

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
KOMPIUTERIŲ TINKLŲ KATEDRA

**Andrijanas Pavlovas**

**Semantinis teksto transformavimas ir jo taikymas  
kompiuterinio vertimo sistemose**

**Magistro darbas**

Darbo vadovas

Doc.dr. B. Tamulynas

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
KOMPIUTERIŲ TINKLŲ KATEDRA

**Andrijanas Pavlovas**

**Semantinis teksto transformavimas ir jo taikymas  
kompiuterinio vertimo sistemose**

**Magistro darbas**

Kalbos konsultantas

Lietuvių k. katedros lekt.  
Jonas Jonušas

2006-05

Vadovas

doc.dr. B. Tamulynas  
2006-05

Recenzentas

doc.dr. A. Lenkevičius

2006-05

Atliko

IFM-0/3 gr. stud.  
Andrijanas Pavlovas

2006-05-17

Kaunas, 2006

# Turinys

1	Įvadas .....	4
2	ANALITINĖ DALIS .....	5
2.1	Probleminės srities bendras aptarimas .....	5
2.2	Sprendžiamo uždavinio aktualumo ir naujų sprendimo variantų poreikio pagrindimas .....	5
2.3	Ką kuriama sistema atliks ?.....	6
2.4	Teksto semantikos transformavimo sistema .....	6
2.5	Teksto semantikos apdorojimo sistemos.....	8
2.6	Semantinio transformavimo galimybės internete.....	13
2.7	Bendri reikalavimai sistemai.....	15
2.8	Funkciniai reikalavimai sistemai.....	15
2.9	Nefunkciniai reikalavimai sistemai.....	16
2.10	Semantinio teksto transformavimo samprata .....	16
2.11	Teksto semantikos transformavimo sunkumai .....	20
2.12	Analitinės dalies rezultatas.....	23
3	PROJEKTINĖ DALIS .....	24
3.1	Reikalavimų specifikavimas panaudojant Volere šablona.....	24
3.1.1	Projekto apribojimai .....	25
3.1.2	Funkciniai reikalavimai .....	27
3.1.3	Nefunkciniai reikalavimai .....	34
3.1.4	Projekto išeiğa .....	39
3.2	Programinės įrangos detali architektūra.....	42
3.2.1	Dokumento paskirtis.....	42
3.2.2	Apibrėžimai ir sutrumpinimai .....	42
3.2.3	Architektūros apžvalga .....	43
3.2.4	Architektūros pateikimas.....	43
3.2.5	Architektūros tikslai ir apribojimai .....	43
3.2.6	Panaudos atvejų vaizdas .....	43
3.2.7	Sistemos statinis vaizdas .....	46
3.2.8	Duomenų vaizdas .....	54
4	Vartotojo dokumentacija.....	56
5	Produkto testavimas .....	60
6	Išvados.....	61
7	Literatūra.....	62
8	Summary .....	64
9	Terminų ir santrupų žodynas.....	65
10	Priedai .....	66

# 1 Įvadas

Šiandien anglų kalba dominuoja internetiniuose resursuose, literatūroje. Patyrę interneto vartotojai subjektyviai teigia, kad anglų kalba pateikiama 95% visos prieinamos informacijos. Sudėtinga objektyviai pamatuoti tai. Skirtingomis kalbomis pateikiamos informacijos tyrimai rodo, kad 80% interneto šaltiniuose dominuoja anglų kalba (mokslinėje tematikoje - arti 100%).

Taip dažnai atvejis kada žmogus ieškodamas jam reikiamos informacijos neranda dėl to, kad paprasčiausiai jo gimtąja kalba tokios informacijos nėra. Atsiranda poreikis turėti galimybę naudotis tokia programine įranga, kuri leistų greitai ir efektyviai išversti anglų kalbos tekstą į gimtąją kalbą. Tokios automatinio vertimo priemonės pakankamai brangiai kainuoja ir nėra įkandamos kiekvienam vartotojui. Taip pat automatizuoto vertimo priemonės neturi kai kurių kalbų vertimo modulių.

Mašininis vertimas pagrįstas automatizuoto vertimo technologija, į kurią įsikiša žmogus - redaguoja tekstą prieš vertimą, po vertimo, sąveikauja su programa.

Šiandien Lietuvos rinkoje kol kas nėra automatinio vertimo priemonių iš įvairių užsienio kalbų į lietuvių kalbą. Bet tokios automatinio vertimo sistemos jau yra kuriamos pavienių žmonių ir organizacijų. Svarbiausia yra tai, kad semantinio vertimo būdai šiandien nėra pakankamai tobuli. Kadangi sukurti teksto semantinio apdorojimo mechanizmą ir automatinio vertimo sistema yra gana sudėtinga – tai mano darbe bus analizuojama anglų kalbos teksto semantinio transformavimo specifika, bus sukurta pagalbinė semantinio apdorojimo programa, kurioje bus realizuota keletas pagrindinių anglų kalbos teksto semantikos apdorojimo būdų.

**Darbo tikslas** – sukurti anglų kalbos teksto pagalbinę semantinio transformavimo priemonę.

**Darbo uždaviniai:**

- 1) Panaudoti anglų kalbos semantinio teksto apdorojimo būdus rašant programą bei aprašyti šių būdų taikymo specifiką;
- 2) Išanalizuoti anglų kalbos teksto semantinio apdorojimo priemones;
- 3) Apžvelgti kūriamos programinės įrangos architektūrą;
- 4) Sukurti semantiniam teksto transformavimui reikalingą žodyną;

## **2 ANALITINĖ DALIS**

### **2.1 Probleminės srities bendras aptarimas**

Teksto semantikos transformavimas ir jo taikymas kompiuterinio vertimo sistemose, kuriose anglų kalbos tekstas verčiamas į lietuvių kalbą, nėra tobulas ir nepakankamai efektyviai panaudojamas, todėl tokios automatinio vertimo sistemos į lietuvių kalbą yra kūrimo stadijose, jų kol kas nėra rinkoje, nors poreikis joms yra. Tai galima paaiškinti tuo, kad automatinio vertimo sistemos, kurios verčia tekstą pažodžiui, nėra sudėtingos, bet norint versti tekstą ne pažodžiui, o sakiniai, stulpeliais, tam tikrais žodžių fragmentais - reikia iš pradžių sukurti semantinio teksto transformavimo metodiką, kuri turi būti realizuota tam tikrų algoritmų pagalba, kas yra nelengvas uždavinys.

Teksto reikšmių, o ne fragmentų paieška ne tik užtikrintų kur kas natūresnę ir efektyvesnę bendravimą, naudojant vieną kalbą, bet ir taptų galinga daugiakalbės informacijos tvarkymo sistema. Teksto, pateikto viena kalba, reikšmę galima pagal tam tikras sąlygas surasti ir kita kalba parašytame tekste.

Tokio pobūdžio paieškos sistema leistų gauti nesudėtingus tekstus, įveikiant kalbinius barjerus. Suformulavus užklausimą viena kalba, atsakymus galima būtų gauti skirtingomis kalbomis. Sistema, be jokios abejonės, turi būti aprūpinta priemonėmis ir funkcijomis, kurios užtikrintų intelektualų informacijos pateikimą vartotojui. Tai ir transkripcija, ir automatinis žodynas, mašinos ir žmogaus atliekamas vertimas.

### **2.2 Sprendžiamo uždavinio aktualumo ir naujų sprendimo variantų poreikio pagrindimas**

Šiuo metu rinkoje nėra lietuviškos anglų kalbos teksto semantinio apdorojimo sistemos, kurios pagalba būtų galima supaprastinti anglų kalbos sakinio struktūrą, panaikinti pasikartojimus.

Toks produktas reikalingas, nes:

- Nėra sukurto Lietuvoje efektyvaus analogiško produkto;
- Kiti sukurti pavydžiai paimti iš užsienio programuotojų, kompanijų – nėra pritaikyti Lietuvos rinkai, taip pat ir programuotojams, nes pigiau išsina sukurti analogišką produktą;
- Šalies vidaus rinkoje jau yra bandoma kurti automatizuoto vertimo sistemą, kuri anksčiau ar vėliau atsiras;

- Produkto atsiradimas vidaus rinkoje galėtų paskatinti vystytis ir pigti automatinio vertimo sistemoms.

### 2.3 Ką kuriama sistema atliks ?

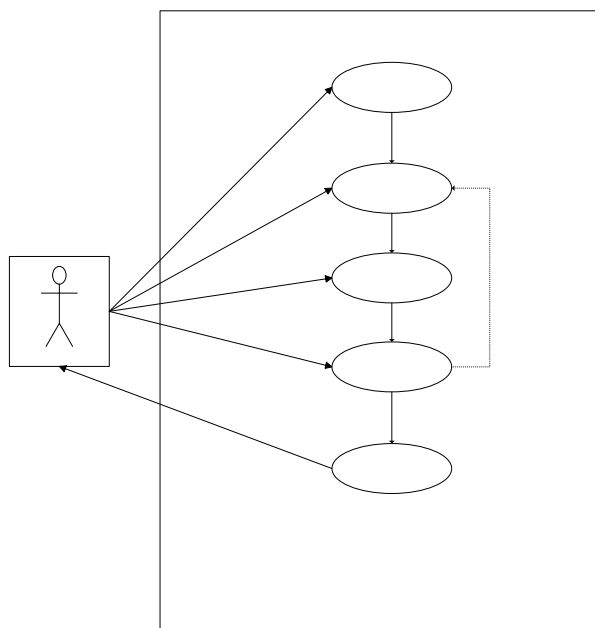
Semantinio teksto transformavimo priemonė transformuos anglų kalbos tekstą, t.y. trumpins sakinius, tikslins daiktavardžius ir pan.

### 2.4 Teksto semantikos transformavimo sistema

Sudarykime UCM („Use case model“) diagramą, vaizduojančią vartotojo ir kompiuterinės sistemos sąveiką, detaliai aprašančią užduotis, informacinius srautus (1 lentelė.).

1 lentelė. Vartotojo ir kompiuterinės sistemos sąveiką

Veiklos dalyvis	Veiklos dalyvio aprašymas
Vartotojas	Vartotojas - asmuo besinaudojantis anglų kalbos teksto semantinio transformavimo sistema. Tai gali paprastas vartotojas, privačios, valstybinės įmonės darbuotojas, moksleivis, studentas. Vartotojas turi turėti įgūdžius dirbti su Windows OS ir MS Office tam, kad galėtų dirbti su anglų kalbos teksto semantinio transformavimo sistema.
Teksto semantikos transformavimo priemonė	Teksto semantikos transformavimo priemonė - tai programinis produktas, veikiantis Windows OS ir MS Office programų aplinkoje. Tai tokia sistema, kur anglų kalbos tekstas semantiškai apdorojamas



1 pav. „Teksto semantikos transformavimo sistemos“ taikomieji uždaviniai

2 lentelė. „Teksto semantikos transformavimo sistemos“ taikomųjų uždavinių aprašymas

Uždavinys	Aprašymas
Teksto paruošimas	Šiame etape vartotojas pateikia semantinio transformavimo sistemai tekstą, kuris turi būti transformuotas
Teksto apdorojimas	Šiame etape teksto semantika transformuojama pagal iš anksto apibrėžtas taisykles
Teksto išvedimas ir kontrolė	Šiame etape tikrinamas transformuotas tekstas, kuris bus atvaizduojamas programoje
Teksto išvedimas	Šiame etape transformuotas tekstas atvaizduojamas programoje
Galutinis rezultatas	Šiame etape vartotojas transformuotą tekstą išsaugoja į tekstinį failą

## 2.5 Teksto semantikos apdorojimo sistemos

### *KANT – anglų kalbos semantinio apdorojimo priemonė*

Ši priemonė susidaro tokių pagrindinių dalių:

- a) Žodynų konstruktorius;
- b) Teksto sakinių konstruktorius;
- c) Kalbos žymėjimų naudojimas (reikalinga gramatinei analizei ir dviprasmiškumų minimizavimui ir kt.).

Kai yra apibrėžta viena kalba ir duomenų failai yra sukurti sistema gali atlikti tokias funkcijas

(1):

- a) Žodynų tikrinimas;

Tikrinimas pradinio žodžio atitikimas analogiško žodžio konstrukcijai, kuris yra žodyne

- b) Gramatikos tikrinimas;
- c) Dviprasmybių šalinimas.

Automatinio vertimo priemonėse žodynai yra specifikuojami į pvz.: techninius, mokslinius.

KANT sistemoje techninis žodynas susideda iš (1):

- a) techninių frazių;
- b) techninių žodžių;
- c) techninių simbolių.

Specializuotų žodynų KANT sistema turi nemažai, taip pat analogiškose sistemose specifikuoti žodynai yra taip pat naudojami.

Į KANT sistema įeina ir leksikos konstruktorius, kuris susideda iš:

- a) ortografijos priemonių;
- b) funkcinių žodžių;
- c) modalinių veiksmažodžių.

Teksto gramatika analizuojama per:

- a) frazių konstruktorių (pvz., turn on gali būti parašyta, kaip start angliškai);
- b) frazių turinčių veiksmažodžius ;
- c) prielinksnes frazes;
- d) naudojant daiktavardinių frazių koordinatorių;
- e) nominalius žodžių junginius.

Teksto sakinių konstruktorius turi atlikti tokias pagrindines funkcijas:

- a) sakinio jungtukų analizė;
- b) giminingų sakinių analizavimas.



Šioje sistemoje dviprasmybių fiksavimas yra realizuotas naudojant specialius žymėjimo kalbos SGML kalbos tegus. Šios formalios kalbos naudojimas svarbus yra tuo, kad:

- formalizuoja dokumento struktūrą;
- riboja painumą analizuojant žodynus;
- sumažina leksinių dviprasmybių skaičių;
- supaprastina teksto analizę.

Žymėjimo kalbos pvz. galėtų būti specialios formos (pvz. cheminės formulės, datos, adresai), matavimų išraiška.

2 pav. ir 3 pav. galime matyti KANTOO (Kant sistemos išplėstas pavadinimas) sistemos architektūrą, jos transformavimo priemonių integraciją ir duomenų struktūrą.

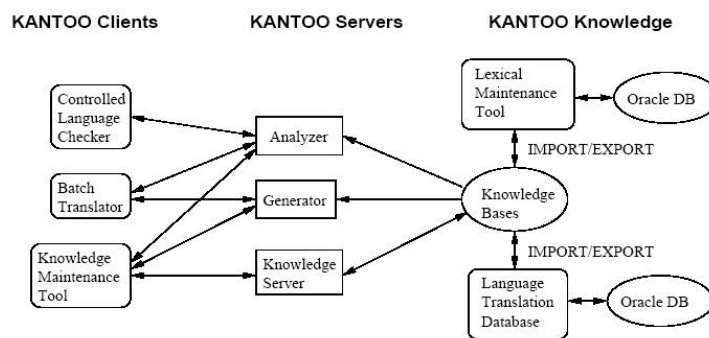


Figure 1: KANTOO Architecture.

2 pav. KANTOO sistemos architektūra

KANTOO sistemoje pradinio teksto įvertinimo priemonės analizuoja:

- Teksto segmentaciją (sakiniai, antraštės);
- Žodžių, frazių atitikimą esamiems žodžiams ir frazėms žodyne;
- Gramatinių teksto atitikimą;
- Automatiškai analizuoja dviprasmybes, šalina jas.

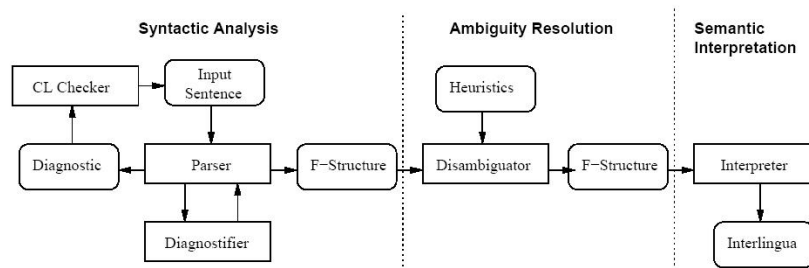


Figure 2: Integration of Diagnostics in the KANTOO Architecture

3 pav. Diagnostikos priemonių integravimas KANTOO architektūroje

## *UCP light priemonė*

UCP light priemonė – tai Upsala chart parser light. Šios priemonės paskirtis yra sintaksinė analizė. Programa parašyta C kalba, bet teksto algoritmai remiasi Lisp programavimo kalba. Duomenų struktūros šioje sistemoje susidaro iš nustatytų viršūnių, briaunų. Šios sistemos fundamentalinė taisyklė pvz. yra tokia (7):

CHART:

|1-----|2-----|3

.np----- d) Passive edge.

+np----- c) Active edge.

+ b) Initial active edge.

.AL-----N----- a) Initial passive edges.

CONFIGURATIONS:

1. active edge (line b) meets passive edge AL (line a)
2. active edge np (line c) meets passive edge N (line a)
3. active edge (line b) meets passive edge (line d)

Figure 2.2: A chart with its

Sakinio „girl handed the baby a toy” rezultatas naudojant šios sistemos gramatinės apdorojimo priemones atrodytų taip:

```
(* = (SUBJ = #0:(SPEC = A
NUM = SG
LEX = GIRL)
TENSE = PAST
PRED = (ACTION = HAND
1 = #0
2 = #2
3 = #1)
#1:OBJ = (SPEC = THE
NUM = SG
LEX = BABY)
#2:OBJ2 = (SPEC = A
NUM = SG
LEX = TOY)))
```

Šis sistema turiu apibrėžtus tokius operatorius, kaip:

- process(arg)
- majorprocess(arg)
- store
- minorstore

Šių operatorių pagalba aprašomi transformuojami duomenis.

## ***SVDA priemonė***

Ši priemonės pagrindu yra SYSTRAN NG technologija (8). Tai yra priemonė skirta sintaksės analizei, vertimui. Šios sistemos pajėgumai:

- a) 30 000 tūkstančių Danų, Švedų, Anglų žodynai;
- b) žodžių rinkinių skaidymas;
- c) sintaksės analizė;
- d) žodžių keitimas abreviatūrinėmis frazėmis.

Sistemos trūkumas – semantinių tegu stoka.

Taisyklės pvz. (Danų žodynas):

```
samtidig      $1->"at the same time";$1~<syntax
type="ADV">+mod_verb</syntax>;
$1-><syntax type="ADV"></syntax>.
```

```
i bedste fald  $1->"at best";$1~<syntax type="ADV"></syntax>;
$1-><syntax type="ADV"></syntax>.
```

Analizės taisyklės pvz:

```
<rule id="AdvP1">
    <match>[ADV:+mod_adv] [ADV]</match>
    <produce pos="ADV" head="$2"></produce>
    <set>$2:+modified_by_adv=$1</set>
</rule>
```

## ***ACE priemonė***

Ši priemonė buvo sukurta programų specifikacijoms rašyti. Iš esmės tai yra tam tikra kalba kuri yra realizuota priemonėje tam tikroms užduotims spręsti. Jos pagrindinis ir esminis skirtumas nuo kitų valdymo kalbų, kad kitose kalbose dviprasmybės analizuojamos leksiškai standartizuojant žodynus, kurios palaiko vertimo sistema, o šioje kalboje dviprasmybės analizuojamos semantiniame, sintaksiniame lygyje (15).

Ši priemonė tris tokius pagrindinius komponentus:

- a) Žodynas su apibrėžtais funkciniais žodžiais (pvz., she, the, and,if,not);
- b) Anglų kalbos gramatikos taisyklių rinkinį;
- c) Priskurimo priemonės, kurios rodo vartotojams kaip konstruoti ir interpretuoti ACE tekstą.

Programos išėjoje tekstas transformuojamas be dviprasmybių į tam tikrą duomenų struktūrą.

Šiose sistemoje sakiniai skirstomi į :

- a) Paprastus (objektas + veiksmažodis + priedinis darinys);
- b) Sudėtinius ( and, or, if-then, who);
- c) Klausiamuosius (not, no).

Taip pat sakiniai gali būti sudaryti iš:

- Daiktavardžio frazės neigimo;
- Veiksmažodžio frazės neigimo;
- Sinonimų ir abreviatūrų.

Veiksmažodžiai gali būti:

- Reiškiantis įvykį;
- Naudojami aktyvioje kalboje
- Ne modaliniai.

Yra ir problemų analizuojant daugiskaitines dviprasmybes, jas transformuojant gali būti prarandamas pirminis sakinio turinys (14). Pvz:

1) The tables as a whole are heavy.

2) Every table is heavy.(anglų k.)

Kaip matome analizuojant ir transformuojant sakinį gali būti prarandama daiktavardžio daugiskaita. Taip 4 pav. struktūriškai galite matyti, kaip šioje sistemoje gali būti pavaizduojama paprastas veiksmas

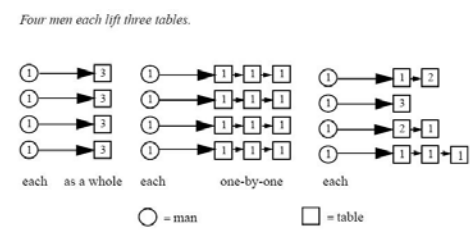


Figure 2 Two Readings of *Four men lift three tables* in ACE.

4 pav. Sakinio „Four men lift three table“ realizacija ACE priemone

### ***CLAT priemonė***

Tai vokiečių sukurtas produktas. Ši produktas yra sukurta profesionaliems redaktoriam ir techninių knygų autoriams. Šio produkto funkcijos yra tokios (10):

- Rašybos tikrinimas;
- Terminologinių ir abreviatūrų tikrinimas;
- Gramatikos tikrinimas pagal pasirinktos kalbos taisykles;
- Stiliaus tikrinimas pagal tam tikrus techninės informacijos rašymo ypatumus.

Tai šis produktas išsiskiria iš kitų tuo, kad gali būti naudojamas šiose platformose:

1. Sun Solaris 7 and higher
2. HP-UX 10 and higher
3. Linux such as SuSE 6.4 or higher
4. Microsoft Windows NT 4.0 with service pack 6
5. Microsoft Windows 2000 with service pack 2
6. Microsoft Windows XP Professional
7. Mac OS X

## 2.6 Semantinio transformavimo galimybės internete

### *Semantinio apdorojimo technologijos internete*

Vieną iš semantinio teksto apdorojimo technologijų internete grindžiama RDF t.y susiję su XML žymėjimo kalbos sintakse. RDF – duomenų modelis meta duomenis pavaizduoti web ištekliuose (11). RDF grindžiamas idėja naudojant URI nuorodas identifikuoti išteklius per RDF sakinius. RDF naudoja tam tikra terminologiją, kurioje predikato apibrėžimas, kuris identifikuoja reikšmę, kurią turi objektas. Štai RDF dokumento kodo pvz:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://www.example.org/about-nic#"
xmlns:bio="http://www.example.org/biology#"
xmlns:con="http://www.example.org/contact#">
<bio:Human rdf:about=#nic>
<con:title>Dr.</con:title>
<con:name>Nic Miller</con:name>
<con:name>Nicolas Miller</con:name>
<bio:pet>
<bio:Labrador>
<con:name>Rex</con:name>
</bio:Labrador>
</bio:pet>
</bio:Human>
</rdf:RDF>
```

O žemiau pateikta RDF/XML ištraukos automatinės transformacijos pvz:

```
@prefix : <http://www.example.org/biology#>.
@prefix con: <http://www.example.org/contact#>.
<#nic> a :Human;
:pet [a :Labrador; con:name "Rex"];
con:name "Nic Miller", "Nicolas Miller";
con:title "Dr."
```

Tam tikri RDF ribojimai lėmė OWL kalbos vystymąsi. pvz:

```
<owl:Class rdf:ID="Man">  
<rdfs:subClassOf rdf:resource="#Human" />  
<owl:equivalentClass rdf:resource=  
"#MaleAdult" />  
</owl:Class>
```

PENG – D valdymo kalbos naudojimas internete apibrėžė taisykles pagal kurias turi būti apdorojamas įvedamas tekstas. Šioje kalboje paprasta sakinio struktūra atrodo taip:

sentence =>subject + predicate  
subject =>nominal head  
subject =>specifier  
{+ pre-nominal modifier} + nominal head  
{+ post-nominal modifier} predicate ! verbal head + complement

Pvz2:

Taip pat yra naudojami konjunkcijos, disjunkcijos ir šios kalbos konstruktoriai

CL: If X is married and X is a woman then X is  
a wife.

HL: wife(X) <=married(X), <= woman(X).

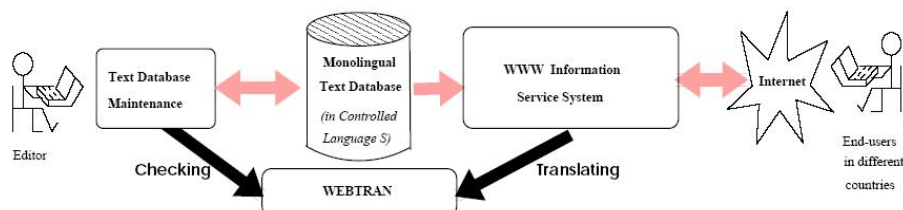
### ***Webtran priemonė***

Šioje priemonėje semantinio teksto transformavimo metodai realizuoti programinėje įrangoje, kuria naudojama internetu (12). Šios priemonės vartotojai gali būti klasifikuojami į tris grupes:

- valdymo kalbos specifikacijų kūrėjai (profesionalūs vertėjai)
- turinio redaktoriai (naudoja webtran sistemos sintaksės arba semantikos tikrinimo priemones)

Webtran programinė įranga susideda iš 2 pagrindinių dalių:

- specifikacijų dalies (Webtran modeller)
- transliatoriaus (Webtran translator)



5 pav. Vertimo sistemos paslaugos naudojimosi struktūra

Kitame, 6 pav. matome šios programinės įrangos struktūrą, kokiais žingsniais apdorojami duomenis ir kaip vyksta informacijos mainai.

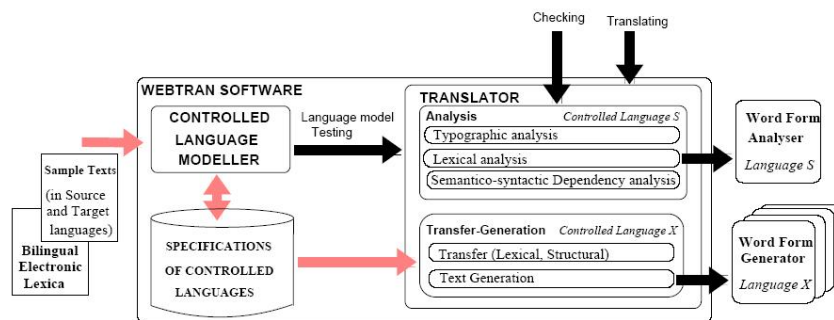


Figure 1: Webtran Software - a building block for multilingual WWW services.

6 pav. Webtran struktūra

## 2.7 Bendri reikalavimai sistemai

### *Produkto apibrėžimas*

Sukurti anglų kalbos teksto semantikos apdorojimo sistema kartu panaudojant semantinio teksto transformavimo būdus.

Ši sistema veiks tokiu principu vartotojas nukopijavęs į atminties buferį angliškai parašyta tekstą, kurio semantika bus apdorota ir pateikta programos informaciniame objekte – vizualinėje formoje.

Kuriama anglų kalbos teksto semantikos apdorojimo sistema pasižymės “draugiška” vartotojo ir kompiuterinės sistemos sąsaja.

### *Vartotojų apibūdinimas*

Programos vartotojui reikalingi darbo su Windows OS ir MS Office įgūdžiai.

## 2.8 Funkciniai reikalavimai sistemai

1. Produktas turi veikti Windows 9X/ME/2000/XP operacijų sistemose, taip pat yra planuojama adaptuoti šią sistemą Linux operacines sistemos „šeimai“.
2. Produktas realizuotas naudojant Borland JBuilder programavimo aplinką
3. Kompiuteryje turi būti instaliuotas Windows XP, Windows 2000, Windows Me
4. Produktas nereikalauja atskiro diegimo
5. Produkte numatytos galimybės:
  - a) pasirinkti semantinio apdorojimo metodus;
  - b) pasirinkti pradinius duomenis.

## 2.9 Nefunkciniai reikalavimai sistemai

1. Produkto elgsena ir valdymas turi atitikti šiuolaikinius grafinės vartotojo terpės reikalavimus ir įgyvendinti šias sąlygas:

- a) suprantamas Windows 9X/ME/XP, MS Office 2000/XP, IE 5/6 vartotojams;
- b) veikti saugiai ir patikimai: nekelti grėsmės kitai programinei įrangai, duomenims, aparatinei įrangai, netrukdyti kitų sistemų darbui, nesukelti MS Ofise „lūžių“, veikti patikimai, estetiškai menu langai, nekenkti vartotojo sveikatai;
- c) iškilusios išimtinės situacijos turi būti tinkamai apdorotos programos viduje, apie jas informuojant vartotoją;
- d) sistemos įdiegimui, vartotojas gali pareikalauti administratoriaus teisių.

2. Produkto modifikavimas, platinimas, kopijavimas ir kopijų laikymas turi būti suderintas su programinio produkto autoriumi. Programinis produktas ginamas autorinėmis teisėmis, pagal Lietuvos Respublikos įstatymus.

## 2.10 Semantinio teksto transformavimo samprata

Semantinis teksto transformavimas – tai yra pradinio teksto apdorojimas pagal iš anksto apibrėžtas taisykles ir metodus. Turint pradinį tekstą yra poreikis jį apdoroti, tai yra supaprastinti, apdoroti gramatinę, sintaksinę, semantinę dalis ir transformuoti į adaptuotą tekstą. Tokiu būdu semantinio teksto transformavimo paskirtis yra:

- a) Apdoroti sintaksinę teksto struktūrą;
- b) Apdoroti sakinių semantines konstrukcijas.

Kita vertus, semantinis teksto transformavimas - apdorojimas pagal tam tikrą metodiką, o ši metodika gali būti apibrėžta, kaip teksto valdymo kalba, kurios galutinis rezultatas - transformuotas tekstas. Transformuojant teksto semantiką pasiekiami tokie rezultatai:

- a) Tekstas gali būti supaprastintas išmetus iš jo pasikartojančias frazes arba „sutraukus“ jas į kitą žodžių kombinaciją;
- b) Pakeičiama teksto semantika, nes apdorojant „įvedamo“ teksto elementai yra keičiami kitais žodžių junginiais, kurie atitinka keičiamų žodžių reikšmes;
- c) Gaunamas tam tikras standartizuotas pagal tam tikras taisykles tekstas, kuris yra gaunamas pasinaudojus valdymo kalbos metodais.

Semantinis teksto apdorojimo metodų naudojimas teksto valdymo kalboje duoda galimybę turint tą patį tekstą skirtingose kalbose jį standartizuoti ir transformuoti pagal vienodas taisykles. Ypač teksto valdymo kalbos naudingumas yra pastebimas komerciniuose



programiniuose paketuose, kuriose vienas tekstas gali būti transformuojamas į kitą naudojant skirtingų kalbų žodynus. Teksto valdymo kalbos naudojimas kartu su programine įranga reikalauja vartotojo įgudimo. Naudojant teksto valdymo kalbą gali būti pasiektas užsibrėžtas tikslas. Tekstas gali būti transformuotas, ir jis gali būti aiškiai suprantamas vartotojui, o tai gali būti pasiekta tik tuo atveju, jeigu semantinio teksto transformavimo mechanizmai yra efektyvūs.

### ***Semantinio teksto transformavimo svarbumas***

Semantinis teksto transformavimas ir jo naudojimas vertimo sistemose yra svarbus šiais atžvilgiais:

a) Komercinis aspektas

Anglų kalbos teksto semantinis transformavimas gali duoti komercinę naudą, sukūrus anglų kalbos teksto semantinio transformavimo priemonę, kuri realiai gali būti paklausi rinkoje.

b) Mokslinis aspektas

Mokslinis aktualumas taip pat yra svarbus, kadangi semantinio teksto transformavimo priemonės yra kuriamos automatinio vertimo sistemų kompanijoms. Bet šios priemonės gali būti neefektyvios. Todėl yra reikalingas mokslinis problemos sprendimas, turi būti parengta programinio produkto specifikacija.

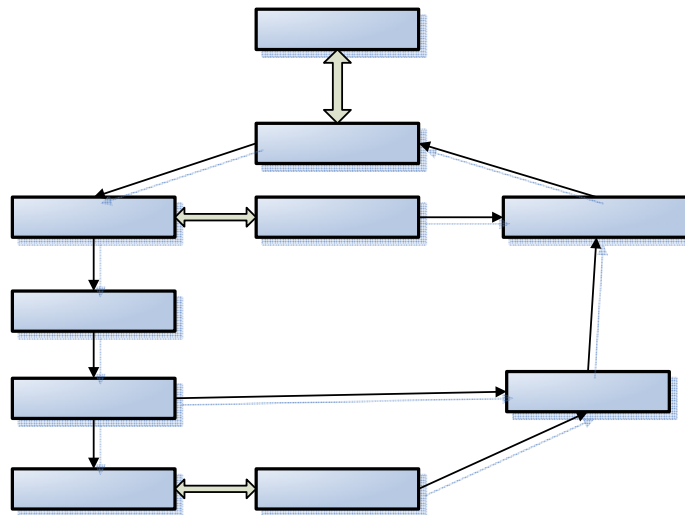
### ***Taisyklė, taikomos apdorojamam tekstui***

Viena iš pagrindinių teksto valdymo kalbos programinės įrangos sudedamųjų dalių yra teksto nagrinėjimo elementas. Šiame elemente yra apibrėžtos taisyklės, kuriomis naudojasi programa, kuri transformuoja tekstą, t.y. ji supaprastina tekstą, įterpia kalbos elementus (2). Yra apibrėžtos tokios pagrindinės taisyklės, kurios yra taikomos tekstui paruošiant jį galutiniam transformavimui į kitą kalbą (3):

- a) Sakinio trumpinimas;
- b) Pasikartojančių žodžių keitimas kitais arba išmetimas;
- c) Loginės sakinio struktūros tikrinimas;
- d) Santrupų analizavimas;
- e) Modalinių veiksmažodžių kontrolė;
- f) Daiktavardinių konstrukcijų analizė.

Teksto sintaksė analizuojama teksto gramatinio apdorojimo priemone. Jos pagalba yra analizuojama teksto sintaksė, žodžių junginiai, daiktavardžiai, frazės. Pagal 7 pav. pateiktus duomenis matome, kad iš pradžių yra paruošiamas tam tikras tekstas, kuris turi būti

transformuojamas. Toliau šis tekstas yra įvedamas į tam tikros automatinio vertimo priemonės dialogo langą. Kitas žingsnis yra teksto paruošimas vertimui, analizuojama teksto sakinių, žodžių struktūra. Kai tai jau yra atlikta pereinama prie kito žingsnio - transformavimo į kitą kalbą. Automatinio vertimo priemonės, PROMT, Systran turi galimybę transformuoti į dešimtis kalbų. Pasirenkama, į kokią kalbą reikėtų transformuoti tekstą. Įvykus teksto transformavimo procesui, informacija „paduodama“ programos algoritmui, kuris per atsakymų generatorių ir dialoginį komponentą pateikia informaciją vartotojui (4).



7 pav. Teksto transformavimo į kitą kalbą schema

Pažymėtini tokie komponentai, kaip lingvistinis žodynas, informacijos saugykla, informacijos absorbuotojas. Jie reikalingi tam, kad vykdant teksto semantikos transformavimo operacijas, gramatinių, morfologinių taisyklių duomenys būtų imami iš šių priemonių. Bet tam, kad būtų galima išvengti tam tikrų netikslumų transformuojant tekstą reikėtų vengti:

- a) gramatinių klaidų;
- b) įvairių sutrumpinimų ;
- c) slengo.

Parseris

Taip pat reikėtų:

- 1) prielinksninių frazių, kurios modifikuoja veiksmažodžio reikšmę;
- 2) gimininguose sakiniuose modifikuoti pirmiausia daiktavardžius;
- 3) santrumpų naudojimo pilna forma.

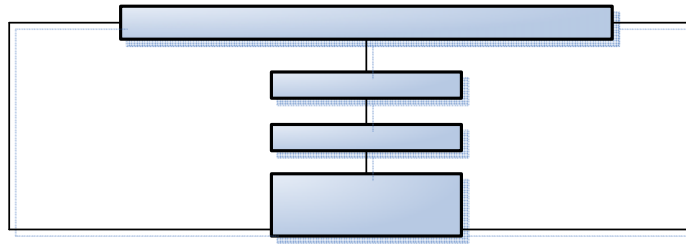
Pranešimų priemonė

Viena iš teksto valdymo kalbos priemonių yra morfologijos tikrinimas. Iš 8 pav. matome, kad iš pradžių yra vykdoma sakinio morfologinė analizė, t.y analizuojama žodžių struktūra (6).

Vykdoma morfologinę

Transliatorius į n kalbą

informacijos  
absorbuotojas



8 pav. Sakinio apdorojimo žingsniai

teksto analizę, žodžiai gali būti apibrėžti kaip derinys. Morfologinės schemas pavyzdžiais gali būti prefiksai, sufiksai ir pan. dalykai. Morfologinės schemas gali būti klasifikuojamos į turinio žodžius ir funkcinius žodžius. Turinio žodžiai - tai tokie žodžiai, kaip daiktavardžiai, veiksmažodžiai, būdvardžiai. Funkciniai žodžiai – tokie žodžiai, kurie yra sintaksinės kalbos dalys (pvz. in, the, and with ir kt.). Morfologinės ypatybės „duoda“ informaciją apie skirtingas kalbos dalis. Pavyzdžiui, sufiksai padeda apibrėžti veiksmažodžio laiką galūnes. Yra taip vadinamasis specialių žodžių junginys, kurį sudaro skirtingos žodinės kombinacijos (pvz., bathroom-vonios kambarys, kur bath-vonia, room-kambarys; screwdriver - atsuktuvus), kurių žodžiais gali būti daiktavardžiai, veiksmažodžiai ir kitos kalbos dalys.

### ***Teksto turinio struktūros loginės formos***

Apdorojant pradinio teksto semantiką yra svarbu tinkamai apdoroti sakinių turinio loginę formą. Yra siūloma naudoti „lygias“ logines formas, kurios pasižymėtų tokiomis charakteristikomis (8).

- 1) Jei pradinio teksto gramatinio nagrinėjimo elementas „nesugeba“ galutinai apdoroti tekstą, svarbi yra bent dalis teksto, kurį įmanoma apdoroti, transformuoti ir prezentuoti;
- 2) Loginių taisyklių didinimas apdorojamam tekstui būdingas tuomet, kai žmogus „valo“ pradines loginės formos tikrinimo taisykles pakeisdamas jas didesniu informacijos kiekiu, bet ne taisykle;
- 3) Paprastas dviejų sakinių semantinių turinių palyginimas apdorojant jų logines formas leidžia supaprastinti sakinio struktūras;

## 2.11 Teksto semantikos transformavimo sunkumai

### *Apie teksto semantika*

Teksto semantika yra tiesiogiai susijusi su žodžių reikšme, kuri turi vieną ar kitą prasmę. Kaip jau ir buvo minėta geriau yra atskirti leksikografinę semantiką ir struktūrinę semantiką.

Tuos pačius žodžius ir iš jų sudarytą sakinį galima suprasti skirtingai. Pvz. angliški žodžiai, man, woman, boy, girl gali būti išreikšti kaip(4):

**man = (+HUMAN, +MASCULINE and +ADULT);**  
**woman = (+HUMAN, -MASCULINE and +ADULT);**  
**boy = (+HUMAN, +MASCULINE and -ADULT);**  
**girl = (+HUMAN, -MASCULINE and -ADULT)**

Tai yra šiuo atveju žodis yra tam tikru būdu užkoduojamas, o tai duoda:

1. Efektyviau išnaudojami kompiuterio našumai;
2. Sutaupoma vieta kompiuterio kietajame diske;
3. Operacijos įvykdomos sparčiau naudojant optimalesnį algoritmą, kuris naudojamas analizuojant duomenis su programinio produkto žodyno informacija.

Štai angliško žodžio game pavyzdys, kuris turi daug prasmių ir gali reikšti:

- 1) a form of play (žaidimo forma);
- 2) a sport (sportą);
- 3) wild animal hunted (laukinio gyvūno medžioklę);
- 4) fished for food (pagautas maistui).

Naudodami apribojimus, naudojančius tam tikrą kriterijų (pvz., maistas – valgomas dalykas), mes galime atmesti nereikalingus variantus, kurių mums nereikia transformavimui. Tokių apribojimo kriterijų naudojimas automatinio vertimo sistemose padeda ženkliai sumažinti dviprasmybių skaičių(2).

### *Leksikografiniai ir struktūriniai neatitikimai*

Kaip buvo minėta, kiekvienas žodis negali turėti tik vienos reikšmės, todėl tokiu atveju atsiranda leksikografinių ir struktūrinių neatitikimo klaidų. Taip yra todėl, kad tarp skirtingų kalbų egzistuoja leksikografiniai skirtumai, dėl kurių ir atsiranda problemos transformuojant tekstą. Kitos problemos atsiranda todėl, kad skirtingos kalbos naudoja skirtingas semantines struktūras rezultatui gauti (11).

### *Analizės sunkumai*

Analizės komponento, naudojamo vertimo sistemose, tikslas yra paimti pradinį tekstą ir apdorojus jį pateikti rezultata. Todėl atsiranda problema nes kiekvienas žodis transformuojamas į kitą pažodžiui, dėl to nukenčia turinys ir sakiny po transformavimo į kitą kalbą gali būti nesuprantamas. Taip pat yra ir kitų sunkumų:

- a) Pradinį tekstą dažnai sudaro sakiniai, kurie yra blogai parašyti, todėl teksto analizės priemonės turi turėti tokį teksto apdorojimo algoritmą, kad galėtų susidoroti su tokia situacija;
- b) Kaip jau buvo minėta ankstesnėje tiriamojo darbo dalyje, tekstas dažnai turėti daugiareikšmių žodžių;

Optimalus variantas būtų toks, kad kiekvienas žodis turėtų vieną reikšmę, bet taip nėra, todėl jeigu žodis turi daugiau negu vieną reikšmę sakoma, kad toks žodis yra leksikografiniu požiūriu yra dviprasmiškas. Kaip pavyzdį būtų galima paminėti tokį atvejį, kai sakinį sudaro dešimt žodžių, kurie turi leksinį dviprasmiškumą, dvi galimas struktūrines dviprasmybes. Sakinys gali turėti 18 skirtingų analizės rezultatų (vertimų). Toks analizių skaičius gali būti problemiškas, nes iš esmės reikia pasirinkti tik vieną iš 18. Tokiu būdu problemiškas yra įvertinimas, nes nežinoma kokia būtent analizė yra geriausia. Štai keltas dviprasmybių pavydžių:

Leksichografinės dviprasmybės:

- (1) a. They are trying to design a better pen. ('writing implement' or 'animal enclosure'?)  
b. Our Jimmy has grown another foot. ('limb' or 'unit of measurement'?)  
c. The post has arrived. ('delivery of mail' or 'piece of wood'?)

Struktūrinės dviprasmybės:

- (2) . Concern has been expressed about conditions in the factory near the river that was polluted last week.  
b. The minister stated that the proposal was rejected yesterday.  
c. Some young girls and boys have arrived.  
d. Sam has joined a student film society.

### *Aprašymo problema*

Kiekviena kalba turi savo gramatiką, sintaksę. O tam, kad būtų įmanomas teksto semantikos transformavimas reikalingas gramatinių, sintaksinių taisyklių detalus aprašymas, kad teksto turinys būtų kaip galima aiškiau transformuotas į kitą kalbą naudojant automatinio vertimo priemones. Todėl automatinio vertimo sistemose teksto semantikos transformavimo priemonės turi turėti nemažą programinio kodo eilučių skaičių. Taip pat teksto transformavimo priemonės turi tendenciją “didėti” priklausomai nuo to, kiek kalbų palaiko vertimo priemonė.

Tam, kad vertimo priemonė vertimą “atlikinėtų” gerai, reikia:

- a) žodynų;
- b) morfologinių taisyklių rinkinio;
- c) sintaksinių/semantinių taisyklių;

Tam, kad būtų minimizuotas aprašymo problemų atsiradimas, reikia:

- a) minimizuoti žodynus, specifikuoti juos;
- b) gramatinės taisyklės turi būti trumpai ir aiškiai apibrėžtos;

## 2.12 Analitinės dalies rezultatas

Aprašius tai, kas yra anglų kalbos teksto semantikos transformavimas, kokie yra šio transformavimo trūkumai, privalumai, apžvelgus kokios teksto semantikos transformavimo programines priemones egzistuoja, šiame darbe pagrindiniu tikslu pasirinkau sukurti anglų kalbos teksto pagalbinę semantinio transformavimo priemonę. Ši priemonė atliks tokias pagrindines operacijas:

- 1) Trumpins sakinį;
- 2) Šalins modalinius veiksmažodžius;
- 3) Šalins sakinyje dviprasmybes, pasikartojimus;
- 4) Detalizuos santrumpas;
- 5) Tikslins daiktavardžius;
- 6) Sudėtinius veiksmažodžius keis kitu veiksmažodžiu atitinkančiu reikšmę;

## 3 PROJEKVINĖ DALIS

### 3.1 Reikalavimų specifikavimas panaudojant Volere šabloną

#### *Sistemos paskirtis*

Anglų kalbos teksto semantinio transformavimo programa skirta atlikti teksto transformacijai turės trumpinti sakinį, supaprastinti jo struktūrą, išmesti pasikartojimus tekste, kt.

#### *Projekto kūrimo pagrindas*

Pats projekto pagrindas pagrįsta tuo, kad realiai jau egzistuoja analogiškos teksto semantinio transformavimo priemonės, kurios yra naudojamos kitose valstybėse. Kitose šalyse jau egzistuoja tokių priemonių kūrimo metodikos, algoritmai yra gan išbaigti, bet didžiausia problema yra tai, kad tie projektai yra komerciniai ir programų source'as nėra prieinamas. Negalima praktiškai pažiūrėti kaip yra aprašytos teksto transformavimo procedūros, funkcijos, yra tam tikra problema semiantis patyrimo, nors teorinė metodika yra paruošta ir egzistuoja. Todėl kuriant semantinio teksto transformavimo programą aš pasistengsiu ne tik teoriškai, bet ir praktiškai ištestuoti programą paeksperimentuoti, rasti trūkumus ir privalumus.

#### *Posistemio paskirtis, tikslai*

Sukurti anglų kalbos teksto semantikos transformavimo praktinę realizaciją - programą, kurią pritaikius ateityje būtų galima sukurti „išbaigtą“ automatinio vertimo sistemą, kurios pagalba būtų galima versti užsienio kalba parašytus tekstus į lietuvių kalbą.

#### *Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys*

Užsakovas:

Doc. Dr B.Tamulynas

KTU, Informatikos fakultetas, Kompiuterinių tinklų katedra

tel: 300369; el. paštas: [bronius.tamulynas@ktu.lt](mailto:bronius.tamulynas@ktu.lt)

Posistemio kūrėja:

IFM-0/3 gr. magistrantas

Pavlovas Andrijanas, [studentas@inbox.ru](mailto:studentas@inbox.ru)

#### *Potencialūs vartotojai*

- Studentai;
- Dėstytojai;



- Aukštiesios mokyklos, Universitetai;
- Juridiniai asmenys;
- Valstybinės įstaigos.

Vartotojo tikslai:

- Turi būti lengvai naudojama;
- Programinių klaidų nebuvimas;
- Intuityviai suprantama;
- Pilnai dokumentuota;
- Programinės priemonės išbaigtumas;
- Interaktyvumas.

### 3.1.1 Projekto apribojimai

#### *Apribojimai sprendimui*

- Kuriama semantinio teksto transformavimo programa anglų kalbos teksto semantikai transformuoti ir paruošti vertimo sistemai
- Programinė įranga kuriama Java programavimo kalba, todėl ji gali būti naudojama ne tik tokiose operacinėse sistemose, kaip Windows XP, Windows 2000, Windows 98 se, bet Linux operacinėse sistemose

#### *Diegimo aplinka*

Programinė įranga veiks bet kokioje operacinėje sistemoje ir bet kokioje platformoje, vienintelis reikalavimas turi būti suinstaliuota JVM (Java virtual machine).

Minimalūs reikalavimai vartotojų programinei įrangai:

- Windows 9x/2000/XP operacinės sistema;
- Procesorius- 500 Mhz

#### *Bendradarbiaujančios sistemos*

Semantinio teksto transformavimo programa nereikalauja specialių palaikymo priemonių, įvairių dll bibliotekų ir pan. Ši programa yra diegiama, kaip vienalytis programinis produktas nereikalaujantis specialaus palaikymo išskyrus JVM.

### ***Komerciniai specializuoti programų paketai***

Programos kūrimui yra naudojama Borland JBuilder X programavimo aplinka, taip pat programos kompiliavimui naudojamos nemokamos priemonės.

### ***Numatoma darbo vietos aplinka***

- Programa nereikalauja specialaus išskirtinio autorizavimo
- Laisvas priejimas prie reikiamų resursų, pasidalinimas resursais
- Spausdintuvas prieinamas per tinklą.

### ***Sistemos kūrimo terminai***

- Reikalavimų analizė (kovas, 2006 m.).
- Programų sistemos kūrimo, vystymo projektas (balandis, 2006 m.).
- Posistemių kūrimas (vystymas) ir testavimas (balandis, 2006 m.).
- Posistemių integracija (balandis, 2006 m.).
- Programų sistemos testavimas (balandis, 2005 m.).
- Programų sistemos diegimas (gegužė, 2005 m.).

### ***Sistemos kūrimo biudžetas***

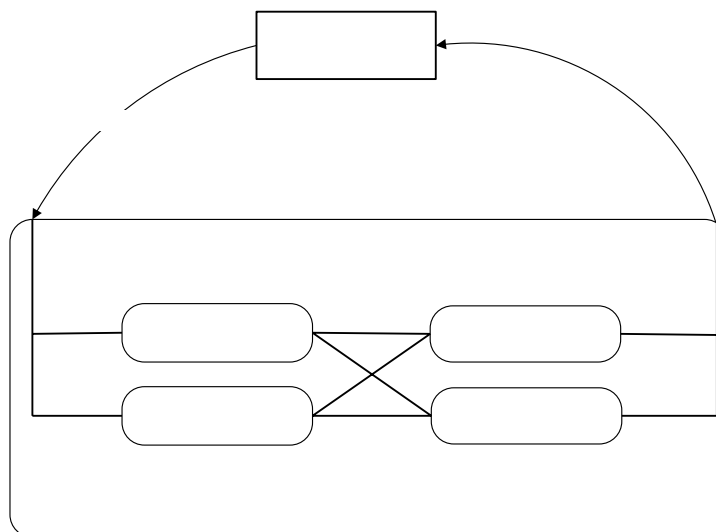
Semantinio teksto transformavimo programinė priemonė kuriama tiriamojo darbo rėmuose, ir yra mokslinis, bet ne komercinis produktas.

### ***Prielaidos***

- Nebus konfliktų tarp kuriamos programinės priemonės procesų
- Kuriama programa yra pilnai suderintai su anksčiau paminėtomis operacinėmis sistemomis
- Vartotojai turi galimybę užsakyti papildomų programinių modulių kūrimą
- Kuriama programinė įranga nebus viešai licencijuota, nes kuriama tiriamojo darbo kontekste

### 3.1.2 Funkciniai reikalavimai

#### Veiklos kontekstas (diagrama)



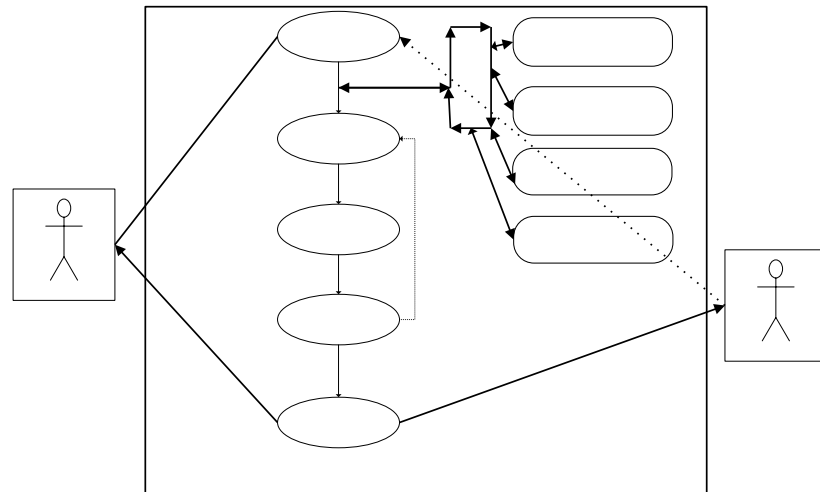
9 pav. Konteksto diagrama **Duomenų transformavimo procesas**

#### Veiklos padalinimas

3. Lentelė. Veiklos įvykių sąrašas

Nr.	Eil.	Įvykio pavadinimas	Įeinantys(in)/Išeinantys(out) informacijos srautai
1		Vartotojas paduoda pradinį tekstą	Procedūras (in)
2		Sistema klausia vartotojo kokį veiksmą pasirinkti	Užklauskimas (out)
3		Vartotojas pasirenka 1 iš 6 pagrindinių operacijų	Pasirinkimas (in) Pagal pasikartojimus
4		Programa atlieka teksto transformacijas	Transformacijos (in)
5		Išvedamas transformuotas tekstas vartotojui	Tekstas (out) Teksto transformavimas Pagal sakinio struktūrą

## Posistemio ribos



10 pav. Posistemio ribos

## Panaudos atvejų sąrašas

Kadangi ne visi diagramoje parodyti panaudojimo atvejai (kaip procedūrų pasirinkimas, pažymėti rodyklėmis ovalinius blokus) gali būti nagrinėjami kaip savarankiškos operacijos, o yra detalizuojami kitų panaudojimo atvejų, tuomet jie sudaryti realių sistemos panaudojimo atvejų sąrašą:

- Semantinis teksto transformavimas pagal pasikartojimus tekste
- Semantinis teksto transformavimas pagal sakinio struktūrą
- Semantinis teksto transformavimas skaidant sakinį
- Semantinis teksto transformavimas pagal kalbos dalis

<b>1. PANAUDOJIMO ATVEJIS:</b>	Semantinis teksto transformavimas pagal pasikartojimus tekste
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Apima procesą, kuriuo metu tekstas transformuojamas pagal tekste pasikartojimų apdorojimo atvejus
<b>Prieš sąlyga:</b>	Tekstas nėra apdorotas
<b>Sužadinimo sąlyga:</b>	Atsirado vartotojas su poreikiu atlikti būtent tokią operaciją
<b>Po-sąlyga:</b>	Pasirinktas tekstas transformuojamas
<b>Alternatyvūs scenarijai:</b>	<ul style="list-style-type: none"><li>- Sistemoje nėra nei vieno failo.</li><li>- Vartotojas nutraukia darbą / pasirenka kitą veiksmą.</li><li>- Sistemos klaida.</li></ul>

Teksto pa

Teksto apc

Teksto iš

Teksto išvedim

Galutinis r

<b>2. PANAUDOJIMO ATVEJIS:</b>	Semantinis teksto transformavimas pagal sakinio struktūrą
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Apima procesą, kuriuo metu tekstas transformuojamas pagal sakinio struktūrą
<b>Prieš sąlyga:</b>	Tekstas nėra apdorotas
<b>Sužadinimo sąlyga:</b>	Atsirado vartotojas su poreikiu atlikti būtent tokią operaciją
<b>Po-sąlyga:</b>	Pasirinktas tekstas transformuojamas
<b>Alternatyvūs scenarijai:</b>	<ul style="list-style-type: none"> <li>- Sistemoje nėra nei vieno failo.</li> <li>- Vartotojas nutraukia darbą / pasirenka kitą veiksmą.</li> <li>- Sistemos klaida.</li> </ul>

<b>3. PANAUDOJIMO ATVEJIS:</b>	Semantinis teksto transformavimas skaidant sakinį
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas
<b>Aprašas:</b>	Apima procesą, kuriuo metu tekstas transformuojamas skaidant sakinį
<b>Prieš sąlyga:</b>	Tekstas nėra apdorotas
<b>Sužadinimo sąlyga:</b>	Atsirado vartotojas su poreikiu atlikti būtent tokią operaciją
<b>Po-sąlyga:</b>	Pasirinktas tekstas transformuojamas
<b>Alternatyvūs scenarijai:</b>	<ul style="list-style-type: none"> <li>- Sistemoje nėra nei vieno failo.</li> <li>- Vartotojas nutraukia darbą / pasirenka kitą veiksmą.</li> <li>- Sistemos klaida.</li> </ul>

<b>4. PANAUDOJIMO ATVEJIS:</b>	Semantinis teksto transformavimas pagal kalbos dalis
<b>Vartotojas/Aktorius:</b>	Sistemos vartotojas.
<b>Aprašas:</b>	Apima procesą, kuriuo metu tekstas transformuojamas pagal kalbos dalis, laikus
<b>Prieš sąlyga:</b>	Tekstas nėra apdorotas
<b>Sužadinimo sąlyga:</b>	Atsirado vartotojas su poreikiu atlikti būtent tokią operaciją
<b>Po-sąlyga:</b>	Pasirinktas tekstas transformuojamas

<b>Alternatyvūs scenarijai:</b>	<ul style="list-style-type: none"> <li>- Sistemoje nėra nei vieno failo.</li> <li>- Vartotojas nutraukia darbą / pasirenka kitą veiksmą.</li> <li>- Sistemos klaida.</li> </ul>
---------------------------------	---

### ***Funkciniai reikalavimai***

Funkcinių reikalavimų sąrašas:

R1 Sistema turi interaktyviai atvaizduoti semantinio teksto transformavimo rezultatą

R2 Sistema turi leisti darbuotojui pasirinkti duomenis, kurie bus transformuojami

R3 Sistema turi „pasiūlyti“ vartotojui, kaip modifikuoti pasirinktą tekstą

R4 Sistema turi galimybę pasiūlyti vartotojui modifikuotą pasirinktą tekstą keliais būdais

R5 Sistemoje turi būti numatyta vartotojui galimybė transformuotą tekstą rankinių būdu keisti

R6 Sistemoje turi būti numatyta duomenų išsaugojimas pagal tam tikrą šabloną priklausomai nuo pasirinktos teksto modifikavimo funkcijos

R7 Sistemoje turi būti numatyto operacijų tikrinimo žingsnių kontrolė

R8 Sistemoje turi būti numatyta vartotojo informavimo procedūros

<b>Reikalavimas #:</b>	1	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	?
<b>Aprašymas:</b>	Sistema turi interaktyviai atvaizduoti semantinio teksto transformavimo rezultatą			
<b>Pagrindimas:</b>	Sistema atvaizduoja vartotojui sistemos darbo veiklos rezultatą			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Į programinės įrangos specialia informacijos išvedimo formą išvedamas rezultatas			
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	2	
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.			

<b>Reikalavimas #:</b>	2	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	?
------------------------	---	---------------------------	--------------------------------------	---

<b>Aprašymas:</b>	Sistema turi leisti darbuotojui pasirinkti duomenis, kurie bus transformuojami		
<b>Pagrindimas:</b>	Failo pasirinkimas		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Įvykdžius operaciją turime tekstinį failą, kuris bus modifikuotas		
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	1
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	3	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4
<b>Aprašymas:</b>	Sistema turi „pasiūlyti“ vartotojui, kaip modifikuoti pasirinktą tekstą			
<b>Pagrindimas:</b>	Vartotojas pasirenka kokias semantinio teksto modifikavimo procedūras pasirinkti, kad transformuoti tekstą			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Sistemoje užkraunamos procedūros pagal kurias bus modifikuojamas tekstas			
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	3	
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.			

<b>Reikalavimas #:</b>	4	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4
<b>Aprašymas:</b>	Sistema turi galimybę pasiūlyti vartotojui modifikuoti pasirinktą tekstą keliais būdais			
<b>Pagrindimas:</b>	Atlikęs teksto modifikavimą pagal pasirinktą vieną funkciją vartotojas turi galimybę			

	papildomai modifikuoti tekstą naudojant kita procedūra		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Sistema yra pasiruošusi atlikti pasirinktos papildomos funkcijos vykdymą teksto modifikavimui		
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	2
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	5	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2
<b>Aprašymas:</b>	Sistemoje turi būti numatyta vartotojui galimybė transformuoti tekstą keisti			
<b>Pagrindimas:</b>	Vartotojas kartais turi poreikį papildomai modifikuoti tekstą rankiniu būdu			
<b>Šaltinis:</b>	Vartotojas			
<b>Tinkamumo kriterijus:</b>	Sistemoje atsiranda vartotojo rankinių būdu atliktos teksto modifikacijų rezultatas			
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	3	
<b>Priklausomybės:</b>		<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.			

<b>Reikalavimas #:</b>	6	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4
<b>Aprašymas:</b>	Sistemoje turi būti numatyta duomenų išsaugojimas pagal tam tikrą šabloną priklausomai nuo pasirinktos teksto modifikavimo funkcijos			
<b>Pagrindimas:</b>	Išsaugomas tekstas			
<b>Šaltinis:</b>	Vartotojas			



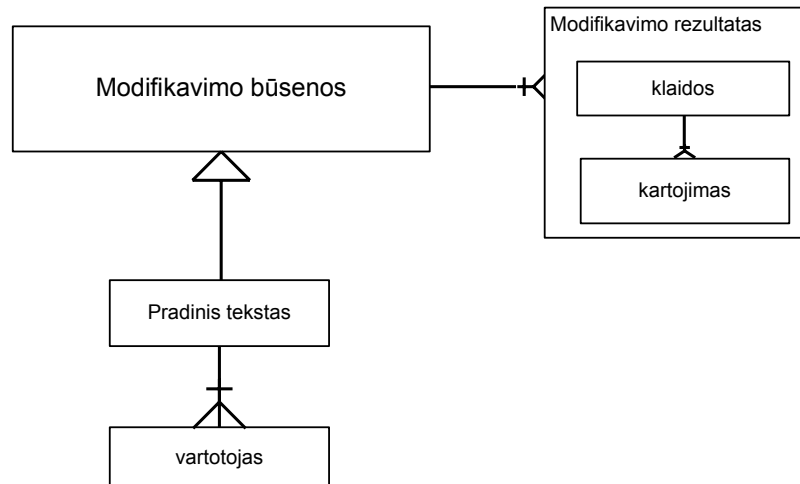
<b>Tinkamumo kriterijus:</b>	Įvykdžius operacija, duomenys išsaugomi į tekstinį failą		
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>	Algoritmų teorija: stipriai susijusios orientuoto grafo komponentės.		
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	7	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4
<b>Aprašymas:</b>	Sistemoje turi būti numatyto operacijų tikrinimo žingsnių kontrolė			
<b>Pagrindimas:</b>	Tikrinama kiekviena vartotojo operacija, kiekviename jo žingsnyje			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Vartotojas atlikęs operaciją ir jei ji yra negalima kontrolės modulis aptinka tai ir informuoja apie klaidą vartotoją			
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	3	
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra	
<b>Papildoma medžiaga:</b>				
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.			

<b>Reikalavimas #:</b>	8	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>	1,2,3,4
<b>Aprašymas:</b>	Sistemoje turi būti numatyta vartotojo informavimo procedūra			
<b>Pagrindimas:</b>	Kiekviena vartotojo operacija užtikrina grįžtamąjį ryšį			
<b>Šaltinis:</b>	Vartotojas.			
<b>Tinkamumo kriterijus:</b>	Sistema išveda informacinius langus apie vartotojo atliktas operacijas			
<b>Užsakovo patenkinimas:</b>	3	<b>Užsakovo nepatenkinimas:</b>	5	
<b>Priklausomybės:</b>	Vartotojo aprašytas invariantas.	<b>Konfliktai:</b>	Nėra	

### ***Reikalavimai duomenims***

Pateikiama pradinis duomenų modelis



11 pav. Duomenų modelis

### **3.1.3 Nefunkciniai reikalavimai**

#### ***Reikalavimai sistemos išvaizdai***

- Programa turi grafinį interfeisą;
- Interaktyvus grafinis interfeisas;
- Profesionali išvaizda (piktogramos, ikonos);
- Paprastumas.

#### ***Reikalavimai panaudojamumui***

Reikalavimų panaudojamumui sąrašas:

R1 Programa turi būti paprasta naudotis eiliniam vartotojui

R2 Programose vykdymo metu vartotojas turėtų galimybę išvengti klaidų

R3 Programoje naudojamas paprastas iškrentantis meniu

R4 Programoje meniu elementai turi turėti komentarus

R5 Programos sąsaja turi būti lietuvių kalba

<b>Reikalavimas #:</b>	1	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programa turi būti paprasta naudotis eiliniam vartotojui		
<b>Pagrindimas:</b>	Vartotojas galės naudotis sistema pagal poreikius esant tam tikromis aplinkybėms turės kasdien, tai susiję su darbo specifika		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Žinodamas apie tai kaip kitiems vartotojams sekasi naudotis programa, vartotojas turi galimybę išsiaiškinti kaip naudotis programa be specialaus apmokymo		
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	2	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programose vykdymo metu vartotojas turėtų galimybę išvengti klaidų		
<b>Pagrindimas:</b>	Programa turi veikti teisingai, be klaidų		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Klaidos turi būti programiškai gaudamos, kad ateityje nepasikartotų		
<b>Užsakovo patenkinimas:</b>	3	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	3	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programoje naudojamas paprastas iškrentantis meniu		
<b>Pagrindimas:</b>	Supaprastina navigaciją, sumažina klaidų tikimybę		

<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Programos vartotojas turi intuityviai suvokti sąsajos elementų reikalingumą, kad galėtų naudotis programa efektyviai		
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	2
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	4	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programoje meniu elementai turi turėti komentarus		
<b>Pagrindimas:</b>	Kartais reikalingas paaiškinimas ką būtent atlieka vienas arba kitas interfeiso elementas		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Turint komentarus padidėja darbo atlikimo greitas, klaidų tikimybė mažėja		
<b>Užsakovo patenkinimas:</b>	3	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	5	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programos sąsaja turi būti lietuvių kalba		
<b>Pagrindimas:</b>	Interfeiso elementų paskirtis turi būti aiški vartotojui		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Vartotojas gali susigaudyti, kaip naudotis esama programa		

<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	1
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

*Reikalavimai vykdymo charakteristikoms*

<b>Reikalavimas #:</b>	1	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Programos darbo efektyvumas		
<b>Pagrindimas:</b>	Sistema turi efektyviai išnaudoti turimus resursus, nekenkiant kitoms gretimai instaliuotoms sistemoms, netrikdant jų veiklos.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Programa netrikdo kitų programų darbo. Leidžia vartotojui lygiagrečiai dirbti su keliomis programomis.		
<b>Užsakovo patenkinimas:</b>	4	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	2	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Integracija		
<b>Pagrindimas:</b>	Sistema turi taisyklingai dirbti su kitomis programomis		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Programa veikti teisingai		
<b>Užsakovo patenkinimas:</b>	3	<b>Užsakovo nepatenkinimas:</b>	4
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			

**medžiaga:**  
**Istorija:** Užregistruotas 2005m. Spalis 16 d.

<b>Reikalavimas #:</b>	3	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Našumas		
<b>Pagrindimas:</b>	Programos operacijos turi būti greitai vykdomos		
<b>Šaltinis:</b>	Vartotojas.		
<b>Tinkamumo kriterijus:</b>	Operacijų vykdymo greitis turi būti proporcingas paduotam duomenų kiekiui.		
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	5
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

<b>Reikalavimas #:</b>	5	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Korektiškumas.		
<b>Pagrindimas:</b>	Sistemoje turi būti kuo mažiau klaidų.		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Klaidų skaičius yra minimalus ir netrikdo vartotojo darbo.		
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	5
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

### *Reikalavimai veikimo sąlygoms*

Programa turi veikti bet kuriame kompiuteryje, kuriame yra virtuali Java mašina.

### ***Reikalavimai sistemos priežiūrai***

Programa reikia dokumentuoti, kad jeigu ja norėtų tobulinti kitas specialistas galėtų susipažinti su jos specifikacija, tuo būdu papildant ja naujomis funkcijomis ir tuo būdu atnaujinant ją.

<b>Reikalavimas #:</b>	2	<b>Reikalavimo tipas:</b>	<b>Įvykis/panaudojimo atvejis #:</b>
<b>Aprašymas:</b>	Dokumentavimo sistema.		
<b>Pagrindimas:</b>	Sukurti tokią programos dokumentaciją, kuri padėtų bet kokiam neparuoštam vartotojui ir programuotojui		
<b>Šaltinis:</b>	Vartotojas		
<b>Tinkamumo kriterijus:</b>	Savalaikiška dokumentacija.		
<b>Užsakovo patenkinimas:</b>	5	<b>Užsakovo nepatenkinimas:</b>	2
<b>Priklausomybės:</b>	Nėra	<b>Konfliktai:</b>	Nėra
<b>Papildoma medžiaga:</b>			
<b>Istorija:</b>	Užregistruotas 2005m. Spalis 16 d.		

### ***Reikalavimai saugumui***

Mano sukurta programa turi tenkinti tokius saugumo reikalavimus:

- 1) sistemoje turi būti licenzinis raktas, tuo būdu išvengiant atvejų, kai programine įranga naudojasi.

## **3.1.4 Projekto išeiga**

### ***Egzistuojantys sprendimai***

Jau egzistuojančių produktų sprendimai nebus taikomi mano kuriamoje programoje, programinis kodas bus rašomas savarankiškai.

### ***Neigiamas vartotojų nusiteikimas***

Prognozuojamos galimos vartotojo neigiamo nusiteikimo priežastys:

- Vartotojas gali nesuprasti kaip tokia programa gali padėti jo darbe;
- Vartotojo nenoras mokytis elgtis su nauja programa.

### ***Kliudantys diegimo aplinkos apribojimai***

Vartotojo kompiuteryje būtina turi būti įdiegta JVM

### ***Sistemos pateikimo žingsniai (etapai)***

1. Reikalavimų nustatymas:

- poreikių surinkimas;
- poreikių specifikavimas.

2. Analizė:

- reikalavimų sistemai specifikavimas.

3. Projektavimas:

- detalus architektūros specifikavimas.

4. Realizavimas:

- funkcinių modulių realizacija;
- veiksmų, formuojančių rezultatus, diagramos modulis

5. Testavimas:

- modulių testavimas;
- bendras sistemos testavimas.

6. Diegimas ir eksploatavimas.

Daugelis šių žingsnių atitinka krioklio modulį, kai vykdomi yra projektai.

### ***Vystymo etapai***

Tam, kad pradėti rašyti programa būtina:

- a) Parengti darbų planą;
- b) Parengti reikalavimų specifikaciją;
- c) Architektūros specifikaciją;
- d) Pratestuoti programą;
- e) Sukurti dokumentaciją;
- f) Parengti programos palaikymo priemones.



## Galimos sistemos kūrimo rizikos

4 lentelė „Sistemos kūrimo rizikos faktoriai“

<i>Rizikos faktorius</i>	<i>Tikimybinis įvertinimas</i>
Reikalavimų specifikacijos pasikeitimas kodo rašymo stadijoje	<b>9</b>
Užsakovų uždavinių pasikeitimai nesuderinti su darbo grafiku	<b>10</b>
Projekto personalo žmogiškųjų išteklių pokyčiai	<b>10</b>
Darbo terminų mažinimas	<b>7</b>

## Atsitiktinumų (rizikų) planas

Rizikos faktoriai ir numatomi planai problemoms spręsti.

5 lentelė „Atsitiktinumų plano faktoriai“

<i>Rizikos faktorius</i>	<i>Problemos sprendimas</i>
Projekte dalyvaujančio personalo patirtis ir sugebėjimai	Programinę įrangą kurti tokiomis priemonėmis, su kuriomis patirtį turėtų didžioji dalis projekto personalo
Programinio kodo naudojimas pirmą kartą	Reikia galutinai ištestuoti programinį kodą, procedūras, išbandyti įvairias situacijas kurioms esant būtų galima rasti klaidas
Reikalavimų pasikeitimas iš užsakovo pusės	Kuo dažniau derinti projekto eigą su užsakovu, kad būtų sutaupytas laikas ir lėšos

## Vartotojo dokumentacija ir apmokymas

Vartotojo dokumentacijoje turėtų būti aprašyti tokie punktai:

- Vartotojui būtina pradinė informacija apie interfeisą ir programos funkcionalumą;
- Informacija apie klaidas;
- Pasiūlymai kaip teisingai elgtis su programa.

## Perspektyviniai reikalavimai

- Konfidencialumo reikalavimas (jis atsiskleis per raktažodžio įvedimą, kuris suteikia teisę naudotis programa tik tiems asmenims, kurie jį naudoja);

## 3.2 Programinės įrangos detali architektūra

### 3.2.1 Dokumento paskirtis

Projektavimo etape parenkama sistemos architektūra ir sudaromas projekto klasių modelis, taip pat yra apibrėžiami paketai. Parenkama sistemos architektūra, taikomi tipiniai architektūros šablonai. Šio dokumento tikslas yra surinkti ir pateikti svarbius architektūrinius sprendimus, kuriuos galima atlikti kuriamoje sistemoje. Šis dokumentas padeda sistemos realizuotojams geriau suprasti architektūrinius sprendimus, kuriuos padarė programinės įrangos architektas. Be to, dalis šiame dokumente esančių diagramų gali padėti generuojant programos kodą naudojantis CASE priemonėmis.

6 lentelė „Galimi vartotojai“

<b>Galimi dokumento vartotojai</b>	<b>Dokumento panaudojimas</b>
Užsakovas	Detaliau susipažindinamas su kuriamu produktu
Sistemos projektuotojas	Sukuria pradinį sistemos architektūros karkasą
Sistemos analitikas	Aptinka sistemos architektūros trūkumus.
Sistemos programuotojas (realizuotojas)	Galimybė generuoti dalį kodo iš pateiktų diagramų.

### 3.2.2 Apibrėžimai ir sutrumpinimai

- Semantinio teksto transformavimo ir taikymo kompiuterinio vertimo sistemoje sistema – pagrindinė schematiškai apibrėžta semantinio teksto apdorojimo priemonė naudojanti apibrėžtas taisykles tam, kad įvestas anglų kalbos tekstas būtų taisyklingai išverstas.
- JVM – Java virtuali mašina, sluoksnis tarp programos ir operacinės sistemos leidžiantis tą pačią programą vykdyti įvairiose operacinėse sistemose (Java Virtual Machine)
- Specifikacija - sistemos funkcionalumo aprašymas formaliais metodais.
- PA - panaudojimo atvejai (Use Case)
- UML – unifikuota modeliavimo kalba (Unified Modeling Language)

### **3.2.3 Architektūros apžvalga**

Šis dokumentas aprašo semantinio teksto transformavimo posistemio programinės įrangos architektūrą. Sistemos nefunkciniai reikalavimai ir apribojimai pateikiami skyriuje „Architektūros tikslai ir apribojimai“. O panaudojimo atvejai pateikiami skyriuje „Panaudojimo atvejų vaizdas“. Sistemos išskaidymas į paketus ir statinė struktūra pateikta skyriuje „Loginis vaizdas“. Sistemos procesai ir jų aprašymai pateikiami skyriuje „Procesų vaizdas“. Sistemos išdėstymas, ir techninė įranga, kurioje bus realizuota sistema, pateikiama skyriuje „Išdėstymo vaizdas“. Skyriuje „Duomenų vaizdas“ pateikiamos duomenų struktūros. Skyriuje „Kokybė“ aprašoma, kaip architektūra įtakoja sistemos išplečiamumą, pernešamumą, patikimumą ir pan.

### **3.2.4 Architektūros pateikimas**

Sistemos architektūra čia pateikiama keliais aspektais:

- Panaudojimo atvejų vaizdu (panaudojimo atvejų diagrama)
- Loginis programos vaizdas (Klasių diagrama, skaidymas į paketus)
- Procesų vaizdas (Būsenų kaitų diagramos, sekų diagramos, bendradarbiavimo diagramos)
- Išdėstymo vaizdas (išdėstymo diagrama)

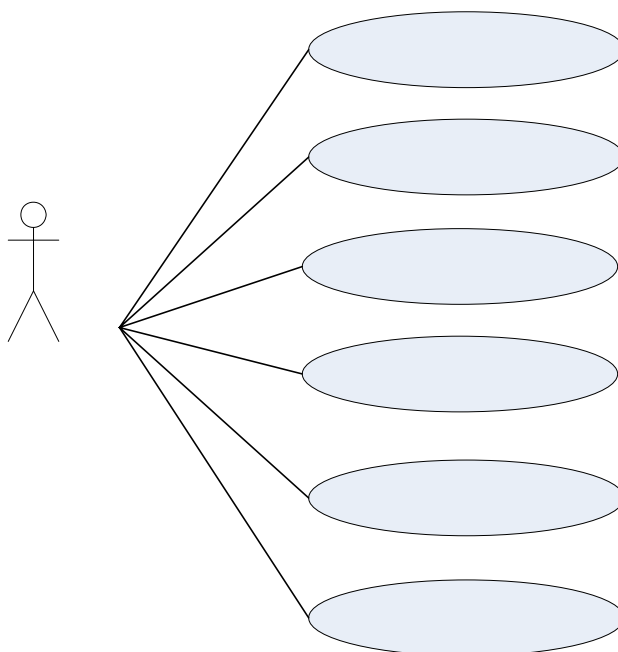
Visos reikalingos diagramos bus vaizduojamos UML notacija.

### **3.2.5 Architektūros tikslai ir apribojimai**

- Programinė įranga yra nekomercinė, kol nėra galutinai išstobulinta ji nėra apmokestinama, neturi komercinio naudingumo, bet ateityje jei ši programinė įranga efektyviai bus taikoma įmanomas jos apmokestinimas.
- Tam, kad programinė įranga nebūtų priklausoma nuo vienos arba kitos platformos, ji kuriama Java kalba, o java virtualias mašinas gali turėti daugelis populiarių operacinių sistemų.

### **3.2.6 Panaudos atvejų vaizdas**

Programinės įrangos panaudojimo atvejų diagrama pateikta paveikslėlyje 1:



12 pav. Panaudos atvejai

### Panaudos atvejis 1

Lentelė Nr. 7 Panaudos atvejis „Teksto paruošimas apdorojimui“

Nr.	1	<b>Vartotojas</b>
Pavadinimas:	Teksto paruošimas apdorojimui	
Vartotojo/aktoriaus pavadinimas:	Vartotojas	
Aprašas:	Randame kietajame diske arba kitoje laikmenoje anglų kalbos tekstą ir iškrauname jį į programos formą	
Prieš sąlyga:	Reikia pasirinkti vietą, kurioje yra failas su tekstine informacija	
Sužadinimo sąlyga:	Nurodomas kelias iki pasirenkamo failo	
Po sąlyga:	Pasirinktas anglų kalbos tekstas yra atvaizduojamas programos interfeiso formoje	

### Panaudos atvejis 2

Lentelė Nr. 8 Panaudos atvejis „Teksto transformavimas pagal pasikartojimus“

Nr.	2	
Pavadinimas:	Teksto transformavimas pagal pasikartojimus	
Vartotojo/aktoriaus pavadinimas:	Vartotojas	
Aprašas:	Tekste ieškoma žodžių, žodžių junginių pasikartojimų, kurie yra pašalinami	
Prieš sąlyga:	Tekste yra pasikartojimai, kurių dalį reikia pašalinti	
Sužadinimo sąlyga:	Vartotojas nori pašalinti pasikartojimus tekste, todėl pasirenka tokį interfeiso mygtukas, kuris sužadina šia operaciją	
Po sąlyga:	Iš teksto yra pašalinami pasikartojimai	

### Panaudos atvejis 3

Lentelė Nr. 9 Panaudos atvejis „Teksto transformavimas pagal sakinio struktūrą“

Nr.	3
Pavadinimas:	Teksto transformavimas pagal sakinio struktūrą
Vartotojo/aktoriaus pavadinimas:	Vartotojas
Aprašas:	Tekstas transformuojamas taip, kad jeigu sakinyje yra veiksmažodis tai potencialiai turi būti sakinyje ir daiktavardis, kuris „neša“ tam tikrą informaciją. Taip pat analizuojamos žodžio dalys pagal tam tikrą atitikimą.
Prieš sąlyga:	Pradinėje stadijoje sakinio struktūra yra visiškai neapdorota
Sužadinimo sąlyga:	Vartotojas apdoroti sakinio struktūrą, rasti tam tikrus dėsningumus
Po sąlyga:	Įsimenama informacija apie žodžių dalis

### Panaudos atvejis 4

Lentelė Nr. 10 Panaudos atvejis „Sakinio skaidymas“

Nr.	4
Pavadinimas:	Sakinio skaidymas
Vartotojo/aktoriaus pavadinimas:	Vartotojas
Aprašas:	Kartais kai sakinyje yra didelis – jo vertimas yra sunkus ir problematinis, todėl reikalingas sakinio paprastinimas, kuris įmanomas per sakinio skaidymą
Prieš sąlyga:	Sakinyje nėra suskaidytas į paprastesnius ir trumpesnius sakinius
Sužadinimo sąlyga:	Vartotojas nori pamatyti trumpus sakinius, kuriuos jis galėtų aiškiau suprasti
Po sąlyga:	Sakinyje yra suskaidomas į trumpesnius sakinius.

### Panaudos atvejis 5

Lentelė Nr. 11 Panaudos atvejis „Teksto transformavimas pagal kalbos dalis, laikus“

Nr.	5
Pavadinimas:	Teksto transformavimas pagal kalbos dalis, laikus
Vartotojo/aktoriaus pavadinimas:	Vartotojas
Aprašas:	Tekstas gali būti transformuotas ir išanalizuotas pagal kalbos dalis, t.y. linksnius, taip pat analizuojami yra ir veiksmažodžių laikai
Prieš sąlyga:	Pradinis tekstas nėra apdorotas įvertinus žodžio dalis, laiką
Sužadinimo sąlyga:	Pasirenkama atitinkama komanda operacijai atlikti
Po sąlyga:	Įsimenama informacija apie veiksmažodžio laiką, kita informacija apie žodžio dalis

### Panaudos atvejis 6

Lentelė Nr. 12 Panaudos atvejis „Galutinis rezultato išsaugojimas“

Nr.	6
Pavadinimas:	Galutinis rezultato išsaugojimas

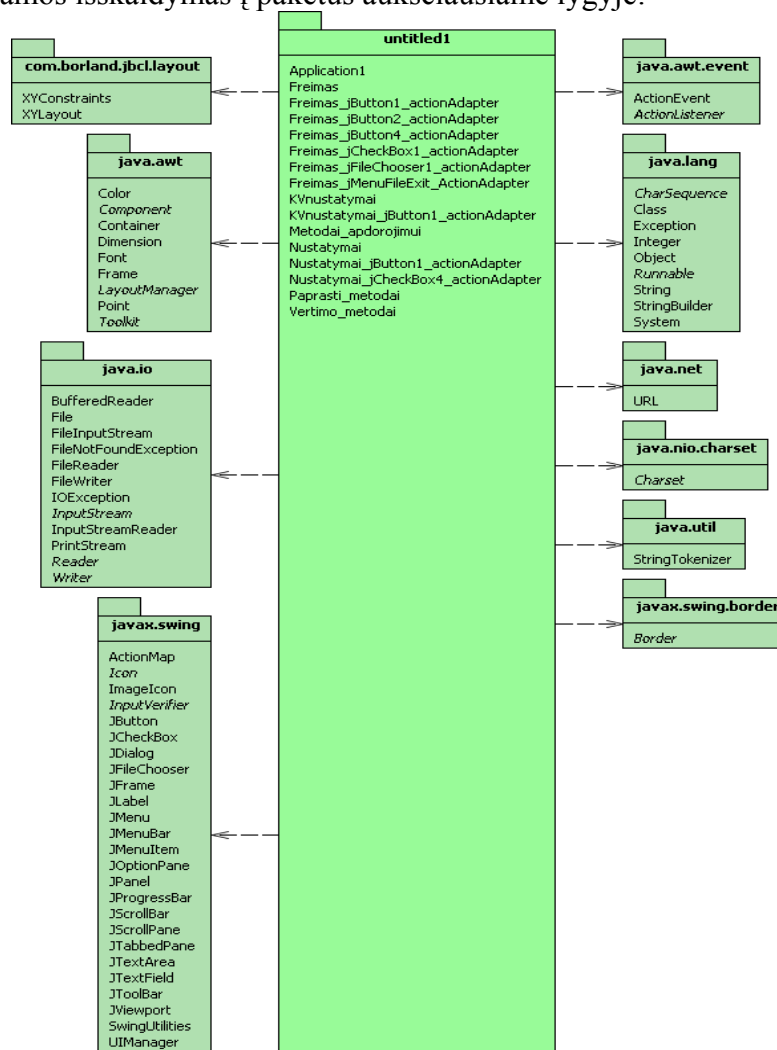
Vartotojo/aktoriaus pavadinimas:	Vartotojas
Aprašas:	Informacija apie sakinių kalbos dalis, laikus yra išsaugojami į atskirą failą, kuris yra panaudojamas tada, kai vyksta teksto transformavimas
Prieš sąlyga:	Informacija apie teksto transformacijas yra paruošta, bet neišsaugota
Sužadinimo sąlyga:	Pasirenkama komanda išsaugojimas
Po sąlyga:	Sukuriamas tekstinis failas su teksto struktūros informacija

### 3.2.7 Sistemos statinis vaizdas

Šis skyrius aprašo sistemos loginę struktūrą, pateikia sistemos išskaidymą į paketus ir juos sudarančias klases.

#### *Sistemos klasių apžvalga*

Kuriamos programos išskaidymas į paketus aukščiausiame lygyje:



13 pav. Sistemos išskaidymas į paketus ir klases

Mano kuriama programinė įranga turės tris pagrindinius paketus:

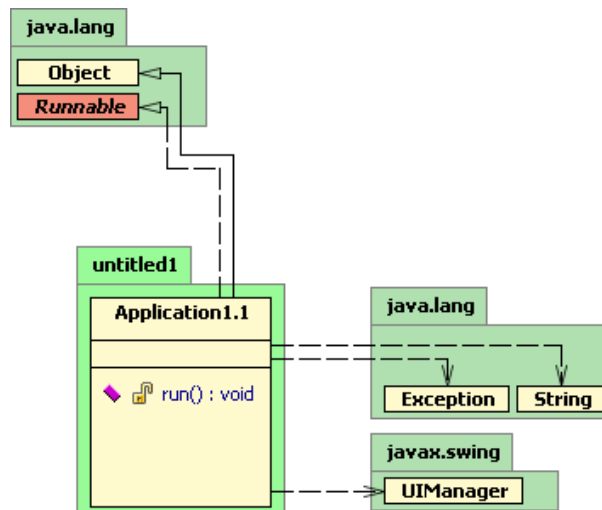
- 1) Paketas „untitled1“, tai toks paketas kuriame yra aprašomos visos teksto paruošimo funkcijos, tai tokios funkcijos, kaip skaitymas iš failo, išsaugojimas į failą;
- 2) Klasė paprasti metodai realizuoja failo skaitymą ir išsaugojimą;
- 3) Klasė metodai apdorojimui realizuoja teksto transformavimą, kur kiekviena šios klasės procedūra atlieka tam tikru būdu semantinę teksto transformaciją;
- 4) Kitos klasės “atsakingos” už vizualinės formos realizacija.

Kiti standartiniai JAVA paketai realizuoja visus kitus bazinius JAVA kalbos metodus.

### ***Paketas „Untitled1“***

Paketą „untitled“ sudaro 7 klasės, klasė “vertimo metodai” nedetalizuojama nes nėra panaudojama.

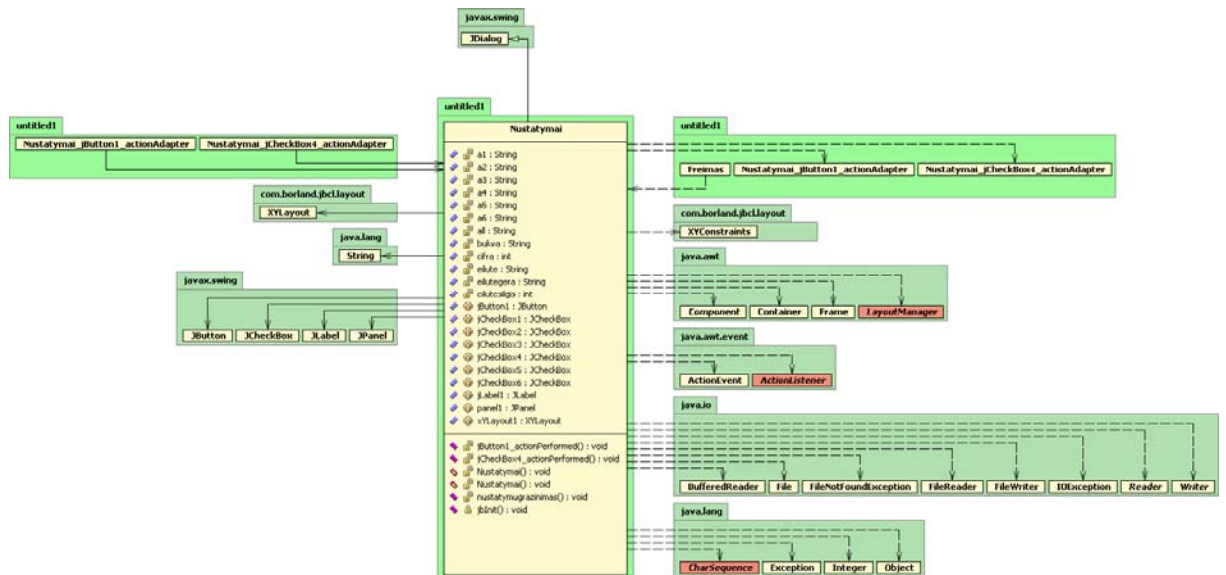
#### *Klasė Application1*



14 pav. Application1 klasės struktūra

#### *Klasė Nustatymai*

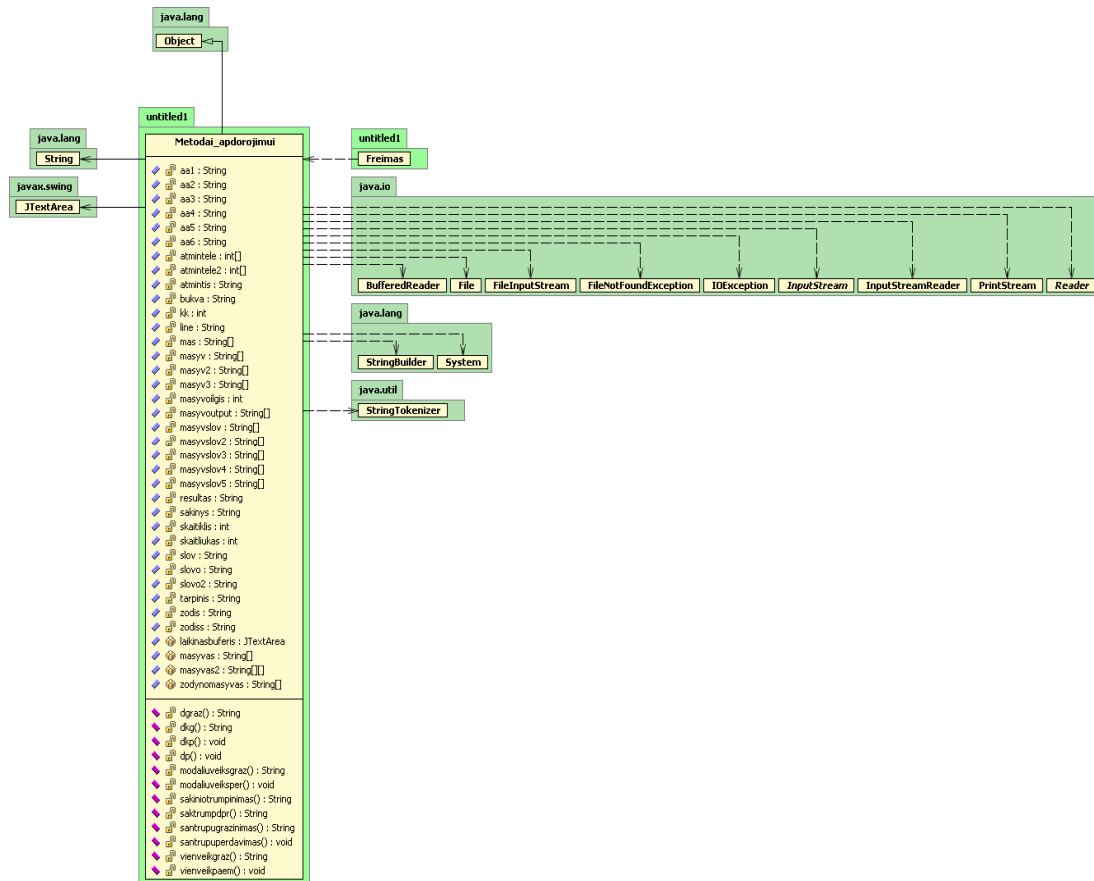
Klasė „Nustatymai“ išsaugo pasirinktus programoje nustatymus.



15 pav. Nustatymai klasės struktūra

### Klasė Metodaiapdorojimui

Klasėje „Metodaiapdorojimui“ apibrėžtos pagrindinės teksto apdorojimo procedūros ir kintamieji.

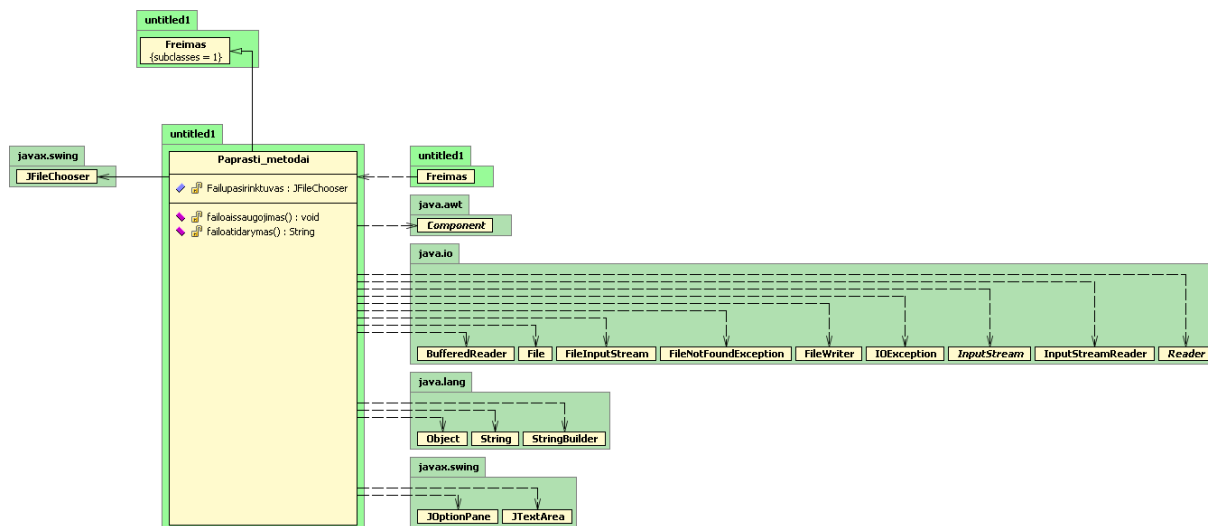


16 pav. Metodaiapdorojimui klasės struktūra

### Klasė Paprastimetodai

Klasė, kuri realizuoja failo skaitymą ir programos tekstinį objektą ir rašymą į tekstinį failą.





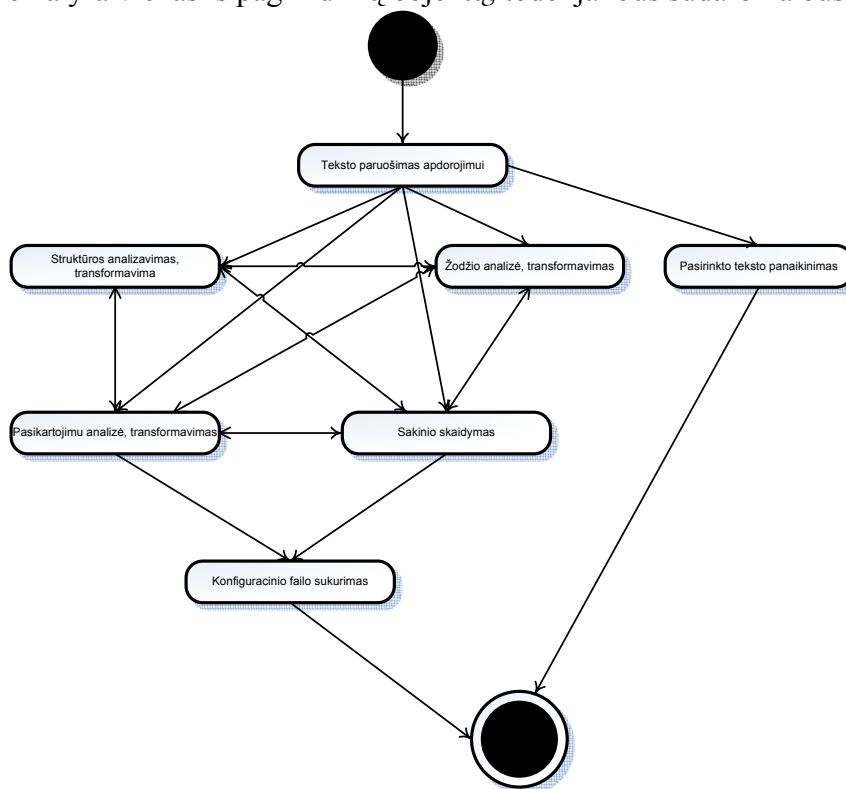
17 pav. Paprastimetodai klasės struktūra

### Procesų vaizdas

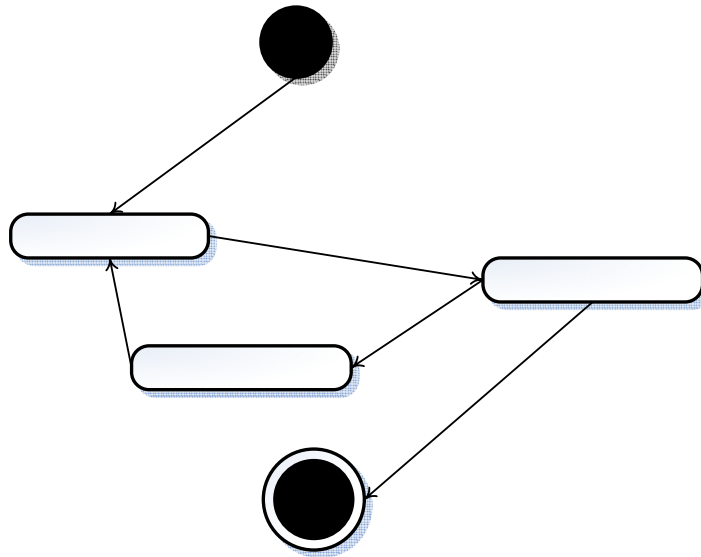
Šiame skyrelyje galima rasti kuriamos sistemos objektų būsenų diagramas, sistemos elementų bendradarbiavimo ir sekų diagramas.

### Būsenų diagramos

Operacijų schema yra vienas iš pagrindinių objektų, todėl jai bus sudaroma būsenų diagrama:



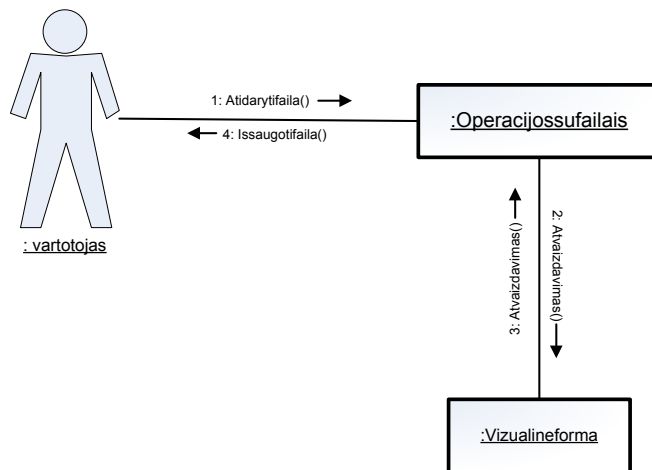
18 pav. Esysbės "Operacijos" būsenų diagrama



19 pav. Esybės "Failai" būsenų diagrama

Atidaryti failą

***Bendradarbiavimo diagramos***



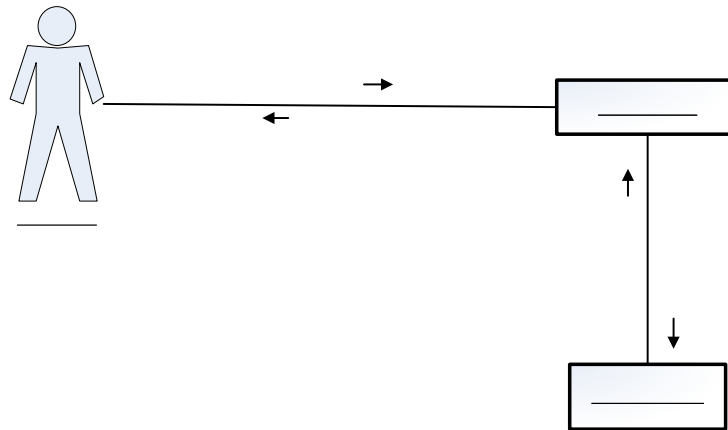
Atidaryto failo atšauk

20 pav. Atidaryti teksto paruošimo schemą

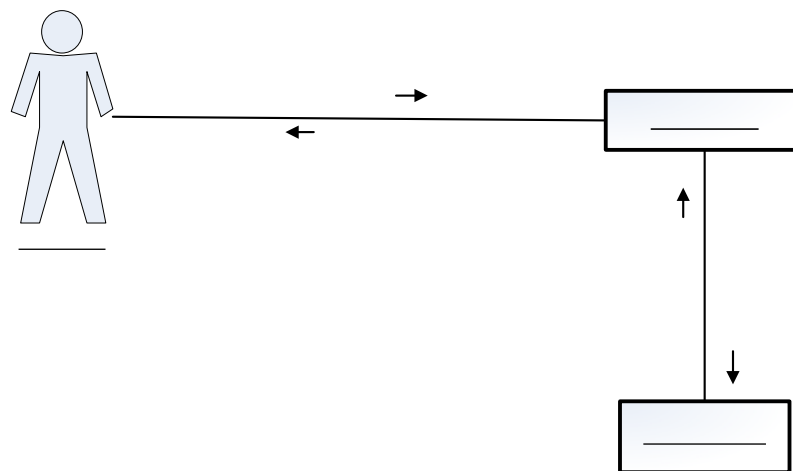
Pažymėtina, kad tokios funkcijos, kaip Atvaizdavimas() nėra, šis teiginys galioja visoms bendradarbiavimo diagramoms, aš ja įvedžiau, kad būtų aiškiau suvokiama informacijos išvedimas į interfeisinę formą, kuri užtikrina java kalbos specifinės klasės, metodai.



21 pav. Teksto transformavimas pagal pasikartojimus schema



22 pav. Sakinio struktūros skaidymas pagal sakinio struktūros schema

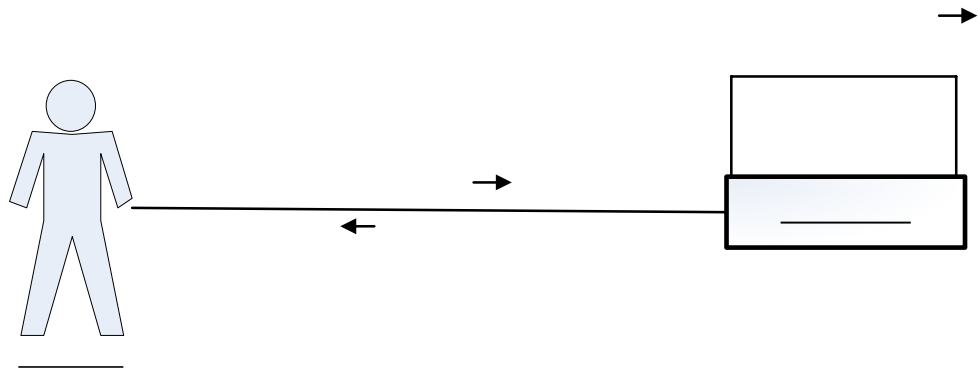


23 pav. Teksto transformavimas pagal sakinio kalbos ir laiko dalis struktūros schema

: vartotojas

1 Atid

5



24 pav. Galutinis rezultato išsaugojimas

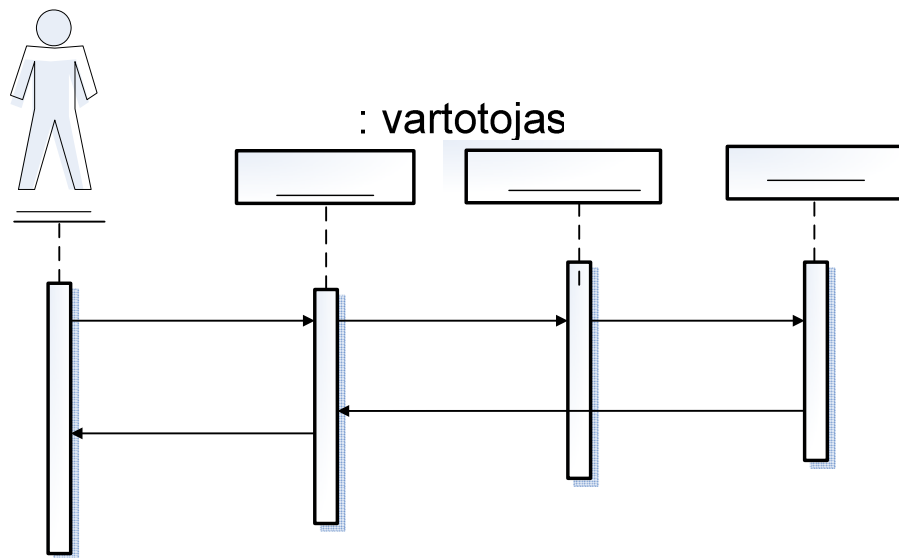
Šią schemą būtų galima suprasti taip:

- 1) Įvykus pasirinktai operacijai, jos rezultatas vizualiai gali būti matoma grafinio interfeiso objekte, taip pat galimas transformuoto teksto išsaugomas į failą.

1: Atidaryti

3: issa

### Sekų diagramos



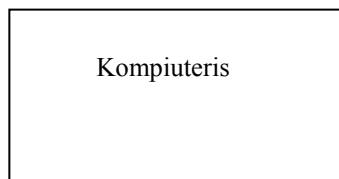
25 pav. Semantinio teksto apdorojimo paruošimo schema

Po N operacija() aš turiu omenyje tas operacijas, kurios yra klasėje „metodai\_apdorojimui“:

- pasikartojimufiksavimas()
- strukturosanalizavimas(){
- sakinoskaidymas()
- kalbosdaliesanalize()
- veiksmožodžiomodalumoanalize()

### *Išdėstymo vaizdas*

Sistemos išdėstymo diagrama:



26 pav. Fizinė sistemos išdėstymo diagrama

Kuriama programinė įranga tiesiog yra įdiegiama vartotojo kompiuteryje. Visi reikalingi duomenys saugojami kompiuterio kietajame diske arba kitoje laikmenoje.

Minimalūs reikalavimai vartotojo darbo vietai (kompiuteriui):

- Procesorius (CPU): 600 MHz
- Operatyvioji atmintis (RAM): 128 Mb
- Operacinė sistema: Windows 9x/200/NT/XP/Linux distribucijos

### 3.2.8 Duomenų vaizdas

Mano darbe duomenų bazė nenaudojama, visi duomenis kuriais disponuoja klasės, bei jų tarpusavio ryšiai parodyti 27 pav.



27 pav. Duomenų struktūrų klasių diagrama

- Pradinė tekstinė informacija – programos *textarea* objekte esanti tekstinė informacija;
- Informacija apie žodžio kalbos dalis saugoma žodyno faile, txt formatu;
- Teksto vienetas- *žodis*.

***Kokybė***

Produktas nereikalauja didelių sisteminių resursų.

## 4 Vartotojo dokumentacija

### *Funkcinis sistemos aprašymas:*

Sistema skirta semantiniam anglų kalbos teksto transformavimui. Ji leidžia supaprastinti anglų kalbos tekstines struktūras, paruošiant jas automatinio vertimo sistemoms.

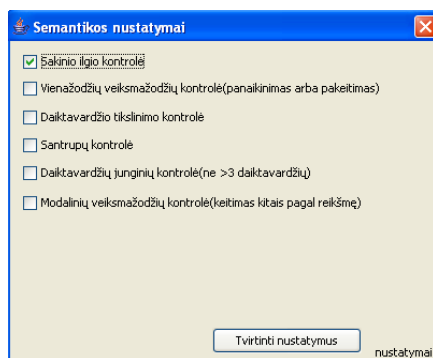
Pagrindinės sistemos galimybės:

- Interaktyvumas
- Semantinio teksto transformavimo procedūrų pasirinkimo galimybė dialogo lange
- Paprastas valdymas.
- Duomenų išsaugojimas į txt formatą

### *Kaip naudotis programa:*

- Meniu.
  - **Failas** – pagrindiniame meniu punkte galima atidaryti, išsaugoti, tekstinę informaciją.
    - **Atidaryti** – atidaro tekstinį failą.
    - **Išsaugoti** – išsaugo tekstinį failą.
    - **Uždaryti** – programos darbo pabaiga.
  - **Nustatymai** – leidžia vartotojui pačiam pasirinkti kokias teksto semantinio transformavimo procedūras jam naudoti.
    - **Semantiniai**– semantinių procedūrų nustatymų dialogo langas.
    - **KV\_nustatymai**– vartotojas jų nenaudoja, bet jie vizualiai matomi, šie nustatymai palikti tobulinimui jeigu šis projektas plėsis.
  - **Informacija**– vartotojo pagalba.
    - **Pagalba** – paleidžia pagalbos posistemę.
    - **Apie** – iššaukia „Apie“ dialogą.
- Dialogai (Nustatymai dialogorealizacija)

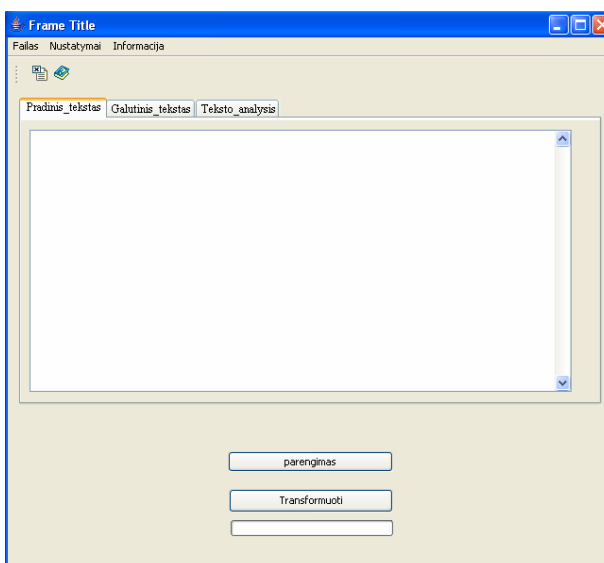




28 pav. Nustatymu dialogo langas

Šiame dialoge vartotojas pats pasirenka kokias teksto semantikos transformavimo procedūras jam naudoti.

*Vizualinis programos interfeisas*

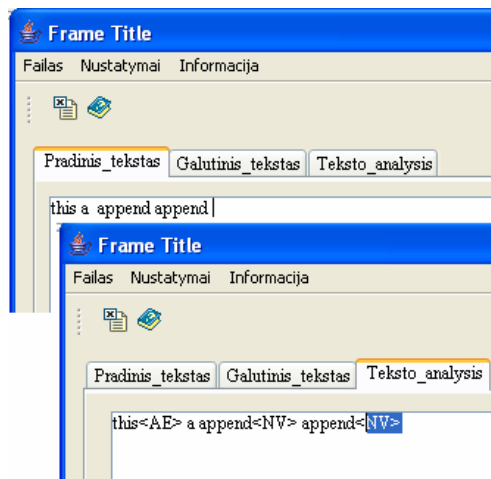


29 pav. Programos dialogo langas

Mygtukas „parengimas“ reikalingas tam, kada būtų paleista anglų dalių nustatymo procedūra, kurios rezultate vartotojas mato kalbos dalis + kokias tai kalbos dalis (užkuoduotoje formoje pagal 1 ają raidę)

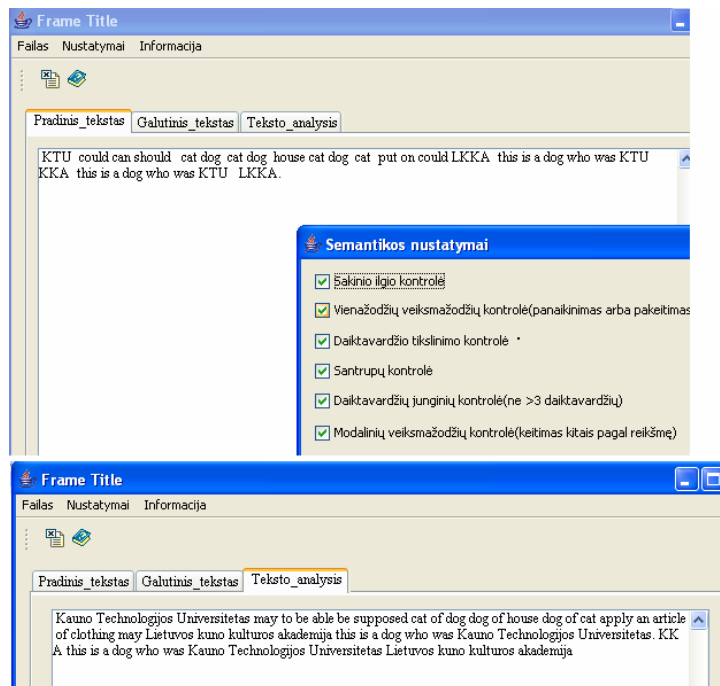
*Priemonės praktinis panaudojimo pavyzdys :*

Kai tekstas yra pasirinkta ir nustatytas paspaudus mygtuką „Parengimas“ mes matome tokį rezultatą, šiuo atveju tekstas tarp skliaustų „< ir >“ rodo, kad žodis turi daugiau nei vieną prasmę ir gali priklausyti skirtingoms kalbos dalim.



30 pav. Programos veikimo dialogų langas

Kitas panaudojimo pvz. yra kai įjungtos visos programos semantinio transformavimo procedūros. Šiuo atveju mes matome tai, kas pavaizduota 31 pav.



31 pav. Praktinio panaudojimo pavyzdys

### ***Žodyno sukūrimas***

Buvo sukurtas žodynas į kuri įėjo anglų kalbos žodžiai. Šie žodžiai buvo patalpinti tokia tvarka: žodis=Kalbos dalis, priklausomai nuo to kokia kalbos daliai gali priklausyti žodis.

*Praktinis žodyno demonstravimas*

happened=V=

state=N=

building=N=

Building=N=

not=J=

see=V=

append=Nv=

(kur N, J reiškia priklausimą kalbos daliai)

*Kalbos dalių paaiškinimas:*

D-daiktavardis(cowboy, theatre, box, thought, tree, kindness, arrival )

I-įvardis(I, you, he, she, it, we, they )

B-būdvardis(big, yellow, thin, amazing, beautiful, quick, important )

P-prielinksniai(pvz, in,on,from,into,off)

A-artikeliai(a,the)

J-jungtukai(but, so, and, because, or )

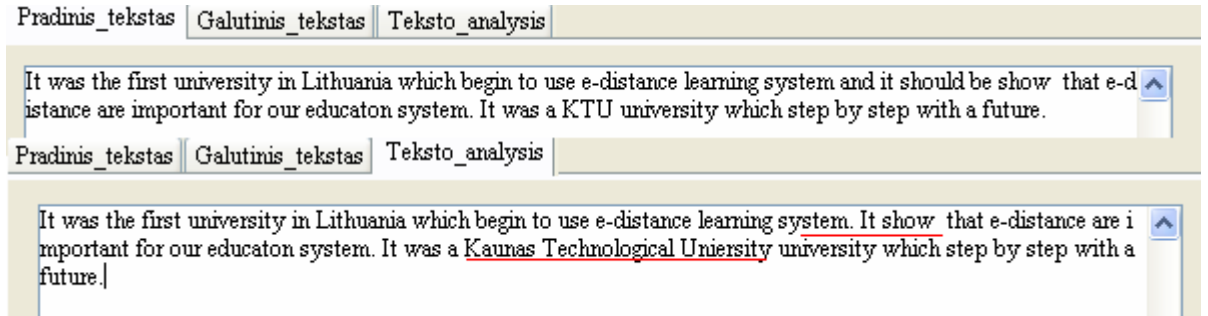
C-adverb(slowly, intelligently, well, yesterday, tomorrow, here, everywhere )

v-veiksmažodis (walk, talk, think, believe, live, like, want )

## 5 Produkto testavimas

Šiame skyriuje bus pratestuotos anglų kalbos teksto semantikos transformavimo procedūros realizuotos programoje.

Kaip matome 32 pav. iš pradžių mes turime pradinį tekstą, kuris paruošiamas transformavimui, bet taip pat matome tekstą po transformavimo.



32 pav. "Testavimas"

Tekstas po transformavimo 32 pav. rodo, kad sakinyje buvo įterptas taškas, pašalintas modalinis veiksmažodis, santrumpa pakeista pilnu universiteto pavadinimu. Detalesnius testavimo rezultatus galite peržiūrėti priede A.

## 6 Išvados

1. Atlikta teksto semantinio transformavimo galimybių analizė. Tai leido apibrėžti anglų kalbos teksto pagalbinės semantinio transformavimo priemonės sukūrimo principus ir technologinius jos įgyvendinimo būdus.
2. Atlikus anglų kalbos teksto semantinio apdorojimo priemonių apžvalgą pavyko nustatyti pagrindines kryptis, kuriomis grindžiamas teksto semantinis transformavimas. Buvo apibrėžta keletas pagrindinių anglų kalbos teksto semantikos apdorojimo būdų, kurie buvo panaudoti kuriant programą.
3. Rašant programą buvo sukurta septynios pagrindinės klasės. Vienoje iš šių klasių buvo aprašytos teksto semantinio transformavimo procedūros, kitoje klasėje buvo aprašytos skaitymo iš failo, rašymo į failą procedūros, kitos klasės buvo sugeneruotos automatiškai.
4. Programos kūrimo metu buvo sukurtas anglų kalbos semantinis žodynas, kurį sudaro byla txt formate. Žodyno struktūra turi tokį pavidalą - žodis, parametro kriterijus, kuris nurodo kokiai kalbos daliai priklauso žodis.

## 7 Literatūra

1. Teruko Mitamura. Controlled English For Knowledge-Based MT: Experience with Kant system. [žiūrėta 2005-05-20]. Prieiga per internetą: <http://citeseer.ist.psu.edu/mitamura95controlled.html>
2. Teruko Mitamura, Eric Nyberg. Multiple strategies for automatic disambiguation in technical translation [žiūrėta 2005-05-23]. Prieiga per internetą: <http://citeseer.ist.psu.edu/mitamura99multiple.html>
3. Teruko Mitamura, Katrin Backer. Diagnostics for interactive controlled language checking [žiūrėta 2005-05-21]. Prieiga per internetą: <http://citeseer.ist.psu.edu/580976.html>
4. Teruko Mitamura, Katrin Backer. Source language diagnostic for MT [žiūrėta 2005-05-23]. Prieiga per internetą: <http://citeseer.ist.psu.edu/692793.html>
5. Eric Nyberg, Teruko Mitamura. An integrated system for source language checking, analysis and term management [žiūrėta 2005-05-22]. Prieiga per internetą: <http://citeseer.ist.psu.edu/662992.html>
6. John Hutchins. Towards a new vision for MT. [žiūrėta 2005-05-23]. Prieiga per internetą: <http://ourworld.compuserve.com/homepages/WJHutchins/MTS-2001.pdf>
7. Anna Sagvaal. Uppsala chart parser light: improving efficiency in a chart parser. [žiūrėta 2005-05-23]. Prieiga per internetą: [http://stp.ling.uu.se/~perweij/papers/ucp3\\_weij\\_nitz\\_thesis.pdf](http://stp.ling.uu.se/~perweij/papers/ucp3_weij_nitz_thesis.pdf)
8. Nano Gough, Andy Way. Controlled generation in example-based machine translation [žiūrėta 2005-05-22]. Prieiga per internetą: <http://www.amtaweb.org/summit/MTSummit/FinalPapers/49-Gough-final.pdf>
9. Aarno Lehtola. Controlled language technology in multilingual user interfaces [žiūrėta 2005-05-20]. Prieiga per internetą: <http://ui4all.ics.forth.gr/UI4ALL-98/lehtola.pdf>
10. Diego Molla, Rolf Schiwitter. From plain english to controlled english [žiūrėta 2005-05-23]. Prieiga per internetą: <http://www.ics.mq.edu.au/~rolfs/papers/anlp2001.pdf>
11. Rolf Schiwitter, Marc Tolbrook. Controlled natural language meets the semantic web [žiūrėta 2005-05-22]. Prieiga per internetą: <http://www.ics.mq.edu.au/~rolfs/papers/alta04-schwiter-tilbrook.pdf>

12. Wladyslaw Janowski. Controlled language: Risk and side effects [žiūrēta 2005-05-24].  
Prieiga per internetą: [http://www.techwriter.de/tc-forum/pdf/editions/tc\\_f982s.pdf](http://www.techwriter.de/tc-forum/pdf/editions/tc_f982s.pdf)
13. Rolf Schiwitter, Norbert E. Fuchs. Attempto controlled language(ACE) for software  
specifikacijos .[žiūrēta 2005-05-23]. Prieiga per internetą:[http://www.ics.mq.edu.  
au/~rolfs/papers/Attempto\\_CLAW98.pdf](http://www.ics.mq.edu.au/~rolfs/papers/Attempto_CLAW98.pdf)
14. Stefan Hoefler. The syntax attempto controlled english: an abstract grammar for ace  
4.0 [žiūrēta 2005-05-23]. Prieiga per internetą: [http://www.ling.ed.ac.uk/~s04503  
13/hoefler/2004theSyntax.pdf](http://www.ling.ed.ac.uk/~s0450313/hoefler/2004theSyntax.pdf)
15. Uta Schwertel. Controlling plural ambiguities in attempto controlled english(ACE)  
[žiūrēta 2005-05-24]. Prieiga per internetą: [http://www.ifi.unizh.ch/groups/re  
q/ftp/papers/claw2000.pdf](http://www.ifi.unizh.ch/groups/research/ftp/papers/claw2000.pdf)

## **8 Summary**

### **Semantic text conversion and using it in computerized automatic translation systems**

Today Lithuania have a real need in having automatic translation system, which can simplificate a process of translation English language to Lithuanian language. But how we can realize this. First of all we must have a text semantic transformation system. It is a main purpose of this work – to create text semantic transformation system. Semantic transformation – process, which can help us in simplificating a sentence structure, but also to save a connections between different parts of sentence and the main means of sentence is not disappear. In my project I selected several trends(realized as functions) how we can transform a text. For example it can be a shorten sentence length or remove from sentence modal verbs, because in described rules this type of verb not need. And I will try to realized this my project.



## 9 Terminų ir santrupų žodynas

- *Semantinio teksto parseris* – specialus programinis modulis skirtas apdoroti tekstui, kuris vėliau turi būti verčiamas
- *Išeiiga* – apdorotas tekstas, kuris gali arba turi būti verčiamas į lietuvių kalbą.
- Semantinio teksto transformavimo ir taikymo kompiuterinio vertimo sistemoje sistema – pagrindinė schematiškai apibrėžta semantinio teksto apdorojimo priemonė naudojanti apibrėžtas taisykles tam, kad įvestas anglų kalbos tekstas būtų taisyklingai išverstas.
- JVM – Java virtuali mašina, sluoksnis tarp programos ir operacinės sistemos leidžiantis tą pačią programą vykdyti įvairiose operacinėse sistemose (Java Virtual Machine)
- Specifikacija - sistemos funkcionalumo aprašymas formaliais metodais.
- PA - panaudos atvejai (Use Case)
- UML – unifikuota modeliavimo kalba (Unified Modeling Language)

## 10 Priedai

### Priedas A

Procedūra 1 – Sakinio ilgio kontrolė (sakinio trumpinimas)

Pradiniai duomenis(tekstas):

This was a beautiful day everywhere where can I see was a deep blue sky but I was alone as in the dark place because my girlfriend left me and I was very dissappointed

The screenshot shows a software interface with three tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Pradinis\_tekstas' tab is active, displaying the original text: 'This was a beautiful day everywhere where can I see was a deep blue sky but I was alone as in the dark place because my girlfriend left me and I was very dissappointed.' The 'Galutinis\_tekstas' tab is also visible, showing the result of the shortening process: 'This was a beautiful day where can I see was a deep blue sky, but I was alone as in the dark place because my girlfriend left me. And I was very dissappointed.'

33 pav. Sakinio trumpinimas

Rezultatas:

This was a beautiful day where can I see was a deep blue sky, but I was alone as in the dark place because my girlfriend left me. And I was very dissappointed.

Procedūra 2 – Vienažodžių veiksmažodžių kontrolė (pakeitimas kitu)

Pradiniai duomenis(tekstas):

Mike, please hand in this letter to my mother because tomorrow I will be very busy.

The screenshot shows a software interface with three tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Pradinis\_tekstas' tab is active, displaying the original text: 'Mike, please hand in this letter to my mother because tomorrow I will be very busy.' The 'Galutinis\_tekstas' tab is also visible, showing the result of the modification: 'Mike, please submit this letter to my mother, tomorrow I will be very busy.'

34 pav. Vienažodžių veiksmažodžių kontrolė

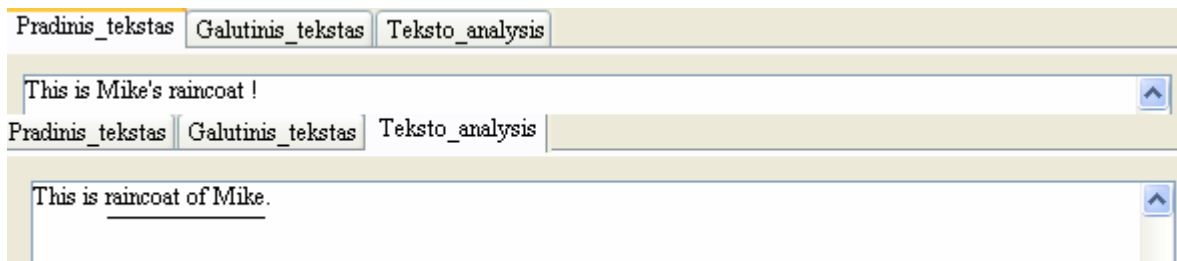
Rezultatas:

Mike, please submit this letter to my mother, tomorrow I will be very busy.

Procedūra 3 – Daiktavardžio tikslinimo kontrolė (jeigu du daiktavardžiai eina vienas po kito tarp jų įterpiamas „of“)

Pradiniai duomenis(tekstas):

This is Mike's raincoat !



35 pav. Daiktavardžio tikslinimo kontrolė

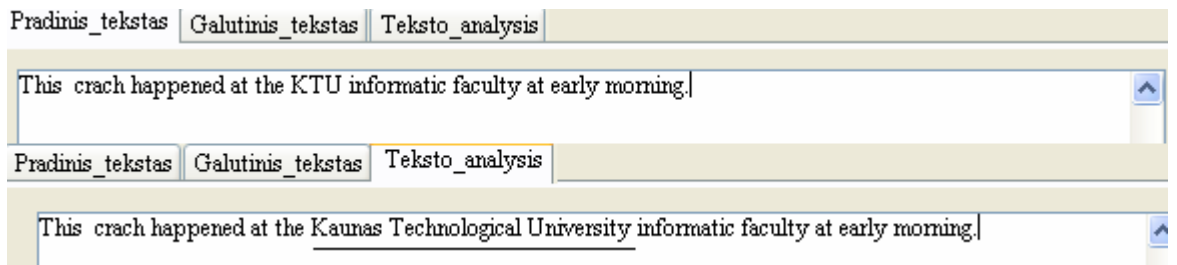
Rezultatas:

This is raincoat of Mike.

Procedūra 4 – Santrumpų kontrolė (ieško sakinyje santrumpą ir pakeičia ją pilnu žodžių junginiu)

Pradiniai duomenis(tekstas):

This crach happened at the KTU informatic faculty at early morning.



36 pav. Santrumpų kontrolė

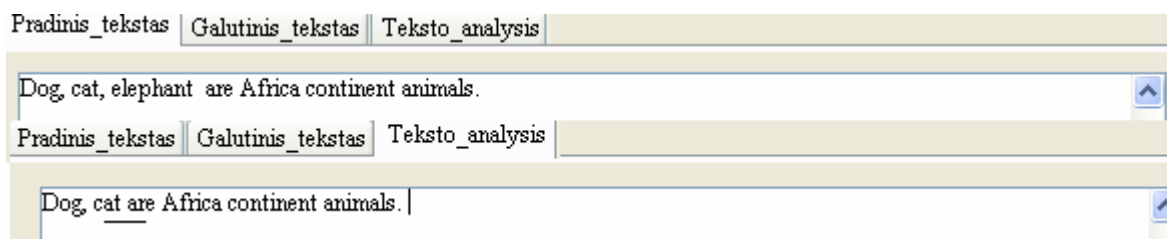
Rezultatas:

This crach happened at the Kaunas Technological University informatic faculty at early morning.

Procedūra 5 – daiktavardžių junginių kontrolė (jei daiktavardžių junginis viršija 3 daiktavardžius - paskutinis daiktavardis yra pašalinamas)

Pradiniai duomenis (tekstas):

Dog, cat, elephant are Africa continent animals.



The screenshot shows a software interface with a light beige background. At the top, there are three tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Pradinis\_tekstas' tab is selected. Below the tabs is a text input field containing the sentence 'Dog, cat, elephant are Africa continent animals.' To the right of the input field is a small blue arrow icon pointing up. Below the input field, there are three more tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Galutinis\_tekstas' tab is selected. Below these tabs is another text input field containing the sentence 'Dog, cat are Africa continent animals.' The word 'cat' is underlined. To the right of this input field is another small blue arrow icon pointing up.

37 pav. Daiktavardžių junginių kontrolė

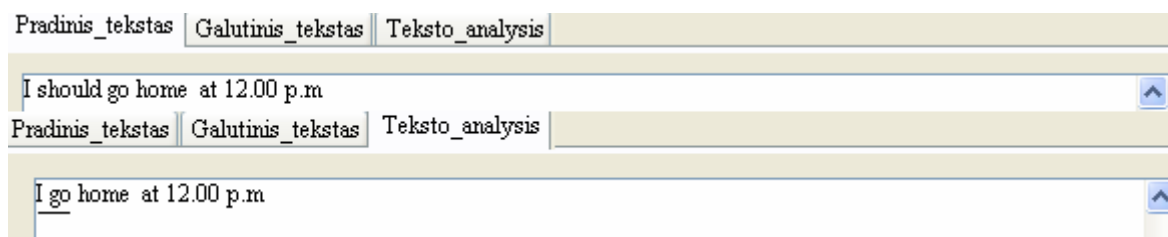
Rezultatas:

Dog, cat are Africa continent animals.

Procedūra 6 – modalinių veiksmažodžių kontrolė (modalinis veiksmažodis pakeičiamas kitu veiksmažodžiu arba tiesiog tarpu, t.y. paprasčiausiai išmetamas ir teksto konteksto)

Pradiniai duomenis (tekstas):

I should go home at 12.00 p.m



The screenshot shows a software interface with a light beige background. At the top, there are three tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Pradinis\_tekstas' tab is selected. Below the tabs is a text input field containing the sentence 'I should go home at 12.00 p.m'. To the right of the input field is a small blue arrow icon pointing up. Below the input field, there are three more tabs: 'Pradinis\_tekstas', 'Galutinis\_tekstas', and 'Teksto\_analysis'. The 'Galutinis\_tekstas' tab is selected. Below these tabs is another text input field containing the sentence 'I go home at 12.00 p.m'. The word 'go' is underlined. To the right of this input field is another small blue arrow icon pointing up.

38 pav. Modalinių veiksmažodžių kontrolė

Rezultatas:

I go home at 12.00 p.m