

Eksperimentinės plėtros veiklos ataskaita

Projekto pavadinimas: Nurodyti

Projekto numeris: J05-LVPA-K-01-XXXX

Projekto vykdytojas: *UAB „Omnisend“*

Projekto partneriai: UAB „Tokenmill“

Veiklos numeris: Nurodyti

Veiklos pavadinimas: *Nurodyti*

Ataskaitos data: 2019-03-18

Lapų skaičius: Nurodyti

**Turinys**

[4.1. Veiklos aprašas 4](#__RefHeading___Toc803_4054088796)

[4.2. Veiklos vykdymo eiga 4](#__RefHeading___Toc805_4054088796)

[4.3. Rezultatai 5](#__RefHeading___Toc807_4054088796)

[5.1. CCG pasirinkimo priežastys 7](#__RefHeading___Toc809_4054088796)

[5.2. Veiklos aprašas 7](#__RefHeading___Toc811_4054088796)

[5.3. Veiklos vykdymo eiga 8](#__RefHeading___Toc813_4054088796)

[5.4. Rezultatai 9](#__RefHeading___Toc815_4054088796)

[6. Įvykdyti paslaugų pirkimai 10](#__RefHeading___Toc823_4054088796)

# Veiklos tikslas

Sukurti metodiką ir prototipą prekių elektroninės prekybos sistemose automatiniam aprašymų generavimui, panaudojant natūralios kalbos generavimo (NLG) ir dirbtinio intelekto (AI) metodus.

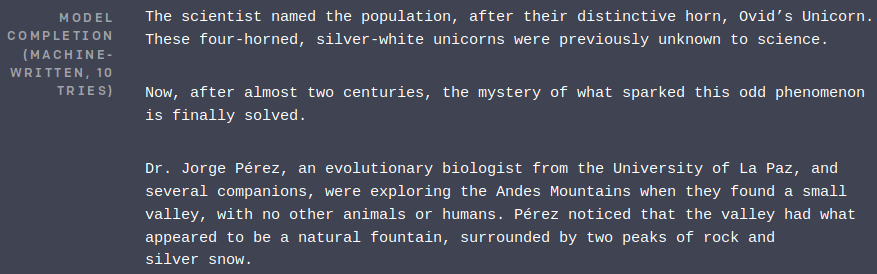
# Veiklos užduotys

Šiame etape buvo apibrėžtos tokios veiklos užduotys:

* 1. *Automatizuotas dokumento plano redaktorius* – tolimesnis vartotojo sąsajos skirtos dokumento plano kūrimui vystymas.
  2. *Formalia gramatika paremtas teksto realizavimo modulio prototipas* skirtas tiksliam pranešimo tikslų realizavimui. naudojamas, formalia gramatika paremtas teksto ir teksto variantų generavimo, metodas.

# Pokyčiai NLG rinkoje – OpenAI NLG rezultatai

2019 vasario 14d Open AI Laboratorija (https://openai.com/) savo žiniatinklyje anonsavo pasiekę itin gerus rezultatus generuojant tekstą pasitelkus dirbtinio intelekto metodą. OpenAI Laborotorijos sugeneruotas tekstas išties pasižymi nuoseklumu, įvairove bei rišlumu, todėl, atrodytų, kad NLG problema jau išspręsta [1].

  
Pav 1: OpenAI sugeneruoto teksto ištrauka

Nors šie rezultatai, tam tikroje srityje, yra iš ties geri, tai nekeičia mūsų projekto tikslų ir neįtakoja pasirenkamų priemonių. Mūsų kuriamo produkto tikslai ženkliai skiriasi nuo technologijos naudojamos OpenAI darbe.

Esminis mūsų darbo tikslas yra sukurti NLG sistemą kuri generuoja tekstą kuris maksimaliai atitinka turimus verslo duomenis. Mes siekiame sukurti technologiją, kuri ne tik pasižymi generuojamo teksto rišlumu, bet ir tiksliai atitinka faktinius duomenis. OpenAI šio reikalavimo netienkina. Pavyzdžiui, OpenAI sugeneruotas sakinys, su faktus neatitinkančiais teiginiais:

*The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These* ***four-horned****, silver-white* ***unicorns*** *were previously unknown to science.*

1. Vienaragiai nėra keturragiai gyvūnai
2. „Ovid’s“ niekaip nenurodo į vieno rago savybę, todėl ši konstrucija net nėra logiška.

Tokio teksto negalima naudoti jokioje verslo aplinkoje, kur tikslumas yra būtina teksto savybė. Galbūt, tolimesniame projekto vystymo etape mes panaudosime OpenAI GPT-2 metodą [2], bet tik lokalių teksto dalių generavimui (frazės, šalutiniai sakiniai).

# Automatizuotas dokumento plano redaktorius

## Veiklos aprašas

Daugialypio dokumento ruošimo (angl: *augmented writer*) vartotojo aplinka, leidžianti aprašyti komunikacijos tikslus ir bendrą dokumento planą, yra vienas iš esminių kuriamo produkto komponentų. Šiame etape mes tęsiame jo vystymą pridėdami daugiau galimybių aprašyti teksto tikslams, kontroliuoti žodyną, bei įvesties duomenis (t. y. tie duomenys apie kuriuos yra kuriamas tekstas).

Antrame etape padarytas Google Blockly [3] vizualinio programavimo kalbos pasirinkimas pasitvirtino ir mes toliau vystome savo sprendimą šio atviro kodo produkto pagrindu.

## Veiklos vykdymo eiga

Daugialypio dokumento kūrimo įgyvendinimas yra atliktas panaudojant vizualinio programavimo paradigma (angl: *visual programming*)*,* kurios vystymui kalbos generavimo plano aprašymo srityje mes naudojame ankstesnėje ataskaitoje aprašytą studiją [4].

Šiame etape mes siekėme suteikti visus instrumentus reikiamus dokumento plano aprašymui (žr paveikslą: Pav 2: Dokumento plano kūrimo priemonė):

1. *Duomenų kontrolė* (ateityje ontologija arba žinių grafas, atspindintis prekių struktūrą) – šiuo metu mes leidžiame įkrauti, paprastą duomenų rinkinį, nekreipdami dėmesio į vidines sąsajas. T.y. paprasta duomenų lentelė (CSV failas). Vėliau šis komponentas plėsis šiomis kryptimis
   1. Atlikti duomenų, jų tipų ir jų sąsajų analizę. Pavyzdžiui aprašyti, kad bėgimo bato dalis yra jo padas, kuris yra itin svarbus komponentas kuriant prekę aprašantį tekstą.
   2. Duomenų praturtinimas naudojant veiklos taisyklių variklį. Pavyzdžiui, leisti sukurti taisyklę kuri sako, *jei* bato kaina yra daugiau nei 150eur, *tai* prie duomenų rinkinio pridėti lauką - ‘brangus’. Šiuos papildomus laukus, NLG sistema gali naudoti generavimo logikos valdymui.
2. *Leksikono kontrolė* – vartotojas turi valdyti kada ir kokie žodžiai ir išraiškos yra naudojamos tam tikrose teksto vietose. Ši galimybė yra svarbi todėl, kad dažnai konkretaus žodyno vartojimas yra patvirtinamas pagal įmonės marketingo strategiją ar atitinka griežtas teisines formuluotes. Šiame komponente, mūsų NLG sistema taip pat leis importuoti sinonimų rinkinius iš WordNet duomenų bazės [5].
3. *Teksto planavimo darbo laukas* – vizualaus programavimo erdvė. Šio etapo metu buvo vystoma gramatikos validavimo veikla. T.y. kuriant dokumento planą sistema neleidžia kurti netinkamų konstrukcijų. Tokiu būdu, kalbos realizavimo komponentas neturės būti daromas nereikalingai sudėtingas. Jam nereikės atlikti bazinių dokumento planto teisingumo patikrinimo operacijų.
4. *Teksto plano komponentų rinkinys* – teksto plano komponentai turi suteikti vartotojui plačias galimybes aprašyti norimas teksto struktūras.
   1. *Document* – dokumento struktūrą valdantys komponentai. Čia esantis *segment* komponentas leidža apibrėžti pagrindines dokumento dalis ir jų rašymo stilių.
   2. *Topics –* teksto temų pasirinkimo komponentas, čia vartotojas gali rinktis apie kokią prekę ar jos savybes reikia kurti tekstą.
   3. *Rhetoric* – ankstesnėse ataskaitose buvo aprašyta, kaip mes planuojame naudoti *Rethorical Structure Theory* [6]. Šie elementai leidžia valdyti kokiu būdu yra susiejamos skirtingos teksto temos. Šiuo metu RST valdymas nėra pilnai aprašytas ir suprojektuotas. Tai bus viena iš pagrindinių sekančio etapo veiklų.

## Rezultatai

Šiuo metu įgyvendintas dokumento plano redakturius yra pasiekiamas, šiuo adresu:

http://augmented-writer-app.s3-website.eu-central-1.amazonaws.com/

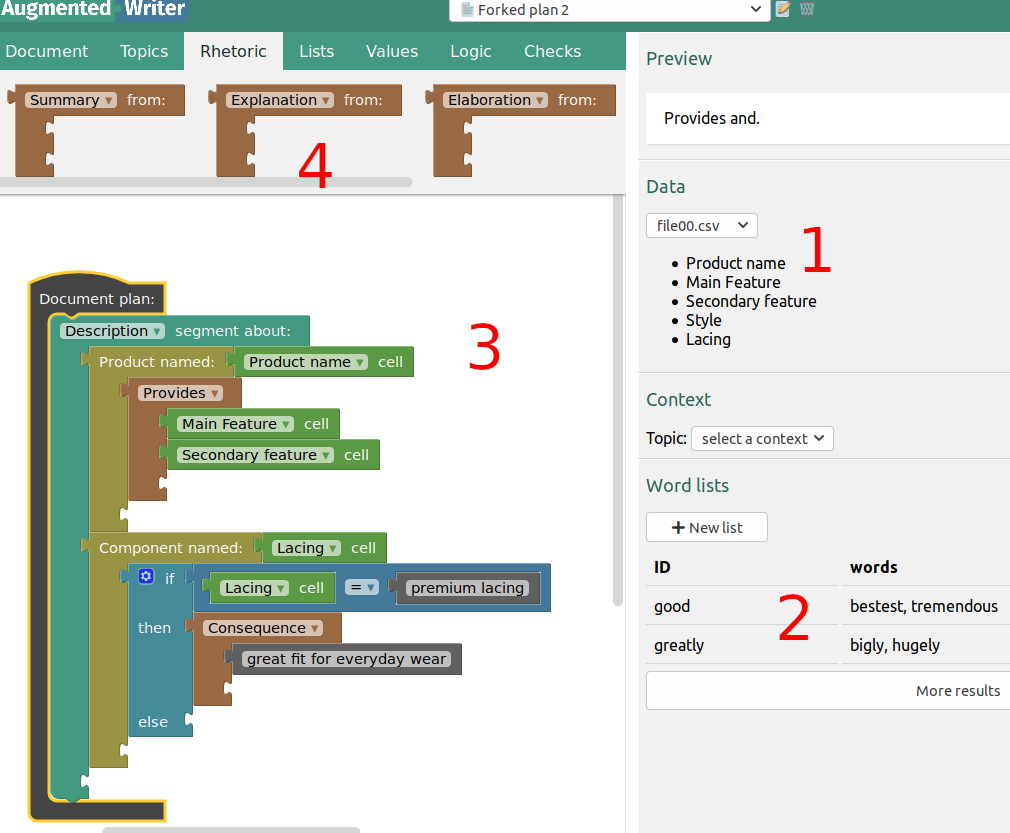
Sukurta vartotojo sąsaja turi visus pagrindinius elementus reikalingus dokumento plano rengimui ir išeities duomenų pakrovimui (reikia turėti mintyje, kad redaktorius naudoja tik nedidelį duomenų pavyzdį; sukūrus dokumentų generavimo planą, jis yra prijungiamas prie elektroninės parduotuvės inventoriaus duomenų bazės).

Sekančiame skyriuje (5.1 CCG pasirinkimo priežastys) bus aprašoma Combinatory Categorial Grammar (CCG) [7] kuri yra naudojama sakinių realizavimui. Dokumento plano redaktorius gali būti suprantamas, kaip aukšto abstrakcijos lygio CCG redaktorius. Tai yra priemonė leidžianti kurti formalias gramatikas asmenims neturintiems kompiuterinės lingvistikos išsilavinimo. Savo ruožtu tai leidžia tiksliau suformuoti šios veiklos rezultatą, šiame ir sekančiuose projekto vystymo etapuose. Dokumento plano redaktoriaus tikslas yra sukurti vizualią programavimo aplinką, kuri:

1. leisti operuoti lingvistinėmis konstrukcijomis;
2. lingvistines konstrukcijas sujungti su vartotojo duomenimis (ontologija);
3. leisti aprašyti komunikacijos tikslus;
4. remiantis ankstesniais žingsniais generuoti programinį kodą, jei įprastiniu vizualinio programavimo atveju, kodas yra: JavaScript, Python ar kita kalba, mūsų atveju tai yra formali CCG teorija aprašyta gramatika.

1,2,3 žingsniai yra formalizuojami didele dalimi naudojant RST [6], o 4 žingsnis atlieka CCG generavimą.

Šio etapo rezultatas yra RST ir CCG naudojimo ir sujungimo prototipas, įgalinantis pradėti eksperimentus, kurių tikslas rasti optimalias aukšto lygio abstrakcijas, leidžiančias kurti formalias gramatikas skirtas implementuoti norimus komunikacijos tikslus.

  
Pav 2: Dokumento plano kūrimo priemonė

# Dokumento plano ir CCG jungimas

## CCG pasirinkimo priežastys

*Combinatory Categorial Grammar* (CCG) [7] yra viena iš daugelio formalios gramatikos teorijų [8]. Šiam projektui mes pasirinkome CCG dėl žemiau išvardintų priežasčių.

*Pirma*. CCG apibrėžia bazinius elementus, pavyzdžiui *veiksmažodžius*, kaip funkcijas, kurios priima argumentus, pavyzdžiui *daiktavardžius*, ir, galiausiai, nusako koks yra šios funkcijos rezultatas, pavyzdžiui *sakinys* ar *daiktavardinė išraišką*. Tokios žemesnio lygio funkcijos gali būti kombinuojamos (naudojant *Lambda Calculus metodologiją* [9]), kuriant vis sudėtingesnes lingvistines konstrukcijas. Šis funkcinis metodas, mums leidžia kurti didelės įvairovės tekstus. Taip pat, tai gerai atspindi funkcinio programavimo principus ir todėl yra lengviau naudojamas kartu su mūsų vizualinio programavimo paradigma.

*Antra*. CCG jungia gramatiką ir ontologiją. T.y. šioje gramatikoje operuojama ne tik gramatinėmis struktūromis bet ir ontologija. Pasirinktoms kalbos ar sakinio dalims galima priskirti ontologines kategorijas. Taip mes turime ne tik tikrinį daiktavardį, bet ir *tikrinį daiktavardį, kuris nurodo į miestą esantį prie vandens telkinio.* Ši CCG savybė yra itin tinkama mūsų tikslams. Mūsų kuriamos generavimo gramatikos turi būti priklausomos nuo prekės tipo ar kitų požymių – kas ir yra aprašoma naudojant prekių ontologiją. Pats ontologijos kūrimas nėra šio projekto dalis.

## Veiklos aprašas

Pagrindinė veiklos kryptis buvo išsiaiškinti CCG veikimo principus ir jų taikymą mūsų reikmėms. CCG gramatika yra konstruojama atsiremiant į leksinius elementus. Pavyzdžiui (labai supaprastinant):

*Nike Air 95* ***suteikia*** *lengvumą.*

CCG modelis remtųsi šiais leksiniais konstruktais:

|  |  |  |
| --- | --- | --- |
| ***Leksinis elementas*** | ***Gramatinė funkcija*** | ***Ontologinė nuoroda*** |
| *Nike\_Air\_95* | **NNP** | bėgimo-batas |
| *suteikti* | **(S/NNP) \ NP** | funkcija |
| *lengvumas* | **NP** | Bato-privalumai |
| Natūrali Kalba | Prasmė | |

Imant pirmąją lentelės eilutę, šiuos CCG aprašus reikia skaityti taip:

* + - *Nike Air 95* – natūralios kalbos elementas, apibrėžiantis konkrečias konstrukcijas, kurios nusako kokiais žodžiais sakinyje bus realizuojama gramatinė išraiška. Šiuo atveju tai yra tik viena fiksuota frazė – prekės pavadinimas, kitais atvejais (kaip žemiau esantis *suteikti)*, gali turėti sinonimų sąrašus, kaip alternatyvas.
    - *NNP* – yra pirmoji prasmės konstravimo dalis, ji nurodo į kalbos dalį atitinkančią parenkamus žodžius. Tačiau reikia nepamiršti, kad tai visų pirma yra *funkcija* ir ji nusako kokio tipo sąryšiai atsiranda naudojant šiuos žodžius tam tikrame kontekste.
    - *bėgimo-batas* – gramatinė funkcija tik dalinai leidžia konstruoti išraiškos prasmę, reikalinga papildoma – ontologinė nuoroda – leidžianti pasakyti, kaip kuriama išraiška siejasi su daiktais esančiais veiklos srityje.

Leksiniai elementai atspindi *natūralios kalbos* dalį CCG gramatikoje. *Gramatinė funkcija* ir *Ontologinė nuoroda* nurodo į prasmę, kurią turi kuriamas gramatinis konstruktas. Tokiu būdu *Prasmė* ir *Natūrali Kalba* tampa dvejomis kryptimis, pagal kurias yra vystoma šios projekto dalies veikla.

## Veiklos vykdymo eiga

***Natūralios kalbos* kryptis** iš dalies jau buvo dalinai įgyvendinta pirmojo etapo veiklose. Tuo metu buvo kuriamas el. parduotuvių tekstynas, leidžiantis kaupti, šioje srityje naudojamus, atskirus žodžius bei sakinius. Taip pat ši veikla atspindima ir dokumento plano redaktoriaus kūrime žr. Pav 2 Dokumento plano kūrimo priemonė (antruoju numeriu pažymėtas grafinės sąsajos elementas).

Šios veiklos vystymo eiga ir numatomi darbai.

|  |  |  |
| --- | --- | --- |
| ***Duomenų šaltinis*** | ***Būsena*** | ***Numatomi darbai*** |
| El parduotuvių aprašų duomenys | Surinkta apie 100 000 tekstų | * Filtravimas pagal teksto kokybę. * Grupavimas pagal kalbos dalis. |
| WordNet žodynų importavimas | Sukurti WordNet duomenų bazės importavimo įrankiai | * Sekančiame etape sukurti įrankius kurie filtruoja reikiamus žodžių rinkinius |
| Word2Vec |  | * Sukurti priemones leidžiančias naudoti Word2Vec modelius naujų žodžių ir jų sinonimų radimui |

**Prasmės konstravimo kryptis**. Norint išvengti ontologijos redaktoriaus įtraukimo į vartotojo sąsają, prasminių sąsajų kūrimas yra atliekamas per dokumento plano struktūrų redagavimą.

Abiejų krypčių implementacijai buvo naudojama OpenCCG [10] atviro kodo programinė įranga. OpenCCG priemonėmis galima aprašyti: ontologines struktūras, natūralios kalbos žodyną, bei gramatines funkcijas [11]. OpenCCG gramatikos deklaravimo pavyzdys:

*Apibrėžiama kokiai kalbos daliai kuriama gramatika*

<family name="Noun" pos="N">

<entry name="Primary">

*Nurodomos ontologinės sąsajos*

<atomcat type="n">

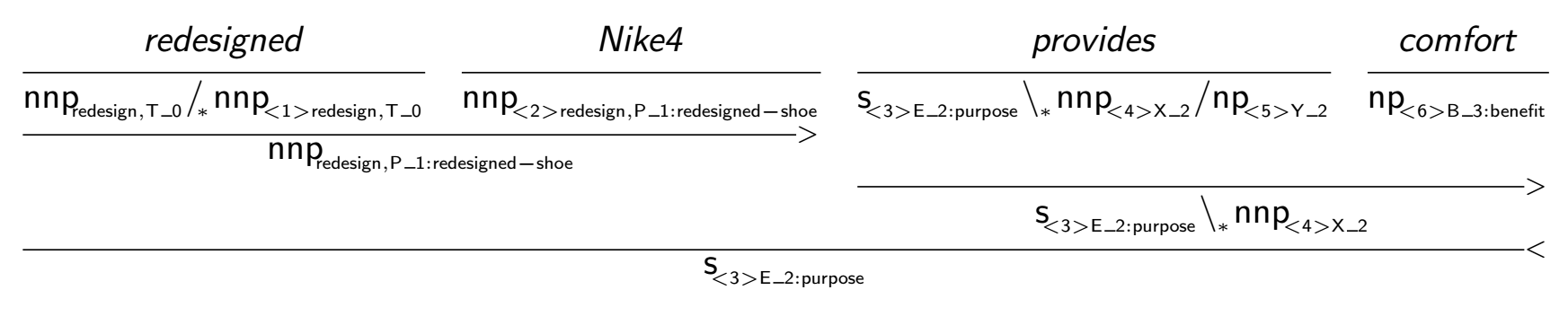
:

</atomcat>

</entry>

</family>

Automatiškai sugeneruotos gramatikos pavyzdys:

  
Pav 3: Paprasto sakinio CCG gramatika

Detalus šios gramatikos paaiškinimas yra suteikiamas CCG literatūroje [7], [11]. Reikėtų atkreipti dėmesį į tai, kad ši gramatika iš esmės aprašo funkcinę kompoziciją, kur įvairios funkcijos priima gramatikos ir ontologijos kombinacijas, kaip įvesties duomenis ir generuoja aukštesnio lygio gramatines išraiškas iki pat pilno sakinio generavimo (paskutinė *‘S<3>’* pažymėta linija). Šiame etape mes sukūrėme bazinę metodologiją tokių gramatikų kūrimui. Tolimesni etapai bus skirti funkcinių struktūrų atradimui, leidžiančių mus generuoti platų spektrą skirtingų sakinių. Sakinių tipus apribojant elektroninės parduotuvės tekstų poreikiams: prekių bei jų komponentų aprašymai, jų teikiamą naudą bei funkciją.

## Rezultatai

Šiame etape buvo sukurtos bazinės gramatikos generuojančios keletą paprastų tekstu, aprašančių turimas prekes. Vieno sakinio variacijų pavyzdžiai:

* 1. The Nike Air 95 provides comfort and support.
  2. The redesigned Air 1001 offers comfort and support.
  3. Comfort and support is provided by the all new Air 1001.

Čia reikia atkreipti dėmesį į Air 95 ir Air 1001 skirtumą. Vėlesnis sporto batas yra naujas modelis, todėl gramatika leidžia panaudoti „*redesinged*“ arba „*all new*“ požymį prieš bato pavadinimą. Tai yra prekės tipo (ontologijos) skirtumas, kuris diktuoja skirtingų sakinio struktūrų generavimą. Air 95 nėra naujas modelis todėl šalia jo negalima naudoti būdvardžių atspindinčių modelio naujumą.

Šiame etape mums pavyko panaudoti semantinių ir prasminių elementų jungimą CCG gramatikoje. Tai mums leidžia tiksliai kontroliuoti kaip yra generuojamas tekstas remiantis konkrečiais parduotuvės duomenimis. Tokiu būdu mes galime išvengti neapibrėžtumo aprašyto *3. Pokyčiai NLG rinkoje – OpenAI NLG rezultatai* skyriuje.

Tolimesnis šios srities (ir didele viso projekto) veiklos tikslas bus aprašyti lanksčias ir abstrakčias CCG gramatikas, tinkančias generuoti įvairių prekių aprašymus. Mes darome prielaidą, kad retorinės struktūros teorija [6] mums leis kurti tokias abstrakčias gramatikas.

# **Įvykdyti paslaugų pirkimai**

Amazon AWS serveriai ir paslaugos (dalis paslaugų šiuo metu neviršija nemokamo naudojimo limito):

1. *Amazon Lambda* – paslaugos ir NLG operacijos
2. *Amazon ES* – tekstynų saugojimas
3. *Amazon VCP* – saugumo infrastruktūra
4. *Amazon DinamoDB* – darbinių duomenų saugykla
5. *UAB Interneto Vizija serveriai* – laikini eksperimentai su pilno sprendimo diegimo aplinka.

# Užduočių atlikimas

Pateikti informaciją, kokios užduotys buvo atliktos projektą vykdančių asmenų, kas jas atliko, kiek laiko tam skyrė, naudoti lentelės formą.

| **Nr.** | **Užduotis** | **Darbuotojai/darbovietė** | **Laiko sąnaudos valandomis** |
| --- | --- | --- | --- |
| 1. | CCG analizė. Pradinių modelių įgyvendinimas | Žygimantas Medelis, UAB “TokenMill” | 168 |
| 2. | NLG sistemos architektūra ir implementacija. | Dainius Jocas UAB “TokenMill” | 431 |
| 3. | CCG gramatikos ir dokumento plano susiejimas ir konvertavimas | Šarūnas Navickas UAB “TokenMill” | 446 |
| 4. | Dokumento plano redaktorius | Tomas Čerkaukas UAB “TokenMill” | 447 |
| 5. | Kiti NLG tyrimai | Žygimantas Medelis, UAB “TokenMill” | 80 |

# Pasiekti rezultatai

Abiejų šio etapo veiklų tikslas buvo sukurti aplinkas, leidžiančios kurti formalia gramatika paremtą, teksto generavimo modelį. Kuriama dokumento plano kūrimo aplinka turi vartotojui nekelti reikalavimo išmanyti gramatikos konstravimo principus, bet tuo pačiu metu suteikti pakankamai instrumentų tiksliai apibrėžti dokumento komunikavimo tikslus.

Šis tikslas yra pasiektas ir, naudojant sukurtą modelį, tolimesniuose projekto etapuose bus toliau vystomas išplėsto dokumento redaktoriaus (Augmented Writer) išraiškų efektyvumas bei šių išraiškų konvertavimas į CCG gramatikos formalizmą.

Savo ruožtu CCG gramatikos turi būti kuriamos taip, kad jos leistų maksimalų perkomponavimą ir perpanaudojimą skirtinguose lingvistiniuose kontekstuose, tokiu būdų užtikrinant viso sprendimo lankstumą bei plėtimo galimybes. Šio etapo metu buvo pasiekta principinė CCG gramatikos naudojimo architektūra. Tolimesniuose projekto etapuose šis rezultatas bus vystomas dvejomis kryptimis:

1. Kadangi siekiama turėti maksimaliai abstrakčias gramatikas (perpanaudojimas ir perkomponavimas), galimų sakinių generavimo erdvė bus didelė, ir iškils poreikis taikyti AI technologijas sakinių rangavimui, pagal jų atitikimą komunikacijos tikslams.
2. CCG ir RST teorijų apjungimas atrandant efektyviausias gramatines išraiškas, tinkančias konkretiems komunikacijos tikslams pasiekti. Ši CCG ir RST kombinacija turės atsispindėti mūsų *Augmented Writer* įrankyje, kur vartotojas galės pasirinkti, kaip jai geriausiai kalbėti apie prekės išskirtinę savybę ar pabrėžti taikomą nuolaidą.

# Nepasiekti rezultatai

Visi numatyti rezultatai buvo pasiekti.

# **Veiklos vykdymo metu atliktų tyrimų eigos pakeitimai**

Veiklos pakeitimų nebuvo.

# **Rezultatų naujumas ir nauda projektui**

Šio etapo metu skirtingos technologijos ir taikomos metodologijos buvo susietos į vieną veikiantį viso sprendimo modelį. Anksčiau aprašyta RAGS architektūra, Retorinės Struktūros Teorija, bei CCG gali veikti kartu, kuriant NLG sprendimą.

CCG ir formalios gramatikos naudojimas natūralios kalbos projektuose nėra naujiena. Formalios gramatikos yra plačiai naudojamos tiksliam teksto struktūros aprašymui. Skirtingai, nuo šio įprastinio panaudojimo, mes formalią gramatiką naudojam ne kaip konkretų tekstą aprašančią struktūrą, o kaip programinį kodą kuris gali generuoti skirtingas dokumento variacijas.

Formalios gramatikos supratimas, kaip programinio kodo, generuojančio tekstus, ir aukšto lygio redaktoriaus, skirto šio kodo rašymui, sukūrimas būtų pagrindinis šio etapo ir viso projekto naujumas NLG srityje. Mes išplėtime vizualinio programavimo paradigmą. Jei įprastai tai suprantama, kaip vizualinės priemonės skirtos naudoti bazinius programavimo elementus (IF, LOOP, ir .t.t) ir generuoti Java, JS ar Python kodą, tai mūsų atveju, baziniai programavimo elementai yra plečiami lingvistinėmis konstrukcijomis (remiantis RST). Šios vizualios programos rezultatas yra kita – CCG gramatika – paremta programa. Kuri savo ruožtu jau yra naudojama sukurti galutiniams tekstams.

# Bibliografija

1. OpenAI. *Better Language Models and Their Implications* (Skaityta: 2019-03-13) <https://openai.com/blog/better-language-models/>
2. Radford, R., et al. *Language Models are Unsupervised Multitask Learners* (2018)
3. Google. *Blockly* (Skaityta: 2019-03-13) https://developers.google.com/blockly/
4. Green T.R.G., Petre M.*, Usability Analysis of Visual Programming Environments: a ‘cognitive dimensions’ framework (1996)*
5. *WordNet* (Skaityta: 2019-03-13), https://wordnet.princeton.edu/
6. *Rethorical Structure Theory* (Skaityta: 2019-03-13), https://www.sfu.ca/rst/
7. *Combinatory Categorial Grammar* (Skaityta: 2019-03-15) http://lsa2017.uky.edu/combinatory-categorial-grammar-introduction
8. *Grammar Frameworks* (Skaityta: 2019-03-15) <https://en.wikipedia.org/wiki/Category:Grammar_frameworks>
9. *Lambda Calculus* (Skaityta: 2019-03-15) http://www.inf.fu-berlin.de/lehre/WS03/alpi/lambda.pdf
10. *OpenCCG* (Skaityta: 2019-03-15) http://openccg.sourceforge.net/
11. C. Bozşahin*,* et al*, Specifying Grammars for OpenCCG:A Rough Guide* (2005) https://users.metu.edu.tr/bozsahin/nli/ceng563/link/grammars-rough-guide.pdf