

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

Ernestas Stravinskas

Virtualaus asistento kūrimas naudojant programinį agentą

Magistro darbas

Vadovas doc. dr. E. Paliulis

Šiauliai, 2013

ŠIAULIŲ UNIVERSITETAS

Informacinių technologijų katedra

TVIRTINU

IT katedros vedėjas doc. dr. M. Bernotas

2013-__-__

Virtualaus asistento kūrimas naudojant programinį agentą

Informatikos inžinerijos magistro darbas

Autorius

ITM-11 gr. studentas
2013 m. gegužės 27 d.

E. Stravinskas

Vadovas

IT katedros docentas
2013 m. gegužės 27 d.

dr. E. Paliulis

Recenzantai

IT katedros lektorė
2013 m. gegužės __ d.
IT katedros lektorė
2013 m. gegužės __ d.

dr. S. Ramanauskaitė

dr. A. Drukteinienė

Šiauliai, 2013



TVIRTINU

Informacinių technologijų

katedros vedėjas

2013 m. _____ mėn. _____ d.

doc. M. Bernotas

Magistro darbo užduotis

Studentui ERNESTUI STRAVINSKUI

Darbo tema: VIRTUALAUS ASISTENTO KŪRIMAS NAUDOJANT PROGRAMINĮ AGENTĄ

(angl. DEVELOPMENT OF VIRTUAL ASSISTANT USING SOFTWARE AGENT)

Patvirtinta 2013 m. _____ mėn. _____ d. potvarkiu Nr. _____

1. Darbe atlikti šias užduotis:
 - 1.1. Atlikti programinių agentų analizę.
 - 1.1.1. Išnagrinėti programinių agentų veikimą ir jų sandarą.
 - 1.1.2. Išnagrinėti programinių agentų tipus ir jų klasifikaciją.
 - 1.1.3. Atlikti programinių agentų palyginimą.
 - 1.1.4. Atlikti programinių agentų kūrimo platformų įvertinimą ir palyginimą.
 - 1.1.5. Išnagrinėti programinių agentų integravimo internetinėse svetainėse galimybes ir praktinį jų panaudojimą.
 - 1.2. Sukurti virtualaus asistento, panaudojant programinį agentą, projektą.
 - 1.2.1. Atlikti reikalavimų specifikacijas.
 - 1.2.2. Sukurti programinio agento funkcionavimo e. parduotuvėje algoritmą.
 - 1.2.3. Suprojektuoti specializuotą duomenų bazę, agento duomenų kaupimui.
 - 1.2.4. Parinkti programinio agento, kaip virtualaus asistento, realizavimo platformas ir įrankius.
 - 1.3. Pasirinkta platforma ir įrankiais realizuoti suprojektuotą virtualų asistentą.
 - 1.4. Integuoti virtualų asistentą į pasirinktą e. parduotuvę.
 - 1.5. Atlikti eksperimentą ir praktinį virtualaus asistento patikrinimą.
2. Aiškinamojo rašto turinys turi atitikti Informacinių technologijų katedros studentų magistro darbų metodinius nurodymus.
3. Magistro darbas pateikiamas su įrišta darbo užduotimi, įdėtu vadovo atsiliepimu ir kompaktine plokštele.
4. Kompaktinėje plokštelėje įrašyti visus su sukurtu virtualiu asistentu susijusius failus ir magistro darbo aiškinamąjį raštą.
5. Gynimo metu pateikiama pristatymo pateiktis ir liudijimas, kad magistro darbas yra įkeltas į Lietuvos ETD informacinę sistemą.
6. Apginto katedroje baigiamojo darbo pristatymo paskutinioji diena – 2013 m. gegužės 23 d.

Baigiamojo darbo vadovas

2013 m. vasario 26 d.

doc. dr. E. Paliulis

(parašas)

Studentas

2013 m. vasario 26 d.

E. Stravinskas

(parašas)

VIRTUALAUS ASISTENTO KŪRIMAS NAUDOJANT PROGRAMINĮ AGENTĄ

Santrauka

Programiniai agentai leidžia automatizuoti užduočių vykdymą, supaprastina sudėtingos sistemos valdymą ir suteikia sistemai intelektualių savybių. Iš agentų sudarytos sistemos tapo nauja ir perspektyvi programų inžinerijos ir dirbtinio intelekto šaka.

Darbo tikslas - ištirti programinio agento panaudojimo el. parduotuvėse galimybes.

Darbo užduotis - išanalizuoti programinius agentus, pasirinkti priemones ir realizuoti virtualų asistentą, bei integruoti jį į kelias pasirinktas el. parduotuves, atlikti eksperimentą palyginant sukurtą virtualų asistentą su standartiniais el. parduotuvių prekių siūlymo moduliais.

Realizuota virtualių asistentų sistema yra pranašesnė už integruotus panašių prekių siūlymo modulius el. parduotuvėse. Ji mažai nusileido spartos atžvilgiu integruotiems moduliams, o vienoje parduotuvėje netgi veikė sparčiau. Sukurta sistema veikia autonomiškai, be administratoriaus įsikišimo, reaguoja į aplinką. Pasiūlymai yra kuriami kiekvienam lankytojui individualiai, ko negalima pasakyti apie standartinius el. parduotuvėse modulius. Be to sistema turi daugiau funkcijų, tokių kaip galimybė uždaryti asistento langelį.

Prasminiai žodžiai: programiniai agentai, integravimo į interneto svetaines būdai, realizavimo technologijos, virtualus asistentas, elektroninė parduotuvė, panašių prekių pasiūlymai.

DEVELOPMENT OF VIRTUAL ASSISTANT USING SOFTWARE AGENT

Summary

Software agents allows to automate tasks, simplify the management of complex systems and provides intelligent properties for system. The system of agents, has become a new and promising software engineering and artificial intelligence branch.

The aim of this work - to examine usage of software agent in e-shops.

The task of this work - to analyze software agents, choose the means to realize a virtual assistant, and integrate it into the chosen e-shops, do an experiment by comparing a virtual assistant with the standard e-shop modules.

Created virtual assistant system is superior to similar products offer module in e-shop stores. It gave a little speed in respect of integrated modules, and one store even had a faster rate. The developed system operates autonomously, without administrator intervention, responding to the environment. Proposals are being developed for each individual visitor, what can not be said about the built-in e-shop modules. Furthermore, the system has more features such as the ability to close the assistant box.

Key words: software agents, integration to websites methods, realizing technology, virtual assistant, an electronic store, similar product offers.

SANTRUMPŲ IR ŽYMĖJIMŲ SĄRAŠAS

- Apache** Apache HTTP serveris, svarbus žiniatinklio plėtrai
- CSS** Programavimo kalba (angl. Cascading Style Sheets), skirta nusakyti kita struktūrine kalba aprašyto dokumento vaizdavimą
- DB** Duomenų bazė (angl. Database)
- FTP** Failų perdavimo protokolas (angl. File Transfer Protocol) – standartas failų perdavimui
- HTML** Hiperteksto žymėjimo kalba (angl. Hyper Text Markup Language) – tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete
- HTTP** Pagrindinis metodas pasiekti informaciją pasauliniame tinkle
- JavaScript** Objektiškai orientuota programavimo kalba, besiremianti prototipų principu
- JVM** Java virtualioji mašina (angl. Java virtual machine) – tai aplinka kurioje veikia „Java“ kodų parašytos programos
- MySQL** Duomenų bazių valdymo sistema, dirbanti SQL kalbos pagrindu
- PHP** Hiperteksto interpretatorius (angl. Hypertext Preprocessor) – tai kompiuterinė programavimo kalba, specialiai pritaikyta interneto svetainių kūrimui
- PhpMyAdmin** Valdymo įrankis, skirtas administruoti MySQL duomenų bazes naudojant interneto naršyklę
- URL** Standartas, adresui nurodyti internete (angl. Uniform Resource Locator)
- WEB** Pasaulinis tinklas, žiniatinklis (angl. World Wide Web arba WWW) – interneto dalis, resursai, kuriuos internete galima pasiekti naudojant URL
- XSS** IT sistemų pažeidžiamumas (angl. Cross-site scripting), kai leidžia įterpti papildomą programinį kodą į vartotojų peržiūrimą puslapį

TURINYS

ĮVADAS.....	9
1. PROBLEMOS ANALIZĖ	10
1.1 Programiniai agentai	10
1.1.1 Programinių agentų istorija	10
1.1.2 Programinių agentų apibrėžimas.....	11
1.1.3 Programinių agentų sandara, veikimas	12
1.1.4 Programinių agentų savybės.....	13
1.1.5 Programinių agentų klasifikacija, tipai	14
1.2 Programinių agentų kūrimo priemonės	21
1.3 Programinių agentų integravimo į interneto svetaines būdai	26
1.4 Programinių agentų panaudojimas	29
1.5 Problemos analizės išvados.....	33
2. TYRIMO REIKALAVIMAI	34
2.1 Programinių agentų reikalavimai	34
2.2 Programinių agentų kūrimo būdų reikalavimai.....	34
2.3 Programinių agentų integravimo į interneto svetaines būdų reikalavimai	34
2.4 Programinių agentų panaudojimo reikalavimai	34
2.5 Sistemos reikalavimų specifikacija	35
2.5.1 Projekto varovai	35
2.5.2 Projekto apribojimai.....	36
2.5.3 Panaudojimo atvejų sąrašas	36
2.5.4 Funkcinių reikalavimų sąrašas	38
2.5.5 Nefuncinių reikalavimų sąrašas	41
3. TYRIMO REZULTATAI.....	44
3.1 Programinių agentų tyrimo metodika.....	44
3.2 Programinių agentų kūrimo būdų tyrimo metodika	45
3.3 Programinių agentų integravimo į interneto svetaines būdų tyrimo metodika.....	47
3.4 Programinių agentų panaudojimo tyrimo metodika.....	49
3.5 Sistemos architektūros specifikacija	50
3.5.1 Dokumento paskirtis	50
3.5.2 Architektūros pateikimas	51
3.5.3 Architektūros tikslai ir apribojimai.....	51
3.5.4 Sistemos dinaminis vaizdas.....	51
3.5.5 Virtualaus asistento duomenų bazės vaizdas	53
3.5.6 Komponentų diagrama.....	54

3. 5. 7 <i>Ontologija</i>	56
3. 5. 8 <i>Kokybė</i>	57
3. 6 Realizuotos sistemos testavimas	57
3. 6. 1 <i>Ivadas</i>	57
3. 6. 2 <i>Sistemos funkcijų testavimas</i>	58
3. 6. 3 <i>Sistemos kokybinių charakteristikų testavimas</i>	58
4. TYRIMO ANALIZĖ	59
4. 1 Programinių agentų tyrimo analizė	59
4. 2 Programinių agentų kūrimo būdų analizė	60
4. 3 Programinių agentų integravimo į interneto svetaines būdų analizė.....	61
4. 4 Programinių agentų panaudojimo analizė	62
4. 5 Realizuotos sistemos testavimo analizė	63
IŠVADOS	68
LITERATŪRA.....	69
PRIEDAI.....	72
1 priedas. Duomenų bazės lentelių detalizavimas	72
2 priedas. Funkcijų testavimas.....	75
3 priedas. Našumo testavimas.....	85
4 priedas. Suderinamumo testavimas	89
5 priedas. Saugumo testavimas.....	91
6 priedas. Vartotojų prisijungimo duomenys.....	94
7 priedas. Kompaktinė plokštelė.....	95

IVADAS

Šiais laikais, kai kompiuteriai naudojami labai plačiai, nuolat ieškoma efektyvesnių ir pažangesnių būdų kaip sutaupyti vartotojo laiką, palengvinti jo darbą, geriau išnaudoti kompiuterinę techniką. Beieškant naujų būdų buvo sukurti programiniai agentai, kurie palengvino sudėtingų kompiuterinių programų realizavimą. Kadangi paprastos informacinės sistemos ar pavieniai programiniai agentai orientuoti paprastų užduočių atlikimui, buvo sukurtos informacinės sistemos, į kurias integruojami programiniai agentai, tokiu būdu sistemos tampa intelektualiomis. Iš agentų sudarytos sistemos tapo nauja ir perspektyvi programų inžinerijos ir dirbtinio intelekto šaka.

Programinių agentų sistemos pradėtos naudoti labai plačiai, kadangi sistemos turi galimybę būti panaudotos internete. Tokios sistemos agentai tarpusavyje bendrauja, keičiasi duomenimis ir ne taip kaip įprastos sistemos, po atlikos užduoties gyvuoja toliau ir panaudoja gautus rezultatus ateityje.

Šiuolaikinės el. parduotuvės dažniausiai naudoja kokią nors turinio valdymo sistemą, todėl teoriškai naudoja ir panašias duomenų bazės lenteles (prekės, klientai, užsakymai ir t.t.), todėl galima realizuoti gana universalią virtualių asistentų sistemą. Kuri galėtų aptarnauti kelias el. parduotuves vienu metu, pateikdamas parduotuvėms lankytojams panašių prekių sąrašus pagal tai kokias prekes peržiūrėjo ar užsisakė. Žinoma, turinio valdymo sistemos turi ir savo integruotus panašių ar susijusių prekių siūlymo būdus, tačiau jie dažniausiai gana riboti, prekes siūlo iš tos pačios kategorijos ir t. t.. Todėl reikalingas eksperimentas, lyginantis virtualių asistentų veikimą su integruotais el. parduotuvių moduliais, siekiant išsiaiškinti ar panaudojus programinį agentą būtų pateikiami tikslesni lankytojo pasiūlymai, greičiau ar lėčiau veiktų pati svetainė, kokie kiekvieno būdo privalumai ir trūkumai.

Šio darbo tikslas išanalizuoti programinius agentus, pasirinkti priemones ir realizuoti virtualų asistentą, bei integruoti jį į kelias pasirinktas el. parduotuves, atlikti eksperimentą palyginant sukurtą virtualų asistentą su standartiniais el. parduotuvių prekių siūlymo moduliais.

Darbo užduotys:

- išanalizuoti programinių agentų sąvokas, technologijas ir veikimo principus, savybes, tipus;
- apžvelgti ir pasirinkti technologijas leidžiančias realizuoti ir integruoti programinius agentus į interneto svetaines;
- peržiūrėti agentų panaudojimo atvejus informacinėse sistemose;
- sukurti virtualaus asistento, panaudojant programinį agentą, projektą;
- pasirinkta platforma ir įrankiais realizuoti suprojektuotą virtualų asistentą, sukurti DB;
- integruoti virtualų asistentą į pasirinktas el. parduotuves;
- atlikti eksperimentą ir praktinį virtualaus asistento patikrinimą.

Tyrimo metodai teoriniai tyrimo metodai, rezultatų analizė, eksperimentas vykdomi taikant matematinės aritmetikos metodus, sudedant tyrimo rezultatų įvertinimų taškus, sudauginant taškus su svarbumo procentais, atrenkant didžiausias reikšmes.

Tyrimo objektas programinių agentų kūrimas ir integravimas į informacines sistemas.

Naujumas virtualus asistentas pritaikytas veikti daugumoje el. parduotuvių, turi savo administravimo valdymo skydą. Esant dileliams prekių kiekiams duomenų bazėse, sukurta sistema veikia sparčiau lyginant su standartiniais el. parduotuvių prekių pasiūlymo moduliais.

1. PROBLEMOS ANALIZĖ

1.1 Programiniai agentai

1.1.1 Programinių agentų istorija

Galima teigti, kad programiniai agentai išsivystė iš multiagentinių sistemų (angl. multi-agent systems). Tuo tarpu multiagentinės sistemos buvo sukurtos iš: dirbtinio intelekto (angl. distributed artificial intelligence), paskirstytojo problemų sprendimo (angl. distributed problem solving) ir paralelinio dirbtinio intelekto (angl. parallel artificial intelligence) sričių. Kadangi šios sistemos, daug ką pasiėmė iš išvardintų sričių, programiniai agentai paveldėjo modulinį projektavimo principą, lygiagretaus skaičiavimo greitį, patikimumą, veikimą žinių lygyje, lengvesnę priežiūrą, daugkartinį vartojamumą ir nepriklausomumą. [12, 13]

Ko gero pirmasis programinis agentas buvo „ELIZA“ sukurtas 1966 metais Džozepo Veizenbamo (Joseph Weizenbaum), programa kuri galėjo bendrauti su vartotoju. Panaši programa „SHRDLU“ 1973 metais leido bendrauti su robotu.

Pirmosios šiuolaikinių programinių agentų užuomazgos atsirado 1973 metais, kai Karlas Hevitas (Carl Hewitt) paskelbė veikėjo modelį (angl. actor model). Karlo pateikta koncepcija skelbė apie save valdantį, interaktyvų ir neatsiejamai vykdančią objektą, kurį jis pavadino „ėagentė“. Šis veikėjas (angl. actor) turėjo apribotą vidinę būseną ir galėjo atsiliepti į kitų panašių objektų pranešimus. Tuo metu, veikėjas arba dar vadinamas „agentas“ turėjo pašto adresą ir elgseną, todėl veikėjai bendravo siųsdami žinutes ir palaikė savo veiksmus, veikiant kartu. [1]

1987 metais Marvinas Minskis (Marvin Minsky) pirmą kartą pasiūlė sujungti atskirus programinius agentus į vieną sistemą. Kadangi jo vizija buvo žmogaus smegenų veikimas, kur atskiros smegenų dalys sąveikauja viena su kita.

1995 N. Jennings ir M. Wooldridge paskirstytos atminties (angl. distributed) algoritmai taip pat prisidėjo prie dirbtinio intelekto kūrimo, suteikiant galimybę ribotos atminties ir galimybių agentus sujungti į vieną didelį mechanizmą, kuris turi bendrą tikslą visiems agentams. Toks programinis karkasas (angl. framework) leido lengvai prijungti naujus agentus prie bendro mechanizmo ir panaudoti lygiagrečius skaičiavimus [27].

Kiek vėliau pasirodė BDI (Belief Desire Intention) programinis karkasas išleistas Anando Rao (Anand Rao) ir Mikaelio Džordžefo (Michael Georgeff) kuris turėjo analizavimo įrankius agentinėms sistemoms. Taip pat buvo patobulinta bendradarbiavimo galimybė agentams specialia bendravimo kalba dar žinoma kaip KIF (knowledge interface format).

Įkurta organizacija „FIPA“ (angl. foundation for intelligent physical agents) kūrė standartizavimą visiems kuriamiems agentams, nes garsios firmos kaip „NEC“, „IBM“, „Siemens“, pradėjo kurti savo programinius agentus [27].

Vėliau agentai buvo tobulinami įvairiose laboratorijose, instituteuose. Gauti rezultatai buvo apžvelgiami kasmetinėse programinių agentų konferencijose tokiose kaip „ICMAS“ ir „AGENTS“. Buvo paskelbta, kad viena pagrindinių agentų pasisekimo priežasčių buvo internetas ir paskirstyto atminties sistemos [27].

Agentų kūrimu pardėjo užsiimti garsios informacinių technologijų kompanijos kaip „HP“ (Hewlett-Packard) ir „Microsoft“. Kurios dar labiau pastūmėjo agentų vystimąsi iki šiuolaikinių mums žinomų programinių agentų.

1. 1. 2 Programinių agentų apibrėžimas

Nėra tvirtai susitarta koks tiksliai turėtų būti programinių agentų apibrėžimas. Tačiau šiais laikais vienas dažniausiai naudojamas yra būtent toks: „Agentas yra programa, kuri egzistuoja aplinkoje arba būdama jos dalis, vykdo paruoštą planą stebėdama aplinką ir adekvačiai imdamasi priemonių aplinkai keisti realizuoja planą“.

Tačiau yra ir kitų apibrėžimų, nuo 1990 metų, jų buvo paskelbta nemažai.

Mobiliųjų agentų technologijas apibrėžė Sankaras Virdhagrisvaranas (Sankar Virdhagriswaran) iš kompanijos „Crystaliz, Inc“. Agentui nusakyti buvo sukurti du apibrėžimai: „Agentas veikia savarankiškai. Agentas orientuodamasis veikia aplinkoje“.

Pasirodęs „AIMA“ („Artificial Intelligence: a Modern Approach) agento apibrėžimas 1995 metais buvo naudojamas apie 200 universitetų ir kolegijų. Jis skambėjo taip: „Agentu vadiname, bet ką, kas sugeba suvokti aplinką savo jutikliais ir veikia aplinkoje darydamas poveikį“. Apibrėžimą sudarę mokslininkai buvo vieni pirmųjų dirbantys su dirbtiniu intelektu [1].

Pati Meis (Paltie Maes) pasiūlė kitokį agentų apibrėžimą: „Agentas yra skaičiavimo sistema, skirta realizuoti apibrėžtai aibei tikslų, kurias agentas pasiekia egzistuojamas kompleksiškoje, dinamiškoje aplinkoje veikdamas savarankiškai ir gebėdamas jausti save supančią aplinką“. Šis apibrėžimas detalizuoja aplinką: ji yra kompleksiška, dinamiška [2].

Darbuotojai iš „Apple“ pasiūlė tokį trumpą agentų apibrėžimą: „Agentas – pastoviai veikianti programa skirta specialioms tikslams“. Toks apibrėžimas nurodė skirtumus tarp paprastos kompiuterinės programos, daugiagenčių sistemų ir programinių agentų [3].

1995 metais Barbara Haies (Barbara Hayes) iš Stanfordo „Knowledge System“ laboratorijos pasiūlė tokį agentų apibrėžimą: „Agentas nuolat atlieka tris funkcijas: visada stebi aplinką ir fiksuoja jos būsenas, paveikia aplinkos būseną, inicijuoja veiksmus reaguodamas į aplinkos pokyčius“ [4].

Kompanija „IBM“ taip pat pateikė savo apibrėžimą: „Intelektualus agentas – programa, kuri tam tikru nepriklausomumo laipsniu arba nepriklausomai gali vykdyti apibrėžtą aibę veiksmų vartotojo ar kitos programos naudai suteikdama žinių arba išpildyti vartotojo užduotį“. Toks apibrėžimas primena „informacinius agentus“, išrenkančius informaciją internete.

1995 metais Mikaelis Volbidžas (Michael Woolbridge) ir Nikolas Dženingsas (Nicholas Jennings) paskelbė agentų savybes: autonomiškumo: agentas veikia be tiesioginio žmogaus ar programos įsikišimo, kontroliuoja savo veiksmus, vidinę būseną; bendravimo: agentas sugeba komunikuoti su kitais agentais ir žmogumi tam tikra komunikavimo kalba; reagavimo: agentas stebi savo aplinką ir reaguoja į aplinkos pasikeitimą; pro-aktyvumas: agentas imasi veiksmų aplinkai pakeisti jei to reikia jo tikslui pasiekti [5].

Mikaelis Konas (Michael Coen) iš Masačusetso technologijų instituto dirbtinio intelekto laboratorijos (angl. Massachusetts Institute of Technology AI Lab) pasiūlė tokį agentų apibrėžimą:

„Programinis agentas – programos, kurios naudodamos dialogą, derybas ir koordinaciją perneša informaciją“.

1991 metais Chosé Brustolonis (Jose Brustoloni) agentus apibūdino taip: „Autonominiai agentai – savarankiškai galinčios veikti sistemos kryptingai veikiančios realiame pasaulyje“. [6]

Tedo Selkreio (Ted Selker) 1994 metais: „Agentai – tai kompiuterinės programos, kurios imituoja žmoniškuosius ryšius, darydamos tai, ką kitas žmogus galėtų padaryti už tave“.

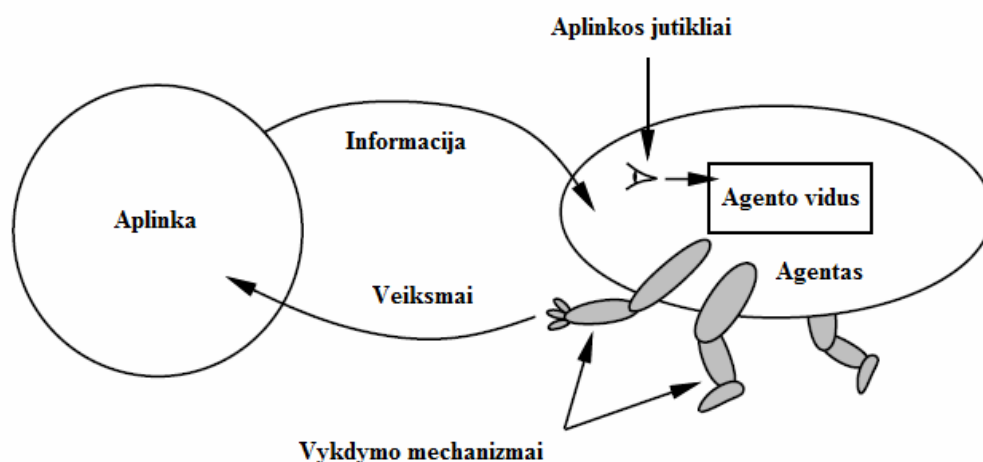
Tais pačiais metais Adrianas Smitas (Adrian Smith) paskelbė kitokį programinių agentų apibrėžimą: „Agentai – nuolat veikiantys programinės įrangos objektai, skirti tam tikram tikslui“.

1996 metais Hiacintas Nvana (Hyacinth Nwana) paskelbė tokį apibrėžimą: „Agentai – programinės įrangos arba techninės įrangos komponentas, kuris sugeba veikti tiksliai siekiant įvykdyti užduotis, kurias jam pateikia vartotojas. [8]

Apibrėžimų būta ir daugiau tačiau galima susidaryti bendrą programinių agentų vaizdą, kad agentai gali išmokyti informaciją, veikti aplinkoje, sąveikauti su kitais agentais, priklausomai nuo nustatytų tikslų.

1. 1. 3 Programinių agentų sandara, veikimas

Programinių agentų rūšių ir sandarų yra įvairių, tačiau visi programiniai agentai turi savo architektūrą, kurioje veikia ir specialią kompiuterinę programą. Kad agentai galėtų tinkamai veikti, jie turi turėti savo jutiklius, aplinkos pasikeitimams pajauti ir vykdymo mechanizmus, aplinkai pakeisti. Pavyzdžiui, agentas stebi įmonės sandėlio vartus, pastebėjęs objektą agentas kamerų pagalba identifikuoja transporto priemonės valstybinius numerius ir vairuotoją pagal veido bruožus. Atpažinęs įmonės darbuotoją ir transporto priemonę įmonės duomenų bazėje agentas siunčia signalą į sandėlių vartų mechanizmą, kad vartai būtų atidaryti. Bendra programinių agentų schema pavaizduota 1 paveiksliuke. [43]



1 pav. Agento sąveika su aplinka [43]

Agentai buvo projektuoti taip, kad atsiradus tam tikroms aplinkybės priimtu teisingiausius sprendimus, todėl gavęs signalą apie pasikeitimus, programinis agentas vadovaujasi nurodytomis instrukcijomis (angl. mapping) ir priima optimaliausią sprendimą. Tačiau agentai naudojami ne tik kūrėjo įvestomis instrukcijomis, agentai gali pasinaudoti ir savo sukauptu patirtimi apie įvykius, priimtus sprendimus ir gautus rezultatus priimant vieną ar kitą sprendimą. Jei agentas neturi

kūrėjo įvestų instrukcijų ir neturi sukauptos savo patirties, pradžioje jis veiksmus vykdo atsitiktinai, tol kol apsimoko.

Agento vidų sudaro architektūra ir programinė įranga. Architektūra, tai įrenginys kuriame veikia programinis agentas, ji gali būti paprastas kompiuteris ar specializuota techninė įranga su specialiais vaizdo ir garso apdorojimo filtrais. Į atmintį diegiama speciali programinė įranga, pasižyminti dirbtiniu intelektu (angl. artificial intelligence). [4]

Pačio paprasčiausio agento programine įrangą sudaro: signalų apdorojimo iš aplinkos algoritmas, taisyklių duomenų bazė, sąlygų ir veiksmų taisyklių (angl. condition-action rule) valdiklis, algoritmas signalui išsiųsti. [26]

Paremti paprasto agento sandara buvo sukurti ir pažangesni modeliai, kurie bet kokiais būdais siekdavo jiems nurodyto tikslo, stebėdavo kaip keičiasi aplinka ir atsimindavo pokyčius. Todėl sujungus visas pažangesnių agentų savybes buvo sukurti modernesni agentai, kaip pavyzdžiui besimokantys (angl. learning). Besimokantys programiniai agentai galėjo ne tik apsimokyti nežinomose aplinkose, tačiau numatyti kas įvyks jei bus priimtas vienas ar kitas sprendimas. Tokie agentai stebėdami aplinką ilgainiui gali pradėti numatyti aplinkos pasikeitimo tendencijas. Pavyzdžiui tokių agentų pagalba dažnai buvo daromos orų prognozių sistemos, sugebančios ganėtinai tiksliai numatyti orus savaitei ar net daugiau. [1, 4]

Besimokantį agentą sudaro tie patys komponentai kaip ir paprasto agento, tačiau šis agentas papildomai turi apsimokymo elementą (angl. learning element), kuris suteikia agentui galimybę dirbti nežinomose aplinkose, vykdymo elementą (angl. performance element), kuris parenka tinkamą veiksmą vykdymui.

Taip pat šis agentas turi savikritikos modulį (angl. critic element), kuris stebi kaip vienas ar kitas priimtas sprendimas paveikė aplinką ir ar buvo sulaukta reikiamų rezultatų. Šis savikritikos modulis kaip ir mokymosi elementas įtakoja kokį galutinį veiksmą pasirinks vykdymo elementas. Dar vienas papildomas elementas tai – problemų generatorius (angl. problem generator), kuris imituoja dar neišbandytas situacijas, eksperimentuoja su galimais atlikti veiksmais ir stebi kokius rezultatus gautų, jei priimtų vienokį ar kitokį sprendimą. Būtent šis elementas ir suteikia agentui galimybę numatyti aplinkos pokyčius į priekį.

Tokie agentai dar vadinami intelektualiniais arba intelektualiniais agentais ir yra labiausiai pažengę, lyginant su kitokios sandaros agentais. Šiuolaikinės intelektualinės informacinės sistemos sujungia keletą intelektualinių agentų norint spręsti sudėtingas užduotis.

Skirtingai nei įprastos kompiuterinės programos programiniai agentai veikia nuolatos. Įprastos programos atlikusios savo užduotį pateikia vartotojui rezultatus ir baigą darbą. O programiniai agentai dirba visada, atlikę tam tikrą užduotį pateikia rezultatus vartotojui ir įsirašo tuos pačius rezultatus ir sprendimo būdą į savo atmintį ir vėl toliau laukia. Kitas skirtumas nuo įprastų programų, kad agentai gali veikti kartu su kitais agentais ir vykdyti skaičiavimus lygiagrečiai [26].

1. 1. 4 Programinių agentų savybės

Programiniai agentai turi daugybę skirtingų apibrėžimų. Tačiau visi programiniai agentai turi panašias savybes, kurias pastebėję galima konstatuoti, kad programa yra būtent programinis agentas.

1998 metais Nikolajus Kasabovas (Nikola Kasabov) apibūdino kokiomis savybėmis dažniausiai pasižymi programiniai agentai [25] :

- Nuolatinis darbas – programiniai agentai visada stengiasi veikti nuolatos, jei tik egzistuoja techninės sąlygos jiems išgyventi.
- Sugebėjimas išmokti – tai viena unikaliausių agentų savybių, agentai per ilgą gyvavimo laikotarpį gali daug pasimokyti ir ateityje priimti geriausius sprendimus vienai ar kitai užduočiai išspręsti.
- Sugebėjimas numatyti – ši savybė susijusi su sugebėjimu išmokti, tačiau ši savybė išsiskiria tuo, kad programiniai agentai ilgai stebėję aplinką gali bandyti numatyti kaip aplinka keisis ateityje.
- Reagavimas – agentai privalo reaguoti į aplinkos pasikeitimus ir imtis priemonių jei to reikalauja jam nustatyti tikslai ir užduotys.
- Autonomiškumas – programiniai agentai savo darbą vykdo be žmogaus įsikišimo, yra savarankiški ir priima reikiamus sprendimus patys.
- Bendravimas – viena svarbesnių savybių, agentai tarpusavyje gali keistis informacija, siųsti užklausas ir nurodymus. Programiniai agentai taip pat gali bendrauti ir su galutiniu vartotoju.
- Mobilumas – agentų galimybė sėkmingai būti perkeltiems į kitas sistemas, bendrauti su vis kitais agentais.
- Savikritika – pastebėję, kad padarė blogą sprendimą agentai priima savo kritiką, išsaugo ją į atmintį ir atsižvelgia
- Sugebėjimas prisitaikyti – agentai patekę į nenumatytas kūrėjo situacijas turi gali sugebėti susitvarkyti su problemomis.

Visomis šiomis savybėmis pasižymintis programinis agentas būtų idealus, tačiau realizuoti tokį agentą tikrovėje yra nepaprastai sudėtinga ir brangu.

Visomis šiomis savybėmis pasižymintis programinis agentas būtų idealus, tačiau realizuoti tokį agentą tikrovėje yra nepaprastai sudėtinga ir brangu. Kita išeitis kurti intelektines informacines sistemas (angl. multi-agent systems), kurios būtų sudarytos iš skirtingomis savybėmis pasižyminčių agentų. Toks sistemos realizavimas taip pat nėra lengvas, tačiau yra pigesnis nei kurti vieną idealų agentą. Tokiu būdu sukurtos sistemos geba spręsti sudėtingas užduotis [11, 25].

1. 1. 5 Programinių agentų klasifikacija, tipai

Brustoloni klasifikacija

1991 metais Chosė Brustoloni (Jose Brustoloni) išskyrė tris agentų klases: savireguliuojantis agentai (angl. regulation agents), planuojantys agentai (angl. planning agents), adaptyvūs agentai (angl. adaptive agents). [6]

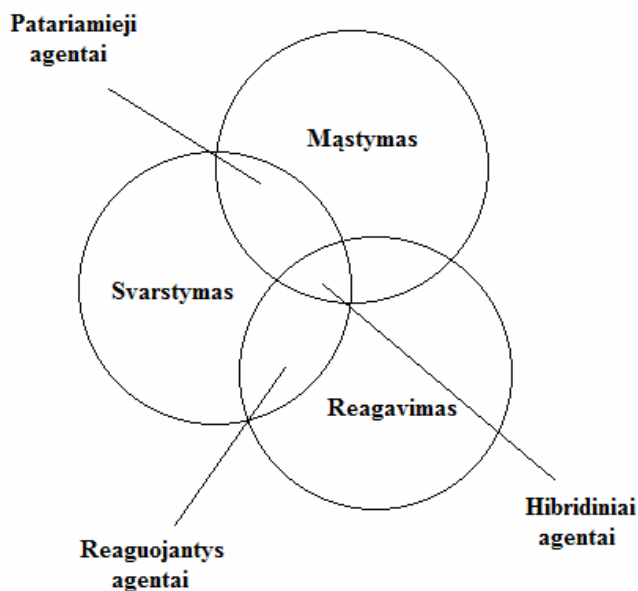
Savireguliuojantis agentas nieko neplanuoja ir neapsimoko, o tiesiog vykdo užduotį, tam tikroje situacijoje. Šie agentai kurti siekiant išvengti darbo sustojimo.

Planuojantis agentas papildomai nei savireguliuojantis gali planuoti, toks agentas turi dirbtino intelekto elementų.

Adaptyvus agentas vykdo tas pačias funkcijas kaip ir savireguliuojantis bei planuojantis agentai, tačiau turi atmintį ir vėliau ja pasinaudoja priimant sprendimus.

Davis klasifikacija

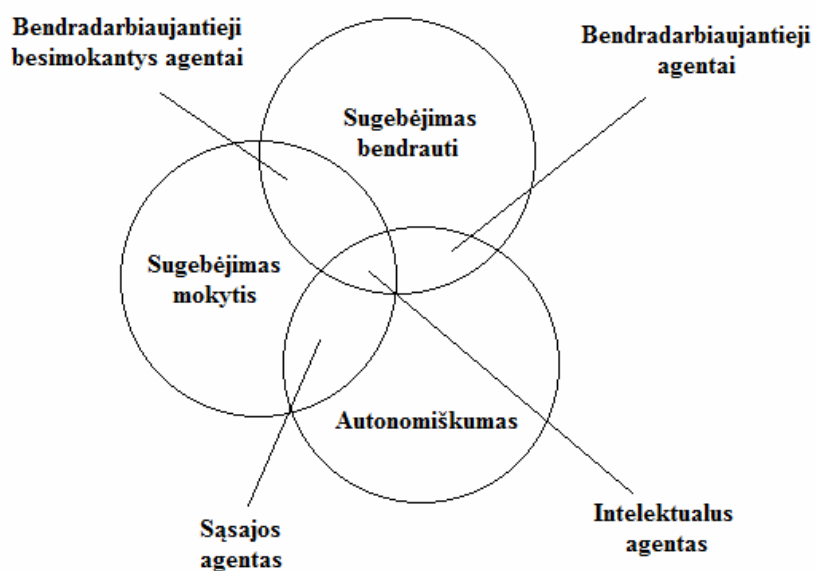
Derelas Davis (Darryl Davis) pasiūlė klasifikuoti agentus pagal jų pagrindines tris savybes: mąstymą, svarstymą ir reagavimą. Sujungus tam tikras agentų savybes D. Davis gavo tris agentų klases: patariamuosius, reaguojančius ir hibridinius agentus. Jo pasiūlyta klasifikacija pavaizduota 2 paveiksliuke. [7]



2 pav. D. Davis pasiūlyta agentų klasifikacija [7]

Nvana klasifikacija

Hiacintas Nvana (Hyacinth Nwana) 1996 metais pasiūlė klasifikuoti agentus pagal jų pagrindines tris savybes: sugebėjimą bendrauti, sugebėjimą mokytis ir autonomiškumą. [8]



3 pav. H. Nwana pasiūlyta agentų klasifikacija [8]

Sujungus tam tikras agentų savybes H. Nwana gavo keturias agentų klases: bendradarbiaujančius besimokančius, bendradarbiaujančius, intelektualius ir sąsajos agentus. H. Nwana pasiūlyta klasifikacija pavaizduota 3 paveiksliuke. [8]

Įvertinant visus klasifikavimo būdus buvo sugalvoti septyni pagrindiniai programinių agentų tipai: bendradarbiaujantis agentas (angl. collaborative agent), sąsajos agentas (angl. interface agent), mobilus agentas (angl. mobile agent), informacinis agentas (angl. informatikon agent), reaguojantis agentas (angl. reactive agent), hibridinis agentas (angl. hybrid agent), intelektualus agentas (angl. smart agent).

Bendradarbiaujantys agentai

Bendradarbiaujantys agentai (angl. collaborative agents) pagal pavadinimą pasižymi bendradarbiavimu su kitais agentais. Norėdamas įvykdyti nurodytas užduotis, agentas stengiasi gauti ir dalintis informacija su kitais sistemoje esančiais agentais. Tačiau dažnai tokiems agentams tenka derėtis tarpusavyje kokius duomenis pateikti ir kada. Šio tipo agentai gali mokytis, bet dažniausiai projektuojant bendradarbiaujančius agentus tokia galimybė nebūna prioritetinga. Pagrindinės šių agentų savybės: autonomiškumas, bendravimo gebėjimas, iniciatyvumas.

Bendradarbiaujančių agentų sistemos pavyzdys galėtų būti „Pleiades“ sistema, sukurta 1994 metais Tomo Mičelo (Tom Mitchell) ir Katios Sykaros (Katie Sycara) „Carnegie Mellon“ universitete. Šio projekto buvo imtasi norint išsiaiškinti kaip veikia automatinė derybų galimybė, pagerinti sistemos efektyvumą, išplečiamumą ir priežiūrą. [7]

Ši sistema padeda vartotojui sukurti studento lankymo grafiką. Architektūra padalinta į tris sluoksnius, darbo agentus, informacinius agentus ir duomenų sluoksnius. Agentai veikia tarpusavyje, kad išsiaiškintų vartotojo mėgstamas vietas, sužinoję kur gyvena vartotojas parenka artimesnes lankymo vietas. [7]

Bendradarbiaujančių agentų privalumai:

- Gebėjimas išspręsti užduotis, kurios yra per sudėtingos vienam programiniui agentui, dėl resursų ribojimo.
- Galimybė sujungti jau egzistuojančias sistemas į vientisą sistemą.
- Padidinamas sistemų patikimumas, gebėjimas surasti įvairius problemos sprendimo būdus.
- Daugkartinis panaudojimas, galimybė tuos pačius agentus panaudojant kelis kartus bendradarbiaujant, tai leidžia sutaupyti.

Tačiau šie agentai nėra labai greiti, taip pat ne taip greitai reaguoja į pakitusią aplinką. Tokių agentų sistemos realizavimas tampa gana sudėtingas, šio tipo agentai nepopuliarūs dinaminėje aplinkoje.

Sąsajos agentai

Sąsajos agentai (angl. interface agents) pasižymi autonomiškumu ir sugebėjimu mokytis. Jų pagrindinis tikslas padėti vartotojui besimokant dirbti kokia nors kompiuterine programa arba palengvinti darbą atliekant už vartotoją pasikartojančius veiksmus. 1994 metais Pati Meis (Pattie Maes) pateikė sąsajos agento apibūdinimą. [14]

Sąsajos agentas stebi vartotojo veiksmus, mokosi ir siūlo geresnius bei greitesnius būdus veiksmams atlikti. Agentas veikia, kaip nepriklausomas asistentas, kuris dirba kartu su vartotoju, siekiant atlikti tam tikrus darbus programoje. Sąsajos agentai mokosi stebėdami ir imituodami vartotoją, gaudami vartotojo atsiliepimus, gaudami tikslias instrukcijas iš vartotojo arba klausdami kitų agentų patarimo. Sąsajos agentų bendradarbiavimas yra apribotas, nevyksta darybos, kaip bendradarbiavimo agentų sistemoje. [14]

Sąsajos agentų privalumai:

- Galimybė apmokyti naujus tam tikros programos naudotojus.
- Sutrumpinamas sugaištas laikas tiems patiems pasikartojantiems vartotojo veiksmams atlikti.
- Palengvinamas programos kūrėjų darbas, integravus sąsajos agentą, mažiau laiko sugaištama rašant smulkias naudojimo instrukcijas.

Tačiau sąsajos agentai veiksmingi tik ten kur yra daug pasikartojančių veiksmų, nes jei nėra daug pasikartojančių veiksmų, agento apsimokymas vyks labai ilgai ir jo efektyvumas bus ženkliai mažesnis. Taip pat agentai nepasiteisina paprastose programose kuriose nėra daug veiksmų, tokiu atveju pigiau ir paprasčiau sukurti paprastą mokymo vedlį ar programėlę.

Mobilieji agentai

Mobilieji agentai (angl. mobile agents) sugeba keliauti po kompiuterinius tinklus. Vartotojui pateikus paieškos užklausą agentas išsiunčiamas į tinklą. Tuo tarpu vartotojas gali netgi atsijungti nuo tinklo, programinis agentas vis tiek toliau tęs informacijos rinkimo darbus. Kai paieškos darbai baigti, agentas grąžina gautus rezultatus vartotojui. Viena iš anksčiausių mobiliųjų agentų programavimo kalbų buvo „Telescript“. Ji palaikė programinių agentų keliavimą po tinklą, tai leido sumažinti tinklo apkrovimą. [24]

Mobilieji agentai veikia savarankiškai, jie taip pat bendrauja tarpusavyje, gali pasidalinti surinkta informacija, nurodyti esamą vietą kompiuteriniame tinkle. [24]

Patekęs užklausą ir kitus mobiliajam agentui reikiamus parametrus vartotojas, pasiunčia agentą klajoti kompiuteriniame tinkle tikslu surasti reikiamos informacijos ar atlikti skaičiavimus. Kol agentas vykdo paiešką vartotojui nereikia būti prisijungus prie kompiuterinio tinklo. Agentas patekęs į serverį duoda užklausą vietiniui serverio agentui, surinkti duomenys siunčiami į centrinį serverį, o agentas keliauja toliau. Kai patikrinami visi serveriai, agentas grįžta į pradinį tašką ir pateikia vartotojui rezultatus.

Mobiliųjų agentų privalumai:

- Sumažinami ryšio kaštai. Naudojant agentus išvengiama bereikalingos informacijos siuntinėjimo iš serverių į kompiuterį.
- Išsprendžiama ribotų vietinių resursų problema. Jei vartotojo kompiuteris yra lėtas, agentas su užduotimi patekęs į galingą serverį gali žymiai greičiau išspręsti uždavinį pasinaudodamas serverio resursais. Galutiniam vartotojui grąžindamas tik skaičiavimo rezultatus.

- Lengvas koordinavimas. Vartotojas gali paleisti keletą mobiliųjų agentų, kurie gali atlikti nepriklausomas užduotis. Gražintus rezultatus apdoroja lokali programa.
- Padidinamas paskirstytos skaičiavimo sistemos lankstumas.
- Suteikiama galimybę vykdyti kelis darbus vienu metu. Vartotojas išsiuntęs mobiliuosius agentus vykdyti užklausos gali užsiimti kitais darbais.

Tačiau mobilieji agentai nėra labai saugūs, agentai keliaudami po kompiuterinius tinklus gali susidurti su agentais sukurtais iš piktavališkų paskatų. Tokiais atvejais agentas gali grąžinti ne tik klaidingus rezultatus tačiau atsiųsti ir kenkėjiško kodo į vartotojo kompiuterį.

Informaciniai agentai

Informaciniai agentai (angl. information agents) valdo, tvarko ir sugretina informaciją iš įvairių šaltinių. Šie agentai labai panašūs ir į bendradarbiaujančius ir sąsajos agentus. Skirtumas tas, kad šie agentai daugiau naudojami internete ir jų pagrindinė užduotis yra darbas su informacija. Informacinių agentų rūšių yra nemažai, vieni agentai nebendradarbiauja, kiti atsimena vykdytas paieškas ir apsimoko, kiti atvirksčiai. Kai kurie informaciniai agentai gali klajoti po tinklą kaip ir mobilieji agentai. Orenas Etzionis (Oren Etzioni) ir Danielius Veldas (Daniel Weld) yra aprašę šiuolaikinį informacinį agentą pavadinimu „Softbot“.

Informacinis agentas dar vadinamas internetiniu pasinaudodamas interneto naršyklės naršymo įrankiais tokiais kaip vorais renka informaciją. Šioje scheme informacinis agentas susiejamas su voru, kuris sugeba surūšiuoti internetinius puslapius ir išsaugoti duomenų bazėje. Vartotojo informacinis agentas, kuris skirtas palyginti tam tikros srities informaciją, paleidžia užklausas pasinaudodamas viena ar keliomis paieškos platformomis, bei vorais, kad susidoroti su vartotojo užduotimi. Informacinis agentas paiešką vykdo ir lokaliaje atmintyje, visą surinktą informaciją apdoroja, surūšiuoja ir pateikia galutiniam vartotojui.

Informacinių agentų privalumai:

- Galimybė dideliame kiekyje informacijos greitai rasti dominančią informaciją.
- Turimos ar rastos informacijos susisteminamas, suglaudindimas ir filtravimas, kad vartotojas gautų kuo tikslesnius duomenis.

Kadangi šie agentai labai panašūs į mobiliuosius ir bendradarbiavimo agentus jie pasižymi tais pačiais trūkumais.

Reaguojantys agentai

Reaguojantys agentai (angl. reactive agents) neturi sugebėjimo planuoti, jie sprendimus priiminėja atsižvelgdami į kūrėjo įvestas galimas situacijas ir sprendimo būdus. Šių agentų pagrindinis bruožas – paprastumas ir lengvas bendravimas su kitais agentais. Reaguojantys agentai pavieniui nepasižymi dirbtiniu intelektu, nes neapsimoko ir nekonvertuoja jutiklių signalų, tačiau jie dažnai jungiami į sistemas, kurios sprendžia gana sudėtingas užduotis. [8]

Reaguojančių agentų privalumai:

- Pigus ir nesudėtingas sistemų realizavimas. Paprastas realizavimas ženkliai sumažina agentų kūrimo kaštus.

- Sutrumpinamas reakcijos laikas. Kadangi agentams nereikia gaišti laiko jutiklių pateiktiems signalams konvertuoti ir nereikia sugalvoti naujo veiksmo, sprendimus agentai gali priimti labai greitai.
- Tolerantiškumas klaidoms. Sugedus vienam reaguojančiam agentui sistemoje, sistema funkcionuos toliau, tik su tam tikrais apribojimais.

Tačiau tokie agentai labai jautrūs aplinkos pasikeitimams, jei susidariusi situacija neaprašyta agento elgsenos modulyje, agentas nežinos kokius veiksmus atlikti, todėl visos sistemos darbas gali sutrikti.

Hibridiniai agentai

Hibridiniai agentai (angl. hybrid agents) savyje apjungia dviejų ar daugiau kitų agentų tipų savybes. Hibridiniai agentai buvo projektuoti tam, kad skirtingais atvejais geriau turėti daugiau agentų savybių viename agente. Tokių agentų elgsenos priklauso nuo pavaldėtų kitų agentų savybių ir kokios savybės numatytos kaip svarbiausios. Hibridiniame agente agentų savybės sujungiamos sluoksniuota struktūra. Kiekviena savybė ar elgsenos modelis hibridinio agento struktūroje sudaro vieną sluoksnį. 1995 metais vokiečių dirbtinio intelekto tyrimų centre buvo sukurta „InteRRaP“ hibridinių agentų architektūra. [8, 13]

„InteRRaP“ architektūrą sudaro trys valdymo sluoksniai išdėstyti vertikaliai. Žemiausias lygis, elgsenos sluoksnis, atsakingas už galutinio veiksmo parinkimą. Planavimo sluoksnis, atsakingas už kasdienių veiksmų kontrolę atliekant užduotį ar siekiant tikslo. Bendravimo sluoksnis, atsakingas už bendravimą su kitais agentais. Kiekvienas sluoksnis turi ryšį su sau pritaikyta atitinkama žinių baze, kuriose kaupia patirtį. Išoriniame modelis dalyje, gali būti tik aplinkos stebėjimo duomenys vidiniame - veiksmų planai, bendravimo – informacija apie kitus agentus, jų atliekamas užduotis. Šioje architektūroje sluoksniai sąveikauja vieni su kitais siekiant bendro galutinio tikslo. [8]

Hibridinių agentų privalumai:

- Sumažinami sistemos perkėlimo, pritaikymo prie kitų sistemų kaštai.
- Galimybė aiškiau pamatyti sistemos veikimą, išskaidant agentus į tam tikrus sluoksnius.
- Padidinamas sistemos universalumas, integruojant kelis agentus į vieną yra didesnė tikimybė, kad agentas vis tiek priims tinkamą sprendimą nors veikimo aplinka jam yra nežinoma.

Tačiau tokių agentų pagrindinė problema sudėtingas sąryšių tarp sluoksnių realizavimas.

Intelektualūs agentai

Intelektualūs agentai (angl. smart agents) – teoriškai idealūs agentai, pasižymintys visomis agentų savybėmis, tačiau tokio agento realizavimas tikrovėje labai sudėtingas ir ne visada tikslingas. Pigiau ir paprasčiau atsieina sujungti atskirus agentus į sistemas. [12]

Apžvelgus visu agentų tipus galima sudaryti agentų palygimo lentelę:

2 lentelė. Programinių agentų palyginimo lentelė

Agento tipas ir savybės	Bendradarbiaujantis agentas	Sąsajos agentas	Mobilus agentas	Informacinis agentas	Reaguojantis agentas	Hibridinis agentas	Intelektualus agentas
Autonomiškumas	Taip	Taip	Taip	Taip	Taip	Taip	Taip
Bendravimas su kitais agentais	Taip	Ne	Taip	Taip	Taip	Taip	Taip
Sparta reaguojant į pakitimus	Lėta, nes reikia kreiptis į kitus agentus	Greita	Lėta, reikia iš naujo paleisti agentą	Lėta	Labai greita	Lėta	Lėta
Gebėjimas mokytis	Ne	Taip	Ne	Taip	Ne	Taip	Taip
Galimybė rinkti informaciją internete	Ne	Ne	Taip	Taip	Ne	Taip	Taip
Saugumas	Bendradarbiaujant su kitais agentais atsiranda grėsmių	Patikimas	Keliaujant po tinklą atsiranda grėsmių	Keliaujant po tinklą atsiranda grėsmių	Patikimas	Patikimas	Patikimas
Rezultatų patikimumas	Aukštas	Aukštas	Aukštas	Aukštas	Žemas	Aukštas	Labai aukštas
Tolerantiškumas klaidoms	Įvykus klaidai kreipiamasi į kitą agentą	Įvykus klaidai nustoja veikti	Įvykus klaidai kreipiamasi į kitą agentą	Įvykus klaidai dirbama toliau	Įvykus klaidai kreipiamasi į kitą agentą	Įvykus klaidai kreipiamasi į kitą agentą	Įvykus klaidai kreipiamasi į kitą agentą
Realizavimo sudėtingumas	Sudėtingas	Vidutinis	Sudėtingas	Vidutinis	Vidutinis	Sudėtingas	Labai sudėtingas
Efektyviausia panaudojimo sritis	Ribotų resursų sistemose sprendžiant sudėtingas užduotis	Sistemose kur yra daug pasikartojančių veiksmų	Rinkti informaciją internete	Rinkti informaciją internete	Vykdyti paprastą budėjimą	Sudėtinguose sistemose	Sudėtinguose sistemose
Ryšiausio privalumai	Gebėjimas išspręsti	Automatizuojami	Mobilumas,	Informacijos	Nesudėtingas	Didelis sistemos	Turi visas

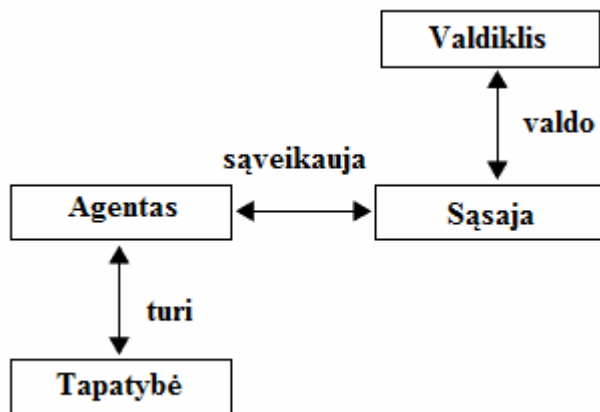
	užduotis, kurios yra sudėtingos vienam agentui	pakartojantys vartotojo veiksmai	galimybė vykdyti kelis darbus vienu metu	susistemintas, filtravimas, greita paieška	sistemų realizavimas, geras reakcijos laikas	universalumas, priimami geresni sprendimai nežinomoje aplinkose	agentų savybės
Ryškesniausi trūkumai	Sudėtingas realizavimas, lėtas reagavimas į pokyčius	Veiksmingai tik ten kur yra daug pasikartojančių veiksmų	Keliaujant po tinklą atsiranda grėsmių	Keliaujant po tinklą atsiranda grėsmių	Įvykus nenumatytam veiksmui gaunami netinkami rezultatai	Sudėtingas sąryšių tarp sluoksnių realizavimas	Realizavimas labai sudėtingas ir ne visada tikslingas

1.2 Programinių agentų kūrimo priemonės

Kuriant programinius agentus svarbu teisingai numatyti jo paskirtį, jam užduodamus tikslus. Tada pasirinkti programines priemones ir programavimo kalbą agentams realizuoti.

Kuriant paprastus agentus, galima naudoti specialiai pritaikytas klases objektiškai orientuotose programavimo kalbose. Dažniausiai naudojamos „Java“ ar „C++“ programavimo kalbos.

Pavyzdžiui, kuriant agentą nenaudojant specializuotų programavimo kalbų ar sistemų, visą kodą tektų rašyti nuo nulio, taip pat reiktų sukurti agento klasių architektūrą. „Java“ programavimo kalboje agento klasių architektūra pavaizduota 4 paveiksliuke. [23]



4 pav. Agento klasių architektūra „Java“ programavimo kalboje [23]

Valdiklio klasė nusako visus agentus sistemoje, valdiklis reikalingas bendradarbiaujant su kitais agentais (kitais sistemų valdikliais). Agento klasė aprašo patį agentą, jo veikimą, atliekamas funkcijas ir priimamus sprendimus. Sąsajos klasėje aprašoma agento valdymo sąsaja, nurodomi laukeliai, langai ir t.t. Taip pat kiekviena agento klasė privalo turėti unikalią tapatybę, kuri aprašyta tapatybės klasėje. Jos pagalba agentai nustato ar verta bendradarbiauti su kitu agentu, ar pavyzdžiui kitas agentas taip pat ieško informacijos internete ar tik apdoroja ją.

Nesvarbu, kuri programavimo kalba pasirenkama, daug ką lemia architektūra, pagal kurią realizuojama sistema. Kuriant sudėtingas sistemas naudojami vidiniai agentai, siekiant sistemas padaryti kiek įmanoma lankstesnes, nes agentai turi būti dinamiškai įdedami, pašalinami arba pakeičiami kai to reikalauja užduotys. Tokia sistemos architektūra vadinama paskirstyta arba centralizuota.

Sudėtingoms sistemoms rekomenduojama naudoti žiedo architektūrą, kuri padeda išspręsti daugybę problemų, kylančių naudojant centralizuotą arba paskirstytą architektūros metodą. Tokioje architektūroje naudojami keli vidiniai agentai, kurių kiekvienas yra atsakingas už nepriklausomas programas. Šioje platformoje agentai yra suskirstomi į tris kategorijas: vidiniai agentai (angl. middle agents), atsakingi už dinamišką paslaugų agentų bei programų komandinių agentų registraciją bei atšaukimą, paslaugų tiekimo agentai (angl. service provider agents), programų komandiniai užklauso agentai (angl. application team-based requester agents).

Tačiau atsiranda problema, kad kuriant agentus nusilaikant standartų, ateityje sunku sujungti tokius agentus į bendras sistemas ir suteikti galimybę jiems bendradarbiauti. Todėl tam tikslui buvo kuriamos programavimo kalbų platformos, palengvinančios agentų ir jų sistemų realizavimą. Vienos garsesnių sistemų ir programavimo kalbų: „DAML“, „Jason“, „3APL“, „Jack“, „Jade“, „GOAL“, „OWL“, „JAT“.

„DAML“ programavimo kalba

Tai viena pirmųjų specializuotų programavimo kalbų skirtų būtent agentams. „DARPA Agent Markup Language“ (DAML) pavadinimas kilo iš JAV projekto vykdyto „US Defense Advanced Research Projects Agency“ agentūroje, kuris buvo pradėtas 1999 metais. Pagrindinis projekto iniciatorius Džeimas Hendleris (James Hendler). Programos tikslas buvo sukurti mašinių skaitomus dokumentus internete.

„DAML“ programa pirmiausiai turėjo savą programavimo kalbą tokiu pačiu pavadinimu „DAML“. Ši programavimo kalba buvo paremta „RDF“ (Resource Description Framework) karkasu. Vėliau 2002 metais ši programavimo kalba buvo patobulinta ir vadinosi „DAML+OIL“, kalba buvo atnaujinta įtraukiant OIL (Ontology Interchange Language) programavimo kalbą. [18]

Tačiau šis projektas buvo nutrauktas 2006 metais, jo vietą užėmė nauja programavimo kalba „Web Ontology Language“ (OWL), kuri iš esmės daug ką paveldėjo ir iš „DAML“ kalbos. [18, 22]

„Jason“ agentų kūrimo platforma

„Jason“ – tai agentų ir jų sistemų kūrimo platforma paremta „Java“ programavimo kalba. „Jason“ naudoja savo programavimo kalbą, kuri vadinasi „AgentSpeak“, ji yra orientuota būtent į agentus. Taip pat „Jason“ turi pildomų įrankių, kaip minčių inspektorius (angl. mind inspector), kurie palengvina agentų kūrimą ir klaidų ieškojimą.

Šios platformos pagrindiniai privalumai:

- Leidžia sukurti stipriai bendradarbiaujančius agentus.
- Pagalba kuriant agentų sprendimo funkcijas ir sukurta biblioteka pagrindinių vidinių veiksmų.
- Galimybė kurti ir tobulinti aplinkas.

- Didelių agentinių sistemų palaikymas.
- Galimybė paleisti agentų sistemą nutolusiame tinkle.
- Pilnas architektūros, funkcijų, klasių redagavimas.
- Papildomi įrankiai kaip „Mind Inspector“, padedantys kurti ir rasti klaidas agentuose.

Tačiau ši platforma daugiausiai orientuota į reaguojančius agentus, todėl norint sukurti kitokio tipo agentą iškyla sunkumų. Taip pat šios programos agentai orientuoti veikimui kliento pusėje. Programavimo kalba „Java“ reikalauja vartotojo kompiuteryje įdiegtos specialios programinės įrangos, norint pasileisti agentą. [21]

„3APL“ agentų programavimo kalba

Ši agentų programavimo kalba buvo sukurta Olandijoje, Utrechto universitete (University of Utrecht). Pavadinimas „An Abstract Agent Programming Language“ arba „Artificial Autonomous Agents Programming Language“ sutrumpintai „3APL“. Ši agentams skirta kalba turi ir specialius įrankius paremtus, „Java“ programavimo kalba, agentų kūrimui. Platformos langas pateiktas 15 paveiksliuke.

Šios platformos pagrindiniai privalumai:

- Leidžia sukurti besimokančius ar mobilius agentus.
- Programavimo kalba ganėtinai paprasta.
- Turi priedus, tokius kaip klaidų ieškiklis ir aplinkos modulis.
- Didelių agentinių sistemų palaikymas.

Tačiau šios platformos agentai sunkiai pritaikomi praktikoje, dėl agentų naudojamų kombinuotų taisyklių ir nenuoseklumo. Platforma daugiausiai orientuota į reaguojančius agentus, todėl norint sukurti kitokio tipo agentą iškyla sunkumų. Taip pat šios programos agentus sudėtinga integruoti į interneto svetaines, nes pagrindinis agento formatas yra specifinis, kurį atidaro tik pati programa. Programavimo kalba „Java“ reikalauja vartotojo kompiuteryje įdiegtos specialios programinės įrangos, norint pasileisti agentą. [20]

„GOAL“ agentų programavimo kalba

„GOAL“ programavimo kalba skirta programiniams agentams, kurie priima sprendimus atsižvelgiant į užduotus tikslus, sukurti. Ši programavimo kalba suteikia pagrindinius agentų kūrimo rėmus. Taip pat padeda kūrėjui įterpdamas programavimo kalbos kodus pagal pasirinktus parametrus, todėl vartotojui užtenka paviršutiniškai išmanyti programavimo kalbą kuriant paprastus agentus. Platforma ir programos veikimas paremtas platforma paremta „Java“ programavimo kalba. Platformos langas pateiktas 16 paveiksliuke.

Šios platformos pagrindiniai privalumai:

- Agentai gali turėti kelis tikslus ar užduotis. Orientuojamasi ne į taip kaip jis pasieks tikslą, o tikslus ir aplinką.
- Agentai nustoja siekti tikslo kai jį visiškai įvykdo.
- Platforma turi integruotų uždavinių sprendimo būdų, todėl kūrėjui kartais užtenka nurodyti tik kokius tikslus norima pasiekti ir agentas tai įvykdo.
- Sukurti agentai bendrauja ir keičiasi informacija tarpusavyje.

Tačiau tokių platformų agentai remiasi veiksmų ir užduočių duomenų baze, todėl kūrėjui tenka įvedinėti nemažai užduočių sprendimo būdų. Programavimo kalba „Java“ reikalauja vartotojo kompiuteryje įdiegtos specialios programinės įrangos, norint pasileisti agentą. [28]

„Weka“ rinkinys

„Weka“ rinkinys skirtas kurti besimokančias mašinas. Pagrindinis „Weka“ tikslas duomenų surinkimas ir jų analizavimas, tačiau jis taip pat tinka besimokančių agentų kūrimui. Šis rinkinys turi specialius įrankius skirtus duomenų surinkimui ir analizavimui. „Weka“ naudoja „Java“ programavimo kalbą, agentų kūrimui. Programos langas pateiktas 17 paveiksliuke.

Šio rinkinio pagrindiniai privalumai:

- Rinkinys turi nemažai integruotų duomenų rinkimo ir apdorojimo technologijų.
- Galimybė tiesiogiai susijungti su SQL duomenų bazėmis.
- Galimybė sukurti visiškai nepriklausomas programas, o jas integruoti ir į svetaines.
- Visa technologija yra atviro kodo.

Tačiau šis rinkinys yra ribotas dėl programų kūrimo savo grafine sąsaja, be to šiuos agentus galima atidaryti tik su pačia programa. [29]

„JACK“ agentų kūrimo platforma

„JACK“ agentų kūrimo platforma paremta „Java“ programavimo kalba, skirta programinių agentų sistemų kūrimui. Ši platforma yra tobulinama jau virš 10 metų, yra nuolat atnaujinama.

Šios platformos pagrindiniai privalumai:

- Galima sukurti įvairių tipų agentų sistemas, suteikti galimybę jiems mokytis.
- Platforma turi nemažai integruotų taisyklių, įvykių, numatytų sprendimo būdų.
- Sukurti agentai bendrauja ir keičiasi informacija tarpusavyje.

Tačiau ši platforma yra komercinė ir yra mokama. [30]

„Jade“ agentų kūrimo platforma

„Jade“ agentų kūrimo platforma paremta „Java“ programavimo kalba, skirta programinių agentų kūrimui. Agentai kuriami „Java“ ir „XML“, kalbomis, kurie atitinka „FIPA“ standartus.

Pagrindiniai „Jade“ privalumai:

- Sistema yra nemokama.
- Jau yra paruoštas agentų veikimo taisyklių rinkinys.
- Galimybė sujungti kelis agentus, taip sukuriant daugiagentes sistemas.
- Papildomi įrankiai tokie kaip: „NBJAD“, leidžiantis kurti agentus „NetBeans“ aplinkoje.

Norint paleisti reikalinga „IDE“ (integrated development environment) sąsaja, pvz.: „Eclipse“, „NetBeans“. [31]

„OWL“ ontologijos kalba

„OWL“ (Web Ontology Language), ontologijos kalba skirta sukurti mašinų ir žmonių skaitomus dokumentus internete. Ši kalba pastūmėjo „XML“ formatų atsiradimą. Susisieti šią kalbą su programiniais agentais galime dėl galimybės sukurti informacinius agentus, ieškančius tam tikros

informacijos internete. Reikia paminėti, kad ši programavimo kalba daug paveldėjo ir iš „DAML“ platformos.

„OWL“ kalba neturi specializuotų agentų kūrimo įrankių, todėl pats kūrėjas pasirenka būdą kurti informacinius agentus. Tačiau toks būdas yra gana sudėtingas, be to kuriant tokį agentą reikia laikytis numatytų standartų „Rule Interchange Format“ (RIF). [9, 32]

„JAT“ šablonai

„JAT“ (Java Agent Template), tai specialūs šablonai paruošti agentams kurti, kurie gali bendrauti p2p (peer-to-peer) tinkle internetu. Šie šablonai parašyti „Java“ programavimo kalba. Sukurti agentai šiais šablonais nėra mobilieji, o statiški. Agentų bendravimo žinutės parašytos „KQML“ protokolu. Tokie agentai gali būti sukurti kaip atskiros programos ir turėti savo grafinę sąsają. [10]

Tačiau šis projektas buvo nutrauktas, agentų šablonai nebetobulinami. [10, 33]

Apžvelgus visas agentų kūrimo platformas ir įrankius galima sudaryti tokią palyginimo lentelę:

3 lentelė. Programinių agentų kūrimo įrankių ir platformų palyginimo lentelė

Agentų kūrimo priemonės ir savybės	„Jason“	„3APL“	„GOAL“	„Weka“	„Jade“	„Jack“
Įrankio programavimo kalba	„Java“	„Java“	„Java“	„Java“	„Java“	„Java“
Kuriamo agento programavimo kalba	„AgentSpeak“	„3APL“	„GOAL“	„Java“	„XML“, „Java“	„Java“
Standartų palaikymas	„KQML“	„FIPA“	Nėra	Nėra	„FIPA“	„FIPA“
Licenzija	„GNU Lesser General Public License“	„GNU Lesser General Public License“	„GNU Lesser General Public License“	„GNU General Public License“	„GNU Lesser General Public License“	Komercinė
Papildomos programinės įrangos reikalavimai	„Java Development Kit“	„Java Runtime Environment“	„Java Runtime Environment“	Nereikia	Interpretatorius „Eclipse“, „Java Runtime Environment“	„Java Development Kit“
Palaikomi bylų formatai	.mas2j, .java, .asl	.3apl	.mas2g, .goal	.arff	.java, .jar, .xml, .jsp	.java, .jar, .prj, .jsp
Išsaugotų bylų atidarymas su kitomis	Pagrindinis bylos failas .mas2j nepalaikomas	Visi programos formatai nepalaikomi	Visi programos formatai nepalaikomi	Visi programos formatai nepalaikomi	Visi programos failai palaikomi	Pagrindinis bylos failas .prj nepalaikomas

programomis	kitų programų	i kitų programų	mi kitų programų	kitų programų	daugelio programų	mas kitų programų
Papildomi įrankiai	„JEdit“, „Mind inspector“	Nėra	Nėra	Weka “Experimenter”	„NBJAD“, kiti įrankiai skirti “NetBeans”	„JACK WebBot“, „JACK Eclipse Plug-in“
Dokumentacija, pavyzdžiai	Taip, yra	Taip, riboti	Taip, yra	Ribota, yra	Taip, yra	Taip, yra
Ryškiausi privalumai	Leidžia sukurti stipriai bendradarbiaujančius reaguojančius agentus	Leidžia sukurti besimokančius, mobilius agentus	Turi integruotų uždavinių sprendimo būdų, agentai bendrauja	Atviro kodo, nemažai integruotų duomenų rinkimo ir apdorojimo technologijų	Agentų veikimo taisyklių rinkinys, priedai	Turi numatytų taisyklių, sprendimo būdų. agentai bendrauja
Ryškiausi trūkumai	Norint sukurti kitokio tipo agentą išskyla sunkumų, sistema paremta agentų kūrimui kliento pusėje	Agentai sunkiai pritaikomi praktikoje	Agentai remiasi veiksmų ir uždavinių duomenų baze	Ribotas dėl agentų kūrimo savo grafine sąsaja	Norint paleisti sistemą reikalinga papildoma programinė įranga	Platforma yra komercinė

1.3 Programinių agentų integravimo į interneto svetaines būdai

Kadangi internetas šiais laikais labai svarbus, todėl taip pat yra svarbu, kad programiniai agentai galėtų būti integruojami į interneto svetaines. Daugiausia interneto svetainių yra sukurtos „PHP“ (Hypertext Preprocessor) ir „HTML“ (Hyper text Markup Language) programavimo kalbomis, o daugiausia programinių agentų kūrimo įrankių ir platformų paremtos „Java“ programavimo kalba. Atsižvelgiant į tai, kad visi agentų realizavimo būdai realizuoja agentus „Java“ programavimo kalba, apžvelgiamos programinių agentų integravimo į interneto svetainių galimas priemonės.

„Java applet“

Norint įterpti „Java“ programavimo kalbos kodą į interneto svetainę dažnai originalus kodas perverčiamas į „Java applet“ programėlę, kuri gali funkcionuoti interneto svetainėje. Ši programėlė gali būti aktyvuojama „HTML“ kode nurodant bylos pavadinimą ir klasę. Tačiau toks „Java“ programavimo kalbos panaudojimas reikalauja, kad interneto svetainės lankytojai savo interneto naršyklėse turėtų įdiegtą „Java Virtual Machine“ arba „AppletViewer“ programinę įrangą, be kurios „Java applet“ neveiktų.

„Applet“ programėlė veikia kliento pusėje (angl. client-side), o tai leidžia mažiau užkrauti serverio darbą. Šią technologiją palaiko visos grasiausios operacinės sistemos ir interneto naršyklės, galima sujungti programėlę su duomenų bazėmis. „Java applet“ veikia gana sparčiai, be to naršyklės įsimena programėlę savo atmintyje (angl. cache) ir pakartotiniai pakrovimai užtrunka žymiai trumpiau. Reikia nepamiršti ir tai, kad šis būdas neveikia serverio pusėje, tai didelis minusas sistemai.

Tačiau šis būdas turi ir trūkumų. Svarbiausias trūkumas, kad ši technologija neleidžia agentui veikti serverio pusėje (angl. server side), todėl šis trūkumas labai svarbus, kadangi agentas negalės funkcionuoti autonomiškai. Taip pat specialios programinės įrangos reikalavimas lankytojams, tai sukelia problemų įstaigų ir ofisų darbuotojams, mat dažniausiai programinę įrangą gali įdiegti tik sistemos administratorius. Taip pat jei kliento įdiegta programinė yra senesnė nei reikalauja programėlė, tenka ilgai laukti svetainės užsikrovimo. Tokios programėlės sumažina saugumą, kadangi dažnai programėlės reikalauja leidimo veikti svetainės lankytojo kompiuteryje, kad galėtų pilnai funkcionuoti. [32, 33]

„Common Gateway Interface“

Kitas būdas integruoti programinį agentą į interneto svetainės tai CGI (Common Gateway Interface) protokolo panaudojimas kuris, nurodo, kaip turi bendrauti interneto serveris ir jo vykdomos programos. Šis protokolas apdoroja perduotus duomenis, nusiunčia juos programai realizuotai „Java“, „C++“, „Perl“ ar „Visual Basic“ programavimo kalbomis ir vėliau gražina programos atsakymą svetainės lankytojui.

Pagrindiniai šio įterpimo būdo privalumai: nėra reikalinga papildoma programinė įranga svetainės lankytojams, užtenka interneto naršyklės. CGI palaikomas programos galima sujungti su duomenų bazėmis. Šis yra protokolas palaikomas populiariausių operacinių sistemų ir interneto naršyklių, tik reikalinga, kad CGI būtų įdiegtas svetainės serveryje. Tokių programinių agentų perkėlimas patogus tarp serverių su įdiegtu CGI protokolu, kadangi nereikia daug integruoti kodo į svetainės bylas. Galima naudoti kitokias programos ir svetainės bylų teises taip padidinant saugumą.

Kadangi kiekviena užklausa yra apdorojama atskiro proceso ir visi veiksmai vykdomi serverio pusėje (angl. server-side), susiduriama su serverio resursų išnaudojimo problema. Ne visi serveriai įdiegia šį protokolą. Taip pat rašant pačią programą tenka atsižvelgti į „Common Gateway Interface“ protokolo reikalavimus. [34, 35]

„Apache Tomcat“

„Java“ programavimo kalbos klasė „Java Servlet“ leidžia išplėsti žiniatinklių serverio galimybes. Naudojant vieną žinomesnių „Apache Tomcat“ žiniatinklio serverį galima integruoti programinį agentą į interneto svetainės. Šis serveris palaiko „Java Servlet“ ir „JavaServer Pages“ (JSP) – technologija leidžianti gauti duomenis iš dinamiškai generuojamų puslapių naudojant „Java“ kalbą.

Svarbu, kad serveris yra atvirko kodo ir jį galima naudoti nemokamai. Jis užima mažai kietojo disko vietos ir yra lengvai suderinamas su „Apache“ žiniatinklio serveriu. „Apache Tomcat“ leidžia „Java“ programavimo kalba parašytoms programoms funkcionuoti svetainėje. Kaip ir kitos priemonės ši taip pat leidžia „Java“ susijungti su duomenų bazėmis.

Tačiau „Apache Tomcat“ serveris yra labai reiklus serverio resursams, ypač jo startavimo metu, todėl nėra dažnai pasitaikantis serveriuose. Stabilumu serveris pasižymi tik „Unix“ operacinėse sistemose. [36, 37]

„Quercus“

Technologija „Quercus“ taip pat leidžia panaudoti „Java“ programas internete. Ši technologija naudojama „Resin“ serveryje. O pats „Resin“ serveris jau turi ir integruotą „PHP“ interpretatorių.

Tokiu būdu sukuriama nauja architektūra, leidžianti glaudžiai bendradarbiauti PHP ir „Java“ programavimo kalboms.

„Quercus“ technologija užtikrina didelę serverio spartą, turi galimybę susijungti su populiariausiomis duomenų bazėmis. Tokiame serveryje „PHP“ byla gali laisvai prieiti „Java“ bibliotekas, kreiptis į klases. Nauja architektūra padidina serverio saugumą, jame išspręsta standartinių „PHP“ interpretatorių spragos. Be to „Quercus“ yra atviro kodo programinė įranga, todėl ją galima naudoti nemokamai.

Vienintelė problema, kad standartinės interneto svetainės „PHP“ kodas turi būti perrašytas į reikalaujamą „Quercus“ technologijos. Todėl perdaryti didelę svetainę gali užimti daug laiko ir pastangų. [38]

„PHP/Java Bridge“

Šis „PHP/Java“ tiltas – „JJE“ (Java Platform, Enterprise Edition) žiniatinklio aplikacija, suteikianti galimybę serveryje vykdyti „Java“ programas. Tiltas paremtas „XML“ (Extensible Markup Language) tinklo protokolu, kuris panaudojamas susiejant skirtingas kalbas.

„PHP/Java Bridge“ veikia sparčiai ir neužkrauna daug serverio resursų. Šis tiltas taip pat yra atviro kodo. Tačiau priešingai nei „Quercus“ technologija nereikalauja keisti interneto svetainės „PHP“ bylų kodo.

Norint naudoti šį tiltą reikalingas specialus „Java“ servisas (angl. Java service), pavyzdžiui jau minėtas „Apache Tomcat“. Iškyla problemų su kai kuriomis duomenų bazėmis, nepavykus prisijungimui kartais tenka vykdyti užklausas iš „PHP“ bylų. [39]

Apžvelgus visas priemones, padedančias integruoti programinius agentus į interneto svetaines, galima sudaryti agentų palyginimo lentelę:

4 lentelė. Priemonių, padedančių integruoti agentus į interneto svetaines, palyginimo lentelė

Priemonės ir savybės	„Java applet“	„CGI“	„Apache Tomcat“	„Quercus“	„PHP/Java Bridge“
Programavimo kalbų palaikymas	„Java“	„Java“, „Perl“, „C++“, „Visual Basic“	„Java“	„Java“	„Java“
Veikimo pusė	Kliento pusėje	Serverio pusėje	Serverio pusėje	Serverio pusėje	Serverio pusėje
Reikalavimai lankytojui	Reikalingas „Java Runtime Environment“ įskiepis	Nėra	Nėra	Nėra	Nėra
Vykdomi bylų formatai	.jar	.cgi	.war (.jar), .jsp	.war (.jar)	.war (.jar)
Suderinamumas su duomenų bazėmis	Suderinama su populiariausiomis duomenų bazėmis	Suderinama su populiariausiomis duomenų bazėmis	Suderinama su populiariausiomis duomenų bazėmis	Suderinama su populiariausiomis duomenų bazėmis	Nesuderinama su kai kuriomis „MySQL“

					versijomis
Papildomi reikalavimai	Nėra	Rašant programas reikia atsižvelgti į „CGI“ reikalavimus	Nėra	„Resin“ serveris, rašant PHP kodą reikia atsižvelgti į „Quercus“ reikalavimus	Servisas pvz.: „Apache Tomcats“
Ryškiausi privalumai	Mažai užkraunamas serveris, populiariausių naršyklių ir duomenų bazių palaikymas	Praktiškai visi serveriai standartiškai turi šį protokolą, nereikalauja lankytojų įsidiegti papildomos programinės įrangos	Populiariausių naršyklių ir duomenų bazių palaikymas	Užtikrinama didelė veikimo sparta, padidinamas saugumas, PHP gali laisvai kreiptis į „Java“ klases	Spartus veikimas, nereiki keisti „PHP“ bylų kodo
Ryškiausi trūkumai	Reikalavimas lankytojams įsidiegti papildomą programinę įrangą, sumažėjęs saugumas, nepalaikymas išmaniuosiuose telefonuose	Serverio resursų išnaudojimo problema, rašant agentą reikia atsižvelgti į „CGI“ reikalavimus	Serverio resursų išnaudojimo problema, reikalingas papildomas serverio servisas, kuris standartiškai nėra įdiegiamas serveryje	„PHP“ ir „Java“ kodas turi būti perrašytas į reikalaujamą „Quercus“ technologijos	Reikalingas papildomas serverio servisas, kuris standartiškai nėra įdiegiamas serveryje

1.4 Programinių agentų panaudojimas

Programinių agentų pritaikymas yra labai platus: kuriant imitacinius modelius, versle, ypač elektroninėje prekyboje, medicinoje, karo pramonėje, informacijos apdorojime ir paieškoje, kuriant asmeninius pagalbininkus, žaidimuose, netgi astronomijoje.

Programiniai agentai gali būti panaudoti automatizuojant informacines sistemas arba palengvinti jų darbą. Agentais įmanoma net ir visiškai pakeisti kai kuriuos žmogaus veiksmus. Pavyzdžiui, klientų pagalbos linija, paprastai prie telefono budi operatorius, klientui paskambinus operatorius dažniausiai remiasi pateiktomis instrukcijomis patariant klientui. Tokį darbą gali atlikti ir programinis agentas, išklausęs pašnekovo problemas, vykdo paiešką elektroninėje instrukcijoje ir pateikia labiausiai tikėtiną problemos sprendimo variantą.

Kitas panaudojimo pavyzdys – namas kurį valdo intelektualinė informacinė sistema. Agentai savo jutikliais gali fiksuoti apšvietimo intensyvumą, kambario temperatūrą, drėgnumą. Agentas stebėdamas šeimnininko elgesį, įpročius priima sprendimus, kurie turėtų tenkinti šeimnininką. Toks namas kamerų pagalba netgi galėtų atskirti įsibrovėlius nuo šeimnininko ir automatiškai pranešti saugumo tarnyboms. Taip pat užrakinti naktį pamirštas užrakinti duris.

Agentai naudojami net ir kompiuteriniuose žaidimuose, kur žaidėjo priešininką valdo programinis agentas. Priešininkas, ne tik apsimoko naujose vietovėse, bet ir pasimoko iš savo klaidų. Tuo pačiu būdu nugalėti priešininkai po truputį keičia savo veiksmus, tuo priversdami žaidėją pakeisti savo veiksmus.

Vienas garsesnių realių agento panaudojimo pavyzdžių, ankstyvais 1997 metais, tai „FireFly“ intelektualinė informacinė sistema, kuri naudojo agentus savo vartotojų profilio informacijai surinkti. Vartotojui užsiregistravus ir nurodžius pavyzdžiui mėgstamą muzikos grupę, programiniai agentai pradėdavo ieškoti panašaus stiliaus ir žanro atlikėjų. Agentai atsižvelgia ką mėgsta ir kiti „FireFly“ vartotojai, kurie nurodė būtent šią ar panašią muzikinę grupę. Atlikę paiešką, agentai siūlo vartotojui panašią jo skoniu muziką, vartotojas gali balsuoti ar jam tai pasiūlytos grupės patinka. Po vartotojo balsavimo rezultatų agentai vykdydami paieškas kitiems vartotojams atžvelgs ir į balsavimo rezultatus. Vėliau „FireFly“ perėjo prie intelektualių anketų kūrimo, kur anketų duomenys analizuojami agentų pagalba. [19]

Kompanija „IBM“ taip pat yra sukūrusi sistemų, į kurias integravo programinius agentus. Vienas tokių atvejų kai į paprastą parduotuvę kūrėjai integravo asmeninį pirkimo pagalbininką (angl. personal shopping assistant). Pirkimo pagalbininkas stebi pirkėjo veiksmus parduotuvėje, kokius produktus dažniausiai žiūri klientas, ką perka. Kitą kartą užsukus tam pačiam klientui sistema jau pasiūlo klientui prekes, kuriomis domėjosi klientas. Jei klientas yra užsiregistravęs, pagal nurodytą gimimo data pagalbininkas ieško įdomesnių prekių pasiūlymų gimtadieniui. Taip pat susisiečia su pirkėju kai atsiranda naujų jo mėgstamų prekių ar išpardavimų ir pasiūlo pirkimą. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo. [15]

Kitas programinių agentų panaudojimo atvejis – karyboje. JAV pėstininkų kovos grupei 2004 metais buvo sukurta speciali simuliacijos sistema „AEGIS Cruiser Air-Defense Simulation“, kuri naudoja programinius agentus. Sistemos pagrindinis tikslas kuo realiau imituoti karo veiksmus jūroje. Sistema ne tik tikroviškai simuliuoja oro ir jūros reidus, tačiau taip pat vertina ir karininkų gynybos veiksmus, todėl po pratybų „AEGIS“ pasiūlo patarimų ir įvertina mūšio pasisėkimą. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo. [16]

Viena žinomiausių sistemų Europoje – „ARCHON“ (Architecture for Cooperative Heterogeneous ON-line systems) sukurta dar 1999 metais sistema naudojama iki šių dienų. Pagrindiniai šios sistemos naudotojai šiaurės Ispanijos elektros kompanija „Iberdrola“. Ši sistema padeda analizuoti ir sutaupyti tiekiant elektrą. Taip pat ji pataria valdymo operatoriams kritiniais atvejais, kai į valdymo pultą gaunama daug klaidų pranešimų. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo. [40]

Dar vienas pavyzdys - prancūzų įmonės „Numericable“, teikiančios televizijos tiekimo paslaugas. Įmonė panaudojo „Chattbot“ agentą savo klientų aptarnavimo puslapyje. Agentas bendrauja su klientais tekstiniais pranešimais. Klientui įvedus klausimą, agentas bando į jį atsakyti pateikdamas su klausimu susijusią informaciją. Tokiu būdu imituojamas gyvas kliento ir klientu aptarnavimo darbuotojo dialogas. Klientai gali bendrauti su agentu paprasta kalba, užuot ieškoję reikiamos informacijos svetainėje. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo,

tačiau galima pastebėti, kad svetainėje naudojama „Flash“ ir „Javascript“ programavimo kalbos. „Numericable“ klientų aptarnavimo agentas pavaizduotas 23 paveiksliuke. [42]

„Metacrawler“ agentais paremta paieškos sistema, kuri veikia ir iki šių dienų. Ši sistema naudoja informacinius agentus paieškoms vykdyti. Vartotojui davus užklausą sistemai, ji ieško informacijos įvairiuose paieškos puslapiuose, gautus rezultatus sujungia, pateikia susistemintus ir pirmiausia išrenka jos manymu tinkamiausius įrašus, pašalina puslapiuose įdėtas reklamas. Taip pat agentai mokosi iš anksčiau lankytojų dažniausiai įvestų frazių ir pasiūlo užbaigti paieškos frazes pats. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo, tačiau internete galima rasti, kad „Metacrawler“ agentai naudoja „Java“ programavimo kalbą. [41]

Vienas garsesnių automatinio duomenų rinkimo agentų yra „Googlebot“. Jis keliauja po visas svetaines ir renka interneto svetainių duomenis. Jei svetainė yra atsinaujinusi, agentas atnaujina jo aprašymą paieškos sistemoje. Agentai tarpusavyje dalinasi informacija ir pasiskirsto ieškojimo zonas. Kiti agentai, gavę pranešimą, kad svetainė atsinaujino, pakeičia informaciją ir išsaugo svetainės atsarginę kopiją savo serveryje. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo, tačiau yra paskelbta, kad pirmos „Googlebot“ versijos realizuotos „Python“ ir „Java“ kalbomis. [17]

Prekių kainų palyginimo „Shopbot-Inc“ kompanijos sukurtas agentas „Shopbot“, keliauja po el. parduotuves ir renka interneto svetainių siūlomas kainas. Vartotojui davus užklausą sistemai, ji ieško informacijos įvairiuose paieškos puslapiuose, gautus rezultatus sujungia, pateikia geriausius pasiūlymus. Per laiką šis agentas pradeda ieškoti dažniausiai vartotojo ieškotas prekes. Tiksliai nėra žinoma realizavimo technologija, nes gamintojas nenurodo. Tiesa šis agentas nebendruoja su pačiu lankytoju gyvai, jis veikia kaip prekių paieškos sistema, kuri automatiškai gali ieškoti tūkstančiuose elektroninių parduotuvių. [11]

Apžvelgus visus agentų panaudojimo atvejus galima sudaryti tokią palyginimo lentelę:

5 lentelė. Programinių agentų panaudojimo atvejų palyginimo lentelė

Agentų panaudojimas ir savybės	„FireFly“	„Personal shopping assistant“	„AEGIS“	„Metacrawler Agent“	„Chattbot“	„Googlebot“	„Shopbot“
Sistemos tipas	Internetinė	Specifinė prekybos sistema	Specifinė simuliacinio programo	Internetinė paieškos sistema	Internetinė klientų aptarnavimo sistema	Internetinė	Internetinė
Agento tipų savybės	Informacinis, sąsajos	Informacinis, sąsajos	Reaguojantis, sąsajos	Informacinis, mobilus	Reaguojantis	Informacinis, mobilus	Informacinis, mobilus
Bendravimas su lankytoju	Ne	Taip	Ne	Ne	Taip	Ne	Ne
Atsižvelgimas į	Taip	Taip	Taip	Ne	Taip	Ne	Taip

lankytojo veiksmus							
Paieškos vykdymas kitose svetainėse	Ne	Ne	Ne	Taip	Ne	Taip	Taip
Informacijos kaupimas	Taip	Taip	Taip	Ne	Ne	Taip	Taip
Agento sąveika su sistema	Sąveikauja su interneto svetaine perduodant surinktus duomenis į serverį	Sąveikauja su prekybos serveriu perduodant surinktus duomenis, gaunant informaciją apie akcijas	Sąveikauja su analizavimo sistema perduodant surinktus mūšių duomenis	Sąveikauja su interneto svetaine perduodant surinktus duomenis ir gaunant užklausą	Sąveikauja su interneto svetaine perduodant atsakymus ekrane ir gaunant užklausą	Sąveikauja su atsarginių kopijų serveriu duodant komandas, gaunant užklausas kurias svetainės tikrinti	Sąveikauja su serveriu perduodant surinktus duomenis
Agento realizavimo technologija	Tiksliai nėra aišku, kai kurie šaltiniai nurodo „Java“ programavimo kalbą	Nežinoma	Nežinoma	Tiksliai nėra aišku, kai kurie šaltiniai nurodo „Java“ programavimo kalbą	Tiksliai nėra aišku, galima pastebėti, kad naudojami „Flash“ ir „JavaScript“ kalbos	Tiksliai nėra aišku, pirmos versijos realizuotos „Python“ ir „Java“ kalbomis	Nežinoma
Atliekami veiksmai	Renkama informacija apie vartotojo pomėgius, vėliau pateikiama labiausiai dominanti informacija	Stebimi pirkėjo veiksmai, mėgstamos prekės, siūlomoms prekių akcijos pagal pirkėjo domėjimąsi	Simuliuojami kariniai veiksmai jūroje, reaguojama į kariškio veiksmus, mokosi iš ankstesnių veiksmų	Vykdo vartotojo paiešką kitose svetainėse	Atsako į lankytojo užduotus klausimus ir pateikia informaciją, remdama savo duomenų baze	Naršo internetą ir tikrina visas svetaines, pastebėjus pokyčius siunčia komandą serveriui, kad padarytų atsarginę svetainės kopiją	Naršo internetą ir tikrina visas svetaines, kaupia prekių kainas savo duomenų bazėje, atsižvelgia į klientų paieškas

1.5 Problemos analizės išvados

Viena pirmųjų problemų su kuria susiduriama prieš projektuojant virtualų asistentą yra programinių agentų pasirinkimas. Reikia išsiaiškinti kas yra programinis agentas, veikimo principą, sandarą, taip pat nuspręsti kokį programinio agento tipą pasirinkti el. parduotuvėms. Pagal Nikolajų Kasabovą (Nikola Kasabov), Chosė Brustolonį (Jose Brustoloni), Derelą Davis (Darryl Davis) ir Hiacintą Nvaną (Hyacinth Nwana) visi programiniai agentai skirstomi pagal veikimo pobūdį į skirtingus tipus, klasifikacijas, todėl reikalinga nustatyti kokie programinių agentų tipai labiausiai tiktų el. parduotuvėje integruotam virtualiam asistentui, kokiomis savybėmis jis turėtų pasižymėti. Reikalingos virtualaus asistento savybės: veikimas autonomiškai be administratoriaus įsikišimo, bendravimas su svetainės lankytojais, gebėjimas surinkti informaciją iš interneto svetainių, rezultatų pateikimo patikimumas, reagavimas į pokyčius, paprasta realizacija. Į visus šiuos reikalavimus bus atsižvelgiama renkantis agentų tipus, savybes.

Pasirinkus programinio agento tipą taip pat būtina išnagrinėti įrankius, skirtus programinių agentų realizavimui. Jų yra nemažai todėl renkantis kūrimo platformą reikia atsižvelgti į: įrankio programavimo kalbą, programinio agento programavimo kalbą, licenziją, palaikomų bylų formatą, papildomų reikalavimų gausą, kitų programų palaikymą, informacijos ir pavyzdžių kiekį. Šis pasirinkimas yra labai svarbus, kadangi turi būti apgalvota ar pavyks tokį programinį agentą paleisti internetiniame puslapyje.

Realizavus patį programinį agentą reikalinga jį integruoti į interneto puslapį. Todėl privalu palyginti esamas integravimo technologijas, kurios leistų aktyvuoti programinį agentą serveryje. Renkantis integravimo būdą atsižvelgiama į šiuos kriterijus: programavimo kalbų palaikymas, bylų formatų palaikymas veikimo pusė (ar agentas veiks kliento kompiuteryje ar serveryje), reikalavimai lankytojui, suderinamumas su duomenų bazėmis. Ši analizė atskleistų kokių būdų virtualus asistentas būtų integruojamas į svetainę, todėl ši analizė priklauso ir nuo agento kūrimo įrankių analizės. Būtina, kad kūrimo įrankis ir integravimo būdai derėtų tarpusavyje.

Pravartu apžvelgti ir kitų programinių agentų panaudojimo atvejus. Jie leistų pastebėti kuriamo agento privalumus ir trūkumus, lengviau nustatyti ką būtų galima pakeisti, patobulinti jau egzistuojančiose sistemose. Analizuojant pavyzdžius atsižvelgiama į: sistemos tipą, bendravimo su lankytojų būdą, atsižvelgimą į lankytojo veiksmus, paieškos vykdymą internetinėse svetainėse, agento sąveiką su sistema, naudojamą technologiją.

Norint sukurti universalų programinį agentą reikalingos ir papildomos priemonės, kaip: virtualaus asistento duomenų bazė, administravimo skydas, papildomas apdorojimo kodas, leidžiantis lengviau įterpti asistentą į el. parduotuves, bei supaprastinantis pačio programinio agento veikimą. Todėl šios priemonės taip pat kuriamos kartu su programiniu agentu ir yra neatsiejama dalis.

2. TYRIMO REIKALAVIMAI

2.1 Programinių agentų reikalavimai

Virtualus asistentas turėtų pasižymėti autonomišku veikimu, bendravimu su lankytoju ir kitais agentais, sparčiu reagavimu į pakeitimus, galimybę rinkti informaciją internete, sugebėjimu mokytis, saugumu, rezultatų patikimumu, tolerantiškumu klaidoms. Taip pat šis programinis agentas turėtų būti lengvai realizuojamas, kadangi sudėtingo programinio agento realizavimas užtruktų per ilgai.

Programinis agentas, atitinkantis visus reikalavimus būtų idealus, tačiau vienos savybės svarbesnės nei kitos. Apžvelgus agentų panaudojimo pavyzdžius, ypač virtualiojo asistento „Personal shopping assistant“ ir „Shopbot“ agentus, taip pat atsižvelgiant į ekspertų nuomonę, nuspręsta, kad svarbiausios virtualiojo asistento savybės yra autonominis veikimas, bendravimas su lankytoju, saugumas, rezultatų patikimumas, informacijos rinkimas, realizavimo sudėtingumas. Ne tokiomis svarbiomis savybėmis laikoma sparta reaguojant į pakeitimus, bendravimas su kitais agentais, tolerantiškumas klaidoms.

2.2 Programinių agentų kūrimo būdų reikalavimai

Programinių agentų kūrimo būdas turėtų pasižymėti populiaria ir suderinama programinių agentų programavimo kalba, programinių agentų standartų palaikymu, nemokama licencija, papildomų įrankių, dokumentacijos, pavyzdžių ir saugomų formatų gausa. Taip pat svarbu ir programos išsaugotų agentų atidarymo kitomis programomis galimybė.

Kūrimo priemonės atitinkanti visus reikalavimus būtų ideali, tačiau vienos savybės svarbesnės nei kitos. Atsižvelgiant į programinio agento savybių reikalavimus, ekspertų nuomonę, nuspręsta, kad svarbiausios programinių agentų kūrimo priemonių savybės yra agentų programavimo kalba, nemokama licencija, papildomi įrankiai, išsaugojimo formatų gausa. Ne tokiomis svarbiomis savybėmis laikoma pavyzdžių, dokumentacijos gausa, standartų palaikymas.

2.3 Programinių agentų integravimo į interneto svetaines būdų reikalavimai

Programinių agentų integravimo į interneto svetaines būdas turėtų pasižymėti programų aktyvavimu serverio pusėje, papildomų ir programinės įrangos reikalavimų nebuvimu lankytojui ir administratoriui, suderinamumu su populiariausiomis duomenų bazėmis. Taip pat šis integravimo būdas turėtų palaikyti kaip įmanoma daugiau programavimo kalbų ir vykdomų bylų formatų.

Programinių agentų integravimo būdas, atitinkantis visus reikalavimus būtų idealus, tačiau vienos savybės svarbesnės nei kitos. Atsižvelgiant į programinio agento savybių reikalavimus, taip pat ekspertų nuomonę, nuspręsta, kad svarbiausios programinių agentų integravimo į interneto svetaines būdų savybės yra programų aktyvavimas serverio pusėje, suderinamumas su duomenų bazėmis, palaikomų formatų gausa. Ne tokiomis svarbiomis savybėmis laikoma papildomų reikalavimų lankytojui ir administratoriui nebuvimas.

2.4 Programinių agentų panaudojimo reikalavimai

Programinių agentų panaudojimo atvejis turėtų pasižymėti atsižvelgimu į lankytojo veiksmus, informacijos kaupimu, atliekamų veiksmų gausa, paieškos vykdymą internete, bendravimu su lankytoju. Taip pat šis atvejis turėtų būti internetinio sistemos tipo ir pasižymėti virtualaus asistento savybėmis.

Programinių agentų panaudojimo atvejais, atitinkantis visus reikalavimus būtų idealus, tačiau vienos savybės svarbesnės nei kitos. Atsižvelgiant į programinio agento savybių reikalavimus, taip pat ekspertų nuomonę, nuspręsta, kad svarbiausios programinių panaudojimo atvejų savybės yra atsižvelgimas į lankytojo veiksmus, informacijos kaupimas, bendravimas su lankytoju. Ne tokiomis svarbiomis savybėmis laikoma papildomų agento tipų savybių gausa, paieška internete galimybė.

2.5 Sistemos reikalavimų specifikacija

2.5.1 Projekto varovai

Projekto kūrimo pagrindas

Programinis agentas kuriamas, todėl kad norima pademonstruoti programinio agento integravimą į informacinę sistemą. Šiuo atveju kuriamas virtualaus asistentas el. parduotuvei.

Sistemos tikslai

Virtualaus asistentas turi užtikrinti lengvą, greitą ir nereikalaujantį gilių žinių informatikos srityje informacijos pateikimą interneto parduotuvės klientams, agento valdymą administratoriams.

Projekto dalyviai

Užsakovas

Šiaulių universitetas, technologinis fakultetas.

Vykdytojai

Pagrindinis ir vienintelis vykdytojas Šiaulių universiteto informatikos inžinerijos studentas Ernestas Stravinskas.

Vartotojai

Tai asmenų, kurie betarpiškai naudosis sistema, sąrašas. Kiekvienai vartotojų kategorijai reikia nurodyti tokią informaciją:

6 lentelė. Asmenų, kurie betarpiškai naudosis sistema sąrašas

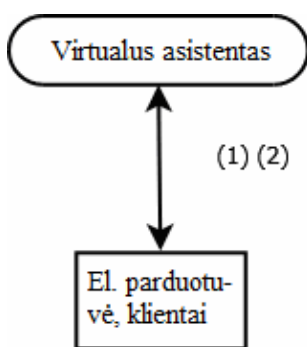
Vartotojo kategorija	Administratorius
Vartotojo sprendžiami uždaviniai:	Galimybė pridėti/redaguoti/ištrinti virtualius asistentus, sustabdyti ar iš naujo paleisti patį agentą. Taip pat galimybė pridėti/redaguoti/ištrinti DB šablonus, Peržiūrėti ir išvalyti sukauptą statistiką.
Patirtis dalykinėje srityje:	Įvairi
Patirtis informacinėse technologijose:	Srities specialistas
Papildomos charakteristikos:	Reikalingas susipažinimas su sistema
Apsimokymo poreikis:	Nereikia
Amžiaus grupė:	18-80
Vartotojo kategorija	Klientas
Vartotojo sprendžiami uždaviniai:	Galimybė uždaryti virtualų asistentą, gauti panašių prekių pasiūlymus.
Patirtis dalykinėje srityje:	Įvairi
Patirtis informacinėse technologijose:	Nėra
Papildomos charakteristikos:	Nereikalingas susipažinimas su sistema
Apsimokymo poreikis:	Nereikia
Amžiaus grupė:	18-80

2. 5. 2 Projekto apribojimai

Pagrindiniai reikalavimai el. parduotuvei, kad ji išsaugotų savo klientus (identifikavimo numerį, vardą), prekes (identifikavimo numerį, kainą, prekės kodą, pavadinimą, kategorijos identifikavimo numerį), kategorijas (identifikavimo numerį, pavadinimą). Kadangi agentas naudosis šiais duomenimis savo statistikai skaičiuoti.

Reikalavimai serverio, kuriame veiks virtualių asistentų sistema, programinei įrangai: „Apache 2.0“ arba naujesnė, „PHP 5.0“ arba naujesnė, „MySQL 5.0“ arba naujesnė. Apribojimai bus keliami duomenų bazės atminties bei serverio disko vietos sunaudojimui. Reikalavimai interneto naršyklėms: Internet Explorer versija 7.0 arba naujesnė, „Mozilla FireFox“ 3.0 arba naujesnė, „Opera“ 10.0 arba naujesnė, „Google Chrome“ 26.0.1410 arba naujesnė. Taip pat serveris privalo palaikyti „Java“ programavimo kalba parašytą agentą (turi turėti įdiegtą vieną iš anksčiau aprašytų).

Pagrindinis našumo apribojimas - kas minutę virtualus asistentas apdoroja tik 40 statistikos įrašų. Toks apribojimas pritaikytas todėl, kad serveris, kuriame veikia sistema yra Šiaulių universiteto bendras, skirtas visiems studentams. Limitavimas buvo privalomas, kadangi serverio resursai riboti.



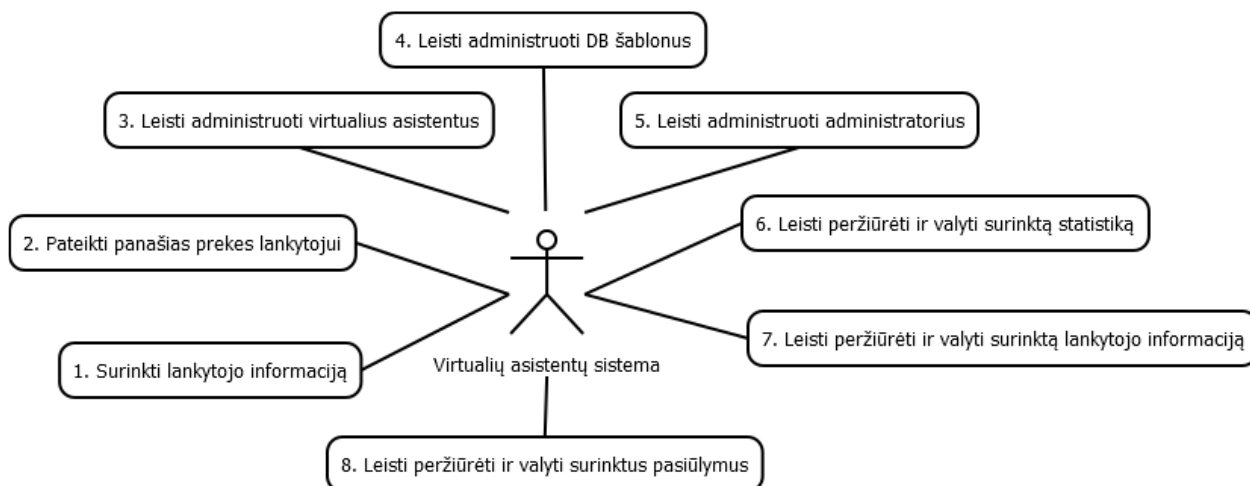
5 pav. Konteksto diagrama

Veiklos padalinimas

7 lentelė. Veiklos padalinimo lentelė

Eil. nr.	Įvykio pavadinimas	Įeinantys/išeinantys srautai
1.	Siūlomos prekės, pateikti pranešimai	Duomenys (out)
2.	Peržiūrėtos, užsakytos prekės	Duomenys (in)

2. 5. 3 Panaudojimo atvejų sąrašas



6 pav. Virtualių asistentų sistemos panaudos atvejų diagrama

8 lentelė. Panaudojimo atvejo „Surinkti lankytojo statistiką“ aprašymas

Pavadinimas	Surinkti lankytojo statistiką	Numeris	1
Aprašymas	Virtualus asistentas fiksuoja prekių peržiūrų, padarytų užsakymų skaičių, lankytojų prisijungimus, sesijos įrašus ir t.t.		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyimo sąlyga	Lankytojas atlikta užsakymą arba peržiūri prekę, lankytojas aplanko svetainę, prisijungia		
Po sąlyga	Asistento duomenų bazėje išsaugoma atlikto veiksmo informacija		

9 lentelė. Panaudojimo atvejo „Pateikti panašias prekes lankytojui“ aprašymas

Pavadinimas	Pateikti panašias prekes lankytojui	Numeris	2
Aprašymas	Virtualus asistentas stebėdamas lankytojo veiksmus (žiūrimas prekes, padarytus užsakymus) pagal statistiką, sukurtas akcijas (jei el. parduotuvė palaiko), atrenka panašias prekes, kurios labiausiai domina lankytoją		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	Turi būti sukaupti bent keli statistikos įrašai apie klientą		
Vykdyimo sąlyga	Lankytojas peržiūri prekes ar padaro užsakymą		
Po sąlyga	El. parduotuvėje lankytojui pateikiamas pranešimas su siūlomomis prekėmis		

10 lentelė. Panaudojimo atvejo „Leisti administruoti virtualius asistentus“ aprašymas

Pavadinimas	Leisti administruoti virtualius asistentus	Numeris	3
Aprašymas	Valdymo skyde administratorius gali ištrinti, pridėti ar redaguoti virtualius asistentus. Kiekvienas asistentas turi savo el. parduotuvės DB šabloną, prisijungimo duomenis, kitus nustatymus		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyimo sąlyga	Administratorius ištrina, prideda ar redaguoja virtualų asistentą		
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai		

11 lentelė. Panaudojimo atvejo „Leisti administruoti DB šablonus“ aprašymas

Pavadinimas	Leisti administruoti DB šablonus	Numeris	4
Aprašymas	Valdymo skyde administratorius gali ištrinti, pridėti ar redaguoti DB šablonus. Kiekvienas asistentas turi savo el. parduotuvės DB šabloną, kuriame nurodyta visa reikalinga lentelių ir laukelių informacija		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyimo sąlyga	Administratorius ištrina, prideda ar redaguoja DB šabloną		
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai		

12 lentelė. Panaudojimo atvejo „Leisti administruoti administratorius“ aprašymas

Pavadinimas	Leisti administruoti administratorius	Numeris	5
Aprašymas	Valdymo skyde administratorius gali ištrinti, pridėti ar redaguoti kitus administratorius, tai gali daryti tik pagrindinis administratorius. Kiekvienas administratorius turi sau priskirtus virtualius asistentus kuriuos gali redaguoti		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyimo sąlyga	Administratorius ištrina, prideda ar redaguoja administratorių		
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai		

13 lentelė. Panaudojimo atvejo „Leisti peržiūrėti ir valyti statistiką“ aprašymas

Pavadinimas	Leisti peržiūrėti ir valyti statistiką	Numeris	6
Aprašymas	Valdymo skyde administratorius gali peržiūrėti ir išvalyti surinktus statistikos įrašus apie lankytojus		

Aktorius	Virtualių asistentų sistema
Prieš sąlyga	-
Vykdyto sąlyga	Administratorius peržiūri ar išvalo statistiką
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai

14 lentelė. Panaudojimo atvejo „Leisti peržiūrėti ir valyti lankytojo informaciją“ aprašymas

Pavadinimas	Leisti peržiūrėti ir valyti lankytojo informaciją	Numeris	7
Aprašymas	Valdymo skyde administratorius gali peržiūrėti ir išvalyti surinktą informaciją apie lankytojus (apsilankymų skaičių, paskutinio apsilankymo datą, pirkimų skaičių ir t.t.)		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyto sąlyga	Administratorius peržiūri ar išvalo lankytojo informaciją		
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai		

15 lentelė. Panaudojimo atvejo „Leisti peržiūrėti ir valyti pasiūlymus“ aprašymas

Pavadinimas	Leisti peržiūrėti ir valyti pasiūlymus	Numeris	8
Aprašymas	Valdymo skyde administratorius gali peržiūrėti ir išvalyti sukurtus panašius prekių pasiūlymus		
Aktorius	Virtualių asistentų sistema		
Prieš sąlyga	-		
Vykdyto sąlyga	Administratorius peržiūri ar išvalo pasiūlymus		
Po sąlyga	Duomenų bazėje išsaugomi administratoriaus atlikti pakeitimai		

2. 5. 4 Funkcinių reikalavimų sąrašas

16 lentelė. Funkcinis reikalavimas 1

Reikalavimas #:	1	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	1
Aprašymas:	Asistentas kaupia statistiką pagal klientų veiksmus				
Pagrindimas:	Viena pagrindinių agento funkcijų, surinkti įvairius duomenis, tokiu būdu leidžiant vykdyti kitas svarbias agento funkcijas				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Asistentas automatiškai renka statistinius duomenis klientams atlikus užsakymą, peržiūrėjus prekes ar įdėjus jas į krepšelį				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	-	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

17 lentelė. Funkcinis reikalavimas 2

Reikalavimas #:	2	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	2
Aprašymas:	Asistentas siūlo lankytojui prekes pagal jo pomėgius				
Pagrindimas:	Viena svarbiausių funkcijų pagal kiekvieno lankytojo poreikius pateikti jį dominančias prekes tokiu būdu padidinant šansus, kad klientas nusipirks prekę				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Asistentas remdamasis statistika automatiškai siūlo klientui jį labiausiai dominančias prekes				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	-		Konfliktai:	Nėra	
Papildoma medžiaga:					
Istorija:	2012-12-01				

18 lentelė. Funkcinis reikalavimas 3

Reikalavimas #:	3	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	3
Aprašymas:	Sistema leidžia administruoti virtualius asistentus				
Pagrindimas:	Ši funkcija leidžia lengviau ir paprasčiau prižiūrėti daugiau nei vieną virtualų asistentą integruotą į el. parduotuves				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali redaguoti, pridėti ar ištrinti virtualius asistentus valdymo skyde				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	-		Konfliktai:	Nėra	
Papildoma medžiaga:					
Istorija:	2012-12-01				

19 lentelė. Funkcinis reikalavimas 4

Reikalavimas #:	4	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	4
Aprašymas:	Sistema leidžia administruoti DB šablonus				
Pagrindimas:	Ši funkcija leidžia lengviau ir paprasčiau prižiūrėti DB šablonus, kurie yra būtini virtualių asistentų veikimui				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali redaguoti, pridėti ar ištrinti DB šablonus valdymo skyde				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	-		Konfliktai:	Nėra	
Papildoma medžiaga:					
Istorija:	2012-12-01				

20 lentelė. Funkcinis reikalavimas 5

Reikalavimas #:	5	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	5
Aprašymas:	Sistema leidžia administruoti administratorius				
Pagrindimas:	Ši funkcija leidžia lengviau ir paprasčiau prižiūrėti kitus administratorius, suteikiant jiems prisijungimo duomenis bei leidimus valdyti asistentus				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali redaguoti, pridėti ar ištrinti, bei suteikti leidimus administratoriams valdymo skyde				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:	-	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

21 lentelė. Funkcinis reikalavimas 6

Reikalavimas #:	6	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	6
Aprašymas:	Sistema leidžia peržiūrėti ir valyti statistikos įrašus				
Pagrindimas:	Ši funkcija leidžia lengvai patikrinti sistemos veikimo teisingumą, galimybė išvalyti įrašus praverčia testavime				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali peržiūrėti ir išvalyti norimus statistikos įrašus				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	3		
Priklausomybės:	-	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

22 lentelė. Funkcinis reikalavimas 7

Reikalavimas #:	7	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	7
Aprašymas:	Sistema leidžia peržiūrėti ir valyti lankytojo informaciją				
Pagrindimas:	Ši funkcija leidžia lengvai patikrinti sistemos veikimo teisingumą, galimybė išvalyti įrašus praverčia testavime				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali peržiūrėti ir išvalyti norimą lankytojų informaciją				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	3		
Priklausomybės:	-	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

23 lentelė. Funkcinis reikalavimas 8

Reikalavimas #:	8	Reikalavimo tipas:	2.5.4	Panaudojimo atvejis #:	8
Aprašymas:	Sistema leidžia peržiūrėti ir valyti pasiūlymus				
Pagrindimas:	Ši funkcija leidžia lengvai patikrinti sistemos veikimo teisingumą, galimybė išvalyti įrašus praverčia testavime				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Administratorius gali peržiūrėti ir išvalyti norimus pasiūlymus				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	3		
Priklausomybės:	-	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

2. 5. 5 Nefunkcinių reikalavimų sąrašas

Reikalavimų sistemos priežiūrai sąrašas

24 lentelė. Nefunkcinis reikalavimas 1

Reikalavimas #:	9	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:	
Aprašymas:	Patogi vartotojo sąsaja				
Pagrindimas:	Reikalavimai agento pranešimams turėtų sudaryti: saikingas informacijos kiekis, lengvai iskaitomas tekstas (šriftas), paprastas dizainas				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Naudojant programinį agentą nereikia gaišti daug laiko ieškant reikimos informacijos, matomas akiai malonus ir neperkrautas vaizdas				
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1		
Priklausomybės:	Nėra	Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

25 lentelė. Nefunkcinis reikalavimas 2

Reikalavimas #:	10	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:	
Aprašymas:	Mažai pastangų priežiūrai				
Pagrindimas:	Svarbu sunaudoti kuo mažiau laiko ir pastangų prižiūrint programinį agentą				
Šaltinis:	Virtualių asistentų sistema				
Tikimo kriterijus:	Sistema veikia autonomiškai, be administratoriaus įsikišimo				
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1		
Priklausomybės:		Konfliktai:	Nėra		
Papildoma medžiaga:					
Istorija:	2012-12-01				

26 lentelė. Nefunkcinis reikalavimas 3

Reikalavimas #:	11	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Neįkyri sąsaja			
Pagrindimas:	Reikalaujama, kad vartotojui nereikėtų kelis kartus patvirtinti to pačio veiksmo			
Šaltinis:	Virtualių asistentų sistema			
Tikimo kriterijus:	Veiksmai atliekami vienu patvirtinimu, nereikalaujama patvirtinti kelis kartus			
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1	
Priklausomybės:	Nėra	Konfliktai:	Nėra	
Papildoma medžiaga:				
Istorija:	2012-12-01			

Reikalavimų panaudojamumui sąrašas

27 lentelė. Nefunkcinis reikalavimas 4

Reikalavimas #:	12	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Nesudėtingas naudojimas sistema			
Pagrindimas:	Norint atlikti veiksmus/pakeitimus susijusius su programiniu agentu turi būti lengva ir intuityvu, pateikti paaiškinimai			
Šaltinis:	A Virtualių asistentų sistema			
Tikimo kriterijus:	Minimalių žinių naudotis sistema, paaiškinimų buvimas			
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1	
Priklausomybės:		Konfliktai:	Nėra	
Papildoma medžiaga:				
Istorija:	2012-12-01			

28 lentelė. Nefunkcinis reikalavimas 5

Reikalavimas #:	13	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Duomenys pateikiami nacionaline – lietuvių – kalba			
Pagrindimas:	Turinio pateikimas nacionaline kalba būtinas			
Šaltinis:	Virtualių asistentų sistema			
Tikimo kriterijus:	Visa sistema aprašyta lietuvių kalba			
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1	
Priklausomybės:		Konfliktai:	Nėra	
Papildoma medžiaga:				
Istorija:	2010-12-01			

Reikalavimų vykdymo charakteristikoms sąrašas

29 lentelė. Nefunkcinis reikalavimas 6

Reikalavimas #:	14	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Sistemos išplečiamumas			
Pagrindimas:	Lengvas virtualaus asistento galimybių išplečiamumas sutaupys nemažai administratoriaus laiko ir pastangų			
Šaltinis:	Virtualių asistentų sistema			
Tikimo kriterijus:	Virtualiam asistentui lengvai galima pritaikyti naujas funkcijas			
Užsakovo tenkinimas:	4	Užsakovo netenkinimas:	1	
Priklausomybės:	Konfliktai:			Nėra
Papildoma medžiaga:				
Istorija:	2010-12-01			

Reikalavimų saugumui sąrašas

30 lentelė. Nefunkcinis reikalavimas 7

Reikalavimas #:	15	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Nustatymus gali keisti tik administratorius			
Pagrindimas:	Siekama apsaugoti nuo sistemos pakeitimo ir panaikinimo			
Šaltinis:	Virtualių asistentų sistema			
Tikimo kriterijus:	Visus duomenis gali keisti tik administratorius su atitinkamais leidimais			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	Nėra	Konfliktai:	Nėra	
Papildoma medžiaga:				
Istorija:	2012-12-01			

31 lentelė. Nefunkcinis reikalavimas 8

Reikalavimas #:	16	Reikalavimo tipas:	2.5.5	Panaudojimo atvejis #:
Aprašymas:	Veikimas ir prie didesnių apkrovų			
Pagrindimas:	Siekama užtikrinti sistemos funkcionalumą net ir esant nemažai serverio apkrovai			
Šaltinis:	Virtualių asistentų sistema			
Tikimo kriterijus:	Sistema vis tiek funkcionuoja nepaisant didesnės serverio apkrovos			
Užsakovo tenkinimas:	5	Užsakovo netenkinimas:	1	
Priklausomybės:	Nėra	Konfliktai:	Nėra	
Papildoma medžiaga:				
Istorija:	2012-12-01			

Reikalavimai esamų duomenų perkėlimui

Svarbu turėti visas programinio agento bylas ir „MySQL“ duomenų bazės eksportavimo bylą. Įkėlus bylas į FTP serverį ir pasinaudojus importavimo komanda duomenų bazėje bus įdiegtas programinis agentas. Būtina iš naujo nustatyti agento konfigūraciją pasikeitus el. parduotuvei. Taip pat būtina perkelti ir patį programinį agentą, realizuotą specializuotomis agentų kūrimo priemonėmis ir iš naujo aktyvuoti jo veikimą serveryje.

3. TYRIMO REZULTATAI

3.1 Programinių agentų tyrimo metodika

Programinių agentų tipų tyrime remiamasi į programinių agentų reikalavimus, savybių galimas mažiausias ir didžiausias taškų skaičius įvertintas 32 lentelėje.

32 lentelė. Programinių agentų savybių galimas įvertinimas taškais

Savybės ir taškai	Mažiausias galimas taškų skaičius	Didžiausias galimas taškų skaičius
Autonomiškumas	0	1
Bendravimas su kitais agentais	0	1
Sparta reaguojant į pakitimus	1	3
Gebėjimas mokytis	0	1
Galimybė rinkti informaciją internete	0	1
Saugumas	1	2
Rezultatų patikimumas	1	3
Tolerantiškumas klaidoms	0	1
Realizavimo sudėtingumas	1	4

Taip pat įvertinama ir programinių agentų tipų savybių svarba 33 lentelėje. Remiamasi į programinių agentų reikalavimus, o svarbumas išreikštas procentais. Visų savybių svarbumo suma yra 100%.

33 lentelė. Programinių agentų savybių svarbumas, įvertintas procentine išraiška

Savybės ir svarba, įvertinta procentais	Svarbumo procentas
Autonomiškumas	5%
Bendravimas su kitais agentais	5%
Sparta reaguojant į pakitimus	5%
Gebėjimas mokytis	15%
Galimybė rinkti informaciją internete	20%
Saugumas	5%
Rezultatų patikimumas	15%
Tolerantiškumas klaidoms	5%
Realizavimo sudėtingumas	25%

Remiantis problemos analizės skyriaus medžiaga programinių agentų tipus galima įvertinti taškais, taškai skirstomi pagal 32 lentelėje nurodytus taškus ir pagal savybės įvertinimą konkrečiam tipui (kuo daugiau savybės bruožų turi tipas, tuo daugiau taškų jam skiriama).

34 lentelė. Programinių agentų savybių įvertinimas taškais

Agento tipas ir savybės	Bendradarbiaujantis agentas	Sąsajos agentas	Mobilus agentas	Informacinis agentas	Reaguojantis agentas	Hibridinis agentas	Intelektualus agentas
Autonomiškumas	1	1	1	1	1	1	1
Bendravimas su kitais agentais	1	0	1	1	1	1	1
Sparta reaguojant į pakitimus	1	2	1	1	3	1	1
Gebėjimas mokytis	0	1	0	1	0	1	1
Galimybė rinkti informaciją internete	0	0	1	1	0	1	1

Saugumas	1	2	1	1	2	2	2
Rezultatų patikimumas	2	2	2	2	1	2	3
Tolerantiškumas klaidoms	1	0	1	1	1	1	1
Realizavimo sudėtingumas	2	3	2	3	3	1	1
Iš viso	9	11	10	12	12	11	12

Galutinis programinių agentų tipų įvertinamas gaunamas 35 lentelėje. Šie taškai gaunami sudauginus savybės svarbumo procentą (33 lentelė) su savybių taškais (34 lentelė).

35 lentelė. Programinių agentų galutinis įvertinimas taškais

Agento tipas ir savybės	Bendradarbiaujantis agentas	Sąsajos agentas	Mobilus agentas	Informacinis agentas	Reaguojantis agentas	Hibridinis agentas	Intelektualus agentas
Autonomiškumas	5	5	5	5	5	5	5
Bendravimas su kitais agentais	5	0	5	5	5	5	5
Sparta reaguojant į pakeitimus	5	10	5	5	15	5	5
Gebėjimas mokytis	0	15	0	15	0	15	15
Galimybė rinkti informaciją internete	0	0	20	20	0	20	20
Saugumas	5	10	5	5	10	10	10
Rezultatų patikimumas	30	30	30	30	15	30	45
Tolerantiškumas klaidoms	5	0	5	5	5	5	5
Realizavimo sudėtingumas	50	75	50	75	75	25	25
Iš viso	105	145	125	165	130	120	135

Daugiausiai galutinių taškų surinko informacinis ir sąsajos agentai.

3.2 Programinių agentų kūrimo būdų tyrimo metodika

Programinių agentų kūrimo būdų tyrime remiamasi į programinių agentų kūrimo būdų reikalavimus, savybių galimas mažiausias ir didžiausias taškų skaičius įvertintas 36 lentelėje.

36 lentelė. Programinių agentų kūrimo būdų savybių galimas įvertinimas taškais

Savybės ir taškai	Mažiausias galimas taškų skaičius	Didžiausias galimas taškų skaičius
Kuriamo agento programavimo kalba	1	2
Standartų palaikymas	0	1
Licenzija	0	1
Papildomos programinės įrangos reikalavimas	0	2
Palaikomi bylų formatai	1	4
Išsaugotų bylų atidarymas su kitomis programomis	0	2
Papildomi įrankiai	0	2
Dokumentacija, pavyzdžiai	1	2

Taip pat įvertinama ir programinių agentų kūrimo būdų savybių svarba 37 lentelėje. Remiamasi į programinių agentų kūrimo būdų reikalavimus, o svarbumas išreikštas procentais. Visų savybių svarbumo suma yra 100%.

37 lentelė. Programinių agentų kūrimo būdų savybių svarbumas, įvertintas procentine išraiška

Savybės ir svarba, įvertinta procentais	Svarbumo procentas
Kuriamo agento programavimo kalba	10%
Standartų palaikymas	10%
Licenzija	25%
Papildomos programinės įrangos reikalavimas	15%
Palaikomi bylų formatai	10%
Išsaugotų bylų atidarymas su kitomis programomis	10%
Papildomi įrankiai	15%
Dokumentacija, pavyzdžiai	5%

Remiantis problemos analizės skyriaus medžiaga programinių agentų kūrimo būdus galima įvertinti taškais, taškai skirstomi pagal 38 lentelėje nurodytus taškus ir pagal savybės įvertinimą konkrečiam tipui (kuo daugiau savybės bruožų turi tipas, tuo daugiau taškų jam skiriama).

38 lentelė. Programinių agentų kūrimo būdų savybių įvertinimas taškais

Agentų kūrimo priemonės ir savybės	„Jason“	„3APL“	„GOAL“	„Weka“	„Jade“	„Jack“
Kuriamo agento programavimo kalba	1	1	1	2	2	2
Standartų palaikymas	1	1	0	0	1	1
Licenzija	1	1	1	1	1	0
Papildomos programinės įrangos reikalavimas	1	1	1	2	0	1
Palaikomi bylų formatai	3	1	2	1	4	4
Išsaugotų bylų atidarymas su kitomis programomis	1	0	0	0	2	1
Papildomi įrankiai	2	0	0	1	2	2
Dokumentacija, pavyzdžiai	2	1	2	1	2	2

Galutinis programinių agentų kūrimo būdų įvertinamas gaunamas 39 lentelėje. Šie taškai gaunami sudauginus savybės svarbumo procentą (37 lentelė) su savybių taškais (38 lentelė).

39 lentelė. Programinių agentų kūrimo priemonių galutinis įvertinimas taškais

Agentų kūrimo priemonės ir savybės	„Jason“	„3APL“	„GOAL“	„Weka“	„Jade“	„Jack“
Kuriamo	10	10	10	20	20	20

agento programavimo kalba						
Standartų palaikymas	10	10	0	0	10	10
Licenzija	25	25	25	25	25	0
Papildomos programinės įrangos reikalavimas	15	15	15	30	0	15
Palaikomi bylų formatai	30	10	20	10	40	40
Išsaugotų bylų atidarymas su kitomis programomis	10	0	0	0	20	10
Papildomi įrankiai	30	0	0	15	30	30
Dokumentacija, pavyzdžiai	10	5	10	5	10	10
Iš viso	140	75	80	105	155	135

Daugiausiai galutinių taškų surinko „Jade“ kūrimo priemonė.

3.3 Programinių agentų integravimo į interneto svetaines būdų tyrimo metodika

Programinių agentų integravimo į interneto svetaines būdų tyrime remiamasi į programinių agentų integravimo į interneto svetaines reikalavimus, savybių galimas mažiausias ir didžiausias taškų skaičius įvertintas 40 lentelėje.

40 lentelė. *Programinių agentų integravimo į interneto svetaines būdų savybių galimas įvertinimas taškais*

Savybės ir taškai	Mažiausias galimas taškų skaičius	Didžiausias galimas taškų skaičius
Programavimo kalbų palaikymas vertinamas	1	4
Veikimo pusė	0	1
Reikalavimai lankytojui	0	1
Vykdomi bylų formatai	1	2
Suderinamumas su duomenų bazėmis	1	2
Papildomi reikalavimai	0	2

Taip pat įvertinama ir programinių agentų integravimo į interneto svetaines būdų savybių svarba 41 lentelėje. Remiamasi į programinių agentų kūrimo būdų reikalavimus, o svarbumas išreikštas procentais. Visų savybių svarbumo suma yra 100%.

41 lentelė. *Programinių agentų integravimo į interneto svetaines būdų savybių svarbumas, įvertintas procentine išraiška*

Savybės ir svarba, įvertinta procentais	Svarbumo procentas
Programavimo kalbų palaikymas vertinamas	10%
Veikimo pusė	30%
Reikalavimai lankytojui	10%
Vykdomi bylų formatai	10%

Suderinamumas su duomenų bazėmis	15%
Išsaugotų bylų atidarymas su kitomis programomis	25%

Remiantis problemos analizės skyriaus medžiaga programinių agentų integravimo būdus galima įvertinti taškais, taškai skirstomi pagal 42 lentelėje nurodytus taškus ir pagal savybės įvertinimą konkrečiam tipui (kuo daugiau savybės bruožų turi tipas, tuo daugiau taškų jam skiriama).

42 lentelė. Programinių agentų integravimo į interneto svetaines būdų savybių įvertinimas taškais

Priemonės ir savybės	„Java applet“	„CGI“	„Apache Tomcat“	„Quercus“	„PHP/Java Bridge“
Programavimo kalbų palaikymas	1	4	1	1	1
Veikimo pusė	0	1	1	1	1
Reikalavimai lankytojui	0	1	1	1	1
Vykdomi bylų formatai	1	1	2	1	1
Suderinamumas su DB	2	2	2	2	1
Papildomi reikalavimai	0	0	2	0	1

Galutinis programinių agentų integravimo į interneto svetaines būdų įvertinamas gaunamas 43 lentelėje. Šie taškai gaunami sudauginus savybės svarbumo procentą (41 lentelė) su savybių taškais (42 lentelė).

43 lentelė. Programinių agentų integravimo į interneto svetaines būdų galutinis įvertinimas taškais

Priemonės ir savybės	„Java applet“	„CGI“	„Apache Tomcat“	„Quercus“	„PHP/Java Bridge“
Programavimo kalbų palaikymas	10	40	10	10	10
Veikimo pusė	0	30	30	30	30
Reikalavimai lankytojui	0	10	10	10	10
Vykdomi bylų formatai	10	10	20	10	10
Suderinamumas su DB	30	30	30	30	15
Papildomi reikalavimai	0	0	50	0	25
Iš viso	0	0	50	0	25

Daugiausiai galutinių taškų surinko „Apache Tomcat“ integravimo būdas.

3.4 Programinių agentų panaudojimo tyrimo metodika

Programinių agentų integravimo į interneto svetaines būdų tyrime remiamasi į programinių agentų integravimo į interneto svetaines reikalavimus, savybių galimas mažiausias ir didžiausias taškų skaičius įvertintas 44 lentelėje.

44 lentelė. Programinių agentų panaudojimo atvejų savybių galimas įvertinimas taškais

Savybės ir taškai	Mažiausias galimas taškų skaičius	Didžiausias galimas taškų skaičius
Sistemos tipas	0	1
Agento tipų savybės	1	2
Bendravimas su lankytoju	0	1
Atsižvelgimas į lankytojo veiksmus	0	1
Paieškos vykdymas kitose svetainėse	0	1
Informacijos kaupimas	0	1
Atliekami veiksmai	1	2

Taip pat įvertinama ir programinių agentų panaudojimo atvejų savybių svarba 45 lentelėje. Remiamasi į programinių agentų panaudojimo atvejų reikalavimus, o svarbumas išreikštas procentais. Visų savybių svarbumo suma yra 100%.

45 lentelė. Programinių agentų panaudojimo atvejų savybių svarbumas, įvertintas procentine išraiška

Savybės ir svarba, įvertinta procentais	Svarbumo procentas
Sistemos tipas	15%
Agento tipų savybės	5%
Bendravimas su lankytoju	15%
Atsižvelgimas į lankytojo veiksmus	30%
Paieškos vykdymas kitose svetainėse	10%
Informacijos kaupimas	10%
Atliekami veiksmai	15%

Remiantis problemos analizės skyriaus medžiaga programinių agentų panaudojimo atvejus galima įvertinti taškais, taškai skirstomi pagal 44 lentelėje nurodytus taškus ir pagal savybės įvertinimą konkrečiam tipui (kuo daugiau savybės bruožų turi tipas, tuo daugiau taškų jam skiriama).

46 lentelė. Programinių agentų panaudojimo atvejų įvertinimas taškais

Agentų panaudojimas ir savybės	„FireFly“	„Personal shopping assistant“	„AEGIS“	„Metacrawler Agent“	„Chatbot“	„Googlebot“	„Shopbot“
Sistemos tipas	1	0	0	1	1	1	1
Agento tipų savybės	2	2	2	2	1	2	2
Bendravimas	0	1	0	0	1	0	0
Atsižvelgimas į lankytojo	1	1	1	0	1	0	1

veiksmus							
Paieškos vykdymas kitose svetainėse	0	0	0	1	0	1	1
Informacij os kaupimas	1	1	1	0	0	1	1
Atliekami veiksmai	1	2	1	1	1	2	1

Galutinis programinių agentų panaudojimo atvejų įvertinamas gaunamas 47 lentelėje. Šie taškai gaunami sudauginus savybės svarbumo procentą (45 lentelė) su savybių taškais (46 lentelė).

47 lentelė. Programinių agentų panaudojimo atvejų galutinis įvertinimas taškais

Agentų panaudoji mas ir savybės	„FireFly“	„Personal shopping assistant“	„AEGIS“	„Metacrawler Agent“	„Chatbot“	„Googlebot“	„Shopbot“
Sistemos tipas	15	0	0	15	15	15	15
Agento tipų savybės	10	10	10	10	5	10	10
Bendravim as	0	15	0	0	15	0	0
Atsižvelgi mas į lankytojo veiksmus	30	30	30	0	30	0	30
Paieškos vykdymas kitose svetainėse	0	0	0	10	0	10	10
Informacij os kaupimas	10	10	10	0	0	10	10
Atliekami veiksmai	15	30	15	15	15	30	15
Iš viso	80	95	65	50	80	75	90

Daugiausiai galutinių taškų surinko „Personal shopping assistant“ ir „Shopbot“ panaudojimo atvejai.

3.5 Sistemos architektūros specifikacija

3.5.1 Dokumento paskirtis

Šio dokumento paskirtis paaiškinti programinio agento architektūrą, veikimo principą ir sudedamąsias dalis.

3. 5. 2 Architektūros pateikimas

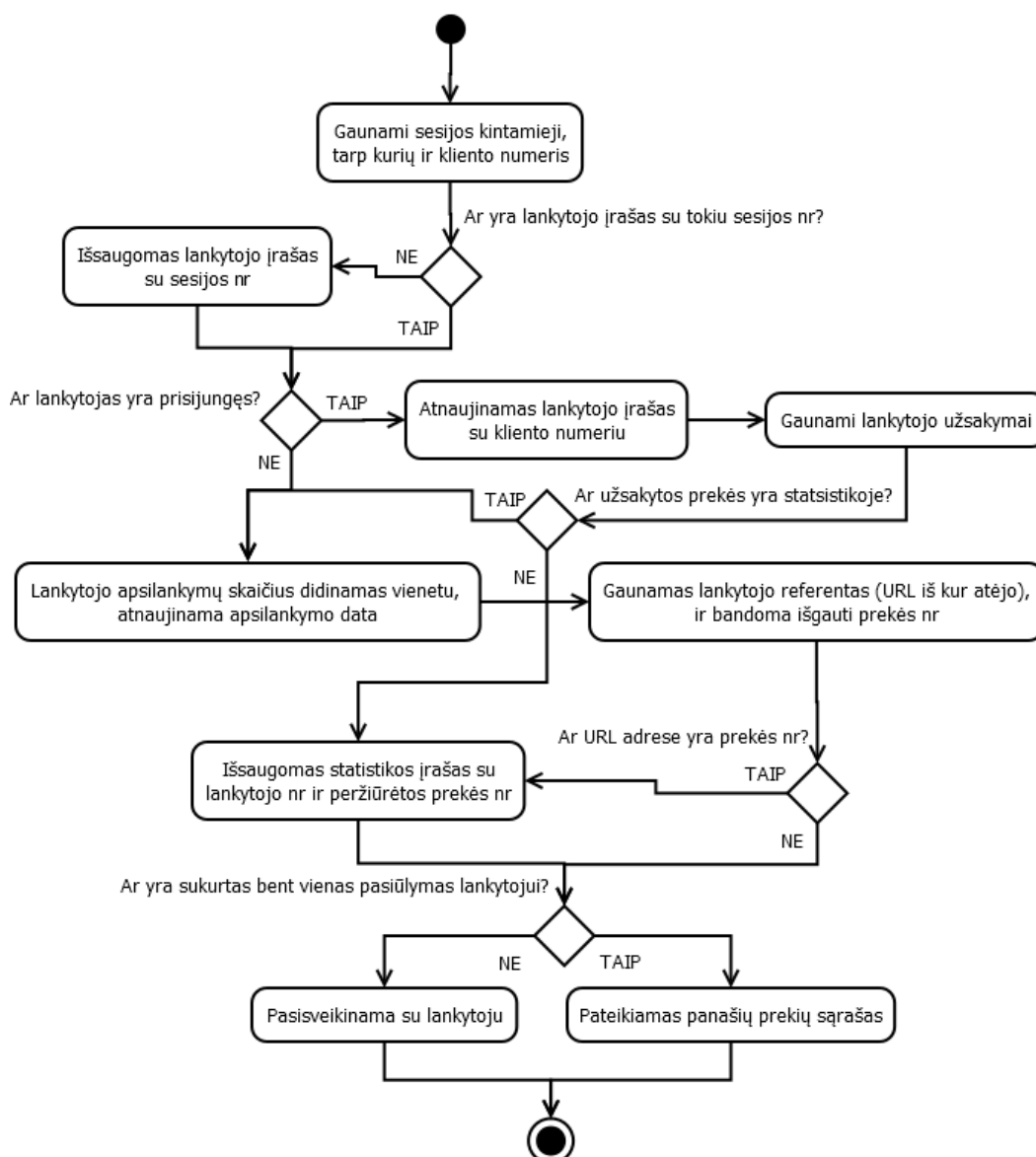
Architektūros specifikacijoje pateikiamos veiklos, būsenų, esybių ryšių bei komponentų diagramos. Jos paaiškina programinio agento architektūrą, jo komponentų tarpusavio veikimo principus, atliekamas funkcijas, veikimo etapus.

3. 5. 3 Architektūros tikslai ir apribojimai

Programinio agento tikslas yra suteikti lankytojui kuo daugiau patarimų, padėti išsirenkant prekę. Priminti apie vykstančias akcijas. Programinis agentas suteikia galimybę administratoriui bei gilių programavimo įgūdžių administruoti virtualaus asistento veikimą. Šis agentas nereikalauja išorinių serverių, svarbiausia yra serverio stabilumas kuriame jis yra talpinamas. Kadangi pats agentas dirba su tik su duomenų baze, visą lankytojų informaciją, peržiūrėtas prekes surenka pagalbinis „PHP“ kodas.

3. 5. 4 Sistemos dinaminis vaizdas

Virtualaus asistento informacijos rinkimas ir prekių siūlymas lankytojui

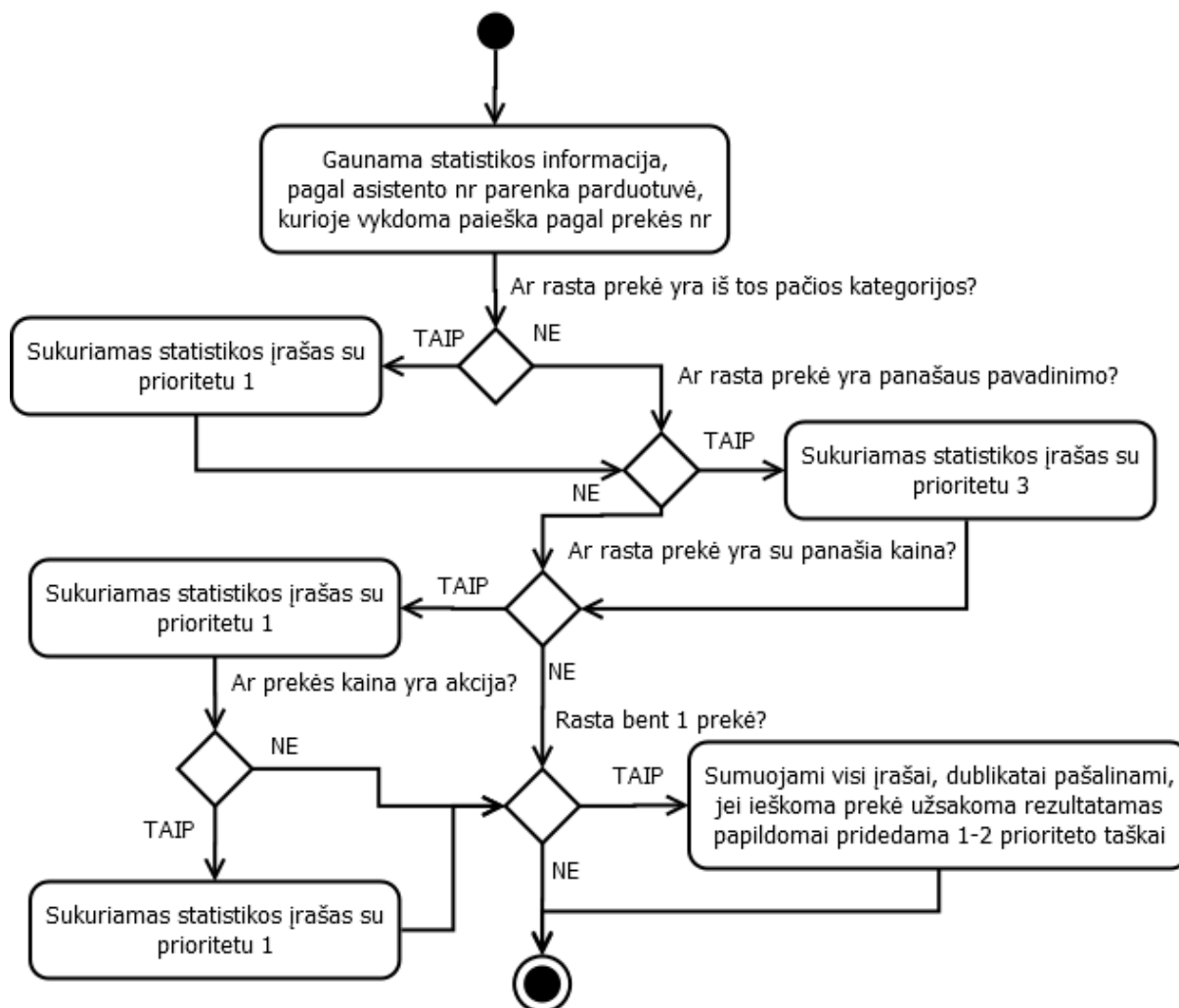


7 pav. Pagalbinio kodo informacijos rinkimo ir prekių siūlymo lankytojui atvaizdavimas

Diagramoje pateiktas el. parduotuvės virtualaus asistento informacijos surinkimo ir prekių siūlymo lankytojui veikimo principas (žr. 7 pav.). Iš pradžių tikrinama ar lankytojas jau yra buvęs

svetainėje, sukurdamas naują lankytojo įrašą jei lankytojas yra pirmą kartą. Po to tikrinama ar lankytojas yra prisijungęs el. parduotuvėje ir ar turi užsakymų, jei taip automatiškai sukuriama pasiūlymai pagal užsakymo prekes. Toliau žiūrima į lankytojo „URL“ adresą iš kur atėjo, ar lankytojas peržiūrėjo prekę, jei taip automatiškai sukuriama pasiūlymai pagal prekės nr. Galiausiai asistentas tikrina ar duomenų bazėje yra lankytojo statistikos įrašų, jei ne – tiesiog pasisveikinama su lankytoju be kreipinio (jei lankytojas yra neprisijungęs), atsižvelgiant į serverio laiką (pvz.: labas vakaras jei serverio laikas rodo virš 17 val.), jei taip – pateikimas pasiūlymų sąrašas surikiuotas pagal prioritetą (didžiausią prioritetą turintys pasiūlymai rodomi aukščiausiai). Šis programinis kodas vykdomas kiekvieną kartą kraunantis bet kokiam el. parduotuvės puslapiui.

Virtualaus asistento pasiūlymų kūrimas



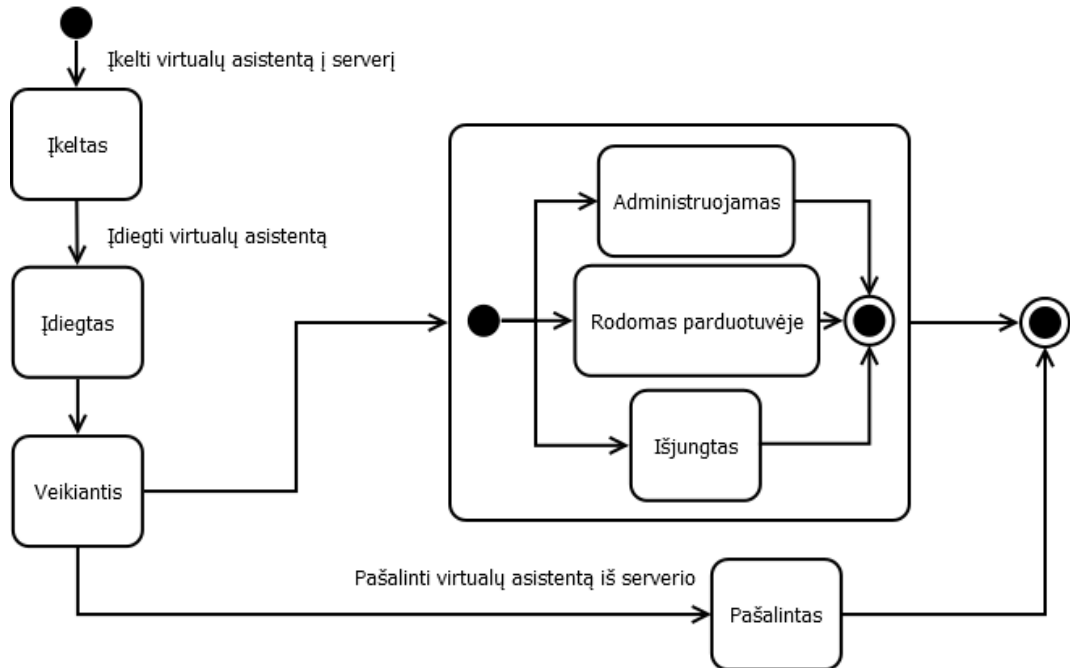
8 pav. Virtualaus asistento pasiūlymų kūrimo atvaizdavimas

Diagramoje pateiktas virtualaus asistento pasiūlymų lankytojui kūrimo veikimo principas (žr. 8 pav.). Iš pradžių tikrinama ar rasta prekė parduotuvėje yra iš tos pačios kategorijos, kaip ir ieškoma, jei taip – sukuriama statistikos įrašas. Toliau vykdoma paieška tikrina ar prekės pavadinimas arba aprašymas yra panašaus, jei taip – vėl sukuriama statistikos įrašas. Toliau tikrinama prekės kaina, jei ji akcija, suteikiama papildomai prioriteto taškų. Pabaigoje tikrinama ar rasta bent viena prekė atitinkanti bent vieną iš visų kriterijų, jei taip – visi vienodo prekės numerio įrašai sumuojami (jų prioritetai), o dublikatai pašalinami. Taip pat atsižvelgiama ar ieškoma prekė buvo peržiūrėta ar užsakyta, jei prekė buvo užsakyta papildomai suteikiama prioriteto taškų, o užsakyta prekė pašalinama

iš pasiūlymų sąrašo (klientui nebesiūloma jau nusipirkto prekės). Šis programinis kodas vykdomas kas minutę automatiškai.

Virtualaus asistento būsenos

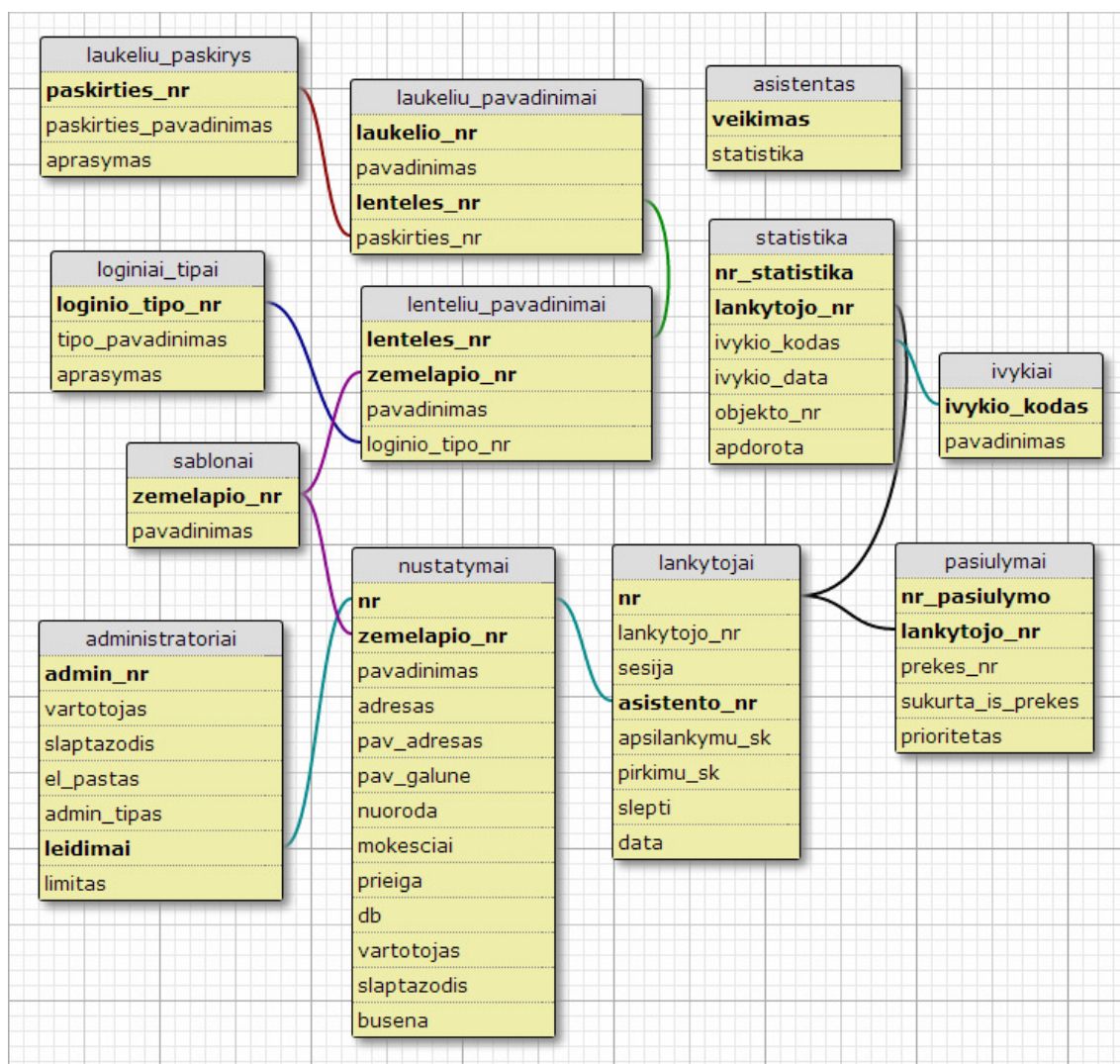
Būsenų diagrama parodo kaip keičiasi virtualaus asistento būseną serveryje (žr. 9 pav.). Iš pradžių programinis agentas turi būti įkeltas (programinės bylos), tik po to kai programinis agentas įdiegiamas, jis tampa veiksnus. Į veikimo būseną įeina administravimo, rodymo el. parduotuvėje, išjungimo būsenos. Virtualus asistentas taip pat gali turėti ir pašalinimo būseną, kai visos sisteminės jo bylos ir duomenų bazės lentelės yra pašalinamos. Taip pat bet kuriuo metu virtualaus asistento veikimas gali būti pristabdytas, o vėliau vėl paleistas.



9 pav. *Virtualaus asistento būsenos*

3. 5. 5 Virtualaus asistento duomenų bazės vaizdas

Pagrindiniai virtualaus asistento nustatymai (žr. 10 pav.) būtinai nurodo agento būseną (įjungtas, išjungtas), taip pat svarbūs ir serverio kuriame talpinama el. parduotuvė duomenys (duomenų bazės prisijungimo informacija), el. parduotuvės duomenų bazės struktūros nustatymai (reikalinga agentui gauti prekių, kategorijų, klientų, užsakymų, informaciją). Kiti svarbūs duomenis – lankytojo informacija, čia reikia užfiksuoti apsilankymo datą, bei sesijos kintamąjį, jei tai tas pats lankytojas ir apsilankymų skaičių, jei lankytojas yra užsiregistravęs, duomenis (kliento identifikavimo numerį, vardą, el. pašto adresą ir t.t.) agentas pasiima jau iš esančios el. parduotuvės duomenų bazės. Saugant statistiką būtina naudoti jau užfiksuoto lankytojo numerį, taip pat išsaugoma įvykio data, tipas (pvz.: prekė), objekto numeris (pvz.: 120, tai prekės identifikavimo numeris paimtas iš el. parduotuvės). Panašiai ir su pasiūlymais, būtina naudoti jau užfiksuoto lankytojo numerį, prekės numeris paimamas iš el. parduotuvės, o prioritetą nustato agentas, pagal tai kiek dažnai ta pati prekė pasikartoja siūlymuose.

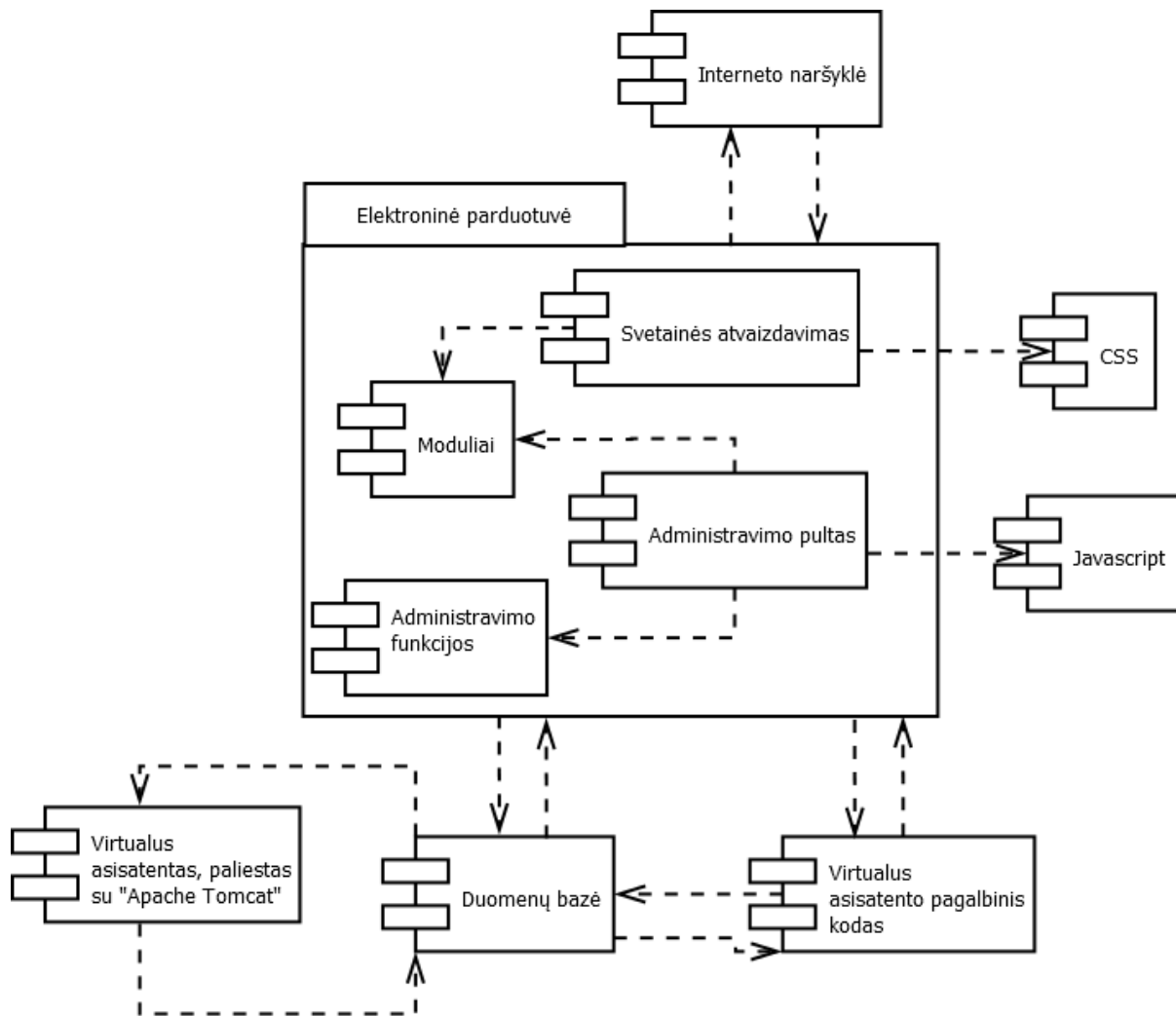


10 pav. Esiybių ryšių modelis

Lentelių detalizavimas pateikiamas 1-me priede.

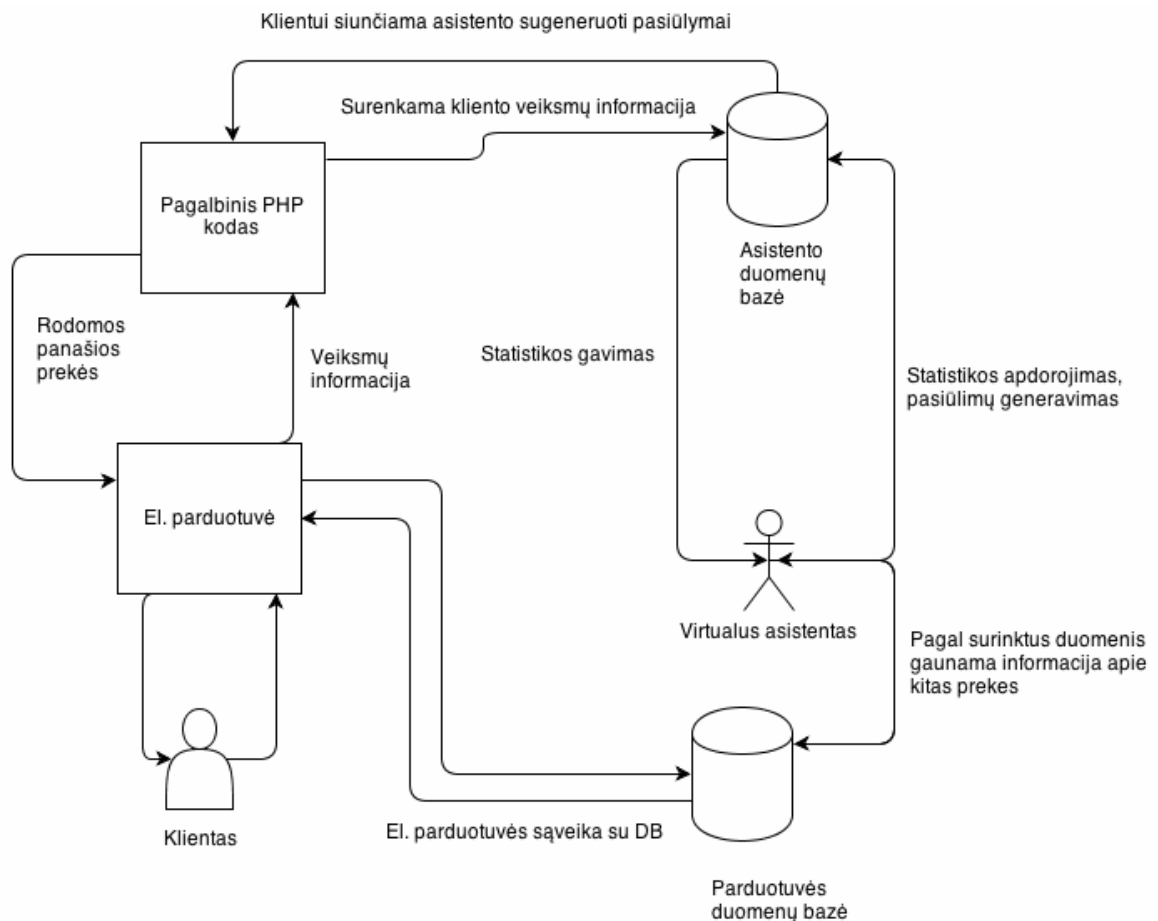
3. 5. 6 Komponentų diagrama

Šioje diagramoje pateikiami sistemos komponentai, bet ryšiai tarp jų (žr. 11 pav.). Interneto naršyklė kreipiasi į el. svetainę. El. parduotuvė jungiasi prie duomenų bazės duomenims gauti ir išsaugoti. Svetainės atvaizdavimui naudojamas dizaino failas „CSS“, taip pat svetainės atvaizdavimui naudojami papildomi moduliai, o administravimo pultas naudoja administravimo ir „JavaScript“ funkcijas. Pats virtualus asistentas jungdamasis į el. parduotuvę pirmiausia naudojasi vienu iš agentų integravimo į interneto svetainės būdų, kadangi jis parašytas „Java“ programavimo kalba. Virtualus asistentas taip pat naudojasi pagalbiniu „PHP“ kodu, kuris padeda lengviau funkcionuoti agentui (agentas dirba tik su informacija iš duomenų bazių). Tokių būdu mažiau apkraunamas serveris, atsiveria galimybė toki agentą naudoti žymiai daugiau serverių, kurie net ir nepalaiko „Java“ kodo. Svarbiausia, kad pagrindinis serveris, kuriame talpinamas agentas turėtų palaikymą. Taip pat nuo el. parduotuvių serverių (jei jie yra kitame serveryje) nuimama nemaža apkrova duomenų bazei.



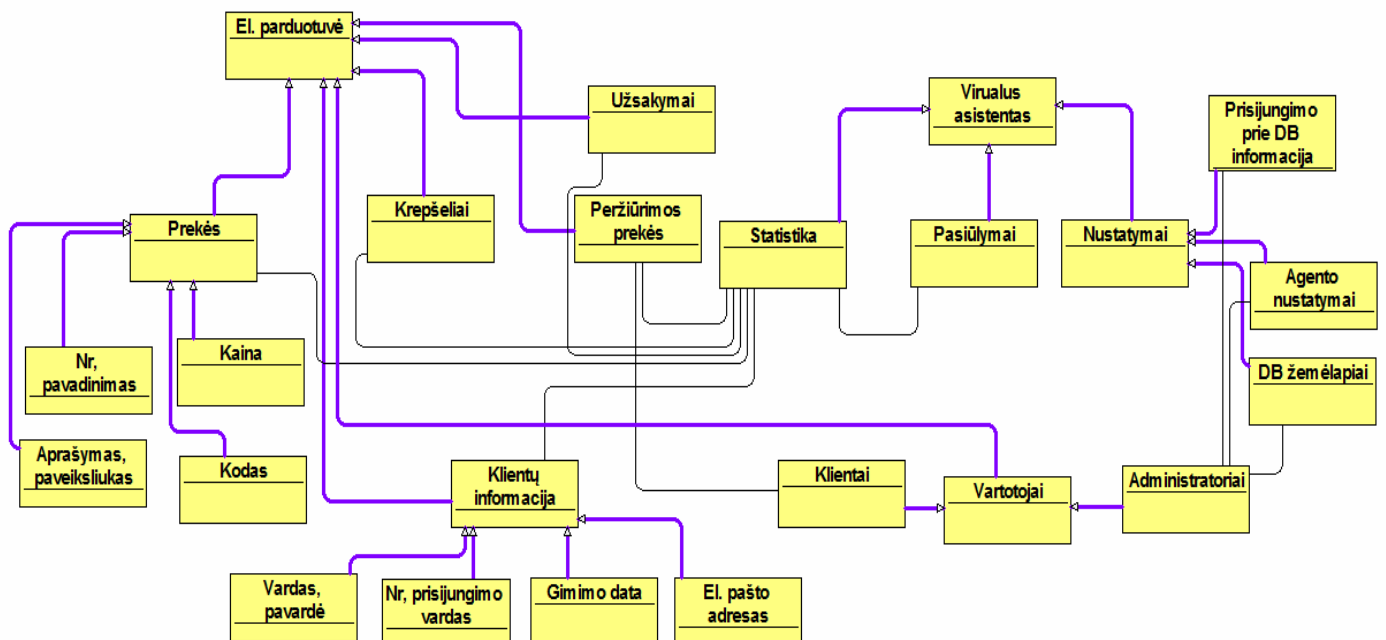
11 pav. Komponentų diagrama

Šioje diagramoje (žr. 12 pav.) pateikiami detalesni sistemos komponentai, bet ryšiai tarp jų (žr. 12 pav.). El. parduotuvė sąveikauja su klientu ir pagalbinio „PHP“ kodu, kuris surenka kliento veiksmų informaciją ir išsaugo ją asistento duomenų bazėje. Toliau kas kiek laiko virtualus asistentas peržiūri įrašus ir sukuria pasiūlymus, juos įrašydamas į savo duomenų bazę. Tada pagalbinis „PHP“ kodas atvaizduoja pasiūlymus klientui. Tokiu būdu buvo išvengta būtinumo el. parduotuvės serveryje naudoti „Java“ palaikymo. Serveris gali būti įprastas tik su „Apache“ ir „MySQL“ servisais, todėl tai leidžia virtualius asistentus naudoti žymiai daugiau el. parduotuvių. Patį virtualų asistentą dar sudaro ir vidiniai agentai: paleidimo agentas, duomenų bazės valymo ir senų įrašų šalinimo agentai, kurie reikalui esant aktyvuojami pagrindinio asistento. Pagrindinis agentas aktyvuojamas kas minutę, valymo agentai kas valandą, jie pašalina neregistruotų lankytojų įrašus, kurie yra 5 ar daugiau valandų senumo.



12 pav. Detalesnė komponentų diagrama

3.5.7 Ontologija



13 pav. Programinio agento ir el. parduotuvės ontologija

Ontologija parodo kaip bendrauja virtualus asistentas su klientais (svetaines lankytojai) ir administratoriais (žr. 13 pav.). Virtualus asistentas kaupia statistiką pagal tai kokius veiksmus atliko klientai, kokias prekes peržiūrėjo. Po to pagal atliktus statistikos skaičiavimus, pateikiami pasiūlymai.

Taip pat asistentas surenka klientų informaciją (vardą, pavardę, el. pašto adresą...), prekių informaciją (pavadinimą, numerį, kainą, kodą...), užsakymų, krepšelių informaciją iš el. parduotuvės duomenų bazės. Žmonės turi turėti arba kliento arba administratoriaus statusą, abiejų turėti negali. Administratoriai identifikuojami, kaip žmonės, turintys galimybę keisti virtualaus asistento nustatymus, o klientai – tie kurie gali peržiūrėti prekes, pateikti užsakymus, dėti prekes į krepšelius.

3. 5. 8 Kokybė

Išplečiamumas

Agentas naudoja funkcijas veiksams atlikti. Todėl galima nesunkiai papildyti veiksmų sąrašą, tereikia sukurtą funkciją įdėti į funkcijų bylą ir tada galima kreiptis į jas iš el. parduotuvės.

Pernešamumas

Virtualų asistentą galima pernešti į kitą importavus duomenų bazę ir nukopijavus visas agento bylas į serverį. Taip pat pasikeitus duomenų bazės susijungimo nustatymus specialioje byloje.

Patikimumas

Sklandus sistemos darbas priklauso nuo serverio stabilumo kuriame talpinama virtualus agentas.

3. 6 Realizuotos sistemos testavimas

3. 6. 1 Įvadas

Dokumentas skirtas ištestuoti kuriamą virtualių asistentų sistemą.

Testavimo tikslai

Pagrindinis tikslas išbandyti realizuotos sistemos funkcionalumą, spartą. Taip pat rasti kuo daugiau klaidų virtualių sistemų sistemos veikime, rasti visas įmanomas saugumo, našumo spragas. Pabandyti sistemą pakeisti specialiu kodu, taip imituojant piktavalių asmenų veiksmus.

Testuojami ir standartiniai el. parduotuvėse integruoti prekių pasiūlymų moduliai, siekiant palyginti sukurtos virtualių asistentų sistemos veikimą.

Testavimo būdai

Funkcionalumas testuojamas tiesiog išbandant visas įmanomas funkcijas praktiškai, bandant įvesti privalomus duomenis, atliekant reikalaujamus veiksmus. Našumas testuojamas į programinį kodą įterpus specialią komandą, kuri nustato puslapių krovimosi, komandų atlikimo trukmę. Suderinamumo testavimas atliekamas bandant pakrauti puslapius skirtingose interneto naršyklėse ir kompiuteriuose, žiūrima ar puslapiai ir virtualaus asistento veikimas nepasikeitė. Saugumas testuojamas imituojant piktavališkų asmenų galimus veiksmus, siekiant pakenkti, sutrikdyti sistemos darbą, stebima ar tai pavyksta padaryti.

Testavimo resursai

Pirmas kompiuteris. Techninės įrangos resursai:

Procesorius: „Intel Core 2 E6600“ 2.40 Ghz, operatyvioji atmintis: 3 Gb, kietasis diskas 320 Gb, vaizdo plokštė: „GeForce GTX 550 Ti“. Monitorius „LG FLATRON IPS235“.

Programinės įrangos resursai:

Operacinė sistema „Windows 8“, naudojamos interneto naršyklės: „Opera“ 11.0 versija, „Mozilla Firefox“ 20 versija, „Google Chrome“ versija 26.0.1410.

Antras kompiuteris. Techninės įrangos resursai:

Procesorius: „Intel Core i3 370M“ 2.4 GHz, operatyvioji atmintis: 2 Gb, kietasis diskas 500 Gb, vaizdo plokštė: „Intel GMA HD“. Monitorius „15.6" LED backlit“

Programinės įrangos resursai:

Operacinė sistema „Windows Vista“, naudojamos interneto naršyklės: „Opera“ 10.0 versija, „Mozilla Firefox“ 20 versija, „Internet Explorer“ versija 10.0.

3. 6. 2 Sistemos funkcijų testavimas

Testavimo atvejai pateikiami 2-me priede.

3. 6. 3 Sistemos kokybinių charakteristikų testavimas

Našumo testavimas

Testavimo atvejai pateikiami 3-me priede.

Suderinamumo testavimas

Testavimo atvejai pateikiami 4-me priede.

Saugumo testavimas

Testavimo atvejai pateikiami 5-me priede.

4. TYRIMO ANALIZĖ

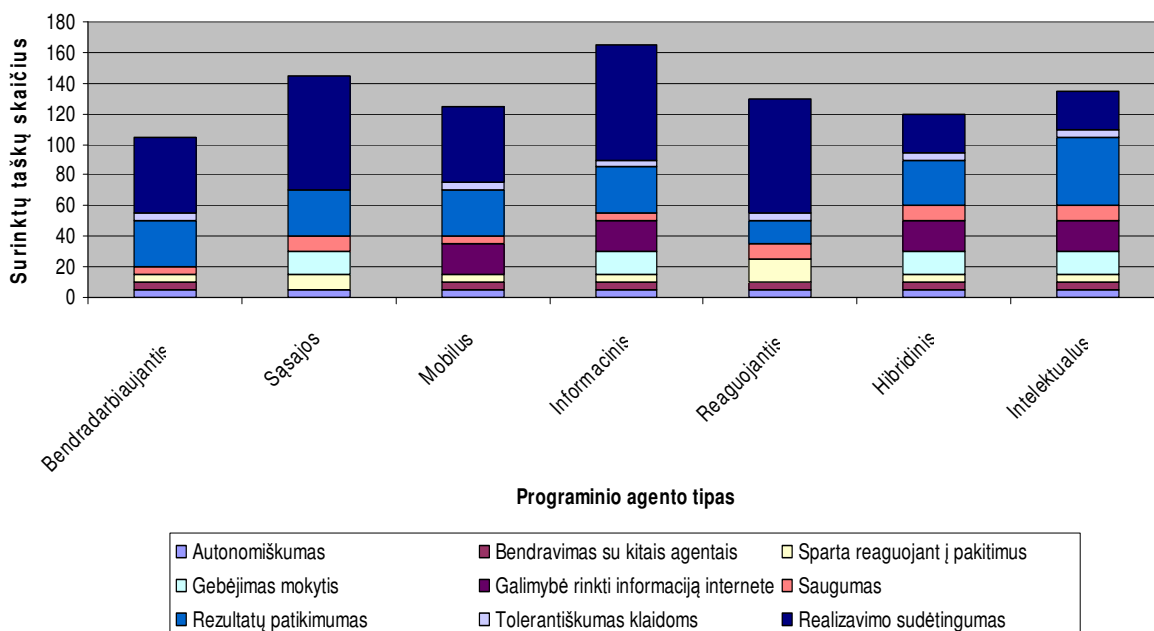
4.1 Programinių agentų tyrimo analizė

Apžvelgus problemos analizę galima pastebėti, kad iš visų programinių agentų tipų labiausiai išsiskiria intelektualieji ir hibridiniai, kurie turi daugiausiai galimybių iš visų agentų tipų, tačiau jų realizavimas yra pats sudėtingiausias ir ne visada tikslingas. Esminis skirtumas tarp hibridinio ir intelektualaus agento tai, kad hibridinis agentas yra sudarytas iš kitų agentų, kurie sąveikauja sluoksnių sąryšiais, o intelektualus veikia kaip vienas agentas.

Mobilus ir informacinis, kurie išsiskiria iš kitų agentų tipų savo paskirtimi ir veikimu. Abejų agentų tipai skirti informacijos paieškai, bet skiriasi jų veikimas. Mobilieji agentai keliauja iš serverio į serverį, tuo tarpu vartotojas gali atsijungti nuo sistemos ar užduoti naują užklausą, tokiu būdu agentai vienu metu gali vykdyti kelias užklausas vienu metu. Informacinis agentas labiau orientuotas į informacijos susistemimą, filtravimą ir informacijos išrinkimą. Jie gali būti tiek mobilūs, tiek veikti vartotojo kompiuteryje. Abu agentų tipai taip pat gali bendradarbiauti su kitais agentais, tačiau dažniausiai tik informaciniai agentai gali mokytis ir numatyti naujas vartotojo paieškas.

Reaguojantys ir bendradarbiaujantys agentai taip pat yra panašūs, nes jie dažniausiai naudojami po kelis. Reaguojantis agentas labai paprastas, jis labai priklausomas nuo kūrėjo aprašytų situacijų kiekio, o bendradarbiaujantis agentas priklausomas nuo kitų agentų. Reaguojantis agentai dažniausiai jungiami į vieną sistemą, tokiu būdu galima lengviau spręsti sudėtingus uždavinius, tuo tarpu bendradarbiaujantys agentai naudoja derybas, stengiamasi sumažinti resursų naudojimą. Tačiau tokių agentų reakcijos laikas yra lėtas, realizavimas sudėtingas lyginant su reaguojančiu agentu.

Programinių agentų palyginimo diagrama



14 pav. Programinių agentų palyginimo diagrama

Iš atlikto tyrimo rezultatų, 2 lentelės, galima sudaryti programinių agentų palyginimo diagramą, kad vaizdžiau būtų galima pamatyti galutinių taškų pasiskirstymą.

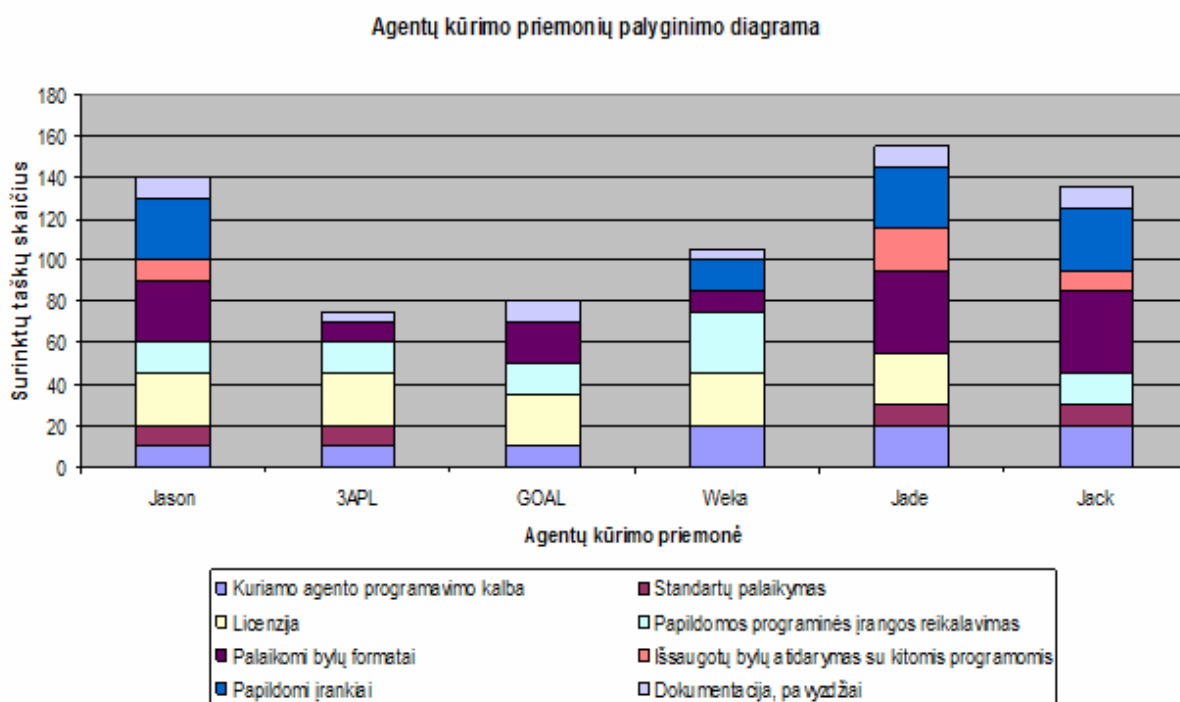
Programinių agentų palyginimo diagramoje (žr. 14 pav.) pastebimi informacinis ir sąsajos agentai. Šie agentų tipai labiausiai atitiktų elektroninės parduotuvės virtualaus asistento reikalavimus. Geriausia būtų panaudoti abiejų agentų savybes kuriant asistentą, tokiu būdu agentas ne tik surinks reikiamą informaciją, ją susistemins, analizuos, bet tuo pačiu galės bendrauti su svetainės lankytojais. Taip pat galima pastebėti intelektualųjį agentą, tačiau šiuo atveju šis agentas bus per daug sudėtingas, o jo realizavimas užtruktų per ilgai.

4.2 Programinių agentų kūrimo būdų analizė

Apžvelgus problemos analizę, galima pastebėti, kad visos programinių agentų kūrimo priemonės paremtos „Java“ programavimo kalba, patys agentai naudoja įvairias kalbas, tačiau dažniausiai pasitaikanti yra „Java“. Dauguma priemonių naudoja „FIPA“ (Foundation for Intelligent Physical Agents) standartą agentų tarpusavio bendravimui. Kai kurie įrankiai turi papildomų įrankių, padedančių agentų realizavime.

„Jason“ agentai naudoja „3APL“ ir „GOAL“ agentai naudoja skirtingas ir savitas programavimo kalbas. Todėl jų pritaikymas tampa ribotas, norint patalpinti tokius agentus serveryje tekstu juo perdaryti. Todėl iš visų išskirti būtų galima „Jade“ ir „Jack“ sistemas. Abejū sistemų parašyti agentai naudoja „Java“ programavimo kalbą, tai užtikriną universalesnį agentų panaudojimą. Taip pat abejos priemonės turi papildomus įrankius pritaikymui interneto svetainėms. Šių sistemų skirtumas, kad „Jade“ yra atviro kodo, o „Jack“ mokama, kuri turi daugiau galimybių ir yra labiau išbaigta.

Taip reikia nepamiršti ir „Weka“ rinkinio, kuris kaip autoriai teigia yra skirtas duomenų rinkimo (angl. data mining) agentams, kurie gali mokytis. Tačiau šis rinkinys gali būti panaudotas ir kitokių besimokančių agentų kūrimui.



15 pav. Programinių agentų palyginimo diagrama

Apžvelgiant atlikto tyrimo rezultatus, 3 lentelę, galima sudaryti programinių agentų kūrimo būdų palyginimo diagramą, kad vaizdžiau būtų galima pamatyti galutinių taškų pasiskirstymą.

Iš diagramos (žr. 15 pav.) galima išskirti „Jade“ kūrimo priemonę. Šis įrankis labiausiai atitiktų virtualaus asistento reikalavimus. Jis palaiko pagrindinius reikiamus formatus, reikiamus integravimui į internetinę svetainę. Taip pat „Jade“ atitinka „Fipa“ standartus. Iš priemonių galima išskirti ir „Jack“, kuri turi daugiausiai geriausių savybių, tačiau didelis trūkumas, kad priemonė yra komercinė. „Jason“ taip pat nedaug atsilieka nuo „Jade“, tačiau vienas didžiausių jos trūkumų yra agentų formatai, kurie palaikomi tik pačios programos, todėl šiuos agentus integruoti būtų žymiai sudėtingiau.

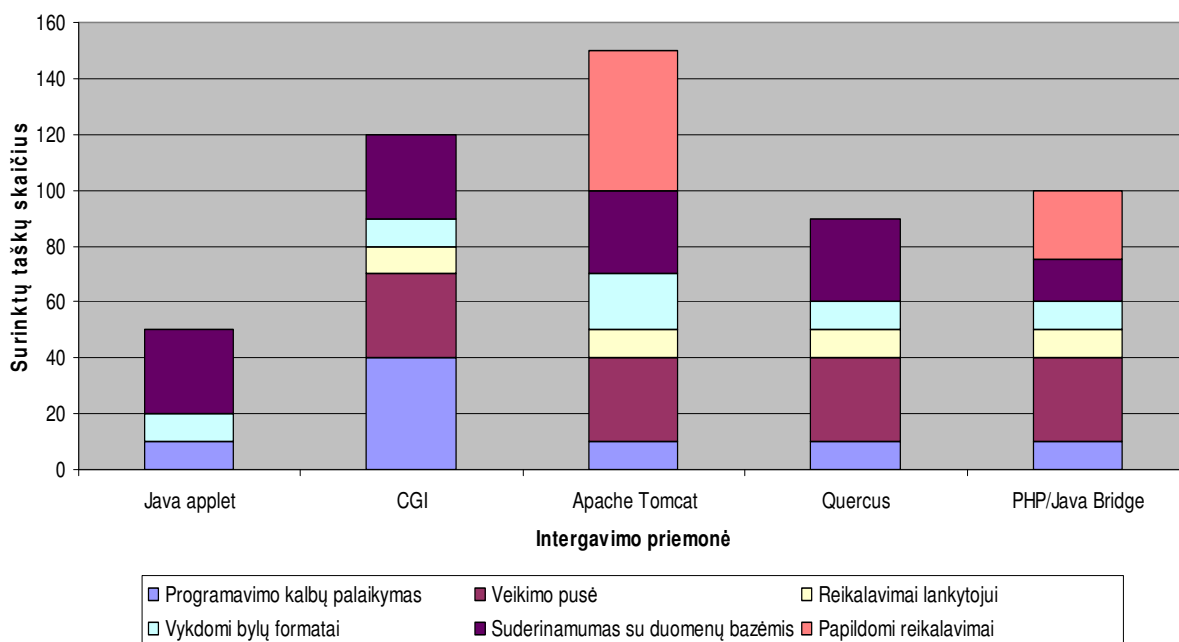
4.3 Programinių agentų integravimo į interneto svetaines būdų analizė

Apžvelgus priemones, padedančias integruoti programinius agentus į interneto svetaines, aprašytas problemos analizės skyriuje, galima išskirti „Java applet“ naudojimą, kuris yra palaikomas praktiškai visų naršyklių, nesudaro sunkumų perverčiant „Java“ kodą į „applet“ programėles. Tačiau agentai veikti galėtų aktyvuojami interneto naršyklės pagalba, tai reiškia, kad tokie agentai prarastu savo autonomiškumą.

„Common Gateway Interface“ protokolas labiausiai palaikomas serveriuose, jis palaiko daug programavimo kalbų. Be to tokie agentai gali veikti pačiame serverio viduje tai leistų jiems veikti autonomiškai, tačiau susiduriama su serverio resursų išnaudojimo problemomis. Taip pat reikia specialiai paruošti programas, kurios bus paleidžiamos serveryje.

Išskirtinė technologija „Quercus“ skirtingai nei visos kitos priemonės naudoja savo architektūrą ir serverį, todėl „Java“ agentai ir svetainės kodai privalo būti perrašomi. Tačiau ši technologija užtikrina spartų ir saugų svetainės veikimą, suderinta su „Java“ programomis.

Priemonių, padedančių integruoti agentus į interneto svetaines palyginimo diagrama



16 pav. Programinių agentų palyginimo diagrama

Peržiūrint atlikto tyrimo rezultatus, 4 lentelę, galima sudaryti programinių agentų integravimo į interneto svetaines būdų palyginimo diagramą, kad vaizdžiau būtų galima pamatyti galutinių taškų pasiskirstymą.

Diagramoje matome (žr. 16 pav.) daugiausiai taškų surinkusią „Apache Tomcat“ priemonę. Šis įrankis labiausiai atitiktų virtualaus asistento reikalavimus, taip pat nereikalauja sudėtingos serverio konfigūracijos ir lankytojo nevargina prašydamas įdiegti papildomų įskiepių. O svarbiausia savybę tai veikimo pusė serveryje, pavyzdžiui „Java Applet“ veikia tik kliento pusėje, jis automatiškai nebegali būti virtualiu asistentu. Taip pat galima pastebėti, kad CGI (Common Gateway Interface) nedaug atsiliko, jis palaiko daugiausiai programavimo kalbų, tačiau svetainės ir agento kodą reikia koreguoti, norint leisti jiems veikti kartu.

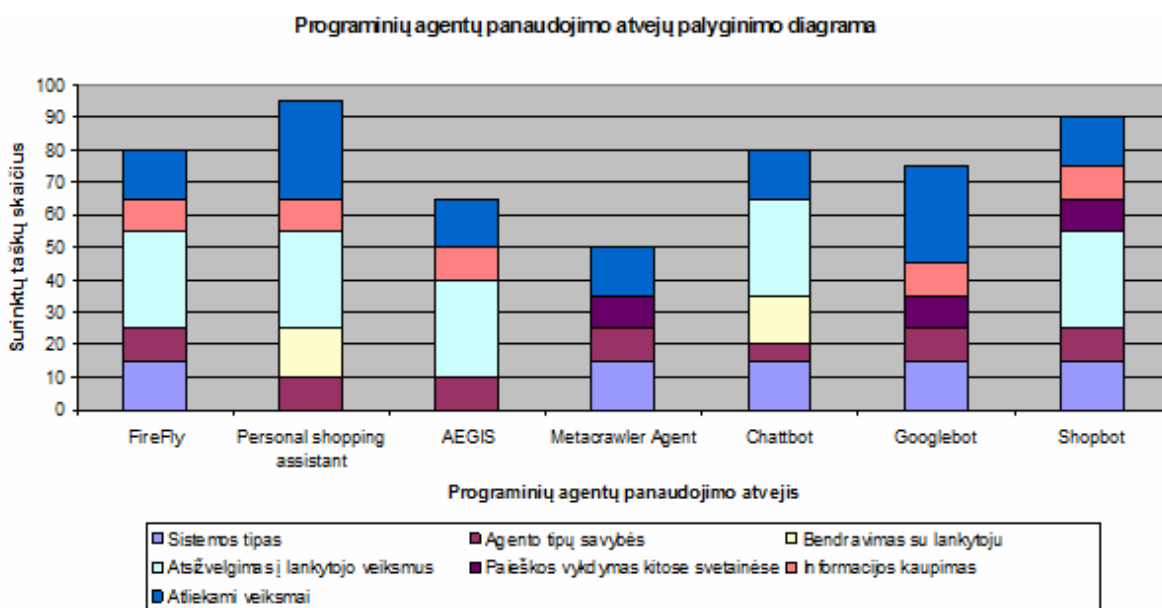
4.4 Programinių agentų panaudojimo analizė

Apžvelgus agentų panaudojimo atvejus, aprašytus problemos analizės skyriuje, galima pastebėti, kad nei vienas agentas nėra visapusiškas. Tačiau dauguma agentų, pasižymi kelių agentų tipų savybėmis. Visi agentai sukurti tam tikram tikslui, todėl tai tik patvirtina teiginį, kad nėra tikslinga kurti idealų intelektualų agentą, kuris pasižymėtų visomis agentų savybėmis. Programiniai agentai padeda sukurti universalesnes sistemas, kurios sugeba priimti sprendimus vienu ar kitu atveju. Kai kurie agentai tiesiog palengvina vartotojo veiklą.

Agentų kūrėjai viešai neatskiedžia savo agentų realizavimo technologijų. Tačiau internete galima rasti informacijos, kad daugiausiai agentų kūrime naudojama „Java“ programavimo kalba.

Taip pat galima pastebėti, kad dauguma agentų yra integruojami į internetines svetaines.

Iš atlikto tyrimo rezultatų, 5 lentelės, galima sudaryti programinių panaudojimo atvejų palyginimo diagramą, kad vaizdžiau būtų galima pamatyti galutinių taškų pasiskirstymą.



17 pav. Programinių agentų panaudojimo atvejų palyginimo diagrama

Diagramoje (žr. 17 pav.) galima išskirti „Personal shopping assistant“ agentą. Šis programinis agentas labiausiai atitiktų virtualaus asistento reikalavimus, sugeba kaupti informaciją savarankiškai ir atsižvelgia į lankytojų veiksmus. Vėliau siūlo lankytojui prekes, pagal lankytojo įpročius. Tačiau ši technologija įdiegta ne internetiniame puslapyje, o uždaroje sistemoje, kuri nenaršo interneto, o tik savo duomenų bazę. „Firefly“ taip pat panašus sprendimas, kuris atsižvelgia į lankytojo veiksmus, tačiau jis veikia gana uždariai, nevykdo paieškų internete. Skirtumas tik tas, kad vienas agentas siūlo prekes, kitas mėgstamas dainas. „Shopbot“ veikimo principas šiek tiek kitoks, jis ieško informacijos

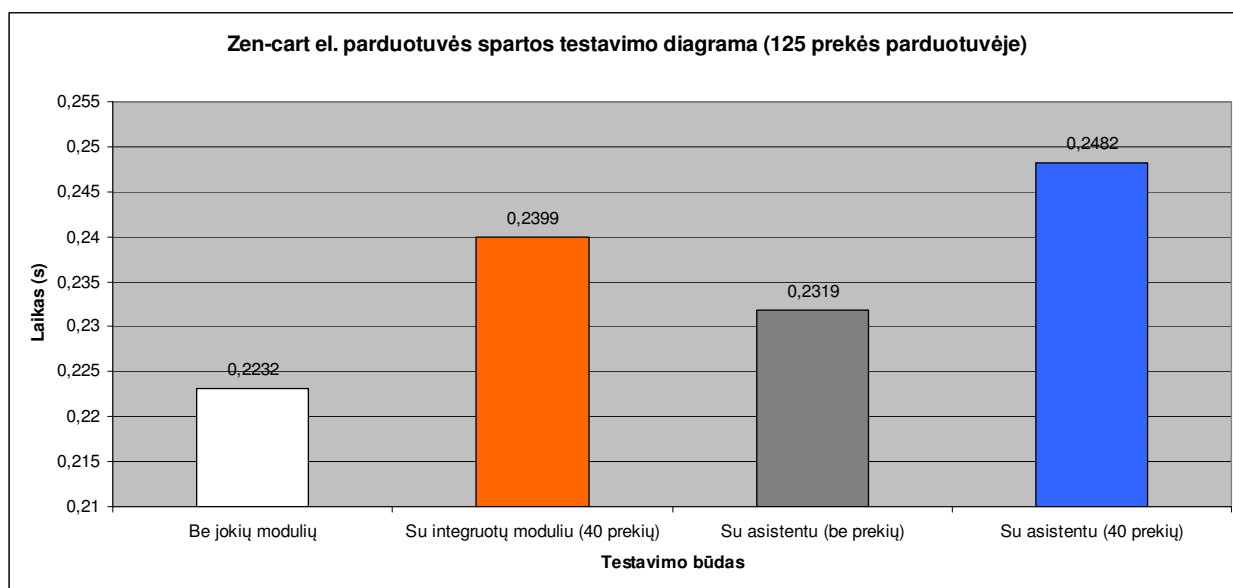
internete ir lankytojui pateikia kainas ir kelių parduotuvių realiu laiku. Taip pat galima pastebėti ir „Chatbot“ agentą, kuris bendrauja su lankytojais, nukreipia į tinkamus puslapius, atsako į klausimus, tačiau jis nekaupia informacijos apie klientą. Geras virtualus asistentas išėtų sujungus kelių agentų, pvz.: „Personal shopping assistant“, „Shopbot“, „Chatbot“ savybes.

4.5 Realizuotos sistemos testavimo analizė

Programinis agentas, virtualių asistentų valdymo skydas ir pagalbinis kodas paruošti naudojimui. Matome iš testavimo rezultatų, kad jie nebeturi didelių spragų. Visos funkcijos atitinka keliamus reikalavimus, visose funkcijose veikia turinio įvedimo filtravimo funkcijos. Valdymo skydas buvo testuojamas, kad puslapiai būtų sugeneruojamai greičiau nei per 0.1 sekundės, testuotų puslapių generavimo laikas atitiko keliamus reikalavimus ir buvo sugeneruojami greičiau nei per 0.1 sekundės. El. parduotuvės buvo testuojamos, kad puslapiai būtų sugeneruojamai greičiau nei per 0.3 sekundės, testuotų puslapių generavimo laikas atitiko keliamus reikalavimus ir buvo sugeneruojami greičiau nei per 0.3 sekundės. Sistemos saugumo testavimas neaptiko jokių saugumo klaidų, sistema atspari SQL, XSS atakoms. Vienintelė likusi problema - „Internet Explorer“ interneto naršyklės dizaino iškraipymai virtualių asistentų valdymo skyde.

Sistemos veikimas testuotas kartu ir su integruotais panašių prekių moduliais, kurie diegiami kartu su el. parduotuvėmis. Tačiau virtualus asistentas praktiškai mažai nusileido spartos atžvilgiu integruotiems moduliams, o vienoje parduotuvėje netgi veikė sparčiau.

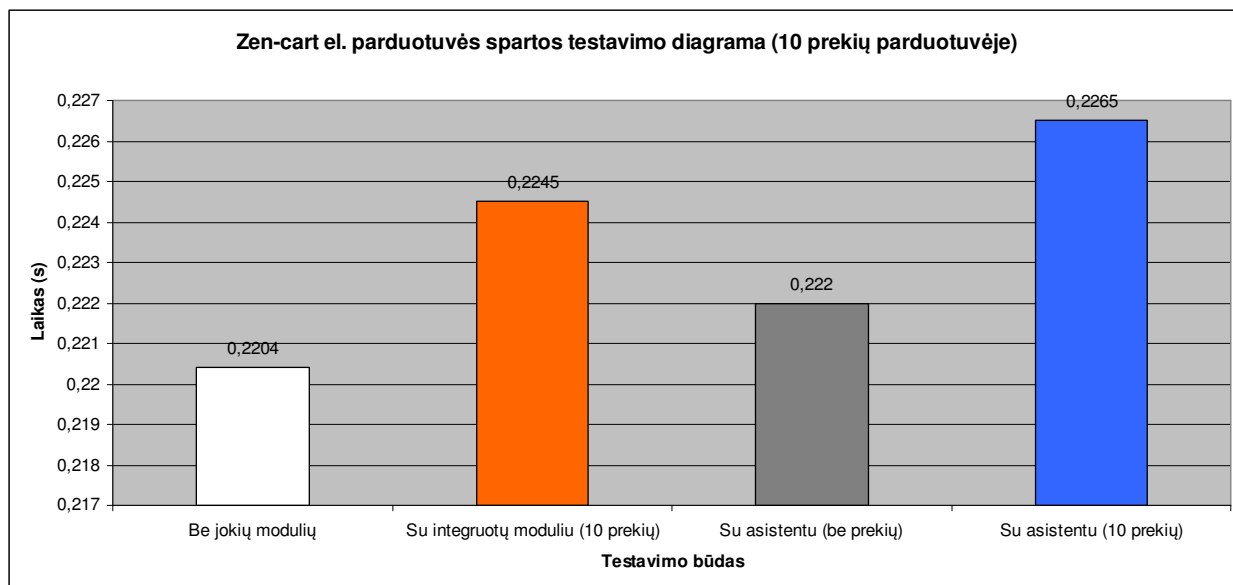
El. parduotuvės „Zen-cart“ testavimo analizė



18 pav. „Zen-cart“ el. parduotuvės spartos testavimo diagrama

El. parduotuvė be integruoto modulio ir virtualaus asistento veikė sparčiausiai. Integruotas modulis šiek tiek sulėtino el. parduotuvės krovimosi laiką (su 40 prekių), tačiau virtualus asistento pagalbinis kodas lyginant su juo veikė dar lėčiau. Taip nutiko dėl to, kad integruotas modulis nieko neskaičiuoja, administratorius turi prie kiekvienos prekės rankiniu būdu pridėti susijusias prekes. Tuo tarpu virtualus asistentas taip pat realiu laiku neskaičiuoja, tačiau sugaištama laiko tikrinant lankytojo prisijungimo duomenis, atnaujinant apsilankymų statistiką, bei pačių prekių atvaizdavimas užtrunka

ilgiau (iš pradžių tikrinama asistento DB, tik po to el. parduotuvės). Tačiau virtualaus asistento be prekių pakrovimas veikia greičiau už integruotą modulį.



19 pav. „Zen-cart“ el. parduotuvės spartos testavimo diagrama

Išbandžius el. parduotuvę su skirtingu prekių skaičiumi (125 ir 10), esminių pokyčių spartos atžvilgiu negalima įžvelgti. Tai yra dėl to, kad tiek integruotas modulis, tiek virtualaus asistento pagalbinis kodas neatlieka skaičiavimų, o tik išveda rezultatus.

Realiai išbandžius el. parduotuvę galima išskirti virtualaus asistento ir integruoto modulio privalumus ir trūkumus, bei atlikti palyginimą 48 lentelėje.

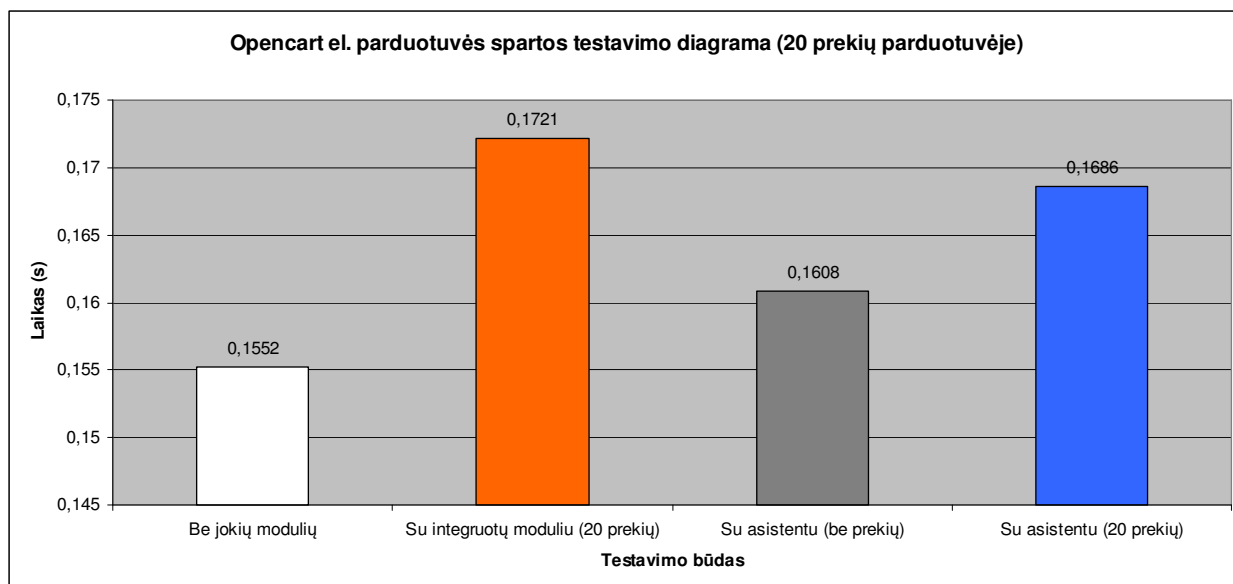
48 lentelė. Integruoto modulio ir virtualaus asistento palyginimo lentelė

Savybės	Integruotas modulis	Virtualaus asistentas
Svetainės pakrovimo sparta	Pakrovimas su pasiūlymų sąrašu 1-3% greitesnis	Pakrovimas su pasiūlymų sąrašu 1-3% lėtesnis
Pasiūlymų rodymas	Rodoma iškart	Pradedama rodyti 1-60 sekundžių laikotarpyje
Panašių prekių kriterijai	Nėra	Atsižvelgiama į prekės pavadinimų, aprašymų, kainų panašumą, taip pat ir į kategorijas
Atsižvelgimas į lankytojo veiksmus	Ne	Atsižvelgiama į prekių peržiūras, padarytus užsakymus, prisijungimą prie parduotuvės
Atsižvelgimas į akcijas	Ne	Taip, pasiūlymai su akcijomis rodomi aukščiau
Priežiūra	Privaloma rankiniu būdu prie kiekvienos prekės sukurti panašių prekių sąrašus	Nereikalinga
Galimybė lankytojui išjungti panašių prekių rodymą	Ne	Taip

Apžvelgus palyginimo lentelę, galima pastebėti, kad integruotas modulis nors ir veikia sparčiau, o pasiūlymai rodomi iškart, o ne 60 sekundžių laikotarpyje, tačiau jis yra ne automatizuotas ir nėra kuriami asmeniniai kiekvieno kliento pasiūlymai. Tuo tarpu virtualus asistentas veikia autonomiškai (net ir pridėjus naujų prekių jo bus pradėdamos siūlyti be administratoriaus įsikišimo),

atsižvelgiama į lankytojo peržiūrėtas ir užsakytas prekes, pasiūlymų sąrašas sukuriamas atsižvelgiant į prekių pavadinimo, aprašymo, kainos ir kategorijos panašumą. Asistentas taip pat atsižvelgia į prekių akcijas. Žinoma, virtualaus asistento įdiegimas užtrunka ilgiau (DB šablono sudarymas, prisijungimo duomenų išsaugojimas ir programinio kodo įterpimas), tačiau vėliau priežiūros jam nebereikia, priešingai nei integruotam moduliui. Integruotas modulis priešingai nei asistentas neturi uždarymo (neberodymo) galimybės.

El. parduotuvės „Opencart“ testavimo analizė

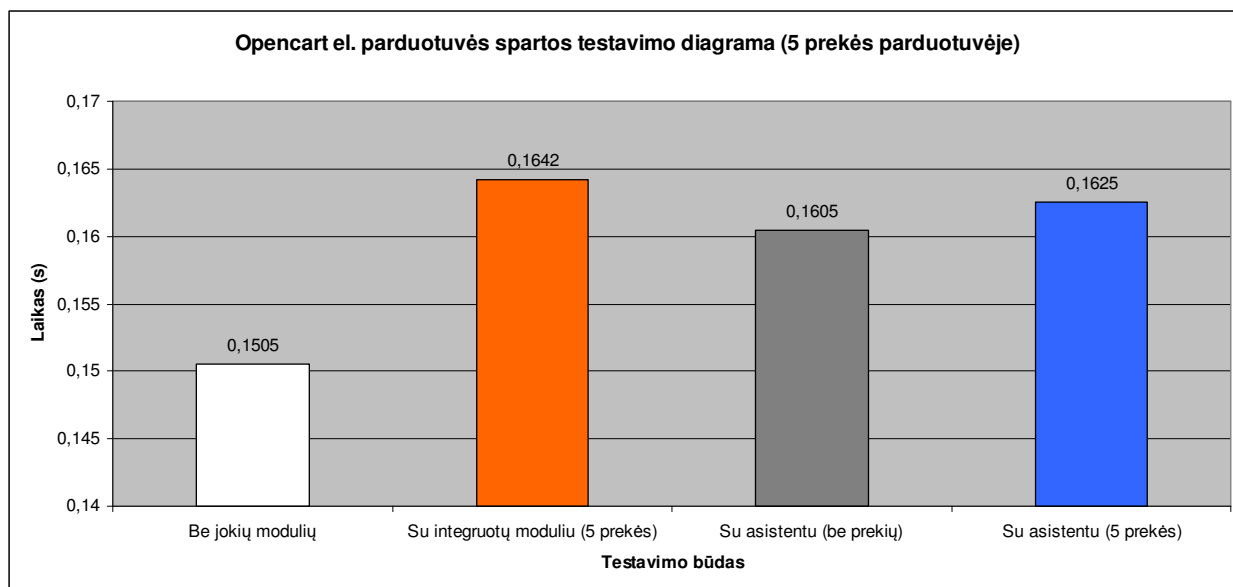


20 pav. „Opencart“ el. parduotuvės spartos testavimo diagrama

El. parduotuvė be integruoto modulio ir virtualaus asistento veikė sparčiausiai. Integruotas modulis labiausiai sulėtino el. parduotuvės krovimosi laiką (su 40 prekių), tuo tarpu virtualus asistentas lyginant su juo veikė greičiau. Taip nutiko dėl to, kad integruotas modulis atlieka panašių prekių paiešką kraunant puslapį (ieškoma pagal gamintoją, kategoriją ir prekės dar renkamos atsitiktiniu būdu). Tuo tarpu virtualus asistentas realiu laiku nieko neskaičiuoja, tačiau sugaištama laiko tikrinant lankytojo prisijungimo duomenis, atnaujinant apsilankymų statistiką, bei pačių prekių atvaizdavimas užtrunka ilgiau (iš pradžių tikrinama asistento DB, tik po to el. parduotuvės). Taip pat virtualaus asistento be prekių pakrovimas veikia greičiau už integruotą modulį.

Išbandžius el. parduotuvę su skirtingu prekių skaičiumi (20 ir 5), spartos atžvilgiu galima pastebėti, kad sumažėjo skirtumas tarp integruoto modulio ir virtualaus asistento. Tai įvyko dėl to, kad integruotas modulis su mažesniu prekių skaičiumi veikia greičiau. O virtualaus asistento pagalbinis kodas veikia praktiškai taip pat.

Realiai išbandžius el. parduotuvę galima išskirti virtualaus asistento ir integruoto modulio privalumus ir trūkumus, bei atlikti palyginimą 49 lentelėje.



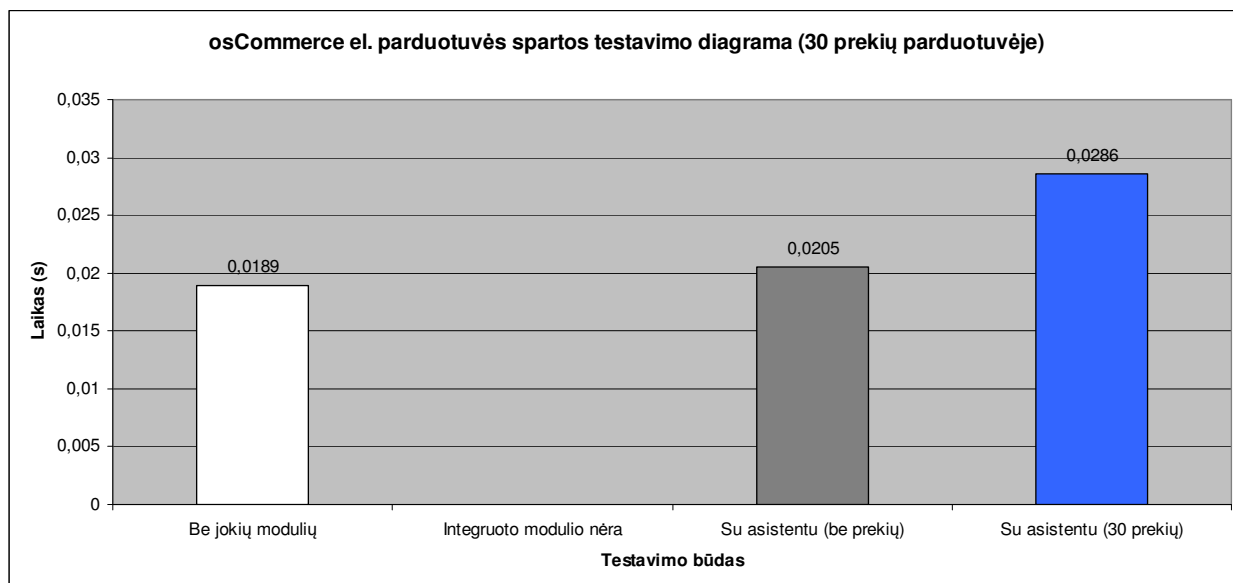
21 pav. „Opencart“ el. parduotuvės spartos testavimo diagrama

49 lentelė. Integruoto modulio ir virtualaus asistento palyginimo lentelė

Savybės	Integruotas modulis	Virtualaus asistentas
Svetainės pakrovimo sparta	Pakrovimas su pasiūlymų sąrašu 2-4% lėtesnis	Pakrovimas su pasiūlymų sąrašu 2-4% greitesnis
Pasiūlymų rodymas	Rodoma iškart	Pradedama rodyti 1-60 sekundžių laikotarpyje
Panašių prekių kriterijai	Atsižvelgiama į prekės gamintoją, kategoriją, taip pat pateikiamos atsitiktinės prekės	Atsižvelgiama į prekės pavadinimų, aprašymų, kainų panašumą, taip pat ir į kategorijas
Atsižvelgimas į lankytojo veiksmus	Ne	Atsižvelgiama į prekių peržiūras, padarytus užsakymus, prisijungimą prie parduotuvės
Atsižvelgimas į akcijas	Ne	Taip, pasiūlymai su akcijomis rodomi aukščiau
Priežiūra	Nereikalinga	Nereikalinga
Galimybė lankytojui išjungti panašių prekių rodyimą	Ne	Taip

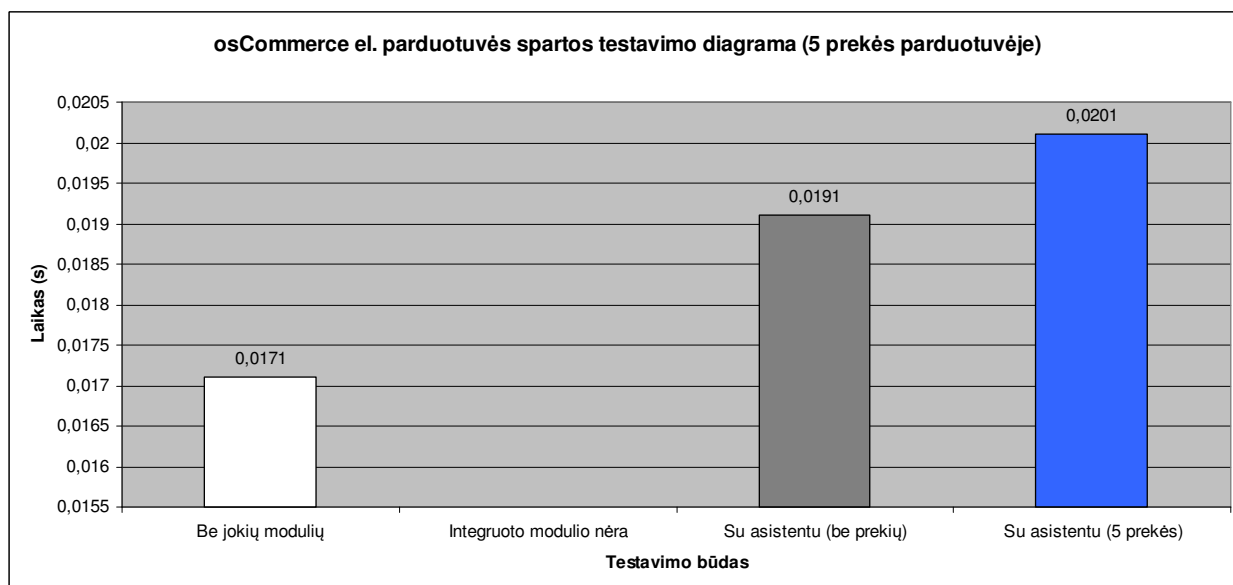
Apžvelgus palyginimo lentelę, galima pastebėti, kad integruotas modulis veikia lėčiau, pasiūlymai rodomi iškart, o ne 60 sekundžių laikotarpyje, tačiau nėra kuriami asmeniniai kiekvieno kliento pasiūlymai. Abu būdai veikia autonomiškai (net ir pridėjus naujų prekių jo bus pradėdamos siūlyti be administratoriaus įsikišimo), tačiau virtualus asistentas atsižvelgia į lankytojo peržiūrėtus ir užsakytas prekes, pasiūlymų sąrašas sukuriama atsižvelgiant į prekių pavadinimo, aprašymo, kainos ir kategorijos panašumą. Asistentas taip pat atsižvelgia į prekių akcijas. Žinoma, virtualaus asistento įdiegiamas užtrunka ilgiau (DB šablono sudarymas, prisijungimo duomenų išsaugojimas ir programinio kodo įterpimas). Integruotas modulis priešingai nei asistentas neturi uždarymo (neberodymo) galimybės.

El. parduotuvės „osCommerce“ testavimo analizė



22 pav. „osCommerce“ el. parduotuvės spartos testavimo diagrama

El. parduotuvė be integruoto modulio ir virtualaus asistento veikė sparčiausiai. Virtualus asistentas šiek tiek sulėtino puslapių krovimosi laiką. Kadangi ši el. parduotuvė neturi integruoto panašių prekių modulio, nebuvo galimybės atlikti jo testavimo.



23 pav. „osCommerce“ el. parduotuvės spartos testavimo diagrama

Išbandžius el. parduotuvę su skirtingu prekių skaičiumi (30 ir 5), esminių pokyčių spartos atžvilgiu negalima įžvelgti. Tai yra dėl to, kad virtualaus asistento pagalbinis kodas neatlieka skaičiavimų, o tik išveda rezultatus.

Apibendrinant galima pastebėti, kad virtualus asistentas, atitinka programinio agento apibrėžimą. Jis veikia autonomiškai, be administratoriaus įsikišimo, reaguoja į aplinką (atsižvelgia į lankytojų peržiūrėtus ir užsakytas prekes). Priešingai nei integruoti moduliai pasiūlymai yra kuriami kiekvienam lankytojui individualiai, to negalima pasakyti apie integruotus el. parduotuvėse modulius. Virtualaus asistento pranašumas spartos atžvilgiu išryškėja kai el. parduotuvėje yra daugiau prekių. Kuo daugiau tuo labiau pranašumas yra pastebimas.

IŠVADOS

- Atlikus programinių agentų savybių, tipų, sandaros, klasifikacijos analize, pastebėta, kad virtualus asistentas pasižymi informacinių ir sąsajos agentų tipų savybėmis: autonomiškumu, bendravimu su lankytojai, informacijos rinkimu, reagavimu į pokyčius, saugumu, rezultatų patikimumu, nesudėtinga realizacija. Todėl virtualus asistentas tampa panašus į hibridinį agentą, visumoje naudojantis kelis programinius agentus.
- Apžvelgus garsiausias ir populiariausias priemones programinių agentų realizavimui, buvo pasirinkta „Jade“ agentų kūrimo platforma, kuri yra nemokama, turinti daugiausiai pavyzdžių, dokumentacijos. Ji taip pat palaiko tarptautinius standartus, o realizuotas agentas gali būti atidaromas ir kitomis programomis.
- Kadangi reikalinga sukurta programinį agentą naudoti internetiniame puslapyje, buvo atlikta integravimo į serverius priemonių analizė. Išsiaiškinta, kad tinkamiausia priemonė sukurtam „Jade“ agentui yra „Apache Tomcat“. Pasirinkus šią priemonę buvo paprasta integruoti ir paleisti virtualų asistentą serveryje.
- Pagal savo veikimo principą, kaip pavyzdys virtualiam asistentui pasirinktas „Personal shopping assistant“, kuris nors ir nėra integruotas į internetinę svetainę (veikia realiuose prekių vežimėliuose), tačiau atitinka viziją, koks turėtų būti virtualus asistentas.
- Atlikus sukurtos sistemos testavimo analizę galima pastebėti, kad virtualus asistentas yra pranašesnis už integruotus panašių prekių siūlymo modulius el. parduotuvėse. Virtualus asistentas mažai nusileido spartos atžvilgiu integruotiems moduliams, o vienoje parduotuvėje netgi veikė 2-4% sparčiau. Didinant prekių skaičių el. parduotuvėje pastebimas didesnis spartos skirtumas virtualiojo asistento naudai.
- Sukurta sistema, atitinka programinio agento apibrėžimą. Ji veikia autonomiškai, be administratoriaus įsikišimo, reaguoja į aplinką. Priešingai nei integruoti moduliai pasiūlymai yra kuriami kiekvienam lankytojui individualiai, ko negalima pasakyti apie integruotus el. parduotuvėse modulius. Be to sistema turi daugiau funkcijų, kaip galimybė uždaryti asistento langelį. Taip pat realizuotas administratoriaus valdymo skydas, leidžiantis lengviau sukurti, valdyti ir prižiūrėti virtualius asistentus.

LITERATŪRA

1. Stuart J. Russell ir Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 1995.
2. Pattie Maes. Artificial Life Meets Entertainment: Life like Autonomous Agents. Communications of the ACM. 1995.
3. D. C. Smith, A. Cypher and J. Spohrer. KidSim: Programming Agents without a Programming Language. Communications of the ACM. 1994.
4. R Hayes-Roth. An Architecture for Adaptive Intelligent Systems. Artificial Intelligence: Special Issue on Agents and Interactivity, 1995.
5. Michael Woolbridge and Nicholas R. Jennings. Agent Theories, Architectures and Languages: a Survey. in Wooldridge and Jennings eds. Intelligent Agents. Springer-Verlag, 1995.
6. Jose C. Brustoloni. Autonomous Agents: Characterization and Requirements. 1995.
7. Carnegie Mellon Technical Report CMU-CS-91-204, Carnegie Mellon University 1991.
8. H. Nwana. Software Agents: An Overview. 1996.
9. OWL Web Ontology Language Use Cases and Requirements [interaktyvus]. 2004 [žiūrėta 2012-01-28]. Prieiga per internetą: <<http://www.w3.org/TR/webont-req/#usecase-agent>>.
10. Cdr.stanford.edu [interaktyvus]. JAT Overview [žiūrėta 2012-01-28]. Prieiga per internetą: <<http://www-cdr.stanford.edu/ABE/documentation/overview.html>>.
11. Shopbot technology. [interaktyvus]. [žiūrėta 2012-05-10]. Prieiga per internetą: <<http://www.shopbot-inc.com/group/en/group/technology.html>>.
12. Kathleen M. Carley. Smart Agents and Organizations of the Future [interaktyvus]. [žiūrėta 2012-01-28]. Prieiga per internetą: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.202.1587&rep=rep1&type=pdf>>.
13. Faisal Alkhateeb, Eslam Al Maghayreh and Iyad Abu Doush MULTI AGENT SYSTEMS MODELING, CONTROL, PROGRAMMING, SIMULATIONS AND APPLICATIONS. 2011.
14. Henry A. Kautz, Bart Selman, Michael Coen, Steven Ketchpel, Chris Ramming. An Experiment in the Design of Software Agents. AI Principles Research Department. 1994.
15. Hanh Tran, Thavoy Tran. Defense [interaktyvus]. Intelligent agent. [žiūrėta 2012-01-29]. Prieiga per internetą: <http://groups.engin.umd.umich.edu/CIS/course.des/cis479/projects/agent/Intelligent_agent.html>
16. Multi-Agent Simulation of Human Behavior in Naval Air Defense [interaktyvus]. [žiūrėta 2012-01-28]. Prieiga per internetą: <<http://faculty.nps.edu/ncrowe/oldstudents/calfeepap.htm>>.
17. Using Googlebot for Crawling Internet Content [interaktyvus]. [žiūrėta 2012-03-21]. Prieiga per internetą: <<http://www.brighthub.com/internet/google/articles/41953.aspx>>.
18. Cheng-Hsiung Hung, Hong-Jie Dai, and Jason Jen-Yen Chen. Intelligent Agent Communication by Using DAML to Build Agent Community Ontology. World Academy of Science, Engineering and Technology 6, 2005.
19. Intelligent Searching Agents on the Web [interaktyvus]. [žiūrėta 2012-01-27]. Prieiga per internetą: <<http://www.ariadne.ac.uk/issue7/search-engines/>>.

20. Johan Kwisthout. Modeling Uncertainty in 3APL. Open Universiteit Masteropleiding Technical Informatics. 2005.
21. Jason Description [interaktyvus]. [žiūrėta 2012-01-29]. Prieiga per internetą: <<http://jason.sourceforge.net/Jason/Description.html>>.
22. Lars Marius Garshol [interaktyvus]. Living with topic maps and RDF [žiūrėta 2012-01-29]. Prieiga per internetą: <<http://www.ontopia.net/topicmaps/materials/tmrd.html>>.
23. Todd Sundsted [interaktyvus]. An introduction to agents [žiūrėta 2012-01-29]. Prieiga per internetą: <<http://www.javaworld.com/javaworld/jw-06-1998/jw-06-howto.html>>.
24. Collecting Data with Mobile Agents [interaktyvus]. [žiūrėta 2012-01-28]. Prieiga per internetą: <<http://www.projectory.de/kaariboga/usage/dataCollection.html>>.
25. N. Kasabov, Introduction: Hybrid intelligent adaptive systems. International Journal of Intelligent Systems. 1998.
26. Alan Bundy, Coping with a Changing World: Structure of Intelligent Agents and Environments. 2003.
27. Boi Faltings. Intelligent Agents: Software Technology for the new Millennium. INFORMATIQUE. 2000.
28. The GOAL Agent Programming Language [interaktyvus]. [žiūrėta 2012-03-23]. Prieiga per internetą: <<http://mmi.tudelft.nl/trac/goal>>.
29. Michael Abernethy [interaktyvus]. Data mining with WEKA [žiūrėta 2012-03-24]. Prieiga per internetą: <<http://www.ibm.com/developerworks/opensource/library/os-weka1/index.html>>.
30. Salman Jamali [interaktyvus]. Experiencing Jack (Agent development Environment). [žiūrėta 2012-03-24]. Prieiga per internetą: <<http://www.codeproject.com/Articles/13644/Experiencing-Jack-Agent-development-Environment>>.
31. L. Braubach, W. Lamersdorf, A. Pokahr. Jadex: Implementing a BDI-Infrastructure for JADE Agents. 2003.
32. Pankaj Kamthan [interaktyvus]. Java Applets in Education [žiūrėta 2012-03-24]. Prieiga per internetą: <<http://www.irt.org/articles/js151/>>.
33. Java applets [interaktyvus]. [žiūrėta 2012-03-21]. Prieiga per internetą: <<http://java.sun.com/applets/>>.
34. Common Gateway Interface (CGI) [interaktyvus]. [žiūrėta 2012-03-23]. Prieiga per internetą: <[http://jabba.edb.utexas.edu/multimedia/TReports/CommonGatewayInterface\(CGI\).pdf](http://jabba.edb.utexas.edu/multimedia/TReports/CommonGatewayInterface(CGI).pdf)>.
35. CGI Made Really Easy [interaktyvus]. [žiūrėta 2012-03-23]. Prieiga per internetą: <<http://www.jmarshall.com/easy/cgi/>>.
36. Sing Li. Tomcat for beginning Web developers. 2005.
37. Apache Tomcat 7. [interaktyvus]. [žiūrėta 2012-03-23]. Prieiga per internetą: <http://www3.ntu.edu.sg/home/ehchua/programming/howto/tomcat_howto.html>.
38. Quercus: php in java [interaktyvus]. [žiūrėta 2012-03-22]. Prieiga per internetą: <<http://www.caucho.com/resin-3.1/doc/quercus.xtp>>.

39. What is the PHP/Java Bridge? [interaktyvus]. [žiūrėta 2012-03-20]. Prieiga per internetą <<http://php-java-bridge.sourceforge.net/pjb/>>.
40. N. R. Jennings , J. M. Corera , I. Laresgoiti , E. H. Mamdani , F. Perriollat , P. Skarek , L. Z. Varga. Using ARCHON to develop real-world DAI applications for electricity transportation management and particle accelerator control. 1996.
41. Digital libraries and software agents. [interaktyvus]. [žiūrėta 2012-03-23]. Prieiga per internetą <<http://scholar.lib.vt.edu/digilib/reports/agents.pdf>>.
42. Numericable Debuts Emilie, the New Face of Support [interaktyvus]. [žiūrėta 2012-03-20]. Prieiga per internetą:
<http://www.chatbots.org/conversational/agent/numericable_debuts_emilie_the_new_face_of_support/>.
43. Nikitos Drakos [interaktyvus]. Artificial Intelligence Week 2: Intelligent Agents [žiūrėta 2012-02-21]. Prieiga per internetą:
<<http://agents.csie.ntu.edu.tw/~yjhsu/courses/u0220/1997/09-24/webnotes.html>>.

PRIEDAI

1 priedas. Duomenų bazės lentelių detalizavimas

50 lentelė. *Detalizuota duomenų lentelė 1*

Pavadinimas	Lankytojai
Aprašymas	Saugomi apsilankiusių lankytojų duomenys
Struktūra	<i>nr</i> saugo statistikos numerį (galimi tik skaitmenys), <i>kliento_nr</i> - kliento identifikavimo numerį iš parduotuvės lentelių. Jei klientas neprisijungęs įrašomas 0 (galimi tik skaitmenys), <i>sesija</i> - lankytojo naršyklės sesijos identifikacijos eilutė (galimi simboliniai duomenys), <i>asistento_nr</i> - lankytojo asistento numerį, tokiu būdu sužinoma kurią svetainę aplankė lankytojas (galimi tik skaitmenys), <i>apsilankymu_sk</i> - lankytojo apsilankymų svetainėje skaičių (galimi tik skaitmenys), <i>pirkimu_sk</i> - lankytojo padarytų užsakymų svetainėje skaičių (galimi tik skaitmenys), <i>slepti</i> - lankytojo asistento rodymo būseną, jei reikšmė 0 asistentas rodomas, jei 1 - ne (galimi simboliniai duomenys), <i>data</i> - lankytojo paskutinio apsilankymo datą (galimi simboliniai duomenys)
Apribojimai	nr – pirminis
Resursai	SQL duomenų bazės lentelė

51 lentelė. *Detalizuota duomenų lentelė 2*

Pavadinimas	Statistika
Aprašymas	Saugomi lankytojų veiklos įrašai
Struktūra	<i>nr_statistika</i> saugo statistikos numerį (galimi tik skaitmenys), <i>lankytojo_nr</i> - lankytojo identifikavimo numerį (galimi simboliniai duomenys), duomenys, <i>ivykio_kodas</i> - statistikos įvykio tipą, reikšmės: 0 – rodo prekės peržiūrą, 2 – prekės užsakymą (galimi tik skaitmenys), <i>ivykio_data</i> - statistikos įvykio datą (galimi simboliniai), <i>objekto_nr</i> – statistikos prekės identifikavimo numerį (galimi tik skaitmenys), <i>apdorota</i> - statistikos apdorojimo būseną, reikšmės: 0 – neapdorotas agento, 1 – apdorotas (galimi tik skaitmenys)
Apribojimai	Nr_statistika – pirminis
Resursai	SQL duomenų bazės lentelė

52 lentelė. *Detalizuota duomenų bazės lentelė 3*

Pavadinimas	Pasiūlymai
Aprašymas	Saugomi lankytojams siūlomos prekės
Struktūra	<i>nr_pasiulymo</i> saugo pasiūlymo numerį (galimi tik skaitmenys), <i>lankytojo_nr</i> - lankytojo identifikavimo numerį (galimi simboliniai duomenys), <i>prekes_nr</i> - siūlomos prekės identifikavimo numerį (galimi tik skaitmenys), <i>prioritetas</i> - prekės siūlymo prioriteto numerį, dydis gali kisti labai įvairiai nuo 1 iki begalybės (galimi tik skaitmenys)
Apribojimai	nr_pasiulymo – pirminis
Resursai	SQL duomenų bazės lentelė

53 lentelė. *Detalizuota duomenų lentelė 4*

Pavadinimas	Nustatymai
Aprašymas	Saugomi administratoriaus išsaugoti nustatymų duomenys
Struktūra	<i>nr</i> saugo nustatymo įrašo numerį (galimi tik skaitmenys), <i>zemelapio_nr</i> saugo agento naudojamo DB žemėlapis numerį (galimi tik skaitmenys), <i>pavadinimas</i> – nustatymo įrašo pavadinimą (galimi simboliniai duomenys), <i>pav_adresas</i> – el. parduotuvės paveikslukų kelio direktorija (galimi simboliniai duomenys), <i>pav_galune</i> – el. parduotuvės paveikslukų galūnės priedėlis (galimi simboliniai duomenys), <i>nuoroda</i> – el. parduotuvės prekių nuoroda (galimi simboliniai duomenys), <i>mokesciai</i> – el. parduotuvės prekių papildomas mokestis prie galutinės kainos (galimi simboliniai duomenys), <i>prieiga</i> - duomenų bazės prieigos adresas (galimi simboliniai duomenys), <i>db</i> - duomenų bazės pavadinimą (galimi simboliniai duomenys), <i>vartotojas</i> - duomenų bazės vartotojo vardą

	(galimi simboliniai duomenys), <i>slaptazodis</i> - duomenų bazės vartotojo slaptažodį (galimi simboliniai duomenys), <i>prisijungimo_prieiga</i> - duomenų bazės prisijungimo prieigą (galimi simboliniai duomenys), <i>busena</i> - saugo agento veikimo būseną, reikšmės: 0 – išjungtas, 1 - įjungtas (galimi skaičiai)
Apribojimai	nr – pirminis
Resursai	SQL duomenų bazės lentelė

54 lentelė. Detalizuota duomenų lentelė 5

Pavadinimas	Ivykiai
Aprašymas	Saugomi įvykių tipai statistikos įrašams
Struktūra	<i>ivykio_kodas</i> saugo įvykio tipo numerį (galimi tik skaitmenys), <i>pavadinimas</i> – įvykio tipo pavadinimą (galimi simboliniai duomenys)
Apribojimai	<i>ivykio_kodas</i> – pirminis
Resursai	SQL duomenų bazės lentelė

55 lentelė. Detalizuota duomenų lentelė 6

Pavadinimas	Administratoriai
Aprašymas	Saugomi valdymo skydo administratorių įrašai
Struktūra	<i>admin_nr</i> saugo administratoriaus identifikavimo numerį (galimi tik skaitmenys), <i>vartotojas</i> – administratoriaus prisijungimo vardą (galimi simboliniai duomenys), <i>slaptazodis</i> – administratoriaus slaptažodį (galimi simboliniai duomenys), <i>el_pastas</i> – administratoriaus el. pašto adresą (galimi simboliniai duomenys), <i>admin_tipas</i> – administratoriaus tipo numerį: reikšmės: 0 – paprastas administratorius, 1 – pagrindinis administratorius (galimi tik skaitmenys), <i>leidimai</i> – saugomi administratoriaus valdomų virtualių asistentų (asistentų numeriai atskirti kabeliais) sąrašas (galimi simboliniai duomenys), <i>admin_tipas</i> – administratoriaus daugiausiai galimų turėti asistentų kiekį (galimi tik skaitmenys)
Apribojimai	<i>admin_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

56 lentelė. Detalizuota duomenų lentelė 7

Pavadinimas	Asistentas
Aprašymas	Saugoma pagrindinio agento, veikiančio serveryje nustatymai
Struktūra	<i>veikimas</i> saugo agento veikimo būseną, reikšmės: 0 – stabdyti veikimą serveryje, 1 – leisti agentui veikti serveryje (galimi tik skaitmenys), (galimi tik skaitmenys), <i>vartotojas</i> – saugo agento statistikos apdorojimo būseną, reikšmės: 0 – stabdyti lankytojų statistikos apdorojimą, 1 – leisti agentui apdoroti statistiką (galimi tik skaitmenys (galimi tik skaitmenys))
Apribojimai	<i>veikimas</i> – pirminis
Resursai	SQL duomenų bazės lentelė

57 lentelė. Detalizuota duomenų lentelė 8

Pavadinimas	Sablonai
Aprašymas	Saugomi DB šablonų pavadinimai ir numeriai
Struktūra	<i>zemelapio_nr</i> saugo šablono identifikavimo numerį (galimi tik skaitmenys), <i>pavadinimas</i> – šablono pavadinimą (galimi simboliniai duomenys)
Apribojimai	<i>zemelapio_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

58 lentelė. Detalizuota duomenų lentelė 9

Pavadinimas	Lenteliu_pavadinimai
Aprašymas	Saugoma DB šablono lentelių informacija
Struktūra	<i>lenteles_nr</i> saugo DB lentelės numerį (galimi tik skaitmenys), <i>zemelapio_nr</i> – DB šablono, kuriam priklauso numerį (galimi tik skaitmenys), <i>pavadinimas</i> – lentelės

	pavadinimą (galimi simboliniai duomenys), <i>loginio_tipo_nr</i> – lentelės loginio tipo numerį (galimi tik skaitmenys)
Apribojimai	<i>lenteles_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

59 lentelė. *Detalizuota duomenų lentelė 10*

Pavadinimas	Laukeliu_pavadinimai
Aprašymas	Saugoma DB šablono lentelių informacija
Struktūra	<i>laukelio_nr</i> saugo DB laukelio numerį (galimi tik skaitmenys), <i>lenteles_nr</i> –DB lentelės, kuriai priklauso numerį (galimi tik skaitmenys), <i>pavadinimas</i> – laukelio pavadinimą (galimi simboliniai duomenys), <i>paskirties_nr</i> – lentelės paskirties numerį (galimi tik skaitmenys)
Apribojimai	<i>laukelio_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

60 lentelė. *Detalizuota duomenų lentelė 11*

Pavadinimas	Loginiai_tipai
Aprašymas	Saugomi DB lentelių loginių tipų įrašai, nurodantys kam skirtos lentelės
Struktūra	<i>loginio_tipo_nr</i> saugo loginio tipo numerį (galimi tik skaitmenys), <i>pavadinimas</i> – loginio tipo pavadinimą (galimi simboliniai duomenys), <i>aprasymas</i> – loginio tipo aprašymo tekstą (galimi simboliniai duomenys)
Apribojimai	<i>loginio_tipo_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

61 lentelė. *Detalizuota duomenų lentelė 12*

Pavadinimas	Laukeliu_paskirtys
Aprašymas	Saugomi DB laukelių paskirties aprašymai
Struktūra	<i>paskirties_nr</i> saugo paskirties numerį (galimi tik skaitmenys), <i>paskirties_pavadinimas</i> – lentelės paskirties pavadinimą (galimi simboliniai duomenys), <i>aprasymas</i> – loginio tipo aprašymo tekstą (galimi simboliniai duomenys)
Apribojimai	<i>paskirties_nr</i> – pirminis
Resursai	SQL duomenų bazės lentelė

2 priedas. Funkcijų testavimas

Virtualių asistentų valdymo skydo testavimas

Testo tikslas:		Patikrinti virtualių asistento valdymo skydo veikimo teisingumą				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
1.	Patikrinti ar pagrindinis administratorius gali įjungti virtualų asistentą serveryje	Nuspaudžiama „Įjungti veikimą serveryje“ mygtukas	Aktyvuojamas agentas serveryje	Aktyvuojamas agentas serveryje	T	Funkcija veikia tinkamai
2.	Patikrinti ar pagrindinis administratorius gali išjungti virtualų asistentą serveryje	Nuspaudžiama „Išjungti veikimą serveryje“ mygtukas	Sustabdomas agentas serveryje	Sustabdomas agentas serveryje	T	Funkcija veikia tinkamai
3.	Patikrinti ar pagrindinis administratorius gali įjungti statistikos apdorojimą serveryje	Nuspaudžiama „Įjungti apdorojimą“ mygtukas	Agentas serveryje pradeda apdoroti statistikos įrašus	Agentas serveryje pradeda apdoroti statistikos įrašus	T	Funkcija veikia tinkamai
4.	Patikrinti ar pagrindinis administratorius gali išjungti statistikos apdorojimą serveryje	Nuspaudžiama „Išjungti apdorojimą“ mygtukas	Agentas serveryje nebeapdoroja statistikos įrašų	Agentas serveryje nebeapdoroja statistikos įrašų	T	Funkcija veikia tinkamai
5.	Patikrinti ar administratorius gali pasikeisti savo slaptažodį ir el. paštą	Nustatymų puslapyje įvedamas slaptažodis: „naujas“, el. paštas: „test@test.lt“	Administratoriaus nustatymai pakeisti duomenų bazėje	Administratoriaus nustatymai pakeisti duomenų bazėje	T	Funkcija veikia tinkamai
6.	Patikrinti ar administratorius gali pridėti virtualų asistentą	Užpildoma virtualių asistentų forma ir spaudžiamas mygtukas „Išsaugoti“	Naujas virtualaus asistento įrašas su visais užpildytais laukeliais išsaugomas duomenų bazėje	Naujas virtualaus asistento įrašas su visais užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai
7.	Patikrinti ar administratorius gali redaguoti virtualų asistentą	Pakeičiami duomenys virtualių asistentų redagavimo formoje ir	Pakeisto virtualaus asistento įrašas su užpildytais laukeliais išsaugomas	Pakeisto virtualaus asistento įrašas su užpildytais laukeliais išsaugomas	T	Funkcija veikia tinkamai

		spaudžiamas mygtukas „Atnaujinti“	duomenų bazėje	duomenų bazėje		
8.	Patikrinti ar administratorius gali ištrinti virtualų asistentą	Ties pasirinktu asistentu spaudžiama nuoroda „Ištrinti“	Pasirinktas asistentas ištrinamas iš duomenų bazės	Pasirinktas asistentas ištrinamas iš duomenų bazės	T	Funkcija veikia tinkamai
9.	Patikrinti ar administratorius gali ieškoti virtualų asistentų	Į paieškos laukelį įvedama frazė: „cart“ spaudžiamas mygtukas „Ieškoti“	Pateikiami asistentai, kurių pavadinime yra frazė „cart“	Pateikiami asistentai, kurių pavadinime yra frazė „cart“	T	Funkcija veikia tinkamai
10.	Patikrinti ar administratorius gali testuoti virtualų asistentą	Asistento redagavimo formoje spaudžiamas mygtukas „Testuoti“, pasirenkant automatinį parinkimą	Pateikiama informacija ar pavyko susijungti su serveriu ir rekomenduojama s naudoti šablonas su kiekiu rastų laukelių, lentelių	Pateikiama informacija ar pavyko susijungti su serveriu ir rekomenduojama s naudoti šablonas su kiekiu rastų laukelių, lentelių	T	Funkcija veikia tinkamai
11.	Patikrinti ar administratorius gali testuoti virtualų asistentą	Asistento redagavimo formoje spaudžiamas mygtukas „Testuoti“, pasirenkant DB šabloną	Pateikiama informacija ar pavyko susijungti su serveriu ir kiek DB laukelių, lentelių rasta naudojamame šablone	Pateikiama informacija ar pavyko susijungti su serveriu ir kiek DB laukelių, lentelių rasta naudojamame šablone	T	Funkcija veikia tinkamai
12.	Patikrinti ar administratorius gali pridėti DB šabloną	Užpildoma DB šablonų forma ir spaudžiamas mygtukas „Išsaugoti“, pasirenkant kurti šabloną kito pagrindu	Naujas DB šablono įrašas su visais užpildytais laukeliais išsaugomas duomenų bazėje, sukuriamos lentelės ir laukeliai pasirinkto šablono pagrindu	Naujas DB šablono įrašas su visais užpildytais laukeliais išsaugomas duomenų bazėje, sukuriamos lentelės ir laukeliai pasirinkto šablono pagrindu	T	Funkcija veikia tinkamai
13.	Patikrinti ar administratorius gali redaguoti DB šabloną	Pakeičiami duomenys DB šablonų redagavimo formoje ir spaudžiamas mygtukas	Pakeisto DB šablono įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	Pakeisto DB šablono įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai

		„Atnaujinti“				
14.	Patikrinti ar administratorius gali ištrinti DB šabloną	Ties pasirinktu DB šablonu spaudžiama nuoroda „Ištrinti“	Pasirinktas šablonas ištrinamas iš duomenų bazės	Pasirinktas šablonas ištrinamas iš duomenų bazės	T	Funkcija veikia tinkamai
15.	Patikrinti ar administratorius gali ieškoti DB šablonų	Į paieškos laukelį įvedama frazė: „cart“ spaudžiamas mygtukas „Ieškoti“	Pateikiami DB šablonai, kurių pavadinime yra frazė „cart“	Pateikiami DB šablonai, kurių pavadinime yra frazė „cart“	T	Funkcija veikia tinkamai
16.	Patikrinti ar administratorius gali pridėti šablono lentelę	Užpildoma lentelių forma ir spaudžiamas mygtukas „Išsaugoti“	Nauja lentelė su visais užpildytais laukeliais išsaugoma duomenų bazėje	Nauja lentelė su visais užpildytais laukeliais išsaugoma duomenų bazėje	T	Funkcija veikia tinkamai
17.	Patikrinti ar administratorius gali redaguoti šablono lentelę	Pakeičiami duomenys lentelių redagavimo formoje ir spaudžiamas mygtukas „Atnaujinti“	Pakeistos lentelės įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	Pakeistos lentelės įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai
18.	Patikrinti ar administratorius gali ištrinti šablono lentelę	Ties pasirinkta lentele spaudžiama nuoroda „Ištrinti“	Pasirinkta lentelė ištrinama iš duomenų bazės	Pasirinkta lentelė ištrinama iš duomenų bazės	T	Funkcija veikia tinkamai
19.	Patikrinti ar administratorius gali ieškoti šablono lentelių	Į paieškos laukelį įvedama frazė: „orders“ spaudžiamas mygtukas „Ieškoti“	Pateikiamos lentelės, kurių pavadinime yra frazė „orders“	Pateikiamos lentelės, kurių pavadinime yra frazė „orders“	T	Funkcija veikia tinkamai
20.	Patikrinti ar administratorius gali pridėti lentelės laukelį	Užpildoma laukelių forma ir spaudžiamas mygtukas „Išsaugoti“	Naujas lentelės laukelis su visais užpildytais laukeliais išsaugomas duomenų bazėje	Naujas lentelės laukelis su visais užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai
21.	Patikrinti ar administratorius gali redaguoti lentelės laukelį	Pakeičiami duomenys lentelės laukelių redagavimo formoje ir	Pakeisto laukelio įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	Pakeisto laukelio įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai

		spaudžiamas mygtukas „Atnaujinti“				
22.	Patikrinti ar administratorius gali ištrinti lentelės laukelį	Ties pasirinktu laukeliu spaudžiama nuoroda „Ištrinti“	Pasirinktas laukelis ištrinamas iš duomenų bazės	Pasirinktas laukelis ištrinamas iš duomenų bazės	T	Funkcija veikia tinkamai
23.	Patikrinti ar administratorius gali ieškoti lentelės laukelių	Į paieškos laukelį įvedama frazė: „product“ spaudžiamas mygtukas „Ieškoti“	Pateikiami laukeliai, kurių pavadinime yra frazė „product“	Pateikiami laukeliai, kurių pavadinime yra frazė „product“	T	Funkcija veikia tinkamai
24.	Patikrinti ar pagrindinis administratorius gali pridėti kitą administratorių	Užpildoma administratorių forma ir spaudžiamas mygtukas „Išsaugoti“	Naujas administratoriaus laukelis su visais užpildytais laukeliais išsaugomas duomenų bazėje	Naujas administratoriaus laukelis su visais užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai
25.	Patikrinti ar pagrindinis administratorius gali redaguoti kitą administratorių	Pakeičiami duomenys administratoriaus redagavimo formoje ir spaudžiamas mygtukas „Atnaujinti“	Pakeisto administratoriaus įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	Pakeisto administratoriaus įrašas su užpildytais laukeliais išsaugomas duomenų bazėje	T	Funkcija veikia tinkamai
26.	Patikrinti ar pagrindinis administratorius gali ištrinti kitą administratorių	Ties pasirinktu administratoriumi spaudžiama nuoroda „Ištrinti“	Pasirinktas administratorius ištrinamas iš duomenų bazės	Pasirinktas administratorius ištrinamas iš duomenų bazės	T	Funkcija veikia tinkamai
27.	Patikrinti ar pagrindinis administratorius gali ieškoti kitų administratorių	Į paieškos laukelį įvedama frazė: „test“ spaudžiamas mygtukas „Ieškoti“	Pateikiami administratoriai, kurių pavadinime yra frazė „test“	Pateikiami administratoriai, kurių pavadinime yra frazė „test“	T	Funkcija veikia tinkamai
28.	Patikrinti ar administratorius gali išvalyti lankytojų įrašus	Pasirenkami norimi šalinti įrašai ir spaudžiamas mygtukas „Valyti“	Pasirinkti lankytojų įrašai ištrinami iš duomenų bazės	Pasirinkti lankytojų įrašai ištrinami iš duomenų bazės	T	Funkcija veikia tinkamai
29.	Patikrinti ar	Į paieškos	Pateikiami	Pateikiami	T	Funkcija

	administratorius gali ieškoti lankytojų įrašų	laukelį įvedama frazė: „12“ spaudžiamas mygtukas „Ieškoti“	lankytojų įrašai, kurių numeris yra „12“	lankytojų įrašai, kurių numeris yra „12“		veikia tinkamai
30.	Patikrinti ar administratorius gali išvalyti statistikos įrašus	Pasirenkami norimi šalinti įrašai ir spaudžiamas mygtukas „Valyti“	Pasirinkti statistikos įrašai ištrinami iš duomenų bazės	Pasirinkti statistikos įrašai ištrinami iš duomenų bazės	T	Funkcija veikia tinkamai
31.	Patikrinti ar administratorius gali ieškoti statistikos įrašų	Į paieškos laukelį įvedama frazė: „12“ spaudžiamas mygtukas „Ieškoti“	Pateikiami statistikos įrašai, kurių numeris yra „12“	Pateikiami statistikos įrašai, kurių numeris yra „12“	T	Funkcija veikia tinkamai
32.	Patikrinti ar administratorius gali išvalyti pasiūlymų įrašus	Pasirenkami norimi šalinti įrašai ir spaudžiamas mygtukas „Valyti“	Pasirinkti pasiūlymų įrašai ištrinami iš duomenų bazės	Pasirinkti pasiūlymų įrašai ištrinami iš duomenų bazės	T	Funkcija veikia tinkamai
33.	Patikrinti ar administratorius gali ieškoti pasiūlymų įrašų	Į paieškos laukelį įvedama frazė: „12“ spaudžiamas mygtukas „Ieškoti“	Pateikiami pasiūlymų įrašai, kurių numeris yra „12“	Pateikiami pasiūlymų įrašai, kurių numeris yra „12“	T	Funkcija veikia tinkamai

Virtualių asistentų pagalbinio kodo testavimas

Testo tikslas:		Patikrinti virtualių asistentų pagalbinio kodo veikimo teisingumą				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
34.	Patikrinti ar el. parduotuvės (zencart) lankytojas gali uždaryti asistento langelį	Spaudžiama nuoroda „Uždaryti“	Virtualaus asistento laukelis uždaromas iki kitos sesijos arba rodymo komandos	Virtualaus asistento laukelis uždaromas iki kitos sesijos arba rodymo komandos	T	Funkcija veikia tinkamai
35.	Patikrinti ar el. parduotuvės (opencart) lankytojas gali uždaryti asistento langelį	Spaudžiama nuoroda „Uždaryti“	Virtualaus asistento laukelis uždaromas iki kitos sesijos arba rodymo komandos	Virtualaus asistento laukelis uždaromas iki kitos sesijos arba rodymo komandos	T	Funkcija veikia tinkamai
36.	Patikrinti ar el. parduotuvės	Spaudžiama nuoroda	Virtualaus asistento laukelis	Virtualaus asistento laukelis	T	Funkcija veikia

	(oscommerce) lankytojas gali uždaryti asistento langelį	„Uždaryti“	uždaromas iki kitos sesijos arba rodymo komandos	uždaromas iki kitos sesijos arba rodymo komandos		tinkamai
37.	Patikrinti ar el. parduotuvės (zencart) lankytojas gali atidaryti uždarytą asistento langelį	Spaudžiama nuoroda „Rodyti virtualų asistentą“	Virtualaus asistento laukelis pradedamas rodyti lankytojui	Virtualaus asistento laukelis pradedamas rodyti lankytojui	T	Funkcija veikia tinkamai
38.	Patikrinti ar el. parduotuvės (opencart) lankytojas gali atidaryti uždarytą asistento langelį	Spaudžiama nuoroda „Rodyti virtualų asistentą“	Virtualaus asistento laukelis pradedamas rodyti lankytojui	Virtualaus asistento laukelis pradedamas rodyti lankytojui	T	Funkcija veikia tinkamai
39.	Patikrinti ar el. parduotuvės (oscommerce) lankytojas gali atidaryti uždarytą asistento langelį	Spaudžiama nuoroda „Rodyti virtualų asistentą“	Virtualaus asistento laukelis pradedamas rodyti lankytojui	Virtualaus asistento laukelis pradedamas rodyti lankytojui	T	Funkcija veikia tinkamai
40.	Patikrinti ar el. parduotuvės (zencart) lankytojo apsilankymas užfiksuojamas duomenų bazėje	Atidaromas el. parduotuvės pradinis puslapis	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	T	Funkcija veikia tinkamai
41.	Patikrinti ar el. parduotuvės (opencart) lankytojo apsilankymas užfiksuojamas duomenų bazėje	Atidaromas el. parduotuvės pradinis puslapis	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	T	Funkcija veikia tinkamai
42.	Patikrinti ar el. parduotuvės (oscommerce) lankytojo apsilankymas užfiksuojamas duomenų bazėje	Atidaromas el. parduotuvės pradinis puslapis	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	Duomenų bazėje išsaugomas lankytojo įrašas su sesijos nr., data, aplankytos el. parduotuvės numeris	T	Funkcija veikia tinkamai
43.	Patikrinti ar el. parduotuvės (zencart) lankytojo pakartotinis apsilankymas užfiksuojamas duomenų bazėje	Atidaromas el. parduotuvės pradinis puslapis dar kartą	Duomenų bazėje lankytojo apsilankymų skaičius padidinamas vienetu	Duomenų bazėje lankytojo apsilankymų skaičius padidinamas vienetu	T	Funkcija veikia tinkamai
44.	Patikrinti ar el.	Atidaromas el.	Duomenų bazėje	Duomenų bazėje	T	Funkcija

	parduotuvės (opencart) lankytojo pakartotinis apsilankymas užfiksuojamas duomenų bazėje	parduotuvės pradinis puslapis dar kartą	lankytojo apsilankymų skaičius padidinamas vienetu	lankytojo apsilankymų skaičius padidinamas vienetu		veikia tinkamai
45.	Patikrinti ar el. parduotuvės (oscommerce) lankytojo pakartotinis apsilankymas užfiksuojamas duomenų bazėje	Atidaromas el. parduotuvės pradinis puslapis dar kartą	Duomenų bazėje lankytojo apsilankymų skaičius padidinamas vienetu	Duomenų bazėje lankytojo apsilankymų skaičius padidinamas vienetu	T	Funkcija veikia tinkamai
46.	Patikrinti ar el. parduotuvės (zen-cart) lankytojo prekės peržiūra išsaugoma duomenų bazėje	Atidaromas el. parduotuvės prekės aprašymo puslapis su prekės nr. 1	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	T	Funkcija veikia tinkamai
47.	Patikrinti ar el. parduotuvės (opencart) lankytojo prekės peržiūra išsaugoma duomenų bazėje	Atidaromas el. parduotuvės prekės aprašymo puslapis su prekės nr. 1	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	T	Funkcija veikia tinkamai
48.	Patikrinti ar el. parduotuvės (oscommerce) lankytojo prekės peržiūra išsaugoma duomenų bazėje	Atidaromas el. parduotuvės prekės aprašymo puslapis su prekės nr. 1	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	Duomenų bazėje lankytojo statistikos įrašas išsaugomas su tipo reikšme 1 ir prekės nr. 1, data	T	Funkcija veikia tinkamai
49.	Patikrinti ar el. parduotuvės (zen-cart) lankytojo prekės užsakymas išsaugomas duomenų bazėje	El. parduotuvėje atliekamas užsakymas, užsakoma prekė nr. 1 ir 2	Duomenų bazėje lankytojo statistikos įrašai išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data	Duomenų bazėje lankytojo statistikos įrašai išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data	T	Funkcija veikia tinkamai
50.	Patikrinti ar el. parduotuvės (opencart) lankytojo prekės užsakymas išsaugomas duomenų bazėje	El. parduotuvėje atliekamas užsakymas, užsakoma prekė nr. 1 ir 2	Duomenų bazėje lankytojo statistikos įrašai išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data	Duomenų bazėje lankytojo statistikos įrašai išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data	T	Funkcija veikia tinkamai
51.	Patikrinti ar el. parduotuvės (oscommerce)	El. parduotuvėje atliekamas	Duomenų bazėje lankytojo statistikos įrašai	Duomenų bazėje lankytojo statistikos įrašai	T	Funkcija veikia tinkamai

	lankytojo prekės užsakymas išsaugomas duomenų bazėje	užsakymas, užsakoma prekė nr. 1 ir 2	išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data	išsaugomi su tipo reikšme 3 ir prekės nr. 1 ir prekės nr. 3, data		
52.	Patikrinti ar el. parduotuvės (zencart) lankytojas pasveikinamas, kai jam pasiūlymų dar nėra sukurta	Atidaromas el. parduotuvės pradinis puslapis	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	T	Funkcija veikia tinkamai
53.	Patikrinti ar el. parduotuvės (opencart) lankytojas pasveikinamas, kai jam pasiūlymų dar nėra sukurta	Atidaromas el. parduotuvės pradinis puslapis	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	T	Funkcija veikia tinkamai
54.	Patikrinti ar el. parduotuvės (oscommerce) lankytojas pasveikinamas, kai jam pasiūlymų dar nėra sukurta	Atidaromas el. parduotuvės pradinis puslapis	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	Lankytojas pasveikinamas pagal laiką (labas rytas, laba diena ar labas vakaras) pagal tikrą laiką	T	Funkcija veikia tinkamai
55.	Patikrinti ar el. parduotuvės (zencart) lankytojas gauna pasiūlymus jei jam DB pasiūlymų yra	Atidaromas el. parduotuvės pradinis puslapis	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	T	Funkcija veikia tinkamai
56.	Patikrinti ar el. parduotuvės (opencart) lankytojas gauna pasiūlymus jei jam DB pasiūlymų yra	Atidaromas el. parduotuvės pradinis puslapis	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	T	Funkcija veikia tinkamai
57.	Patikrinti ar el. parduotuvės (oscommerce) lankytojas gauna pasiūlymus jei jam DB pasiūlymų yra	Atidaromas el. parduotuvės pradinis puslapis	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	Lankytojui pateikiamas prekių sąrašas pagal prioritetą mažėjančia tvarka	T	Funkcija veikia tinkamai
58.	Patikrinti ar el. parduotuvės (zencart) lankytojo prisijungimas užfiksuojamas duomenų bazėje	El. parduotuvėje prisijungiama kaip registruotas klientas su nr. 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	T	Funkcija veikia tinkamai

59.	Patikrinti ar el. parduotuvės (opencart) lankytojo prisijungimas užfiksuojamas duomenų bazėje	El. parduotuvėje prisijungiama kaip registruotas klientas su nr. 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	T	Funkcija veikia tinkamai
60.	Patikrinti ar el. parduotuvės (oscommerce) lankytojo prisijungimas užfiksuojamas duomenų bazėje	El. parduotuvėje prisijungiama kaip registruotas klientas su nr. 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	Duomenų bazėje lankytojo su tokiu pačiu sesijos nr. atnaujinamas kliento numeris į 2	T	Funkcija veikia tinkamai

Programinio agento testavimas

Testo tikslas:		Patikrinti programinio agento veikimo teisingumą				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
61.	Patikrinti ar agentas sukuria pasiūlymus iš prekių statistikos įrašų kas minutę	Sukūrus prekių statistikos įrašą laukiama 1 minutė su prekės nr. 1	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymų įrašai pagal prekės nr.	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymų įrašai pagal prekės nr.	T	Funkcija veikia tinkamai
62.	Patikrinti ar agentas sukuria pasiūlymus iš prekių užsakymo statistikos įrašų kas minutę	Sukūrus prekių užsakymo statistikos įrašą laukiama 1 minutė su prekės nr. 1	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymų įrašai pagal prekės nr.	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymų įrašai pagal prekės nr.	T	Funkcija veikia tinkamai
63.	Patikrinti ar agentas sumuoja tos pačios prekės prioritetus ir ištrina dublikatus	Sukūrus kelis tos pačios prekės pasiūlymus laukiama ar jie bus sumuojami, o dublikatai trinami	Minutės laikotarpyje duomenų bazėje ištrinami vienodos prekės dublikatai, o jų prioritetai sumuojami	Minutės laikotarpyje duomenų bazėje ištrinami vienodos prekės dublikatai, o jų prioritetai sumuojami	T	Funkcija veikia tinkamai
64.	Patikrinti ar agentas ištrina užsakytą prekę iš pasiūlymų	Padarius užsakymą el. parduotuvėje laukiama iki 1 minutės kol bus ištrinta užsakyta prekė iš pasiūlymų sąrašo	Minutės laikotarpyje duomenų bazėje ištrinama prekė, kurią užsakė lankytojas	Minutės laikotarpyje duomenų bazėje ištrinama prekė, kurią užsakė lankytojas	T	Funkcija veikia tinkamai

65.	Patikrinti ar agentas ištrina neregistruotų lankytojų sukauptą informaciją kas 1 valandą, kurie lankėsi seniau nei prieš 5 val.	Suakauptus pasiūlymų neprisijungiam a ir laukiama 5 valandas kol agentas ištrins senus įrašus	Valandos laikotarpyje duomenų bazėje ištrinama informacija, kurios lankytojas yra neregistruotas ir lankėsi seniau nei prieš 5 valandas	Valandos laikotarpyje duomenų bazėje ištrinama informacija, kurios lankytojas yra neregistruotas ir lankėsi seniau nei prieš 5 valandas	T	Funkcija veikia tinkamai
66.	Patikrinti ar agentas sukuria pasiūlymus atsižvelgdamas į panašių prekių pavadinimą	Peržiūrima prekė pavadinimu „Matrox G200 MMS“	Minutės laikotarpyje duomenų bazėje sukuriamą prekė panašiu pavadinimu „Matrox G400 32MB“	Minutės laikotarpyje duomenų bazėje sukuriamą prekė panašiu pavadinimu „Matrox G400 32MB“	T	Funkcija veikia tinkamai
67.	Patikrinti ar agentas sukuria pasiūlymus atsižvelgdamas į panašių prekių kategoriją	Peržiūrima prekė kategorijoje „Software“	Minutės laikotarpyje duomenų bazėje sukuriama įrašai, pateikiamos kitos programos	Minutės laikotarpyje duomenų bazėje sukuriama įrašai, pateikiamos kitos programos	T	Funkcija veikia tinkamai
68.	Patikrinti ar agentas sukuria pasiūlymus atsižvelgdamas į panašių prekių kainą	Peržiūrima prekė su kaina 59.99	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymai, kurių kaina yra nuo 24,99 iki 94,99	Minutės laikotarpyje duomenų bazėje sukuriama pasiūlymai, kurių kaina yra nuo 24,99 iki 94,99	T	Funkcija veikia tinkamai
69.	Patikrinti ar agentas sukuria pasiūlymus atsižvelgdamas į panašių prekių kainos akcijas	Peržiūrima prekė, kurios kaina yra akcija	Minutės laikotarpyje duomenų bazėje sukuriama prekės kurių kaina panaši	Minutės laikotarpyje duomenų bazėje sukuriama prekės kurių kaina yra nuo 24,99 iki 94,99	T	Funkcija veikia tinkamai
70.	Patikrinti ar agentas riboja pasiūlymų skaičių iki 40 vienam lankytojui	Sukuriamą daugiau nei 40 pasiūlymų įrašų	Minutės laikotarpyje duomenų bazėje pašalinami mažiausio prioriteto pasiūlymai, paliksiami 40, turintys aukščiausią prioritetą turintys pasiūlymai	Minutės laikotarpyje duomenų bazėje pašalinami mažiausio prioriteto pasiūlymai, paliksiami 40, turintys aukščiausią prioritetą turintys pasiūlymai	T	Funkcija veikia tinkamai

3 priedas. Našumo testavimas

Virtualių asistentų valdymo skydo testavimas

Testo tikslas:		Patikrinti virtualių asistentų valdymo skydo veikimo našumą				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
71.	Patikrinti valdymo skydo pakrovimo spartą	Atveriamas pagrindinis valdymo langas su jame esančiu specialiu programiniu kodu, kuris nustatys puslapio generavimo laiką	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.10 sekundės	Puslapis sugeneruotas per 0.0017 sekundės	T	Puslapis sugeneruojamas tinkame intervale
72.	Patikrinti valdymo skydo pakrovimo spartą	Atveriamas virtualių asistentų sąrašas (3 įrašai) su jame esančiu specialiu programiniu kodu, kuris nustatys puslapio generavimo laiką	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.10 sekundės	Puslapis sugeneruotas per 0.0032 sekundės	T	Puslapis sugeneruojamas tinkame intervale
73.	Patikrinti valdymo skydo pakrovimo spartą	Atveriamas DB šablonų sąrašas (3 įrašai) su jame esančiu specialiu programiniu kodu, kuris nustatys puslapio generavimo laiką	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.10 sekundės	Puslapis sugeneruotas per 0.0040 sekundės	T	Puslapis sugeneruojamas tinkame intervale

Virtualių asistentų pagalbinio kodo testavimas

Testo tikslas:		Patikrinti virtualių asistentų pagalbinio kodo veikimo našumą skirtingose interneto naršyklėse				
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos

74.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų jokių pasiūlymų modulių ir agento, 10 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2204 sekundės	T	Puslapis sugeneruojamas tinkame intervale
75.	Patikrinti el. parduotuvės pakrovimo spartą, kai įjungtas integruotas panašių prekių modulis (10 įrašų), 10 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2245 sekundės	T	Puslapis sugeneruojamas tinkame intervale
76.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytoji, 10 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2220 sekundės	T	Puslapis sugeneruojamas tinkame intervale
77.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 10, 10 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai pasiūlymų yra 40	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2265 sekundės	T	Puslapis sugeneruojamas tinkame intervale
78.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų jokių pasiūlymų modulių ir agento, 125 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2232 sekundės	T	Puslapis sugeneruojamas tinkame intervale
79.	Patikrinti el. parduotuvės pakrovimo spartą, kai įjungtas integruotas panašių prekių modulis (40 įrašų), 125 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2399 sekundės	T	Puslapis sugeneruojamas tinkame intervale

80.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytoji, 125 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2319 sekundės	T	Puslapis sugeneruojamas tinkame intervale
81.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 40, 125 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (zen-cart), kai pasiūlymų yra 40	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.2482 sekundės	T	Puslapis sugeneruojamas tinkame intervale
82.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų jokių pasiūlymų modulių ir agento, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1505 sekundės	T	Puslapis sugeneruojamas tinkame intervale
83.	Patikrinti el. parduotuvės pakrovimo spartą, kai įjungtas integruotas panašių prekių modulis (5 įrašai), 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1642 sekundės	T	Puslapis sugeneruojamas tinkame intervale
84.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytoji, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1642 sekundės	T	Puslapis sugeneruojamas tinkame intervale
85.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 5, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai pasiūlymų yra 5	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1605 sekundės	T	Puslapis sugeneruojamas tinkame intervale
86.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų	Atveriamas el. parduotuvės pagrindinis langas	Tikimasi, kad puslapis bus sugeneruotas greičiau nei	Puslapis sugeneruotas per 0.1625 sekundės	T	Puslapis sugeneruojamas tinkame intervale

	jokių pasiūlymų modulių ir agento, 20 prekių parduotuvėje	(opencart), kai prekių dar nėra siūloma lankytojui	0.30 sekundės			
87.	Patikrinti el. parduotuvės pakrovimo spartą, kai įjungtas integruotas panašių prekių modulis (40 įrašų), 20 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai prekių dar nėra siūloma lankytojui	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1721 sekundės	T	Puslapis sugeneruojamas tinkame intervale
88.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytojui, 20 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai prekių dar nėra siūloma lankytojui	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1608 sekundės	T	Puslapis sugeneruojamas tinkame intervale
89.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 40, 20 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (opencart), kai pasiūlymų yra 40	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.1686 sekundės	T	Puslapis sugeneruojamas tinkame intervale
90.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų jokių pasiūlymų modulių ir agento, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai prekių dar nėra siūloma lankytojui	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0171 sekundės	T	Puslapis sugeneruojamas tinkame intervale
91.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytojui, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai prekių dar nėra siūloma lankytojui	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0191 sekundės	T	Puslapis sugeneruojamas tinkame intervale
92.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 5, 5 prekės parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai pasiūlymų yra 5	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0201 sekundės	T	Puslapis sugeneruojamas tinkame intervale

93.	Patikrinti el. parduotuvės pakrovimo spartą, kai nėra įjungtų jokių pasiūlymų modulių ir agento, 30 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0189 sekundės	T	Puslapis sugeneruojamas tinkame intervale
94.	Patikrinti el. parduotuvės pakrovimo spartą, kai prekių dar nėra siūloma lankytoji, 30 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai prekių dar nėra siūloma lankytoji	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0205 sekundės	T	Puslapis sugeneruojamas tinkame intervale
95.	Patikrinti el. parduotuvės pakrovimo spartą, kai pasiūlymų yra 30, 30 prekių parduotuvėje	Atveriamas el. parduotuvės pagrindinis langas (oscommerce), kai pasiūlymų yra 30	Tikimasi, kad puslapis bus sugeneruotas greičiau nei 0.30 sekundės	Puslapis sugeneruotas per 0.0286 sekundės	T	Puslapis sugeneruojamas tinkame intervale

4 priedas. Suderinamumo testavimas

Virtualių asistentų valdymo skydo testavimas

Testo tikslas:		Patikrinti virtualių asistentų valdymo skydo užkrovimą skirtingose interneto naršyklėse				
96.	Patikrinti valdymo skydo pakrovimo teisingumą su „Internet Explorer“ 10 interneto naršykle	Atveriamas pagrindinis valdymo langas	Puslapio užkrovimas be iškraipymų	Puslapio su užkrovimas iškraipymais	N	Puslapis atvaizduojamas netinkamai
97.	Patikrinti valdymo skydo pakrovimo teisingumą su „Google Chrome“ 26 interneto naršykle	Atveriamas pagrindinis valdymo langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai
98.	Patikrinti valdymo skydo pakrovimo teisingumą su „Opera“ 10 interneto naršykle	Atveriamas pagrindinis valdymo langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai
99.	Patikrinti valdymo skydo pakrovimo teisingumą su „Mozilla Firefox“ 20 interneto naršykle	Atveriamas pagrindinis valdymo langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai

Virtualių asistentų pagalbinio kodo testavimas

Testo tikslas:		Patikrinti virtualių asistentų valdymo pagalbinio kodo užkrovimą skirtingose interneto naršyklėse				
100.	Patikrinti asistento pasiūlymų langelio pakrovimo teisingumą su „Internet Explorer“ 10 interneto naršyklė	Atveriamas el. parduotuvės pagrindinis langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas netinkamai
101.	Patikrinti asistento pasiūlymų langelio pakrovimo teisingumą su „Google Chrome“ 26 interneto naršyklė	Atveriamas el. parduotuvės pagrindinis langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai
102.	Patikrinti asistento pasiūlymų langelio pakrovimo teisingumą su „Opera“ 10 interneto naršyklė	Atveriamas el. parduotuvės pagrindinis langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai
103.	Patikrinti asistento pasiūlymų langelio pakrovimo teisingumą su „Mozilla Firefox“ 20 interneto naršyklė	Atveriamas el. parduotuvės pagrindinis langas	Puslapio užkrovimas be iškraipymų	Puslapio užkrovimas be iškraipymų	T	Puslapis atvaizduojamas tinkamai

5 priedas. Saugumo testavimas

Virtualių asistentų valdymo skydo testavimas

Testo tikslas:		Patikrinti virtualių asistentų valdymo skydo saugumą				
104.	Patikrinti valdymo skydo prisijungimo saugumą	Bandoma prisijungti neteisingais prisijungimo duomenimis	Pranešama apie klaidingus prisijungimo duomenis, neleidžiama patekti į valdymo skydą	Pranešama apie klaidingus prisijungimo duomenis, neleidžiama patekti į valdymo skydą	T	Apsaugota tinkamai
105.	Patikrinti valdymo skydo pasiekiamumą	Bandoma URL adrese įvesti administravimo failų adresus: admin/skydas.php	Nukerpiama į prisijungimo puslapį	Nukerpiama į prisijungimo puslapį	T	Apsaugota tinkamai
106.	Patikrinti valdymo skydo administratorių kūrimo limitą	Bandoma sukurti daugiau asistentų nei leido pagrindinis administratorius	Pranešama apie klaidą, neleidžiama sukurti daugiau asistentų	Pranešama apie klaidą, neleidžiama sukurti daugiau asistentų	T	Apsaugota tinkamai
107.	Patikrinti valdymo skydo administratorių leidimų ribojimą virtualiems asistentams	Bandoma prieiti prie virtualių asistentų, kuriems pagrindinis administratorius nesuteikė leidimo	Nukreipiama į virtualių asistentų sąrašą	Nukreipiama į virtualių asistentų sąrašą	T	Apsaugota tinkamai
108.	Patikrinti valdymo skydo administratorių leidimų ribojimą DB šablonams	Bandoma prieiti prie DB šablonų, kuriems pagrindinis administratorius nesuteikė leidimo	Nukreipiama į DB šablonų sąrašą	Nukreipiama į DB šablonų sąrašą	T	Apsaugota tinkamai
109.	Patikrinti valdymo skydo administratorių leidimų ribojimą lentelės, laukeliams	Bandoma prieiti prie lentelių ir laukelių, kuriems pagrindinis administratorius nesuteikė leidimo	Nukreipiama į lentelių sąrašą	Nukreipiama į lentelių sąrašą	T	Apsaugota tinkamai
110.	Patikrinti valdymo skydo administratorių leidimą redaguoti administratoriaus duomenis	Bandoma URL adrese įvesti pagrindinio administratoriaus nr. taip siekiant redaguoti paskyrą	Pranešama apie klaidą, neleidžiama redaguoti pagrindinio ar kito administratoriaus	Pranešama apie klaidą, neleidžiama redaguoti pagrindinio ar kito administratoriaus	T	Apsaugota tinkamai
111.	Patikrinti valdymo skydo administratorių leidimą ištrinti administratoriaus	Bandoma URL adrese įvesti pagrindinio administratoriaus nr. taip siekiant	Pranešama apie klaidą, neleidžiama ištrinti pagrindinio ar	Pranešama apie klaidą, neleidžiama ištrinti pagrindinio ar	T	Apsaugota tinkamai

	duomenis	ištrinti paskyrą	kito administratoriaus	kito administratoriaus		
112.	Patikrinti valdymo skydo atsparumą XSS atakoms virtualių asistentų kūrimo formoje	Bandoma pavadinime įvesti „<tag>pav</tag>“	Pavadinimo laukelyje išsaugoma frazė „pav“	Pavadinimo laukelyje išsaugoma frazė „pav“	T	Apsaugota tinkamai
113.	Patikrinti valdymo skydo atsparumą XSS atakoms DB šablonų kūrimo formoje	Bandoma pavadinime įvesti „<tag>pav</tag>“	Pavadinimo laukelyje išsaugoma frazė „pav“	Pavadinimo laukelyje išsaugoma frazė „pav“	T	Apsaugota tinkamai
114.	Patikrinti valdymo skydo atsparumą XSS atakoms laukelių kūrimo formoje	Bandoma pavadinime įvesti „<tag>pav</tag>“	Pavadinimo laukelyje išsaugoma frazė „pav“	Pavadinimo laukelyje išsaugoma frazė „pav“	T	Apsaugota tinkamai
115.	Patikrinti valdymo skydo atsparumą XSS atakoms lentelių kūrimo formoje	Bandoma pavadinime įvesti „<tag>pav</tag>“	Pavadinimo laukelyje išsaugoma frazė „pav“	Pavadinimo laukelyje išsaugoma frazė „pav“	T	Apsaugota tinkamai
116.	Patikrinti paieškos SQL injekcijų atsparumą	Adreso laukelyje vietoj paieškos frazės įrašome „nr' OR 'nr'='x1”	Nepateikiamas nei vienas įrašas paieškoje	Nepateikiamas nei vienas įrašas paieškoje	T	Apsaugota tinkamai
117.	Patikrinti virtualių asistentų redagavimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nepateikiamas nei vienas įrašas redagavimui	Nepateikiamas nei vienas įrašas redagavimui	T	Apsaugota tinkamai
118.	Patikrinti virtualių asistentų trynimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nei vienas įrašas nepašalinamas iš duomenų bazės	Nei vienas įrašas nepašalinamas iš duomenų bazės	T	Apsaugota tinkamai
119.	Patikrinti DB šablonų redagavimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nepateikiamas nei vienas įrašas redagavimui	Nepateikiamas nei vienas įrašas redagavimui	T	Apsaugota tinkamai
120.	Patikrinti DB šablonų trynimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nei vienas įrašas nepašalinamas iš duomenų bazės	Nei vienas įrašas nepašalinamas iš duomenų bazės	T	Apsaugota tinkamai

121.	Patikrinti lentelių redagavimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nepateikiamas nei vienas įrašas redagavimui	Nepateikiamas nei vienas įrašas redagavimui	T	Apsaugota tinkamai
122.	Patikrinti lentelių trynimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nei vienas įrašas nepašalinamas iš duomenų bazės	Nei vienas įrašas nepašalinamas iš duomenų bazės	T	Apsaugota tinkamai
123.	Patikrinti laukelių redagavimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nepateikiamas nei vienas įrašas redagavimui	Nepateikiamas nei vienas įrašas redagavimui	T	Apsaugota tinkamai
124.	Patikrinti laukelių trynimo SQL injekcijų atsparumą	Adreso laukelyje vietoj numerio įrašome „nr' OR 'nr'='x1”	Nei vienas įrašas nepašalinamas iš duomenų bazės	Nei vienas įrašas nepašalinamas iš duomenų bazės	T	Apsaugota tinkamai

Virtalių asistentų pagalbinio kodo testavimas

125.	Patikrinti ar galima pasisavinti ir naudoti virtualaus asistento kodą	Iš vienos el. parduotuvės nukopijuojamas sugeneruotas asistento kodas ir bandomas naudoti kitoje	Gaunas pranešimas, kad asistentas veikia ne toje svetainėje	Gaunas pranešimas, kad asistentas veikia ne toje svetainėje	T	Apsaugota tinkamai
126.	Patikrinti ar galima iššifruoti el. parduotuvėje naudojamą raktą	Gautas raktas f3061da6fc31fa86bed9aad4d8141961 Bandomas iššifruoti dekodavimu	Md5 kodavimo nepavyksta iššifruoti	Md5 kodavimo nepavyksta iššifruoti	T	Apsaugota tinkamai

6 priedas. Vartotojų prisijungimo duomenys

Realizuotos sistemos ir testuotų el. parduotuvių adresas:

<http://rude.su.lt/~stravinskas>

Virtualių asistentų valdymo skydo prisijungimo duomenys

Pagrindinis administratorius

Direktorija prisijungimui: /asistentas/index.php

Prisijungimo vardas: admin

Slaptažodis: stravinskas

Papildomas administratorius (ribotos teisės)

Prisijungimo vardas: admin2

Slaptažodis: stravinskas

Pirmosios parduotuvės (Zen-cart) prisijungimo duomenys

Pagrindinis administratorius

Direktorija prisijungimui: /admin_test/login.php

Prisijungimo vardas: admin

Slaptažodis: zencart2013

Klientas

Prisijungimo vardas: test@test.lt

Slaptažodis: stravinskas

Antrosios parduotuvės (Opencart) prisijungimo duomenys

Direktorija prisijungimui: /admin/index.php

Pagrindinis administratorius

Prisijungimo vardas: admin

Slaptažodis: opencart2013

Klientas

Prisijungimo vardas: test@test.lt

Slaptažodis: stravinskas

Trečiosios parduotuvės (osCommerce) prisijungimo duomenys

Direktorija prisijungimui: /admin/login.php

Pagrindinis administratorius

Prisijungimo vardas: admin

Slaptažodis: oscommerce2013

Klientas

Prisijungimo vardas: test@test.lt

Slaptažodis: stravinskas

7 priedas. Kompaktinē plokštelē