

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: 2019-09-03

Lapų skaičius: Nurodyti

Turinys

[1. Veiklos tikslas 3](#_Toc11242468)

[2. Veiklos užduotys 3](#_Toc11242469)

[3. Esamo NLG sprendimo architektūros apžvalga 3](#_Toc11242470)

[3.1. Architeltūros komponentai 3](#_Toc11242471)

[3.2. Technologiniai komponentai 5](#_Toc11242472)

[4. Testavimo strategijos parengimas 5](#_Toc11242473)

[4.1. Veiklos aprašas 5](#_Toc11242474)

[4.2. Veiklos vykdymo ega 7](#_Toc11242475)

[5. Abstrakti prasmės reprezentacija dokumento plane 7](#_Toc11242476)

[5.1. Veiklos aprašas 7](#_Toc11242477)

[5.2. Veiklos vykdymo eiga 8](#_Toc11242478)

[5.3. Rezultatai 11](#_Toc11242479)

[6. Vartotojo modelio komponentas 12](#_Toc11242480)

[6.1. Veiklos aprašas 12](#_Toc11242481)

[6.2. Veiklos vykdymo eiga 12](#_Toc11242482)

[6.3. Rezultatai 13](#_Toc11242483)

[7. Galutinio teksto generavimas 14](#_Toc11242484)

[7.1. Veiklos aprašas 14](#_Toc11242485)

[7.2. Veiklos vykdymo eiga 14](#_Toc11242486)

[7.3. Rezultatai 14](#_Toc11242487)

[8. Įvykdyti paslaugų pirkimai 15](#_Toc11242488)

[9. Užduočių atlikimas 15](#_Toc11242489)

[10. Pasiekti rezultatai 16](#_Toc11242490)

[11. Nepasiekti rezultatai 16](#_Toc11242491)

[12. Veiklos vykdymo metu atliktų tyrimų eigos pakeitimai 16](#_Toc11242492)

[13. Rezultatų naujumas ir nauda projektui 16](#_Toc11242493)

[Bibliografija 17](#_Toc11242494)

# Veiklos tikslas

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

# Veiklos užduotys

Šiame etape buvo apibrėžtos tokios veiklos užduotys:

* 1. Bandomasis diegimas testinėje el-komercijos platformoje. Buvo sukurta bandomoji el-parduotuvė Shopify.com platformoje, kurioje prekių aprašymas yra atliekamas naudojant mūsų teksto generavimo variklį.
  2. Abstrakčios prasmės reprezentavimo modelio plėtimas. Tęsiamas praeitų etapų darbas (2019-06-14 dienos ataskaita 5 skyrius [1]), aprašant galimas prasmės reprezentavimo struktūras.
  3. Teksto generavimo komponento tolimesnis vystymas ir taikymas.
  4. Teksto plano kūrimo vartotojo sąsajos tobulinimas.

# Bandomoji el-parduotuvė

## Veiklos aprašas

Kuriama teksto generavimo sistema turi integruotis su įvairiomis, turinį publikuojančiomis, platformomis. Dabartinis tikslas yra turėti sistemą veikiančią, bet neapsiribojančią, su pagrindinėmis el-komercijos platformomis. Bandomajam diegimui buvo pasirinkta *Shopify* platforma.

Teksto publikavimo integracijos įgyvendinimui reikalingi žingsniai:

1. *Duomenų sąsaja*. NLG sistema turi pasiimti pradinius, prekę aprašančius, duomenis.
2. *Teksto generavimas*. Prekę aprašančio teksto paruošimas, pagal pirmame žingsnyje gautus duomenis
3. *Teksto perdavimo sąsaja*. Sugeneruotas tekstas turi būti grąžinamas į el-parduotuvę.

Pirmojo ir trečiojo žingsnio vykdymo eiga aprašyta sekančiame skyriuje. Antrasis žingsnis čia neaprašomas, nes yra detaliai aptartas ankstesnėse ataskaitose.

## Veiklos vykdymo eiga

Tiek pradinių duomenų gavimas tiek jų grąžinimas į el-prekybos platformą yra atliekamas per GraphQL interfeisą aptartą 2019-06-14 ataskaitos *3.2 Technologiniai komponentai* skyriuje. Shopify platformos API taip pat yra įgyvendintas GraphQL protokolu. Tai palengvina integraciją su mūsų produktu bei patvirtina mūsų apsiriktos architektūros tinkamumą el-prekybos kontekste. Toliau GraphQL naudojimo specifika nebus aptariama, fokusuojantis į labiau funkcionalius tokios integracijos vykdymo eigos žingsnius.

### **Duomenų struktūros apibrėžimas**

Duomenų struktūra, su kuria yra dirbama yra labai paprasta. Ją galima atspindėti paprasta lentele, kurioje yra išvardinami visi prekės atributai. Prekę aprašantis tekstas yra toks pat atributas kaip ir kiti duomenys.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Authors** | **Kiti atributai** | **Teksto variantai** |
| Building Search Applications | Manu Konchady | ... | Manu Konchady is the author of Building Search Applications with Lucene, LingPipe, and Gate… |
| Moving to the Cloud | Dinkar Sitaram, Geetha Manjunath | Moving to the Cloud watch with Developing Apps in the New World of Cloud Computing was written by Dinkar Sitaram, Geetha Manjunath… |
| Text Processing with GATE | Hamish Cunningham, Kalina Bontcheva, Diana Maynard | Text Processing with GATE watch with Version 6. Authors are Hamish Cunningham, Kalina Bontcheva, Diana Maynard... |

Lentelė 1 Knygos tipo prekių duomenų struktūros iliustracija

Duomenų apsikeitimo veiklos tikslas yra rast sprendimą, kuris leidžia maksimaliai lanksčiai keistis tokio tipo duomenimis. Todėl, NLG sistemoje tokios, prekių aprašymo lentelės yra konvertuojamos į įrašų masyvus, kurie saugo duomenis tokiu formatu:

**Eilutė**, turi daug **Laukų**, kiekvienas laukas sudarytas iš **Vardo** ir **Reikšmės** elementų. Formaliau (JSON formatu):

[

[

{ "field-name" : "title", "field-value" : "Building Search Applications },

{ "field-name" : "authors", "field-value" : "Manu Konchady" }, …

]

…

]

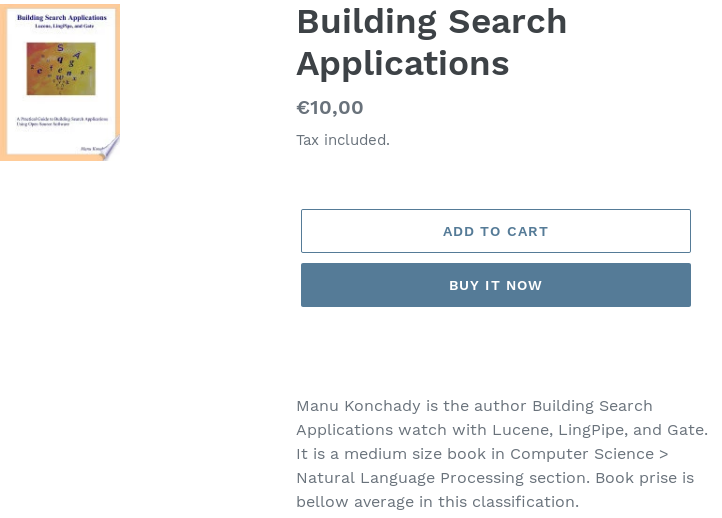
Tokiu būdu mes galime lanksčiai integruotis su el-komercijos sistemomis, kurios skirtingai reprezentuoja prekių duomenis.

### **Shopify integracija**

Shopify, kaip ir kitų panašių el-prekybos platformų, darbo procesas susijęs su prekių įkėlimu vyksta per prekių inventoriaus valdymo darbą. Todėl aprašančio teksto generavimo sistema neturėtų, keisti tokio darbo įpročių ir nusistovėjusių procesų.

Pav 1: Prekių inventoriaus ekranas Shopify platformoje

Mūsų diegiama NLG sistema gali gauti pranešimus apie prekių pakeitimus inventoriaus lentelėje ir papildyti prekės duomenis, prekės automatinio aprašo lauku. Naudojant mūsų sistemos API susiejimą su Shopify API [2], mes galime visiškai automatizuoti tokį procesą ir atsidarius prekės langą rodyti mūsų sugeneruotą aprašą.

  
Pav 2: Prekės pardavimo langas Shopify el-parduotuvėje, su automatiškai sugeneruotu aprašo tekstu.

### **Veiklos Rezultatai**

|  |  |  |
| --- | --- | --- |
| **Tema** | **Rezultatai** | **Iššūkiai** |
| Integracijos duomenų struktūros | a) GraphQL leidžia lankstų integracinio API apibrėžimą. Tai yra populiarus sąsajų apibrėžimo formatas, kas leidžia tikėtis, kad daugiau duomenų platformų bus įgyvendinta šiuo būdu. Tai įgalina ir platesnį mūsų NLG sistemos diegimą.  b) Pačių duomenų dekomponavimas į atskirų laikų masyvus, leidžia neapibrėžti fiksuotos duomenų schemos, kas savo ruožtu nereikalauja, kad NLG naudojanti sistema būtų keičiama. | a) Integracija visada bus tik dalina, negalime turėti sistemos, kuri be pakeitimų integruojasi į bet kokią išorinę platformą. |
| Shopify integracija | a) Pilna integracija per prekių inventoriaus pasikeitimų stebėjimą  b) Generuojami tekstai automatiškai įkeliami į el-parduotuvę. | Iššūkių nėra, iš dalies todėl, kad mes patys pasirinkome Schopify integraciją. Pats Shopify turi itin išvystytą integracijos platformą. |

# Abstrakti prasmės reprezentacija dokumento plane

## Veiklos aprašas

## Veiklos vykdymo eiga

## Rezultatai

# Vartotojo modelio komponentas

## Veiklos aprašas

## Veiklos vykdymo eiga

## Rezultatai

# Galutinio teksto generavimas

## Veiklos aprašas

## Veiklos vykdymo eiga

## Rezultatai

Šis komponentas vystomas pagal tokias kryptis:

|  |  |  |
| --- | --- | --- |
| **Tema** | **Rezultatai** | **Iššūkiai** |
| Sintaksės realizavimas | Šiame etape neplanuota įgyvendinti. | - neigimo konstravimas |
| Morfologinis realizavimas | Šiame etape neplanuota įgyvendinti. | - daugiskaitos ir išimčių valdymas: cat-cats, women-woman; |
| Ortografinis realizavimas | - Sakinio pradėjimas didžiąją raide;  - Elementų vardinimo jungimas per ‘,’ ar ‘ir’ konstruktus.  - Sakinio užbaigimas naudojant reikiamą skyrybos ženklą. |  |

# 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

# 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

# Nepasiekti rezultatai

Visi numatyti rezultatai buvo pasiekti.

# Veiklos vykdymo metu atliktų tyrimų eigos pakeitimai

Veiklos pakeitimų nebuvo.

# Rezultatų naujumas ir nauda projektui

# Bibliografija

1. 2019-06-14 dienos projekto ataskaita
2. *GraphQL examples for products* (Skaityta 2019-09-03) https://help.shopify.com/en/api/graphql-admin-api/examples/product