>has</i> <u>reputation</u> <u>Loyal</u> ?	+

## 5.2. Eksperimento rezultatai

Eksperimentas parodė, kad redaktorius atpažįsta klausimus iš sudaryto šablono. Klausimai turi savyje turėti taisyklinga fakto tipą (fakto tipą aprašo SBVR standartas). 5.3 lentelėje palygintos esamos sistemos su sukurtu įrankiu.

5.3 lentelė. Natūralios kalbos užklausų apdorojimo sistemos

Teiginys	Sistema	VETIS	Nlp-Reduce	Querix	Ginseng	Semantic Crystal
Kalba panaši į natūralią		6	9	7	5	1
Klausimus priima tekstiniu formatu		+	+	+	+	-
Sistema neleidžia įvesti nekorektiškų klausimų		+	+	+	-	-
Sistema pateikia raktinius žodžius		+	-	-	+	-
Sistema padeda suformuluoti		+	+	+	-	-

klausimus					
Klausimams atpažinti naudojamų sinonimų žodynas	-	-	+	-	-
Klausimams įvesti nereikia papildomos informacijos	-	-	+	+	+

**5.4 lentelė. Paaiškinti simboliai**

Simbolis	Paaiškinimas
+	Sistema tenkina teiginį.
-	Sistema netenkina teiginio.
0-10	Įvertinimas nuo 0 iki 10, kaip sistema atitinka teiginį. 0 – visiškai nesutampa; ... 10 – visiškai sutampa.

## 6. IŠVADOS

1. Literatūros šaltinių analizė parodė, kad aktualu sukurti klausimų redaktorių, kuriuo galėtų naudotis vartotojai, nežinantys ontologijų užklausų kalbos, ir tą galima padaryti *SBVR* struktūrizuotos kalbos pagrindu, kadangi ja užrašytus klausimus galima transformuoti į *SPARQL* užklausas.
2. Sukurtas įrankis *SBVR* struktūrizuotos kalbos klausimams sudaryti, formuojant juos iš veiklos žodyno konceptų ir taikant šablonus, sudarytus *SBVR* metamodelio pagrindu.
3. Kadangi klausimai sudaromi naudojant dalykinės srities žodyną, jis padeda dalykinės srities žinovams rašyti užklausas nenaudojant *SPARQL* užklausų kalbos, kuri nepriimtina ne IT specialistams
4. Sukurtas klausimų redaktorius pranašesnis už esamas panašias sistemas, nes leidžia formuluoti tikslius klausimus, kurie sudaromi iš dalykinės srities konceptų, ir formalizuojami, nes remiasi šablonais, sudarytais pagal *SBVR* metamodelį.
5. Sukurtu redaktoriumi sudaromi klausimai savo struktūra panašesni į natūralią lyginant su *Ginseng* ir *Semantic Crystal* formuojamais klausimais ir formalesni negu *Nlp-Reduce*, *Querix* užklauso. Lyginant su šiomis sistemomis, *SBVR* natūralios kalbos klausimų struktūra labiausiai subalansuota.
6. Klausimų redaktorius integruotas į *VETIS* įrankį ir bus pritaikytas semantinės paieškos sistemoje.

## 7. LITERATŪRA

- [ 1 ] Šukys, Algirdas; Nemuraitė, Lina; Šinkevičius, Edvinas; Paradauskas, Bronius. Querying ontologies on the base of semantics of business vocabularies and business rules // Information Technologies' 2011 : proceedings of the 17th international conference on Information and Software Technologies, IT 2011, Kaunas, Lithuania, April 27-29, 2011. 2011, p. 247-254.
- [ 2 ] A. Šukys, L. Nemuraitė, E. Šinkevičius, B. Paradauskas SBVR Based Representation of SPARQL Queries and SWRL Rules for Analyzing Semantic Relations. Bustech 2011 [elektroninis išteklius]: the First International Conference on Business Intelligence and Technology, September 25-30, Rome, Italy. IARIA, 2011, p. 1–6.
- [ 3 ] Nemuraitė, Lina; Skersys, Tomas; Šukys, Algirdas; Šinkevičius, Edvinas; Ablonskis, Linas. VETIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models // Information Technologies' 2010 : proceedings of the 16th international conference on Information and Software Technologies, IT 2010, Kaunas, Lithuania, April 21-23, 2010. 2010, p. 377-384.
- [ 4 ] VeTIS User Guide. Prieiga per internetą, <http://www.magicdraw.com/files/manuals/VeTISUserGuide.pdf>
- [ 5 ] OMG. Semantics of Business Vocabulary and Business Rules, 2008. Prieiga per internetą, <http://www.omg.org/spec/SBVR/1.0/PDF>
- [ 6 ] Alexandros, Marinos; Sotiris, Moschoyiannis; Paul, Krause. An SBVR to SQL Compiler. Prieiga per internetą <http://ceur-ws.org/Vol-649/paper7.pdf>
- [ 7 ] Esther Kaufmann, Abraham Bernstein. How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? Prieiga per internetą, <http://www.cs.xu.edu/csci390/08s/NLInterfacesUsefulToEndUsers.pdf>
- [ 8 ] Šukys, Algirdas; Nemuraitė, Lina; Paradauskas, Bronius; Šinkevičius, Edvinas. Transformation framework for SBVR based semantic queries in business information systems // Bustech 2012 [elektroninis išteklius] : the second International Conference on Business Intelligence and Technology, July 22-27, 2012, Nice, France. IARIA, 2012, p. 1–6.
- [ 9 ] A. Šukys, L. Nemuraitė, B. Paradauskas Representing and Transforming SBVR Question Patterns into SPARQL. // Information and software technologies : 18th International Conference, ICIST 2012, Kaunas, Lithuania, September 13-14, 2012 : proceedings / [edited by] Tomas Skersys, Rimantas Butleris, Rita Butkiene. Berlin, Heidelberg : Springer, 2012. (Communications in computer and information science, Vol. 319, ISSN 1865-0929). ISBN 9783642333071. p. 436-451.
- [ 10 ] Ginseng: A Guided Input Natural Language Search Engine. Prieiga per internetą, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.9665&rep=rep1&type=pdf>
- [ 11 ] Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. NLP-Reduce: A “naïve” but Domain-independent Natural Language Interface for Querying

- Ontologies. Prieiga per internetą, [http://gate.ac.uk/sale/dd/related-work/Kaufmann\\_nlp+reduce\\_ESWC2007.pdf](http://gate.ac.uk/sale/dd/related-work/Kaufmann_nlp+reduce_ESWC2007.pdf)
- [ 12 ] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. Prieiga per internetą, <http://gate.ac.uk/sale/dd/related-work/2006+Kaufmann+Querix.pdf>
- [ 13 ] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. Prieiga per internetą, <http://www.larkc.eu/wp-content/uploads/2008/01/2010-Natural-Language-Interfaces-to-Ontologies-Combining-Syntactic-Analysis-and-Ontology-Based-Lookup-through-the-User-Interaction.pdf>
- [ 14 ] Esther Kaufmann, Abraham Bernstein. Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases. Prieiga per internetą, <http://gate.ac.uk/sale/dd/related-work/2009+Kaufmann+Usability+of+interfaces+for+sw.pdf>
- [ 15 ] Silvie Spreeuwenberg and Keri Anderson Healy. SBVR's Approach to Controlled Natural Language. Prieiga per internetą, <http://ceur-ws.org/Vol-448/paper26.pdf>