

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

ТИЖДЕНЬ НАУКИ-2019.
Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції серед студентів,
викладачів, науковців, молодих учених і аспірантів

15–19 квітня 2019 року

Електронне видання комбінованого
використовування на DVD-ROM

м. Запоріжжя

УДК 004(06)
Т39

*Рекомендовано до видання Вченою радою
Запорізького національного технічного університету
(Протокол №10 від 03.06.2019 р.)*

Упорядник Голуб Т.В.

Редакційна колегія:

Наумик В. В., д-р техн. наук, професор (відпов. ред.)
Прушківський В. Г., д-р екон. наук, професор
Кузькін О.Ф., канд. техн. наук, доцент
Глушко В.І., канд. техн. наук, доцент
Климов О.В., канд. техн. наук, доцент
Антонов М.Л., канд. техн. наук, доцент
Савченко В.О., канд. техн. наук, доцент
Кабак В.С., канд. техн. наук, доцент
Касьян М.М., канд. техн. наук, доцент
Корольков В.В., канд. екон. наук, доцент
Дєдков М.В., канд. іст. наук, доцент
Васильєва О.О., канд. фіз.-мат. наук, доцент
Пуцина І.В., канд. пед. наук, доцент
Філей Ю.В., канд. юр. наук, доцент
Гайворонська Т.О., канд. філос. наук, доцент
Сажєєв В. М., канд. техн. наук, доцент
Висоцька Н. І., начальник патентно-інформаційного відділу

Тези доповідей друкуються методом прямого відтворення тексту, представленою авторами, які несуть відповідальність за його форму і зміст.

Т39 Тижень науки-2019. Факультет комп'ютерних наук і технологій.
Тези доповідей науково-практичної конференції, Запоріжжя, 15–19 квітня 2019 р. [Електронний ресурс] / Редкол. :В. В. Наумик (відпов. ред.) Електрон. дані. – Запоріжжя : ЗНТУ, 2019. – 1 електрон. опт. диск (DVD-ROM); 12 см. – Назва з тит. екрана.
ISBN 978-617-529-224-2.

Зібрані тези доповідей, заслуханих на щорічній науково-практичній конференції серед студентів, викладачів, науковців, молодих учених і аспірантів. Збірка відображає широкий спектр тематики наукових досліджень, які проводяться на факультеті комп'ютерних наук і технологій Запорізького національного технічного університету. Збірка розрахована на широкий загал дослідників та науковців.

ISBN 978-617-529-224-2.

© Запорізький національний
технічний університет (ЗНТУ), 2019

ЗМІСТ

СЕКЦІЯ «КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»	8
<i>Грушко С.С., Зеленьова І.Я.</i> спосіб підвищення показників надійності інформаційно-керуючих систем на базі FPGA	8
<i>Проскурін М.П.</i> оптичні аналого-цифрові перетворювачі прямої дії для цифрових систем	10
<i>Sgadov S.A.</i> Implementation of protocol of serial port exchange with a Hercules development KIT board.....	14
<i>Сирота А.В.</i> Інтернет речей і хмарні обчислення	16
<i>Єфименко М.В., Луценко Н.В.</i> Динамічне рівняння для кватерніона в задачах керування кутовим рухом космічного апарата	18
<i>Golub T. V.</i> The influence of the training set size on the feature vector size for the text document classification	20
<i>Балюшко О.С., Киричек Г.Г.</i> Розгортання мережних ресурсів за допомогою технології GRON	22
<i>Верескун Ю.М., Проскурін Н.П.</i> Основні вимоги створення медичної інформаційної комп'ютерної системи	25
<i>Касьян К.М., Єльський В.В.</i> Спеціалізована система пошуку аудіо файлів на базі спектрального аналізу	29
<i>Здоровецький І.В., Куликовська Н.А.</i> SEO-аналіз тексту.....	32
<i>Тягунова М.Ю., Львіна Н.О.</i> Розробка методу автоматичного збору предметів у приміщенні	34
<i>Іващенко А.В., Куликовська Н.А.</i> Системи, засновані на знаннях	35
<i>Кармелюк К.О., Чубіч А.І., Паромова Т.О.</i> Методи оцінки продуктивності СУБД	37
<i>Кармелюк К.О., Юнусов О.І., Польська О.В.</i> Розробка GUI програмного забезпечення дослідницького стенду	39
<i>Луценко Н.В., Кошарна К.О., Роєнко О.І.</i> Використання системи контролю версій GIT при розробці програмного забезпечення	41
<i>Мазур Д.С., Куликовська Н.А.</i> Застосування нейронної мережі для відновлення зображень	43
<i>Павлішин М.А., Зеленьова І.Я.</i> Застосування апаратних прискорювачів на FPGA при розробці нейронних мереж.....	45
<i>Padalko B.O., Polska O.V., Kudermetov R.K.</i> State of the art of application performance management.....	47

<i>Потапов С.О., Касьян К.М.</i> Розпізнавання об'єктів інтер'єру по фотографії з подальшим їх трансформування в креслення для побудови ІОТ систем	50
<i>Предко В.Ю., Паромова Т.О.</i> Налаштування та усунення проблеми некоректного відображення даних в СУБД	52
<i>Тягунова М.Ю., Семенова К.В.</i> Математична модель оптимізації системи збору предметів у приміщенні	54
<i>Касьян К.М., Сичов В.В.</i> Автоматизація теплиці з дистанційним керуванням на основі ІОТ систем	56
<i>Тіменко А.В., Тіменко К.І., Божко С.С.</i> VPN для мобільних пристроїв на рівні мережі	58
<i>Тімохін А.А., Паромова Т.О.</i> Аналіз рейтингу систем управління базами даних	62
<i>Тіменко А.В., Романюта М.М.</i> З чого складається ІОТ	65
<i>Точилін С.Д., Демченко М.М.</i> Кросплатформний віртуальний амперметр-ресуратор	69
<i>Фалькевич В.Г., Киричек Г.Г.</i> Система на базі LARAVEL, оптимізована за допомогою FRONT END технологій	70
<i>Федько А.О., Зеленьова І.Я., Грушко С.С.</i> Аналіз апаратно-температурних характеристик при різних підходах опису керуючого алгоритму на мікросхемах XILINX FPGA	73
<i>Чиж С.Ю., Скруський С.Ю.</i> Побудова n-арних дерев із булевих виразів	75

СЕКЦІЯ «ПРОГРАМНІ ЗАСОБИ»..... 77

<i>Туленков А.В., Пархоменко А.В., Калініна М.В., Колпаков Д.А.</i> Дослідження особливостей апаратно-програмної платформи Legrand для реалізації систем Розумний будинок	77
<i>Залюбовський Я.І., Пархоменко А.В., Поздняков О.А.</i> Хмарні рішення SAP для реалізації технологій Інтернету речей	79
<i>Каврін Д. А., Субботін С. О.</i> Відновлення пропусків у вибірках з використанням методів еволюційного пошуку	80
<i>Субботін С.О., Корнієнко О.В., Наривський О.Е.</i> Нейромережа для моделювання корозії сталі	82
<i>Левада І.В.</i> Моделювання дискурс-процесу при проектуванні інтелектуальних систем	84
<i>Ткачук Є.М., Пархоменко А.В., Каплієнко Т.І.</i> Розробка керуючого пристрою системи «Розумний Будинок» для користувачів з особливими потребами	85
<i>Коломоєць Р.Р., Пархоменко А.В., Гладкова О.М.</i> Дослідження та розробка програмного забезпечення для заливки графічних зображень	86

<i>Білов О.І., Пархоменко А.В., Соколянський О.В.</i> Розробка віртуальної моделі для віддаленої лабораторії Розумний будинок.....	87
<i>Левченко О.А., Пархоменко А.В.</i> Дослідження особливостей розробки вбудованих систем на основі програмованих кристалів.....	89
<i>Льовкін В.М., Ананченко Є.М.</i> Розпізнавання емоційної забарвленості коментарів користувачів відеохостингу	90
<i>Льовкін В.М., Головізніна А.О.</i> Метод визначення рекомендацій кінофільмів на основі розпізнавання образів.....	92
<i>Льовкін В.М., Лукашенко Ю.О.</i> Розроблення програмного забезпечення пошуку друзів для спільного відвідування подій.....	94
<i>Льовкін В.М., Радєв П.А.</i> Ризик-орієнтований підхід до розробки програмної системи для автоматизації керування тепличними комплексами	96
<i>Льовкін В.М., Федічкін І.О.</i> Розроблення програмного забезпечення керування діяльністю станції технічного обслуговування	98
<i>Гулько Е.В., Гладкова О.М.</i> Розробка інтелектуальної системи керування світлофором	100
<i>Куц А.В., Пархоменко А.В., Гладкова О.М.</i> Розробка рекомендаційної бази знань медичного обладнання	102
<i>Попович В.В., Каплієнко Т.І.</i> Програмна реалізація клієнтської та адміністративної частин веб-сайту для продажу реквізитів для фокусів.....	103
<i>Симоненко О.С., Каплієнко Т.І.</i> Програмна реалізація інтернет-сайту для продажу годинників на основі власного фреймворку.....	105
<i>Волинська А.В., Зайко Т.А.</i> Використання концепції екстрагування при розробці програмного забезпечення	107
<i>Трубічина Д. І., Зайко Т.А.</i> Алгоритми та принципи вирішення задач за допомогою ТБВЗ.....	108
<i>Суха К.С., Зайко Т.А.</i> Принципи боротьби з протиріччями в інформаційних технологіях за допомогою технологій ТБВЗ.....	110
<i>Гринченко П.В., Зайко Т.А.</i> Функціональне розгортання технічної системи на основі ТБВЗ	112
<i>Мамутов Д.Ю., Зайко Т.А.</i> Проблеми підготовки та реалізації STARTUP у сфері ІТ	114
<i>Субботін С.О., Хохлова В.С.</i> Дослідження та програмна реалізація методів машинного навчання для вирішення задачі медичної діагностики	116
<i>Табуницик Г.В., Горелікова Т.О.</i> Програмна реалізація інтерактивного навчального середовища для роботи з RASPBERRY PI	118
<i>Миронова Н.О., Сергієнко О. С.</i> Програмна реалізація алгоритмів машинного навчання для обробки зображень	120
<i>Миронова Н.О., Сиротенко І.О.</i> Програмна реалізація медіасерверу для прийому та обробки відеоданих в реальному часі.....	121

<i>Миронова Н.О., Чадаєва А.К.</i> Програмна реалізація онлайн-сервісу для заміток та планування задач	123
<i>Миронова Н.О., Шемберко Ю.А.</i> Програмна реалізація методів декодування відео.....	124
<i>Миронова Н.О., Зінов'єв С.Д.</i> Розроблення програмного забезпечення прогнозування спортивних змагань.....	126
<i>Миронова Н.О., Лавренко А.А.</i> Розробка програмного забезпечення для обліку та аналізу відвідування учнями школи.....	127
<i>Сердюк С. М., Басанець М. І., Каменський Д. В., Діденко А. Є.</i> Розробка системи управління рестораном.....	129
<i>Сердюк С.М., Вічева Г.Ю.</i> Системний підхід до розробки системи моніторингу туристичних маршрутів.....	131
<i>Сердюк С.М., Козлов В.В.</i> Розробка концепту «Каса самообслуговування».....	133
<i>Сердюк С.М., Джрагацпаян Н.В.</i> Розробка інтерфейсу банкомата	135
<i>Сердюк С.М., Полумієнко Д.О., Дзандзава Г.М.</i> Розробка концепту «Смарт-ранцю».....	137
<i>Скачко Л.П., Чемерис К. М.</i> Розробка інтерфейсу бортового комп'ютера автомобіля.....	139
<i>Камінська Ж.К., Соколовський Д. В.</i> Розробка концепту «Робот-пилосос».....	141
<i>Камінська Ж.К., Цибульський О.Г.</i> Системний підхід до розробки спортивного застосунку	143
<i>Камінська Ж.К., Бережний О. Ю.</i> Розробка концепту гри «Навчальні шахи».....	145
<i>Степаненко О.О., Федорченко Є. М., Качан О.І., Борисенко В.О., Шевчук М.С.</i> Автоматизована система керування та моніторингу устаткування очищення води установкою зворотнього осмосу РО/ДАТ - 2/0,4	147
<i>Олійник А. О., Федорченко Є. М., Рудь М. С.</i> Дослідження та програмна реалізація еволюційних методів розв'язання задачі комівояжера при розпізнаванні образів	148
<i>Степаненко О.О., Федорченко Є.М., Харченко А.С., Гончаренко Д.А., Косміна О.В.</i> Розробка програмного забезпечення для аналізу та прогнозування стану здоров'я курців.....	150
<i>Степаненко О.О., Федорченко Є.М., Харченко А.С., Гончаренко Д.А., Кожушаний С.І.</i> Прогнозування погоди	152
<i>Степаненко О.О., Федорченко Є.М., Гончаренко Д.А., Харченко А.С., Малашок Н.М.</i> Експертна система співвідношення ціна-якість лікарських препаратів	154

<i>Рисіков В.П., Степаненко О.О., Федорченко Є.М., Скачко Л.П., Харченко А.С., Гончаренко Д.А., Плешко П.В.</i> Програмна модель виживання агентів у штучному середовищі.....	156
<i>Степаненко О.О., Федорченко Є.М., Гончаренко Д.А., Харченко А.С., Короткий О.В.</i> Програмна реалізація методів побудови нейронних мереж для задач медичної статистики.....	158
<i>Степаненко О.О., Федорченко Є.М., Харченко А.С., Гончаренко Д.А., Бережнюк О.О.</i> Дослідження використання мов програмування серверних скриптів та реалізація застосунку реального часу з використанням NODEJS та WEBSOCKET	159
<i>Дубровін В.І., Дейнега Л.Ю., Джрагацпанян Н.В., Соколовський Д.В.</i> Оптимізація оподаткування.....	161
<i>Короткий О.В., Українцева Є.С.</i> FLUTTER – новий підхід у розробці мобільних додатків.....	163
<i>Короткий О.В., Горбачов В.С.</i> Засоби штучного інтелекту для розширення можливості людей.....	165
<i>Зубко Е.О., Колпакова Т.О.</i> Перспективи використання сучасних фреймворків для швидкої розробки інтернет-магазину.....	168

СЕКЦІЯ «СИСТЕМНОГО АНАЛІЗУ ТА ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ» 171

<i>Пархоменко Л.О.</i> Аналіз структурного стану пористих систем.....	171
<i>Денисенко О.І.</i> Використання модуля LiveLink for MATLAB для вирішення задач оптимізації геометрії теплообмінних пристроїв.....	173
<i>Терещенко Е.В.</i> Ґратки формальних понять для аналізу біометричних даних	174
<i>Бакурова А.В., Ропало Г.М.</i> Концепція побудови інформаційної системи управління аптечною мережею	175
<i>Вакар Д.С., Терещенко Е.В.</i> Статистична обробка біометричної інформації польового експерименту.....	178
<i>Рябенко А.Є., Ляшенко А.Г.</i> Створення функціонально-аналітичного модуля Python для автоматизації роботи СУБД SQLite.....	179
<i>Бакурова А.В., Пасічник М.С., Терещенко Е.В., Філей Ю.В.</i> Багатофакторний аналіз динаміки злочинності в Україні	180

СЕКЦІЯ «КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»

УДК 658:004

Грушко С.С.¹, Зеленцова І. Я.²

¹канд. техн. наук, старш. викл. ЗНТУ

² канд. техн. наук, доц. ЗНТУ

СПОСІБ ПІДВИЩЕННЯ ПОКАЗНИКІВ НАДІЙНОСТІ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ НА БАЗІ FPGA

Однією з найбільш важливих частин складних технологічних систем є системи вимірювання та управління. Наявність великої кількості складних компонентів і підсистем, які забезпечують вимірювання параметрів, внесення необхідних поправок, моніторинг продуктивності, електроспоживання та ін., призводять до все більшого ускладнення таких систем. Незважаючи на це, інформаційні системи вимірювання та управління (СІУ), що застосовуються, наприклад, в бортовому обладнанні, повинні забезпечувати надійність, відповідність стандартам і безпеку технологічних процесів.

Програмовані логічні інтегральні схеми (ПЛІС) є сучасною елементної базою, що використовується у світовій промисловості. ПЛІС дозволяють реалізовувати як прості і портативні пристрої, так і складні цифрові системи контролю та управління. Різні функції СІУ реалізуються в базисі ПЛІС надійним і зручним способом. Використання ПЛІС має ряд переваг у вирішенні питань підвищення надійності проєктованих систем шляхом резервування, а також розпаралелювання процесів управління [1-2]. Тому розробка методів проєктування керуючих алгоритмів на ПЛІС, що дозволяють підвищити надійність систем управління, є актуальним завданням, що вимагає системного підходу при вирішенні.

На підставі аналізу особливостей проєктування відмовостійких об'єктів інформаційних технологій можна зробити висновок, що основні тенденції проєктування таких об'єктів спрямовані на попередження появи відмов системи засобами автоматичної нейтралізації і усунення їх усередині функціональних блоків системи, а також на запобігання ненавмисних і зловмисних вторгнень в процес роботи систем за рахунок апаратної, програмної і часової надмірності. Застосування технології «система-на-кристалі» є одним із ефективних заходів щодо забезпечення достатнього рівня надійності. У таких системах за рахунок високої оптимізації досягається зниження енергоспоживання, підвищення надійності та продуктивності, зменшення обсягу апаратного налагодження.

Застосування «систем-на-кристалі» дозволяє реалізувати в межах одного кристала як умовно-розподілений обчислювальний комплекс, так і модуль виявлення несправності і управління перемиканням.

На підставі наведеного вище аналізу проблеми і шляхів розв'язання, спрямованих на підвищення показників надійності СІУ, а також з огляду на той факт, що висока обчислювальна складність процесів управління бортовими системами визначає складність алгоритмів управління цими процесами, можна сформулювати завдання про необхідність розробки таких методів синтезу пристрою управління, які ефективно використовують структурні особливості ПЛІС і дозволяють підвищити показники надійності пристрою в цілому.

В якості вирішення поставленого завдання пропонується розглянути композиційний мікропрограмний пристрій керування (КМПК) з модифікацією структури, орієнтованою на елементний базис FPGA і ProASIC. Дослідження авторів показують, що структура композиційного мікропрограмного пристрою керування дозволяє найбільш повно використовувати як особливості внутрішньої архітектури елементного базису ПЛІС, так і особливості алгоритмів, що використовуються для апаратного забезпечення СІУ.

Композиційний мікропрограмний пристрій управління поєднує властивості моделей автоматів з жорсткою і програмованою логікою.

Беручи до уваги архітектурні особливості мікросхем FPGA, дану структуру КМПК пропонується адаптувати до властивостей елементного базису таким чином: комбінаційна схема (КС) може бути реалізована на елементах LUT, регістр і лічильник – на розподілених внутрішніх тригерах кристала FPGA, керуюча пам'ять (КП) - на блоках вбудованої пам'яті (embedded memory block, EMB). Схема формування мікрооперацій може бути реалізована на блоках вбудованої пам'яті, так і на елементах LUT. Структура композиційного мікропрограмного пристрою керування, модифікована з урахуванням внутрішньої структури FPGA, запропонована та проаналізована в рамках науково-дослідної роботи кафедри. При використанні запропонованого підходу до імплементації схеми керуючого автомата, можна поліпшити показники надійності за рахунок зменшення кількості внутрішніх між'єднань кристала.

Дослідження показали, що дана структура так само підходить для мікросхем ProASIC, зважаючи на наявність в них вбудованої пам'яті і логічних блоків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Logic Synthesis for FPGA-based Finite State Machines - Studies in Systems, Decision and Control, 38./ A.Barkalov, Titarenko L., M. Kolopienczyk, K. Mielcarek, G.Bazydło – New York: Springer, 2016. – 280 pp.
2. Control System Design Based on Modern Embedded Systems / Khamis, A., Zydek, D., Borowik, G., and Subbaram Naidu, D. // Springer Berlin Heidelberg. – 2013. – vol. 8112. – pp. 491–498.

УДК 621.3.087.92

Проскурін М.П.¹

¹ канд. техн. наук, доц. ЗНТУ

ОПТИЧНІ АНАЛОГО-ЦИФРОВІ ПЕРЕТВОРЮВАЧІ ПРЯМОЇ ДІЇ ДЛЯ ЦИФРОВИХ СИСТЕМ

Сучасна обчислювальна техніка, комп'ютерні системи, мережі і телекомунікаційні технології оперують в більшості цифровими форматами даних (ЦФД), в основі яких лежить двійкова (основа $q = 2$) системи числення (СЧ) і/або споріднені з нею ($q = 8 = 2^3$, $q = 16 = 2^4$, ін.) та їх комбінації і/або модифікації (наприклад, бінарно кодована десяткова СЧ, т.д.). Але навколишнє середовище (НС) має в більшості аналогову природу своїх характеристик (тиск, температура, швидкість повітря, інтенсивність/потужність сонячного випромінювання, ін.). Широке застосування персональних комп'ютерів (ПК), вбудованих комп'ютерних систем (ВКС [1]) в різні типи пристроїв, в тому числі і в технології IoT (Internet of Things [2]), ін., потребує швидкої і якісної взаємодії їх з елементами НС і іншими прикладними технологіями, наприклад, завдання: обробки відеозображень, медико-діагностичні оптоелектронні комплекси аналізу плазми крові, лімфи, їх компонентів; лідари, як елементи підсистем просторової орієнтації ВКС (керування авто без водія); промислові і/або побутові роботи, тощо, які неможливо уявити без використання аналого-цифрових перетворювачів (АЦП) [3]. Саме ці пристрої здатні якісно і вчасно вирішити задачі вводу параметрів і характеристик НС у цифрові пристрої, ПК, ВКС.

Серед багатьох типів АЦП в останній час приділено увагу саме оптичним АЦП, (людина отримує через очі більш ніж 90% об'єму інформації про НС з подальшим запам'ятовуванням, аналізом візуалізованого явища). Серед них особливо надшвидкісними є так звані паралельні АЦП прямої дії [4], які вирізняються між іншими тим, що використовують принципи безпосереднього перетворення вимірювальної величини і мають відповідну

внутрішню будову (архітектуру), яка це забезпечує. Теоретично-практичною моделлю такого надшвидкісного оптичного АЦП може бути сукупність Y-подібних волоконно-оптичних каналів (ВОК) у вигляді перевернутого стовбура дерева (вхід АЦП, куди направлене випромінювання), кожен з яких поділяє вхідну оптичну потужність випромінювання $P_{\text{випр.}}$ на половину (тобто на вдвічі меншу: принцип формування вагів двійкових розрядів в СЧ з $q = 2$) і аж до останнього «неподільного» фотона (в кожній такій гілці), яке може бути продетектоване на будь якому етапі (перетині) цих гілок, що формують багаторівневу структуру. Коли $P_{\text{випр.}}$ має мале значення, то «активованими» буде невелика кількість з n рівнів; коли $P_{\text{випр.}}$ має середнє значення, то «активованими» буде середня кількість з n рівнів; а коли ж $P_{\text{випр.}}$ буде великим, то «активованими» буде велика кількість з n рівнів ВОК. Таким чином по кількості «активованих» рівнів ВОК можливо швидко, точно і якісно оцінити значення вхідної оптичної потужності випромінювання $P_{\text{випр.}}$. Але практичне створення такої конструкції оптичного АЦП прямої дії потребує спеціальних прецезійних розподільників у вигляді Y-подібних ВОК, точного ділення $P_{\text{випр.}}$ в них навпіл і різних типів (по параметру чутливості) фотоприймачів (ФП) на кожному з n рівнів.

Оригінальною конструкцією оптичного АЦП, що можливо віднести до такого типу є пристрій по патенту РФ [5], який зображено на рис. 1, що містить: непрозорий корпус (1) з блоком відповідних світлофільтрів (2), мікролінзу об'єктива (3), що закріплена в непрозорому перетині (4), послідовно розташовані (по осі до мікролінзи об'єктива) напівпрозорі мікродзеркала (НМД) (51 - 5n) під відповідним до осі кутом по числу розрядів в коді і ФП (61 - 6n), що оптично з'єднані зі своїми НМД, імпульсні підсилювачі (71 - 7n) по числу ФП, послідовно з'єднаний регістр (8) з числом розрядів по числу імпульсних підсилювачів (7), дешифратор (ДШ) (9), блок індикації (10) і послідовно з'єднані генератор тактових імпульсів (11), дільник частоти (12) і логічний елемент 2І (13).

Принцип його дії полягає в тому, що кожне попереду розташоване НМД (5) пропускає на наступне за ним потік випромінювання, послаблений у два рази (відповідає принципу формування двійкового коду). У НМД застосовується світлорозподільне покриття, що забезпечує відношення відбитого випромінювання у ФП (6) до пропущеного як 1:0,5 (або 2:1). Вхідне випромінювання входить в непрозорий корпус (1), проходить світлофільтрів (2), збирається мікролінзою (3), що закріплена перетином (4) і направляється на центри n НМД (51 - 5n). Потік випромінювання проходить крізь них, яких набрано стільки, щоб після нижнього НМД (5n) випромінювання не було, тобто їх число повинне відповідати найбільшій величині вимірюваної потужності (яскравості) $P_{\text{випр.}}$. Результатом вимірювання є багаторозрядний двійковий код з одних «1» (код Джонсона),

бо кожне НМД послаблює потік вдвічі і т.д. Відбите кожним НМД (5) випромінювання надходить в свій ФП (6), електричний сигнал з якого надходить в свій імпульсний підсилювач (7), де посилюється до необхідної величини, формується за тривалістю, амплітудою і надходить в свій розряд регістра (8). Число розрядів n в регістрі (8) відповідає числу НМД (5). Старшим розрядом є сигнал «1» з першого імпульсного підсилювача (7₁), молодшим розрядом є сигнал «1» з останнього імпульсного підсилювача (7_n). Сигнали з імпульсних підсилювачів надходять паралельно і синхронно в розряди регістра. Частота видачі F результатів вимірювань встановлюється вимірником в дільнику частоти, з виходу якого сигнали надходять на перший вхід елемента 2І (13), на другий вхід якого надходить сигнал з виходу імпульсного підсилювача (7), а вихідний сигнал $U_{\text{вих.}}$ з елемента 2І (13) видає код з регістра (8) у ДШ (9) і в пристрій реєстрації накопичених результатів вимірювань. У ДШ (9) код дешифрується і надходить в блок індикації (10), що висвітлює результат вимірювання в десятковому (або ін.) коді. Процес АЦП визначається часом спрацювання ФП і становить $t_{\text{ФП}} \leq 10\text{-}6\text{с}$.

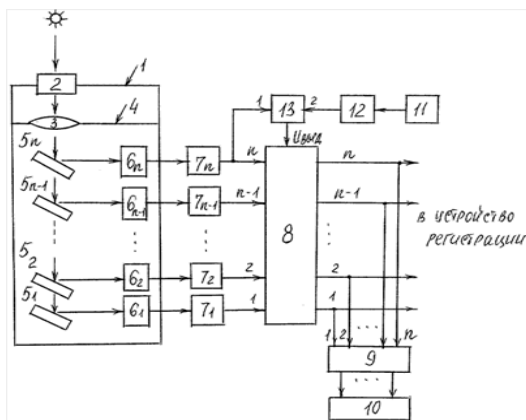


Рисунок 1 - Блок-схема АЦП по патенту №2419116 (RU), МПК G02A7/00, патентотримач Волков Б.И.

Однак вказаний АЦП має суттєві недоліки, які полягають в наступному:

- АЦП не має властивості адаптивності (підлагоджуваності) під потужність (яскравість) вхідного випромінювання $P_{\text{випр.}}$ при вимірюванні більшої потужності, ніж діапазон АЦП (n розрядний цифровий регістр і вихід якого має діапазон $N = 2n-1$);

- невисока точність і значна абсолютна похибка Δ вимірювання АЦП: виникає невизначеність при спрацюванні 1,2 ... і-го ФП з n , але не

спрацьовуванні $i+1$, якщо деяка частина випромінювання увійде в $i+1$ ФП, але не призведе до його спрацьовування. При цьому максимальне значення похибки може бути оцінено виразом $\Delta \leq (2^{i+1})$, тобто становить значення, що дорівнює вазі старшого розряду коду Джонсона (дивись вище) вимірюного значення $R_{\text{випр.}}$ і особливістю якого є те, що вага його старшого розряду задається кількістю одиниць «1» в молодших розрядах;

- використання ВЧ ФП без вбудованого в них підсилення, що підвищує кількість каскадів перетворення;

- відсутністю мікроконтролера (МК) управління електронної частини АЦП (з вбудованими в нього генератором імпульсів, лічильниками, внутрішніми блоками пам'яті, ін.) та схеми довготривалої зовнішньої пам'яті (для зберігання результатів вимірювань: попередніх, теперішніх, ін., їх угруповання в блоки даних).

В основу створення автором заявки на винахід з робочою назвою «Оптичний АЦП адаптивного типу з підвищеною точністю вимірювань» поставлено завдання розробки підлагоджувального оптичного пристрою, підвищення частоти перетворення до УВЧ, розширення діапазону вимірювання потужності (яскравості) випромінювання $R_{\text{випр.}}$ і точності вимірювання через зниження значення абсолютної похибки (до величини $\Delta \leq 2^{i-1}=0$ для цілочисельних форматів результату, тобто ваги молодшого розряду коду Джонсона), введення функції управління електронних частин АЦП швидкісним малорозрядним МК, а за рахунок цього - поліпшення експлуатаційних, технічних і його споживчих властивостей.

Було знайдено оригінальні підходи щодо вирішення вказаних недоліків оптичного АЦП прямої дії (рис.1), наприклад, використанням попередньої розробки (автора і його колег) - твердотільного ФП у вигляді інтегрального фотоприймального пристрою [6] УВЧ діапазона і зменшення кількості елементів в приймальній частині АЦП. Матеріали заявки оформлені і подані ним у інформаційно-патентний відділ ЗНТУ наприкінці 2018р. для їх розгляду, перевірки і відсилки в ДП «Укрпатент» для прийняття рішення по суті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вбудована_система [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Вбудована_система
2. Інтернет_речей. [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Інтернет_речей
3. Зибчук В.И. Справочник по цифровой схемотехнике. / В.И. Зибчук, В.П. Сигорский, А.Н. Шкуро // Киев:Тэхника. – 1990. – 448с.
4. Х.-К. Найтцерт. Оптическое аналого-цифровое преобразование: современное состояние. Ж. Датчики и системы (Измерения. Контроль.

Автоматизация). [Електронний ресурс]. Режим доступу: <http://naukarus.com/opticheskoe-analogo-tsifrovoe-preobrazovanie-sovremennoe-sostoyanie>

5. Волков Б.И. Преобразователь «яркость излучения - код»: патент № 2419116 Россия, G02F7/00. Подача заявки: 2010-03-16; публикация патента 2419116: 20.05.2011; [Електронний ресурс]. Режим доступу: <http://www.freepatent.ru/patents/2419116>

6. Проскурін М.П. Інтегральний фотоприймальний пристрій: деклараційний патент 68540А Україна: МПК G02B 6/12 . № 2003 30764962004 // Білявська О.С., Костенко В.Л., Проскурін М.П.; Заявл. 11.07.2003; опубл.16.08.2004, Бюл. №8.3 с.

UDK 681.527.2

Sgadov S.A.¹

¹senior lecturer ZNTU

IMPLEMENTATION OF PROTOCOL OF SERIAL PORT EXCHANGE WITH A HERCULES DEVELOPMENT KIT BOARD

The Hercules Development Kit development board is constructed on the basis of the TMS570LS310 microcontroller and can be used as a base of the control onboard computer. In this regard there is a problem of implementation of debugging by the serial protocol for data exchange with this board.

In this work author offer a software protocol of data which assumes that the microcontroller is an initiator of data exchange. Before data exchange the microprocessor sends a line command through a serial port which defines type and the direction of data. The modeling program on the PC listens to the specified serial port and after receipt of a command performs the corresponding operations on exchange.

The following algorithm of data exchange with the PC was used:

1. The microcontroller sends a request to GET
2. It expects for some time of arrival of data on port
3. If the timeout is exceeded, then repeats from p. 1
4. The accepted data are processed.
5. The microcontroller sends a request to SEND
6. The microcontroller sends data.

Data are transferred in internal representation. At the same time it is necessary to remember that in the TMS microprocessor uses reverse order of bytes both integer and float data types. Development have been provided in C++ and the

class system working with a serial port was created for PC and for the microcontroller.

The class **TTY** is responsible for initialization and byte data transmission through a serial port. For transfer of data types volume more than one byte created the derivative class **EnhTTY**. It expands a base class with the methods oriented to transfer of typified data.

Thus:

- **virtual int ReadInt(void) virtual void WriteInt(int data)** are intended for transfer of integer numbers.

Methods

- **virtual void WriteIntVector** (const int data [], int size);

- **virtual void ReadIntVector** (int * data, int size); - for transfer of arrays of integer numbers.

- **virtual double ReadDouble(void)** and **virtual void WriteDouble(double data)** - real numbers, and

- **virtual void WriteDoubleVector** (const double data [], int size);

- **virtual void ReadDoubleVector** (double data [], int size); - for transfer of arrays of real numbers and structures from real numbers.

- **virtual void ReadString(string & sendestring)** and

- **virtual void WriteString(string recievedstring)** - transfer zero - the named lines.

The library for transfer by serial port consists of modules: armtty.h, armtty.cpp, typedtty.h, typedtty.cpp, proconst.h, typestruct.h.

The architecture of classes for work with a serial port is similar to the classes considered above with the following changes in the class TTY.

- Work with port is based on hardware drivers created by HALCoGen. The Connect method is absent because the initialization of port is made from the constructor implicitly by sciInit call () from sci.h - so the speed of port is established in the 9600th baud.

- **TIMEOUT** is set in terms of cycles of the processor, but not in milliseconds.

- **uint32 _sciReceiveByte** method appeared which receives byte from port at a low level

- There is a member of the class **bool IsTimeout** which is established in true, at achievement of a timeout of input, reset is recommended to be carried out **ClearTimeout method call ()**.

Thus, the developed class library is invariant in relation to using program, and implementation of methods of a class differs for different architecture (the PC and TMS). Therefore, similar approach can be implemented also in relation to other boards and microprocessors.

ІНТЕРНЕТ РЕЧЕЙ І ХМАРНІ ОБЧИСЛЕННЯ

Інтернет речей (Internet of Things, скорочено IoT) - це глобальна мережа підключених до Інтернету фізичних пристроїв - «речей», оснащених сенсорами, датчиками і пристроями передачі інформації.

Ці пристрої об'єднані за допомогою підключення до центрів контролю, управління та обробки інформації.

Інтернет речей - одна з найпопулярніших концепцій у сучасній футурології. І більш того, одна з тих небагатьох, що вже перестають бути концепціями і втілюються в життя.

Згідно з найбільш поширеною формулюванні, Інтернет речей - це концепція обчислювальної мережі фізичних предметів (тобто власне, речей), які оснащені технологіями для взаємодії один з одним.

Концепція має на увазі, що Інтернет речей здатний серйозно вплинути на розвиток сучасного суспільства, оскільки дозволить багатьом процесам відбуватися без участі людини.

За даними Ericsson Mobility Report, сьогодні в світі налічується більше 16 млрд підключених пристроїв. До 2022 року - це число досягне 29 млрд, і 18 з яких будуть пристроями світу IoT [1].

Інтернет речей зв'яже між собою мільярди пристроїв без участі людини

Інтернет речей об'єднує реальні речі в віртуальні системи, здатні вирішувати абсолютно різні завдання. Ключова ідея концепції - з'єднати між собою всі об'єкти, які можна з'єднати, підключити до мережі

Для реалізації IoT необхідна екосистема, яка включала б у себе «розумні речі» - різні пристрої, оснащені датчиками; мережу доступу і передачі інформації (мобільну або фіксовану - не важливо); а також платформи для управління мережею, пристроями і додатками.

Умовно можна розділити всі IoT- проекти на дві групи в залежності від типу комунікації пристроїв: масові (Massive MTC) і критичні (Critical MTC). Перед кожним типом стоять свої завдання, і кожен з них має свої вимоги до мережі.

Масові IoT-проекти - це «розумні» будинки, лічильники, рішення для відстеження вантажоперевезень або сільського господарства, та ін.

Такі рішення передбачають передачу невеликої кількості даних від величезної кількості сенсорів. Також ці рішення характеризуються некритичністю гарантованої передачі-отримання інформації.

Інтернет речей буде складатися з мережі хмар або буде виглядати, як туман хмар - таке ось нове цікаве поняття. Туман (fog) дуже схожий на хмару, але його мета - взяти сервіси, навантаження, додатки і дані великого обсягу і перенести все це на периферію мережі.

Ідея туманних обчислень в тому, щоб розподілити дані, перемістивши їх ближче до кінцевих користувачів, щоб виключити затримку інформації і повторну їх передачу, а також підтримати мобільні обчислення і потокову передачу даних на мобільні пристрої.

З ростом обсягу інформації та використання хмарних сервісів туманні обчислення гратимуть ключову роль в зниженні затримок і в підвищенні зручності роботи користувачів. Тепер ми по справжньому розподіляємо дані і виносимо складні сервіси на периферію. В кінцевому підсумку це буде означати більш високу якість доступу до даних, вдосконалення корпоративної аналітики і загальне поліпшення роботи користувача з обчислювальним середовищем [2].

Сказане раніше підводить нас до розгляду платформи Everyware Device Cloud (EDC) - наступним кроком Euro - tech на шляху побудови готових рішень для Інтернету речей. Платформа Everyware Device Cloud забезпечує механізм доставки і управління даними в рамках бізнес підприємства. Everyware Device Cloud поєднає в собі хмару, M2M і Інтернет речей. Фахівці Eurotech пропонують новий тип хмари - DaaS (хмара пристроїв) = Device (пристрій) + SaaS (ПО на вимогу) [2]. Хмарна платформа спрощує управління пристроями і даними. Вона з'єднає розподілені пристрої і корпоративні бізнес-додатки, використовуючи безпечні, відкриті і надійні протоколи з підтримкою послуг.

Платформа дозволяє бізнес рішенням миттєво спрацьовувати на базі складних математичних і статистичних правил, що застосовуються до потоку даних від пристрою. EDC забезпечує унікальні можливості, що включають фільтрацію даних, безперервність запитів, агрегування і пошук кореляції між даними від пристроїв, розпізнавання подій за шаблоном, а також швидке реагування на основі сценаріїв виявлення критично важливих ситуацій.

В умовах високої конкуренції розробники таких сервісів перманентно працюють над продуктом. Використання IoT в хмарі "як послуги" змушують користувача переглянути інструменти в пошуках рішень, які є гнучкими, масштабованими і економічними.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сайт Ericsson Mobility Report [Електронний ресурс].- Режим доступу: <https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf>

2. Сайт Radiosovet [Електронний ресурс].- Режим доступу: <https://www.cta.ru/cms/f/460221.pdf>

УДК 681.5.01

Єфіменко М.В.¹, Луценко Н.В.²,

¹ канд. техн. наук, головний конструктор НВП "Хартрон-Юком", доц. ЗНТУ

² старш. викл. ЗНТУ

ДИНАМІЧНЕ РІВНЯННЯ ДЛЯ КВАТЕРНІОНА В ЗАДАЧАХ КЕРУВАННЯ КУТОВИМ РУХОМ КОСМІЧНОГО АПАРАТА

Успіх у вирішенні задач керування рухом космічного апарату (КА) відносно центру мас багато в чому залежить від обраної моделі руху КА. Математична модель кутового руху КА складається з двох груп рівнянь: динамічних рівнянь Ейлера і кінематичних рівнянь, записаних в тих чи інших кінематичних параметрах (кутах Ейлера-Крилова, напрямних косинусів, параметрах Родріга-Гамільтона). Залежно від того, які використовуються кінематичні параметри для опису кутової орієнтації КА можна виділити наступні види моделей кутового руху: 1) динамічне рівняння Ейлера і кінематичне рівняння в направляючих косинусів; 2) динамічне рівняння Ейлера і кінематичне рівняння в кутах Ейлера-Крилова; 3) динамічне рівняння Ейлера і кінематичне рівняння в параметрах Родріга-Гамільтона.

Серед цих моделей найпоширенішою є модель, в якій динаміка описується рівнянням Ейлера, а кінематика – кінематичним рівнянням в параметрах Родріга-Гамільтона. Рівняння цієї моделі в проекціях на осі зв'язаної системи координат мають такий вигляд:

$$J\dot{\omega} + \omega \times J\omega = M_u, \quad (1)$$

$$\dot{A} = \frac{1}{2} \begin{bmatrix} 0 & -\omega^T \\ \omega & -\Phi(\omega) \end{bmatrix} \cdot A, \quad (2)$$

де ω – абсолютна кутова швидкість обертання КА; J – момент інерції КА; M_u – керуючий момент; A – вектор параметрів Родріга-Гамільтона (нормований кватерніон), що визначає кутове положення КА в просторі; $\Phi(\omega)$ – лінійний кососиметричний оператор векторного добутку, що визначається рівністю $\Phi(\omega): y = \omega \times y$.

Перевагою цієї моделі є відсутність обчислювальних особливостей і мінімальна надмірність вектору стану. Недолік – нелінійність моделі, що істотно ускладнює синтез законів керування.

Крім перерахованих вище моделей кутового руху КА для побудови керування можна використовувати і модель руху, що має вигляд системи диференціальних рівнянь другого порядку щодо параметрів Родріга-Гамільтона:

$$\ddot{A} = f(A, \dot{A}, M_u) \quad (3)$$

Питання застосування такого роду моделей в задачах керування кутовим рухом КА розглянуті в роботах В.М. Кошлякова [1] і Ю.Н. Челнокова [2-4]. Однак в моделях, наведених в цих роботах праві частини динамічного рівняння Ейлера описуються безпосередньо через параметри Родріга-Гамільтона, що призводить до того, що моделі стають істотно нелінійними і їх важко використовувати для синтезу керування. Ці труднощі можна обійти, якщо при побудові моделі використовувати динамічне рівняння для кватерніона [5].

$$\dot{A} = (I_4 - AA^T)U - \|A\|^2 A \quad (4)$$

що описує безліч всіх допустимих керованих переміщень точки на поверхні сфери з одиничним радіусом в чотиривимірному просторі.

У доповіді з використанням рівняння руху точки по сфері приведено розв'язання основних задач керування кутовим рухом КА: задачі стабілізації і задачі термінального керування.

При розв'язанні задачі термінального керування запропонована методика синтезу алгоритмів просторової переорієнтації КА за заданий інтервал часу, яка не накладає ніяких обмежень на клас кутових рухів КА. Запропонований підхід на основі принципу максимуму дозволяє отримати аналітичний розв'язок задачі просторової оптимальної переорієнтації космічного апарату. Методика пройшла експериментальну перевірку.

На базі запропонованого підходу були реалізовані режими програмних поворотів космічних апаратів "Egypsat-1" і "Січ-2". Льотно-конструкторські випробування цих космічних апаратів показали високу ефективність запропонованої методики. Методика може бути корисною розробникам систем орієнтації КА.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кошляков В.Н. Задачи динамики твердого тела и прикладной теории гироскопов. / В.Н. Кошляков – М.: Наука, 1985. – 286 с.

2. Челноков Ю.Н. Кватернионное решение кинематических задач управления ориентацией твердого тела: Уравнения ошибок, законы и алгоритмы коррекции (стабилизации) / Ю.Н. Челноков // Изв. РАН. Механика твердого тела. –1994.–№ 4.– С.3-12.

3. Челноков Ю.Н. Управление ориентацией космического аппарата, использующее кватернионы / Ю.Н. Челноков // Космические исследования. – 1994. –№32, вып.3. –С.21-32.

4. Челноков Ю.Н. Построение управлений угловым движением твердого тела, использующее кватернионы и эталонные формы уравнений переходных процессов. Ч.1,2 / Ю.Н. Челноков // Изв. РАН. Механика твердого тела. – 2002. – №1. –С.3-17.

5. Кириченко Н.В. Алгоритмы асимптотической, терминальной и адаптивной стабилизации вращательных движений твердого тела/Н.В.Кириченко, В.Т.Матвиенко//Проблемы управления и информатики.- 2003. – № 1. – С.5-15

UDC 004.4
Golub T. V.
assistant ZNTU

THE INFLUENCE OF THE TRAINING SET SIZE ON THE FEATURE VECTOR SIZE FOR THE TEXT DOCUMENT CLASSIFICATION

One of the factors of the text document classification process complexity is the size of the feature vector, which is used as input to assess the ownership of a category document.

The feature vector is defined on the basis of the training sample processing results presented in the text document form. The text in this case is processed by a number of preliminary steps, including the removal of stop words, stemming, and others. Then we weigh the obtained terms to evaluate their degree of importance.

Statistically, the size of the training set affects the size of the feature vector and, as a result, the complexity and quality of the classification. A feature of the prize vector for a separate category is that to classify a text document into this category, it is necessary to match the terms of the analyzed document with the elements of this vector. So, there are a finite number of key terms that describe a separate category. Therefore, a significant increase in the size of the training set may not provide a significant qualitative increase in the size of the feature vector.

Table 1 shows the dependence of the size of the feature vector on the size of the training sample separately for each category. The number of words in the training collection is indicated in the first column for each category. The resulting

size of the feature vector obtained as a result of the analysis of the training set presented in the second column separate for each category.

Table 1 - Dependence of the feature vector size on the training set size

1 category		2 category		3 category		4 category	
Word summa	Term vector	Word summa	Term vector	Word summa	Term vector	Word summa	Term vector
10336	2036	12475	1273	8629	2084	6715	1955
14138	2491	13859	1649	14004	2681	10392	2439
19973	3305	15760	1806	16238	2956	14648	2758
23207	3734	16743	1955	19947	3536	15243	2894
29333	4448	17459	2074	21641	3804	19427	3457
32982	4931	24714	2796	24017	4183	23158	3885

Figure 1 shows a plot of the normalized size of the feature vector for separate categories relative to the first category, taken as a reference, from the size of text documents of the training set. This calculation is presented in table 2.

Table 2 – Normalized values of dependence of the feature vector size on the training set size

1 category		2 category			3 category			4 category		
Word summa	Term vector	Word summa	Term vector	Norm. vector	Word summa	Term vector	Norm. vector	Word summa	Term vector	Norm. vector
10336	2036	12475	1273	1536	8629	2084	1740	6715	1955	1270
14138	2491	13859	1649	1616	14004	2681	2656	10392	2439	1793
19973	3305	15760	1806	1425	16238	2956	2403	14648	2758	2023
23207	3734	16743	1955	1410	19947	3536	3039	15243	2894	1901
29333	4448	17459	2074	1234	21641	3804	2806	19427	3457	2290
32982	4931	24714	2796	2095	24017	4183	3046	23158	3885	2728

As can be seen from Table 1 and Figure 1, the size of the training set (Word sum) increases and the increase in the size of the feature vector decreases (lines of 1-4 categories). This indicates the inexpediency of a significant increase in the size of the training sample to increase the set of the terms variety of the feature vector.

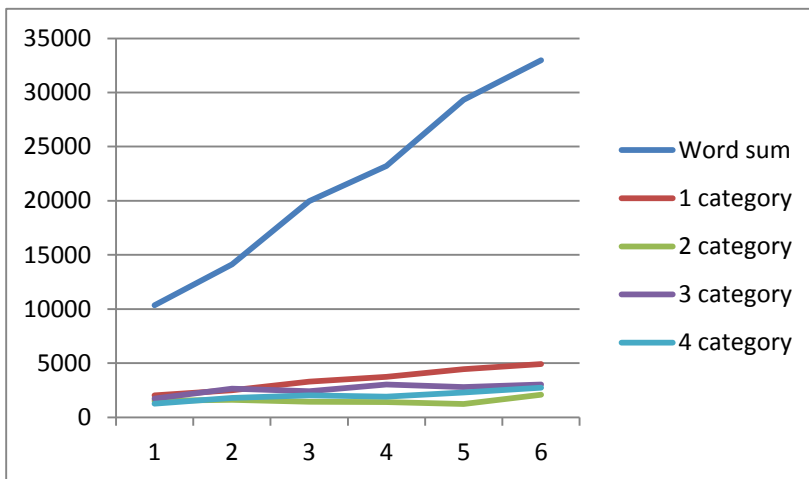


Figure 1 - Dependence of the term vector size on the learning set size

Therefore, further study of the determination of the sampling set optimal size is appropriate.

УДК 004.7

Балюшко О.С.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-626сп ЗНТУ

² канд. техн. наук, доц. ЗНТУ

РОЗГОРТАННЯ МЕРЕЖНИХ РЕСУРСІВ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ GPON

У сучасному світі ринок послуг зв'язку є дуже затребуваним та вимагає вдосконалення механізму передачі великих обсягів даних, гарантуючи надійність каналів і стійкість їх до відмов у роботі. У вік хмарних технологій, стандарт Fast Ethernet вже не задовольняє запитам користувачів, які використовують корпоративні та глобальні мережі.

Gigabit Passive Optical Network (GPON) є представником технологій оптичних мереж доступу PON. В основу PON, при розробці мереж доступу, закладені принципи пасивної оптичної мережі і ці переваги перейняла технологія GPON [1]. До переваг технології GPON відносимо: активне обладнання використовується мінімально; кабельна інфраструктура

мінімізована; вартість обслуговування низька; дуже добра масштабованість та абонентські порти мають високу щільність (рис.1).

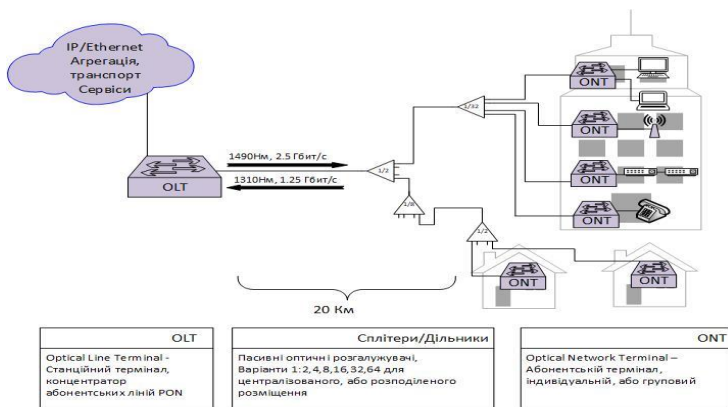


Рисунок 1 - Принципіальна схема роботи технології GPON.

Етапи які треба виконати для розрахунку оптичної мережі, наступні: розрахунок суми затухань для кожної лінії без урахування втрат в розгалужувачах; визначення та калькуляція коефіцієнтів розподілу кожного розгалужувача, починаючи з найбільш віддалених и закінчуючи першим сплітером; розрахунок бюджету втрат для кожного абонентського терміналу з урахуванням втрат у всіх елементах ланцюга та порівняння його з динамічним діапазоном системи [2].

У зв'язку з тим, що абоненти знаходяться на різній відстані від головної станції, то при рівномірному розподілі потужності в розгалужувачі, вона на вході кожного ONU є різною. Виходячи з цього треба підібрати такі дільники, які на вході абонентського пристрою створюють приблизно однаковий рівень сигналу, який є не нижче вказаного в технічному паспорті ОНТ. Це є важливим з двох причин. По-перше, для подальшого розвитку мережі треба мати приблизно рівномірний запас по загасанню в кожній гілці «дерева» PON. По-друге, у випадку коли мережа не є збалансованою, на термінал OLT від різних ONU, в загальному потоці, прийдуть сигнали, які дуже сильно відрізняються за рівнем. При цьому система детектування не зможе відпрацьовувати значні перепади ($> 10-15\text{дБ}$) сигналів (табл.1), тому це набагато збільшить кількість помилок при прийомі зворотного потоку.

Таблиця 1 – Втрати при використанні спліттерів.

Дільник	1 x 3	1 x 4	1 x 5	1 x 6	1 x 8	1 x 12	1 x 16	1 x 32
Максимальне затухання, дБ	6.1	7.2	8.5	9.4	10.7	12.5	14.4	18.5

Для визначення втрат, які вносять розгалужувачі з великою кількістю вихідних портів, можна скористатися формулою:

$$A_i = 10 \lg \left(\frac{100\%}{D\%} \right) + \lg 2(N - 1) \cdot 0,4 + 0,2 + 1,5 \cdot \lg \left(\frac{100\%}{D\%} \right), \quad (1)$$

де D% - відсоток потужності на певному вихідному порту; N- кількість вихідних портів; i - номер вихідного порту.

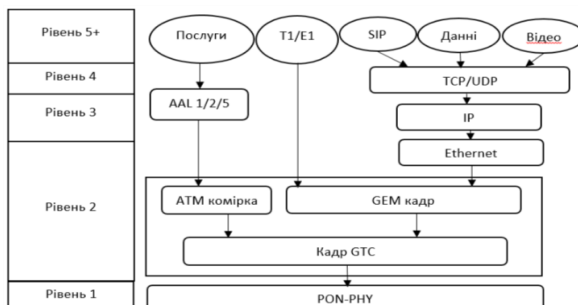


Рисунок 2 - Формування кадру в мережі GPON.

GPON використовує звільнений ресурс та при передачі кадрів Ethernet через мережу GPON не виконує їх фрагментацію. На вході до мережі GPON до кадру додається GPON-мітка, а на виході преамбула кадру перетворюється до стандартного вигляду та мітка змінюється [3]. OLT модифікує преамбулу кожного вихідного кадру, зокрема в преамбулу додається спеціальний тег LLID. Цей тег витягується відповідним підрівнем на ONT, де відбувається відновлення преамбули. Кінцевий вузол ONT при нормальному режимі роботи, коли він є зареєстрованим, обробляє тільки кадри, які мають в преамбулі ідентифікатор, що збігається з їх власним LLID. Інші поля кадру GPON відповідають полям стандартного кадру Ethernet.

В мережі GPON, разом з кадрами даних, передаються і службові кадри фіксованої довжини 64 байти. Поля преамбули і адрес керуючого кадру і кадру даних GPON однакові. Інші поля керуючого кадру мають наступні

дані: L / T – розмір поля 2 байти, для керуючого кадру поле містить значення 0x8809, по ньому вузол GPON відрізняє керуючий кадр від інформаційного кадру; orcode (optional code) - поле 2 байти, задає тип керуючого кадру (повідомлення GATE, які задаються OLT і повідомлення REPORT, які задаються ONT); TS (time stamp) - поле 4 байти, є тимчасовою міткою та поле розміром 40 байт, яке містить службову інформацію, яка необхідна для роботи протоколу MPCP [4].

За рахунок цього і досягається щільність до 126 ONT на порту та швидкість 2.5 Гб/с на низхідному потоці.

Висновки. Мережа, яка побудована за таким принципом є легко масштабованою. Дана технологія по своїй економічності і можливостям надавати негарантований канал до 2.5Гб/с, ідеально підходить для провайдера при наданні недорогої послуги доступу до мережі інтернет та на 100% покриває потреби користувача.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hood D. Gigabit-capable Passive Optical Networks / D. Hood, Elmar Trojer - Basic Books, 2012 – 448 p.
2. Леонов А. Технология PON - эффективная сеть доступа / А. Леонов, В. Кобышев. – М.: CONNEKT, 2007. – С.110-114.
3. ITU-T Recommendation G.984.2 – Gigabit-capable Passive Optical Networks (GPON): Physical Media Dependent (PMD) layer specification – approved by ITU-T Study Group 15 (2001-2004) under the ITU-T Recommendation A.8 procedure on 16 March 2003. – 38с.
4. Красько О.В. Методи та алгоритми підвищення ефективності функціонування конвергентних оптичних мереж доступу: дис. канд. техн. наук: 05.12.02 – телекомунікаційні системи та мережі / О.В. Красько; МОНУ, НУ«Львівська політехніка». – Львів, 2018. – 153 с.

УДК 004.942

Верескун Ю.М¹, Проскурін Н.П.²

¹ студ. гр. КНТ-524з ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ОСНОВНІ ВИМОГИ СТВОРЕННЯ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ

Медико-діагностична інформаційна комп'ютерна система (ІКС) використовує стандартизовані рішення для зберігання і передачі діагностичної інформації. База даних (БД), що зберігає інформацію про

відвідування пацієнтів в строгому ієрархічному порядку, створена на файлової основі у вигляді ієрархічної структури каталогів (папок), які в залежності від її місця в ній, зберігає файли з описом, додаткову інформацію і папки нижчого рангу.

Така структура є найпростішою і дозволяє організовувати легкий доступ до БД по мережі, без необхідності установки додаткових драйверів і розміщення її на сервері установи дозволяє організовувати віддалений доступ до будь-якого її елементу. Така БД легко переноситься на різні носії і при цьому на одному носії можна зберігати їх кілька.

ІКС типу «Сімейний лікар» при роботі з відвідувачами створює його тимчасову копію на локальному носії і, при всіх змінах в процесі роботи, вносить змінені дані в цю локальну копію. При збереженні відвідування локальна копія переноситься в БД. Цей механізм спільно з обробкою журналу транзакцій дозволяє автоматично відновлювати втрачені дані в разі будь-яких збоїв, наприклад, при виключенні живлення ПК. Вся інформація про пацієнта і відвідини, крім сигналів записаних з обладнання комплексу, зберігається в стандартизованому для передачі даних форматі XML.

Клієнтська частина - це комп'ютерні апаратно-програмні комплекси (АПК) реєстрації даних. Це можуть бути такі ж АПК, як і в центрі обробки, так і спрощені їх версії, встановлені на планшетах, кишенькових комп'ютерах, смартфонах, ін., що забезпечує мобільність пунктів реєстрації даних (ПРД). Однак спрощені версії не можуть використовуватися для детального аналізу зареєстрованих даних, але дозволяють переслати їх для розгорнутого аналізу в центр обробки даних (ЦОД) і прийняти результати обробки.

Основні вимоги до медико-діагностичної ІКС типу «Сімейний лікар»:

- асинхронна передача даних. Сьогодні вже існують системи телемедицини, що використовують телефонні канали зв'язку або радіоканали. Але в цих випадках необхідна синхронізація передавальної і приймальної сторін каналу, потрібно, щоб вони були готові до обміну інформацією і мали активовані програмні комплекси. У пропонованому нами варіанті сторона може передавати дані без попереднього узгодження з приймальною. Передавати дані можливо одночасно з багатьох ПРД і з тією швидкістю, яку може забезпечити локальний канал підключення i-го пункту. Також будь-яка кількість ПРД може одночасно завантажувати результати обробки своїх даних з сервера;

- передача даних здійснюється через Інтернет і при цьому використовується зашифроване з'єднання (https). Використовуються надійні і налагоджені механізми TLS-з'єднань. При необхідності, можлива організація свого виділеного / захищеного каналу передачі. Залежно від конкретних умов, завжди можна вибрати найбільш прийнятний варіант підключення: локальна

мережа, wi-fi, мобільний зв'язок або супутниковий канал, ін.. Крім того, Інтернет відкриває і інші можливості, наприклад, віддалена підтримка користувачів, оновлення ПЗ, відеозв'язок з ЦОД, ін., причому прийом / обробка даних може бути з фіксованим часом або цілодобово;

- можливість передавати кілька записів одночасно. При проведенні в фельдшерсько-акушерському пункті періодичного медогляду сеанс передачі даних, наприклад, проходить в кінці робочого дня, а результати обробки завантажуються наступного дня у зручний час;

- можливість обробки записів декількома фахівцями. У складних випадках в ЦОД може бути організовано режим консультацій, коли запис одного і того ж дослідження пред'являється кільком фахівцям одночасно. Після завершення консультації кожен фахівець буде працювати у власному режимі, завантажуючи дані чергового дослідження з БД сервера;

- територіально всім фахівцям центру зовсім не слід перебувати в одному місці. Кожен може працювати на своєму робочому місці, і, навіть, вони можуть перебувати в різних містах;

- організація багаторівневої обробки. Будь-центр обробки може одночасно бути клієнтом іншого центру. Наприклад, лікар районного центру обробки може в складних випадках посилати записи в обласний / федеральний центр для консультації з більш кваліфікованими фахівцями;

- централізоване ведення статистики на сервері. Записи з сервера можуть бути видалені, але на ньому залишається інформація про те, хто і коли надіслав запис, хто і коли її прийняв і обробив. Видалити / змінити подібні записи буде неможливо ні лікарям центру, ні клієнтам. При необхідності, записи з сервера можуть переноситися в спеціально організований архів даних, або, наприклад, переноситися в БД відповідних лікувальних установ.

Відповідно до виділених функціональних підсистем та з урахуванням вимог управління визначається склад задач, що розв'язуються. У функціональних задачах, для автоматизації яких призначена підсистема, відображаються специфічні особливості кожної конкретної підсистеми певного об'єкта. Саме задача є об'єктом розробки, впровадження та експлуатації кінцевим користувачем. Її зміст визначає сукупність вихідних показників, які формуються і обчислюються в задачі за відповідними алгоритмами. З появою нових ІТ інструментів поняття "задача" розглядається як закінчений комплекс опрацювання інформації, що забезпечує видачу її для прийняття рішень.

Функціональна складова автоматизованої ІКС є домінуючою. Вона завжди пов'язана з проблемними сферами і фактично є моделлю системи управління конкретним об'єктом. До функціональної частини належать ті елементи, які визначають її функціональні можливості, а саме: призначення,

виконання управлінських функцій та обробка інформації. Основними елементами функціональної частини автоматизованої ІКС є: функціональні системи, підсистеми, блоки (або комплекси задач) та окремі задачі.

Функціональна підсистема - це відносно самостійна частина системи, виділена за спільністю функціональних ознак управління. В автоматизованих інформаційних системах органів казначейства функціональні підсистеми, як правило, виокремлюють за такими ознаками:

- стадіями управління (планування, облік, звітність, аналіз, контроль тощо);
- видами основної діяльності (доходи, видатки, трансферти тощо);
- організаційною структурою (структурні підрозділи);
- функціональною ознакою (виконувані функції).

Вибір та обґрунтування складу функціональних задач є одним з найважливіших елементів створення інформаційних систем. Розвинені автоматизовані інформаційні системи дають змогу реалізувати багатоваріантні алгоритми розрахунку показників на основі вибору з банку моделей ефективних математичних методів, моделей і алгоритмів для конкретного об'єкта управління. При цьому кожна конкретна задача при розробці автоматизованих ІКС повинна розглядатися в інформаційному зв'язку з іншими задачами даної підсистеми та інших підсистем, а також із зовнішніми автоматизованими ІКС (наприклад, ІКС «Сімейний лікар»). Такий підхід вирішується створенням окремих модулів (підмодулів), що формують простір рішення завдань.

- Суб'єктами інформаційної взаємодії в рамках інтеграції медичних ІКС є:
- медичні організації (здійснюють первинний облік надання допомоги);
 - територіальні органи управління охорони здоров'я (здійснюють збір звітних даних) і фонди обов'язкового медичного страхування;
 - регіональне відділення Фонду соціального страхування, яка здійснює збір даних саме медичної складової обслуговування окремих категорій громадян;
 - територіальний фонд казначейства для сплати за медичні послуги.

Висновки.

1. Розроблена ескізна модель ІКС (з можливістю використання сервісу «телемедицина») для її застосування в розвитку напрямку «Сімейний лікар» реформи медичного обслуговування.

2. Акцентовано, що обмін електронними даними в рамках ІКС здійснюється на основі стандартних технологій і протоколів із шифруванням даних.

3. Наведено склад модулів, підмодулів ІКС, функціональних підсистем.

СПЕЦІАЛІЗОВАНА СИСТЕМА ПОШУКУ АУДІО ФАЙЛІВ НА БАЗІ СПЕКТРАЛЬНОГО АНАЛІЗУ

В ході розвитку комп'ютерних технологій великої популярності набула глобальна мережа «Інтернет». В наш час важко представити людину, яка б не користувалася можливостями мережі. Одна із таких можливостей – веб-сервіси, які пропонують користувачам мережі різні можливості, такі як збереження на віддаленому сервері музики та вільний доступ до неї у будь-який проміжок часу.

Оскільки користувачів мережі багато, то і об'єм даних, який треба зберігати великий. Виникає проблема – колізійні аудіо файли. Декілька різних користувачів можуть завантажити, незалежно один від одного, один і той-самий аудіо файл. Така інформація в базі даних зайва. Результатом пошуку будь-яким користувачем такого файлу буде безліч однакових файлів.

Існуючі системи використовують комерційну систему звіту для відстеження колізійних файлів, тому процедура додавання нового файлу до системи може затягнутись на довго. Інші ж системи не відшукують колізії. Користуватись такими системами незручно. Запропонована система, при додаванні користувачем нового аудіо файлу на сервер, проаналізує акустичний відбиток за допомогою аналізу спектра сигналу на всьому проміжку часу та сформує запит до бази для перевірки колізій. Лише після перевірки система завантажить новий аудіо файл на сервер.

В ході аналізу існуючих систем з'явилась ідея використання дещо модифікованої системи для користувацького спеціалізованого пошуку аудіо файлів. За допомогою акустичного відбитку на базі спектру можна знаходити різні аудіо файли, які будуть дещо схожими на еталонний файл. Кожний користувач подібних сервісів чи не раз хотів знайти музику за його смаком, але яку він не знає. В ході дослідження було виявлено, що можна додати коефіцієнт схожості, тобто наскільки еталонний аудіо файл може відрізнятись від результуючого.

Для успішного виконання етапу завантаження файлу на сервер необхідно отримати файл та форматувати його у зручний, для подальшої обробки, вид. Основою запропонованого алгоритму є амплітудно-частотна характеристика на кожному проміжку часу, яка описується як одномірний масив для кожного моменту часу. Для виділення АЧХ необхідно поділити

сигнал на кадри та для кожного кадру виконати дискретне пряме перетворення Фур'є:

$$X_k = \frac{1}{N} \sum_{i=0}^{N-1} x_n e^{-j \frac{2\pi k i}{N}} = \frac{1}{N} \sum_{i=0}^{N-1} x_n [\cos \frac{2\pi k i}{N} - j \sin \frac{2\pi k i}{N}]. \quad (1)$$

Після цього звуковий сигнал буде представлений у вигляді коефіцієнтів амплітуд та фаз частот:

$$\text{Re } X_k = \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi k i}{N}, \quad (2)$$

$$\text{Im } X_k = \frac{1}{N} \sum_{i=0}^{N-1} \sin \frac{2\pi k i}{N}, \quad (3)$$

$$\text{Abs}[X_k] = \sqrt{\text{Re } X_k^2 + \text{Im } X_k^2}, \quad (4)$$

$$\phi[X_k] = \tan^{-1} \frac{\text{Im } X_k}{\text{Re } X_k}, \quad (5)$$

$$A[k][i] = X_i \sqrt{\cos^2 \frac{2\pi k i}{N} + \sin^2 \frac{2\pi k i}{N}}, \quad (6)$$

де $i=1,2,\dots,n$.

Виконавши таке перетворення на всьому проміжку часу, розбитому на кадри, отримаємо двомірний масив $A[T][n]$, де T – кількість моментів часу; n – кількість частотних діапазонів. Отримаємо допоміжний масив:

$$\text{AVF}[i] = \frac{1}{T} \sum_{k=1}^T A[k][i], \quad (7)$$

де $i = 1,2,\dots,n$, n – кількість частотних діапазонів.

Для пошуку схожих файлів необхідно отримати масив AVF для еталонного файлу, який назвемо EVF . Виконаємо пошук у базі за умовою $\text{EVF}=\text{AVF}$. Якщо результат існує, то виконати точний пошук за $E=A$, де E – масив, утворений таким самим чином, як і масив A . Для проведення

користувачького пошуку додається коефіцієнт схожості $L \in [0;1]$. Замість точного пошуку виконується порівняльна функція $abs(AVF[i] - EVF[i]) \geq L$.

Спектр випадкової музичної композиції, розкладений на масив AVF наведено на рисунку 1.

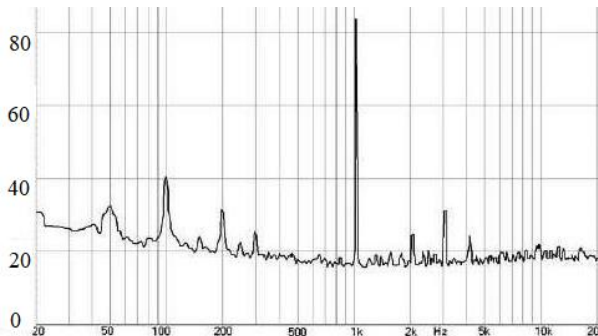


Рисунок 1 – Графічне представлення масиву AVF випадкового файлу.

Для перевірки та подальшого дослідження було створено прототип системи:

- звичайний пошук за акустичним відбитком не є оптимальним, тому необхідно використовувати двох етапний пошук;
- спеціалізований користувачький пошук працює справно, але використовувати його, як основний варіант пошуку аудіо файлів на сервері не можна. Найкращого результату можна досягти використовуючи звичайний пошук за тегами на кшталт «автор композиції» та «жанр» з функцією спеціалізованого пошуку схожих файлів;
- виконання пошуку файлу у базі за частотним аналізом зайняло на експериментальній системі 2.124 с., виконання двох етапного пошуку зайняло всього 0.129 с. На великому об'ємі даних система буде вести себе інакше.

SEO-АНАЛІЗ ТЕКСТУ

За обсягами даних World Wide Web і безліч інших систем передачі даних в мережі Інтернет є самим великим з коли-небудь існували в історії людства сховищ даних. Головне завдання пошукових систем - надавати користувачам сторінки, які максимально відповідають їхнім запитам. Головна проблема власників сайтів, які хочуть пробитися в топ видачі - створювати контент, який точно "догодить" пошуковцям.

Багато компаній все більше і більше цікавляться методами аналізу текстів. Це можуть бути запити клієнтів, внутрішнє листування, чати з клієнтами, внутрішня база знань, огляд преси або дані парсинга соцмереж. Все це ставить на перше місце питання, яким чином можна аналізувати в автоматичному режимі великі обсяги даних і надалі використовувати отримані дані.

Пошукова оптимізація (англ. Search engine optimization, SEO) - комплекс заходів щодо внутрішньої і зовнішньої оптимізації, для підняття позицій сайту в результатах видачі пошукових систем по певних запитах користувачів, з метою збільшення трафіку і потенційних клієнтів (для комерційних ресурсів) і подальшої монетизації цього трафіку.

Переваги SEO:

- якісний і актуальний контент забезпечить сайту високе місце в пошуковій видачі і стабільний приплив трафіку на сайт;
- створення якісного контенту забезпечує не тільки SEO-оптимізацію сайту, але і розширення присутності в соціальних мережах і social media;
- просування сайту по низькочастотних запитах призводить на сайт цільової трафік, який можна конвертувати, пропонуючи пропозиції;
- SEO-оптимізація спрямована на підвищення позицій сайту в пошукових системах і репутації сайту за рахунок посилань на сайт на зовнішніх джерелах;
- SEO-технології забезпечують корисність сайту для відвідувачів.

Недоліки SEO:

- створення та адміністрування якісного контенту вимагає багато часу і ресурсів (фінансових, трудових та ін.);
- для ефективної SEO-оптимізації сайту крім створення якісного контенту необхідні навички роботи з технічними аспектами оптимізації (редирект, робота з Хостін-гами і ін.), що вимагає залучення веб-розробників;

- у різних пошукових системах переважають різних чинники ранжирування, тому необхідно розробляти кілька SEO-стратегій;
- необхідно постійно проводити моніторинг сайту і використовувати веб-аналітику.

На жаль, більшість SEO-оптимізаторів не використовують статистичну близькість між своєю оптимізацією (входами) та відповідними їй пошуковими результатами (виходами). Більш того, їх входи або оптимізації знаходяться перед компонентами обходу в пошуковій системі. Фактично типова оптимізація повинна пройти через 4 рівня: обхід вмісту, індексування, скорингу та, в кінцевому підсумку, рівень запиту в реальному часі.

Для вирішення оптимізаційних задач використовується поєднання семантичних і математичних моделей. Семантична модель використовується для того, щоб відобразити все багатство атрибутів і зв'язків елементів системи, що моделюється. На моделі проводяться логічні і математичні обчислення, які дозволяють знайти оптимальне рішення. Як правило, таке моделювання є козним: модель постійно оновлюється за рахунок надходження нових відомостей про стан системи, що моделюється, і, відповідно, автоматично перераховується оптимальне рішення, що призводить до корекції програми дій.

Серед величезного числа алгоритмів, які використовуються для пошуку і аналізу інформації, особливе місце займають ті, чия мета - виявлення прихованих закономірностей або неочевидних залежностей.

Про один з методів, який застосовується для рекомендаційних систем (коллаборативна фільтрація), інформаційного семантичного пошуку, поділу текстів по тематиках без навчання та багатьох інших і піде мова далі. Метод цей називається латентно-семантичним аналізом (LSA - Latent semantic analysis). Можна сказати, що це просунутий SEO аналіз тексту.

Його суть полягає в тому, що для нас не важлива порядок слів у документі, в яких морфологічних формах вони представлені, а важливо тільки кількість входжень конкретних слів.

На обґрунтуванні тільки цієї інформації, будується таблиця «слово-документ». Де рядки відповідають словам (а точніше, їх леми), а стовпці - документам. У кожному осередку зберігається 1, якщо слово є в документі, і 0 - якщо ні. Хоча такий варіант і найпростіший, але не найкращий. Замість 0 і 1 можна використовувати, наприклад, частоту слова в документі або tf-idf слова. Такий спосіб представлення текстів у вигляді таблиці (або матриці) називається векторною моделлю тексту. Тепер, для того щоб порівняти два документи, потрібно визначити міру схожості двох стовпців таблиці.

Висновки. LSA дозволяє підтримувати SEO-оптимізацію контенту. Після використання LSA ми отримуємо семантичне уявлення слів і документів - це дозволяє знаходити неочевидні зв'язки між словами і

документами. До недоліків алгоритму можна віднести дуже велику обчислювальну складність методу.

УДК 621.38

Тягунова М.Ю.¹, Ільїна Н.О.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-514м ЗНТУ

РОЗРОБКА МЕТОДУ АВТОМАТИЧНОГО ЗБОРУ ПРЕДМЕТІВ У ПРИМІЩЕННІ

На сьогоднішній день пристрої зі штучним інтелектом набувають багато нових корисних для людини можливостей та стають невід’ємною частиною життя суспільства [1], тому актуальним є створення системи, яка дозволить збирати предмети в приміщенні з мінімальним втручанням людини.

Саме з цією метою було розроблено систему автоматичного збору предметів на базі робоплатформи Alphabot та апаратно-програмного комплексу Arduino Uno [2]. Для реалізації системи було створено метод збору предметів в приміщенні, завдяки якому робот рухається за оптимальним шляхом, що сприяє не тільки ефективному функціонуванню системи, а й значному енергозбереженню.

Сутність метода полягає в тому, що кімната, в якій проводиться збір предметів, являє собою матрицю точок, розмір яких відповідний системі збору предметів. Робот розташовується в тих точках, де він має можливість розвернутися ліворуч, далі ультразвуковий датчик визначає відстань до найближчого предмета або стіни кімнати. Точку, що розташована поруч зі стіною, робот відзначає як «корзину» для збору предметів, досягнувши її, робот обертається на 180 градусів та прямує до наступної перешкоди.

При знаходженні предмета, робот здійснює його захват, визначає мінімальний шлях від свого місцезнаходження до «корзини», куди відвозить предмет, та повертається на точку, де він був розташований, з якої здійснює пошук нових предметів для збору. Якщо знайдений предмет має завеликі розміри, що перевищують допустимі параметри, робот об’їжджає його.

Коли система автоматичного збору предметів досягає кінця кімнати, вона визначає мінімальний шлях до точки початку своєї роботи, повертається на неї та автоматично вимикається.

Тому, розроблений метод автоматичного збору предметів в приміщенні дозволяє з мінімальним втручанням людини та оптимальними енергозатратами збирати малогабаритні предмети в домашніх або промислових умовах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рассел С. Искусственный интеллект: современный подход / С. Рассел. - М. : Диалектика-Вильямс, 2018. - 1408 с.
2. Момот М. Мобильные роботы на базе Arduino. 2-е издание. / М. Момот. - СПб. : БХВ-Петербург, 2016. - 336 с.

УДК 004.22

Іващенко А.В.¹, Куликовська Н.А.²

¹ студ. гр. КНТ-514м ЗНТУ

² асист. ЗНТУ

СИСТЕМИ, ЗАСНОВАНІ НА ЗНАННЯХ

Невід'ємною частиною будь-якої інтелектуальної системи є знання. Проблема вилучення знань, їх збереження і розподілу - сьогодні одна з найголовніших проблем організацій даних для комп'ютерних систем.

Для вирішення такої проблеми є системи, засновані на знаннях. Існують наступні типи систем, заснованих на знаннях, як: експертні системи; нейронні мережі; системи, засновані на прецедентах; генетичні алгоритми, інтелектуальні агенти та інші. Найвідоміші серед них це перші три.

Експертні системи моделюють процес прийняття рішення, властивий людському мозку. Вони використовуються для того, щоб діяти подібно експертам-людям, допомагаючи фахівцям прийняти рішення, ґрунтуючись на експертних знаннях. До переваг експертних систем можна віднести: процес прийняття рішення відбувається на основі заздалегідь встановлених правил; швидкість отримання рішення. Недоліками є: висока вартість розробки; орієнтація на вузьку область знань; потреба в експертах для придбання знань.

Нейронні мережі моделюють роботу людського мозку на біологічному рівні. Це означає, що нейронні мережі мають вбудовану здатність до самонавчання. До переваг штучних нейронних мереж відносимо: можливість вирішення завдань, які не вирішуються ніякими іншими методами; самонавчання. Недоліками є: дорожняча апаратної реалізації; для великих мереж неможливість заздалегідь приблизно оцінити час навчання мережі.

У системах, заснованих на прецедентах, в базі знань містяться самі ситуації або прецеденти, а не опис неузгаальнених ситуацій. У цьому випадку рішення проблеми зводиться до пошуку по аналогії. Переваги системи: скорочення трудомісткості придбання нових знань; навчання системи відбувається на основі додавання нових прецедентів, а не зміни або додавання нових правил. Недоліки системи: необхідність суворо

відстежувати правильність оформлення прецедентів; необхідність розробки методів пошуку, що однозначно ідентифікують відповідні прецеденти.

Всі ці системи пов'язані з перспективною галуззю інженерії знань, яка займається розробкою моделей, методів і алгоритмів аналізу даних для отримання знань і навчання. Отримання знань є головним процесом в інженерії знань. Без цього процесу всі інші технології стають маловживаними. Оскільки експертна система або будь-яка інша система, заснована на знаннях, повинна містити в собі в першу чергу знання експертів-людей тієї чи іншої предметної області, процес отримання знань в першу чергу передбачає залучення експертів в діалог з інженером по знаннях.

Для отримання знань від експертів широко використовуються різні способи співбесіди (інтерв'ю). Виділяють чотири основних типи інтерв'ю: неструктуроване інтерв'ю, при якому інженер отримує базові уявлення про предметну область; структуроване інтерв'ю, яке має чіткий план і використовується для отримання детальної інформації про предметну область; випадок з минулого, при якому інженер по знаннях просить експерта згадати конкретний випадок з минулої практики, коли він застосовував свої навички в предметній області; думки вголос, при якому так само, як і в випадку з минулого, розбирається конкретна ситуація (з реального життя або змодельована), але експерта просять усі міркування робити явно, вголос.

Комп'ютерні системи використовуючи інженерію знань стають більш адаптивними. Ядром адаптивної комп'ютерної системи є модель проблемної області (підприємства), що постійно розвивається, підтримувана в спеціальній базі знань, на основі якої здійснюється генерація або конфігурація програмного забезпечення. Таким чином, проектування і адаптація комп'ютерних систем, заснованих на знаннях, зводиться, перш за все, до побудови моделі проблемної області та її своєчасному коректуванню.

Висновки. Використання інженерії знань в комп'ютерних системах вирішує питання формалізації, вилучення, зберігання, розподілення й отримання необхідних знань. Окрім того, комп'ютерні системи, використовуючи інженерію знань, набуває здатність до розвитку системи у відповідності з об'єктивними змінами моделі проблемної області.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Башмаков А.И. Интеллектуальные информационные технологии / А.И. Башмаков. – М.: Издательство МГТУ им. Баумана, 2005. – 324 с.
2. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хоросhevский. – СПб.: Питер, 2000. – 201 с.

3. Джанетто К. Управление знаниями / К. Джанетто, Э. Уилер. – М.: Добрая книга, 2005. – 180 с.

4. Емельянова Н.З. Основы построения автоматизированных информационных систем / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. – М.: ФОРУМ-ИФРА-М, 2007. – 270 с.

5. Волкова В.Н. Информационные системы в экономике / В.Н. Волкова, В.Н. Юрьев, С.В. Широкова, А.В. Логинова. - М.: Издательство Юрайт, 2017. - 402 с.

УДК 004.384

Кармелюк К.О.¹, Чубіч А.І.¹, Паромова Т.О.²

¹ студ. гр. КНТ-526 ЗНТУ

² старш. викл. ЗНТУ

МЕТОДИ ОЦІНКИ ПРОДУКТИВНОСТІ СУБД

В даний час продуктивність СУБД є вирішальним фактором ефективності управлінських і комерційних додатків. Як правило, продуктивністю називають величину, обернено пропорційну часу, який СУБД витрачає на певну операцію по обробці даних. Важливо враховувати відмінності між класами СУБД (розрізняють системи загального призначення і спеціалізовані) при виборі програмного середовища, що має відповідні йому переваги і недоліки, так як орієнтованість функціоналу системи і способи його реалізації впливають на продуктивність системи.

Тому метою дослідження є визначення методів тестування та оцінки продуктивності СУБД, а також розгляд методів її підвищення.

Головною проблемою продуктивності є складність її оцінки. До метрик, за якими оцінюється СУБД, відносять:

- кількість транзакцій в одиницю часу, час відгуку або затримки;
- масштабованість – підтримка стабільної продуктивності системи в умовах мінливого навантаження;
- цілісність і коректне відображення даних – останній стан системи повинен відображатися скрізь однаково, зв'язки в БД повинні відповідати проєктній документації;
- ACID (атомарність, послідовність, ізоляція і міцність) властивості транзакцій;
- час потрібний для завантаження додатку [1].

Єдиний спосіб перевірки якості роботи СУБД – тестування продуктивності системи в різних умовах. Існує два типи еталонного тестування: повне (тестування всієї програми) і покомпонентне.[2] В MySQL

є зручна функція BENCHMARK(), яка визначає швидкість виконання певних типів операцій. Але вона ніколи не використовується для реального еталонного тестування через нечітку визначеність границь вимірювань.[3]

Повне тестування зазвичай є кращим методом отримати чітке уявлення про продуктивність системи. Для цього використовують наступні програмні інструменти, наведені в таблиці 1.

Таблиця 1 – Програмні інструменти для тестування

Інструменти повного тестування	Інструменти тестування компонентів
<p><i>ab</i> – показує, скільки запитів в секунду здатний обслуговувати HTTP-сервер;</p> <p><i>http_load</i> – дозволяє створити вхідний файл, що включає багато різних адрес URL, при цьому утиліта буде вибирати їх випадковим чином;</p> <p><i>JMeter</i> – дозволяє гнучко емулювати поведінку реальних користувачів, керуючи часом наростання навантаження.</p>	<p><i>mysqlslap</i> – емулює навантаження на сервер і видає дані хронометражу;</p> <p><i>sysbench</i> – дає уявлення про продуктивність ОС в термінах факторів, істотних для роботи сервера бази даних;</p> <p><i>Super Smack</i> – дозволяє емулювати кількох користувачів, завантажувати тестові дані в базу і заповнювати таблиці випадковими даними.</p>

Крім того, за необхідності програміст може самостійно створювати інструменти, які б повністю підходили для тестування обраної СУБД.

Найбільш ефективними способами підвищення продуктивності є індексація, яка прискорює запити, відмова від використання циклів з великою кількістю ітерацій, використання механізму блокування таблиць в MySQL, застосування оператора EXISTS() для перевірки наявності запису в таблиці, використання типів даних фіксованого розміру та типів необхідних розмірів для зберігання даних, використання для атрибутів типу ENUM та оголошення їх як NOT NULL.[4]

Отже, ми з'ясували, що еталонні тести відображають реальну поведінку додатка, що дозволяє оцінити різні характеристики продуктивності компонентів СУБД окремо і в цілому. У подальшому планується виконати кількісну оцінку продуктивності СУБД двома описаними способами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тестування баз даних [Електронний ресурс]. - Режим доступу: <http://getbug.ru/testirovanie-baz-dannyih/>

2. MySQL. Оптимізація продуктивності, 2-е видання / Б. Шварц, П. Зайцев, В. Ткаченко та ін. - Пер. з англ. - СПб.: Символ-Плюс, 2010. - 832с.

3. MySQL: інструменти та утиліти для тестування сервера [Електронний ресурс]. - Режим доступу: <https://oracle-patches.com/db/mysql/3835-mysql-instrumenty-i-utility-dlya-testirovaniya-servera>

4. Документація по MySQL [Електронний ресурс]. - Режим доступу: <http://www.mysql.ru/docs/>

УДК 004.4

Кармелюк К.О.¹, Юнусов О.І.¹, Польська О.В.²

¹ студ. гр. КНТ-526 ЗНТУ

² старш. викл. ЗНТУ

РОЗРОБКА GUI ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОСЛІДНИЦЬКОГО СТЕНДУ

Існує чимало підходів до розробки GUI із властивими їм перевагами та недоліками, однак початкові етапи розробки майже не залежать від застосовуваних у подальшому технологій та інструментів. Інфраструктура GUI будується на базі архітектури програмного забезпечення (ПЗ), проте процеси їх створення тісно пов'язані між собою. Таким чином, здійснюються наступні кроки:

- визначається завдання або набори задач, для яких продукт призначений [1];
- формуються вимоги і загальний алгоритм роботи програми;
- обирається архітектурний шаблон програми та шляхи його реалізації в потоку (потоках) виконання;
- формуються уявлення про функціональні та інформаційні елементи як основу інтерфейсу.

Важливим при визначенні задач є дотримання правила: прості задачі повинні залишатися простими незалежно від рівня складності всієї системи. Складні задачі можуть вимагати складних інтерфейсів, але це не виправдовує ускладнення простих задач [2].

Наступним етапом розробки є макетне проектування, яке супроводжується написанням контекстного, ключового та перевірочних сценаріїв поведінки програми.

Контекстні сценарії використовуються для формування бачення того, як продукт може найкращим чином задовольнити потреби користувача. В результаті додавання до контекстного сценарію більш детальних описів взаємодії користувача з продуктом і застосування «проектної лексики» він

стає ключовим сценарієм, який фокусується на найбільш важливих моментах взаємодії. За мірою уточнення образу продукту розробки ці сценарії паралельно з макетним проектуванням проходять ітераційну доробку. Крім того, у ході всього процесу застосовують перевірочні сценарії – для тестування проектних рішень у різноманітних ситуаціях [3].

Для наочності вичерпного опису функціональних вимог до системи використовують методiku візуального моделювання за допомогою уніфікованої мови моделювання UML, де будуються діаграми поведінки, взаємодії тощо. Важливим є те, що на основі UML-моделей можлива генерація коду.

У рамках роботи над розробкою GUI дослідницького стенду було розглянуто підходи з використанням наступних типів програмних інструментів (IDE): GUI-конструктори; IDE з редактором коду (без GUI-конструктора); IDE з редактором коду і з можливістю застосування GUI-конструктора.

Кожен з підходів має свої переваги та недоліки. Плюсом GUI-конструктора є можливість графічно упорядковувати елементи інтерфейсу та відразу бачити результат впроваджених змін, при цьому редагування параметрів елементів відбувається за допомогою меню. Деякі конструктори, наприклад, Glade, автоматично генерують весь вихідний код для елемента інтерфейсу.

Однак, використовуючи GUI-конструктор, не завжди можливо передбачити усі зміни, які мають відбуватися з елементами інтерфейсу, рівно як і всі зв'язки, що формуються між ними. Без GUI-конструктора графічний інтерфейс користувача створюється шляхом написання коду із вказанням параметрів кожного елемента інтерфейсу в вихідному коді без візуального зворотного зв'язку до запуску програми [4].

У наслідок аналізу можливостей розглянутих програмних інструментів обрано середовище розробки Qt Creator, що поєднує застосування редактору коду і GUI-конструктора та разом з тим спрощує роботу з вбудованим кросплатформним фреймворком Qt.

Під час розробки GUI ПЗ дослідницького стенду використано, зокрема, такі модулі бібліотеки: QtWidgets, QObject та QThread. Два останніх компоненти надають розробнику можливість підвищити продуктивність ПЗ за рахунок розпаралелювання процесорних обчислень і операцій вводу-виводу. Особливістю організації сполучення обчислювального потоку програми з потоком GUI є використання сигнально-словових зв'язків.

Прийняті рішення дозволять розробити GUI дослідницького стенду, який відповідатиме сучасним вимогам до графічних інтерфейсів користувача.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Проектування графічного інтерфейсу користувача [Електронний ресурс]. - Режим доступу: <https://habr.com/ru/post/208966/>
2. Raskin, Jef. The Humane Interface: New Directions for Designing Interactive Systems. USA: Addison-Wesley Professional, 2000 – 233 p.
3. About Face: The Essentials of Interaction Design, 4th Edition / A. Cooper, R. Reimann, D. Cronin, C. Noessel. – 4th edition – USA: Wiley, 2014 – 720 p.
4. GUI-конструктор [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/GUI-конструктор>

УДК 004.384

Луценко Н.В.¹, Кошарна К.О.², Роєнко О.І.²

¹ старш. викл. ЗНТУ

² студ. гр. КНТ-525 ЗНТУ

ВИКОРИСТАННЯ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ GIT ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес розробки програмного продукту складний і дуже важливий, один з основних напрямків у світі. Залежно від проекту, кількість рядків коду може варіюватися від декількох сотень до кількох тисяч, а іноді і мільйонів. При розробці програмного забезпечення (ПЗ) використовують різні системи контролю версій, які значно поліпшують та прискорюють процес розробки.

Система контролю версій (СКВ) – це набір консольних утиліт, які відстежують і фіксують зміни в файлах проекту, що розробляється, (найчастіше мова йде саме про вихідний код програм, але можна використовувати його для будь-яких файлів). З допомогою СКВ можна повернутися на попередню версію проекту, бачити зміни, порівнювати, аналізувати і багато іншого.

На сьогоднішній день при розробці ПЗ використовуються централізовані СКВ (CVS, Subversion, Perforce), децентралізовані СКВ (Git, Mercurial, Bazaar, Darcs, BitKeeper) [1]. Найбільш популярною серед розробників ПЗ є система контролю версій Git [2].

Система Git є розподіленою, тобто не залежить від одного центрального сервера, на якому зберігаються файли. Замість цього система працює повністю локально, зберігаючи дані в папках на жорсткому диску, які називаються репозиторієм. Потім даними, що збережені в локальному репозиторії, можна обмінюватися з іншими користувачами. Часто при роботі

з Git створюють центральний репозиторій, з яким інші розробники синхронізуються.

Одна з ключових особливостей Git – гнучкість. Ця система підтримує різні види нелінійних технологічних процесів розробки, ефективна як в невеликих, так і в великих проєктах та сумісна з багатьма існуючими системами і протоколами. У Git розгалуження і операції над гілками (наприклад, злиття або повернення) зберігаються як частина історії змін. Не всі системи контролю версій використовують цей рівень відстеження. Це дозволяє в процесі роботи бачити, який учасник проєкту вносив конкретні зміни, мати можливість швидко повернути зміни файлу в початковий стан, мати зручні інструменти для аналізу внесених змін в кожен файл, кожен рядок вихідного коду.

В порівнянні з іншими системами контролю версій система Git має свої переваги та недоліки [3].

Переваги при використанні системи Git:

- незалежна робота над проєктом кожного учасника;
- просте вирішення конфліктів – при виконанні операції "merge" можна порівняти зміни в двох файлах і прийняти рішення по об'єднанню внесених змін;
- швидке виправлення помилок, виявлених під час тестування;
- можливість швидкого оновлення release-версії проєкту – внесення змін не вимагає зупинки робочого рішення, що сприяє безперебійній роботі;
- простий консольний інтерфейс для вибору вхідних в commit змін;
- двостороння синхронізація з CVS і Subversion (але commit в CVS експортуються наполовину вручну);
- Git може бути в якості CVS-сервер.

Недоліки системи Git:

- не відстежується зміна окремих файлів, а тільки всього проєкту в цілому, що може бути незручно при роботі з великими проєктами, які містять безліч незв'язаних файлів. Але є рішення – треба розбити проєкт на частини, кожна з яких складається з взаємозв'язаних файлів;
- при початковому (першому) створенні сховища та синхронізації його з іншими розробниками, потрібно досить тривалий час для скачування даних, особливо, якщо проєкт великий, так як необхідно скопіювати на локальний комп'ютер весь репозиторій;
- високий поріг входження (навчання) для тих, хто раніше використовував інші СКВ;
- обмежена підтримка Windows (в порівнянні з Linux).

На підставі вище сказаного, можна зробити такі висновки:

- система контролю версій Git відповідає всім вимогам сучасного процесу розробки програмного продукту;

- використання даної системи полегшує командну роботу розробників, прискорюючи пошук помилок та їх виправлення;
- в порівнянні з іншими системами – Git більш гнучка, сучасна та більш підходить для розробки великих програмних проєктів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Обзор систем контроля версий [Електронний ресурс]. – Режим доступу: http://all-ht.ru/inf/prog/p_0_1.html
2. Pro Git [Електронний ресурс]. – Режим доступу: <https://git-scm.com/book/ru/v2/>
3. Девять причин перейти на Git [Електронний ресурс]. – Режим доступу: <http://tarantsov.com/2007/12/nine-reasons-to-use-git.html>

УДК 004.8:004.92

Мазур Д.С.¹, Куликовська Н.А.²

¹ студ. гр. КНТ-514м ЗНТУ

² асист. ЗНТУ

ЗАСТОСУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ

Розвиток сучасних технологій дозволив значно спростити різні галузі виробництва. Використання комп'ютерних технологій дозволило підвищити рівень якості виробництва та зменшити вплив людського фактору. Зараз дуже великої популярності набувають нейронні мережі, що можуть виконувати рутинні для людини дії.

Штучна нейронна мережа (ШНМ) - математична модель, а також її програмне або апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж - мереж нервових клітин живого організму. Такі мережі здатні навчатися, що дозволяє використовувати їх різних сферах, включаючи і обробку зображень.

Під обробкою зображень мається на увазі ряд дій, що направлені на зміну якості зображення. Сюди можна віднести і ретушування, зменшення розміру зображення, збільшення розміру зображення, стискання зображення, зміну палітри кольорів тощо.

Зараз досить розповсюдженим є відновлення старих чорно-білих фотографій. Тобто відновлюється цілісність зображення старої фотокартки відновлюється. Також зображення розфарбовується кольорами відповідно до оригіналу. Цей процес виконується за допомогою засобів редагування зображень, таких як Photoshop. Але відновлення зображень вручну досить

монотонне, складне для людини. Тому актуально використовувати ШНМ для полегшення процесу відновлення зображень.

В даному випадку буде розглянуто простий алгоритм з застосуванням згорткової нейронної мережі. Такий алгоритм може не дати стовідсоткового результату, тому його зазвичай беруть за основу, але додаються інші засоби для більш точного відновлення та розфарбовування старих зображень.

Основна логіка для застосування нейронних мереж досить проста. Чорно-білі зображення можна представити у вигляді сітки з пікселів. У кожного пікселя є значення яскравості, що лежить в діапазоні від 0 до 255, від чорного до білого.

Як і в чорно-білому зображенні, пікселі кожного шару кольорового зображення містять значення від 0 до 255. Нуль означає, що у цього пікселя в даному шарі немає кольору. Якщо у всіх трьох каналах стоять нулі, то в результаті на зображенні виходить чорний піксель.

Нейронна мережа встановлює взаємозв'язок між вихідними та результуючими значеннями, тобто вона повинна знайти сполучні риси між чорно-білими і кольоровими зображеннями. Тобто необхідно знайти властивості, за якими можна зіставити значення з чорно-білої сітки зі значеннями з трьох кольорових.

Далі в якості вихідних даних треба взяти шар з градаціями сірого, і на його основі згенерувати кольорові шари a і b в колірному просторі.

Для отримання двох шарів з одного шару, треба скористатися згортковими фільтрами. Їх можна представити як синє і червоне скло в 3D-окулярах. Фільтри визначають, що буде на картинці. Вони можуть підкреслювати або приховувати якусь частину зображення, щоб наше око витягнув потрібну інформацію. ШНМ теж може за допомогою фільтра створити нове зображення або звести кілька фільтрів в одну картинку.

У згортковій нейронній мережі кожен фільтр автоматично підлаштовується, щоб легше було отримати потрібні результуючі дані. Треба нанести сотні фільтрів, а потім звести їх воедино і отримати шари a і b .

На вході є сітка, що представляє чорно-біле зображення. А на виході - дві сітки зі значеннями кольорів. Між вихідними та результуючими значеннями треба створити сполучні фільтри. Для навчання мережі використовуються кольорові зображення. Найпростішим буде перетворити їх у чорно-білі та подати на вхід, а на виході отримуються два розфарбованих шари.

Далі в одному діапазоні зіставляються (\tanh) обчислені значення з реальними, тим самим порівнюючи їх один з одним. Для зіставлення обчислених значень використовується функція активації \tanh (гіперболічна тангенціальна). Якщо застосувати її до якогось значення, то функція поверне значення в діапазоні від -1 до 1.

Реальні значення кольорів змінюються від -128 до 128. Якщо кожне значення розділити на 128, то всі вони виявляться в межах від -1 до 1. Така «нормалізація» дозволяє порівнювати похибки обчислення.

Після обчислення результуючої похибки мережа оновлює фільтри, щоб скорегувати результат наступної ітерації. Вся процедура повторюється циклічно, поки похибка не стане мінімальною.

Код такого алгоритму найпростіше буде реалізувати за допомогою мови Python, тому що він досить простий у використанні та має велику кількість бібліотек для роботи з нейронними мережами та зображеннями будь-яких форматів. Також слід відзначити, що для виконання цього алгоритму необхідна досить потужна відеокарта. Тому найвдалішим буде використовувати сервіси хмарного графічного обчислення, наприклад, FloydHub.

Правильна реалізація такого алгоритму дозволить відновлювати старі зображення за лічені хвилини, коли для відновлення цих же зображень вручну людині знадобиться значно більше часу та зусиль.

УДК 658:004

Павлішин М.А.¹, Зеленцова І.Я.²

¹ студ. гр. КНТ-514м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ЗАСТОСУВАННЯ АПАРАТНИХ ПРИСКОРЮВАЧІВ НА FPGA ПРИ РОЗРОБЦІ НЕЙРОННИХ МЕРЕЖ

Актуальність даної теми обумовлена тим, що штучний інтелект безумовно, має перспективне майбутнє в галузі комп'ютерних обчислень, притому досить велика кількість досліджень проводиться у спробах створити надійний та корисний штучний інтелект. Останнім часом у світі стрімкими темпами розвивається нова область прикладної математики, що спеціалізується на штучних нейронних мережах (ШНМ). Актуальність досліджень у цьому напрямку підтверджується великою кількістю різних галузей використання ШНМ. Прикладами цих галузей є: автоматизація виробництва, апроксимація, прогнозування, створення експертних систем, організація асоціативної пам'яті, системи комп'ютерного зору, IoT, Blockchain та багато інших.

Сфери використання штучних нейронних мереж є досить різноманітними. При цьому така сфера задач ШНМ, як розпізнавання образів, є достатньо значимою у побудові комп'ютерного зору, але вона є досить трудомісткою, у зв'язку із природою зображень та швидкістю їх

надходження у систему. Частина досліджень ШНМ включає в себе вивчення області, що називається глибоке навчання (Deep Learning), вона є трудомісткою та потребує багато ресурсів і паралельної обробки. Наприклад, за допомогою ШНМ можна прогнозувати показники бірж, діагностувати хвороби або, наприклад, робити оцифровку рукописних документів, що мають важливе технічне значення або зберігшихся до наших днів давніх рукописів. Також процес оптичного розпізнавання символів прискорить введення персональних даних про громадян до систем обліку та баз даних і допоможе позбутися помилок, які притаманні людині. Це як мінімум, прискорить роботу та зробить сервіси більш зручними.

Необхідність використання апаратних прискорювачів в нейронних мережах зумовлена певними причинами. Оскільки у задачах штучного інтелекту грає роль швидкодія та ресурси доцільно обирати системи із використанням додаткових апаратних засобів для обчислення. Пов'язано це з тим, що обсяги оброблюваної інформації ростуть досить великими темпами, а час очікування щодо збільшення кількості ядер або частоти центральних процесорів є достатньо великою величиною (від року та більше, в залежності від виробника), тому є необхідність у використанні апаратних прискорювачів. Економічна складова також грає важливу роль, тому зменшення енерговитрат є однією з пріоритетних задач. Останнім часом у світі спостерігається бум нейронних мереж. Щоб навчання складних нейронних мереж відбувалось не тижнями/днями/часами, а годинами/хвилинами/секундами, без апаратних прискорювачів не обійтись. Тільки спеціалізоване апаратне забезпечення виконає, наприклад, паралельне множення високої точності із одночасними транзакціями до пам'яті. У майбутньому спеціалізовані ASIC для штучного інтелекту будуть всюди, причому міститимуть на одному кристалі як звичні процесори, так і частково реконфігуровувані блоки [1].

Актуальність проектування апаратних прискорювачів FPGA зумовлена тим, що на відміну від CPU і GPU, мікросхему FPGA можна перепрограмувати під конкретну задачу. Також одна із особливостей FPGA – зменшення енерговитрат у розрахунку на одиницю обчислювальної потужності. Ще однією важливою особливістю FPGA є відносно велика ступінь інтеграції, реконфігурованість та можливість перепрограмування. У свою чергу, мікросхеми FPGA мають відносно доступну ціну, що робить їх ще популярнішими. На рисунку 1 можна побачити вигляд FPGA – прискорювача.



Рисунок 1 – Приклад апаратного прискорювача на FPGA.

В навчальній лабораторії університету доступна плата DE0 із встановленою на ній мікросхемою Cyclone III FPGA. Вона вміщує у себе 200 тис. логічних елементів, 4 Мб пам'яті, 396 множників 18x18, а також багату периферію (семисегментні індикатори, кнопки, ключі, VGA, PS/2, UART), що дозволяє спроектувати та відлагодити простий апаратний прискорювач, що може бути застосовано в процесі розробки нейронної мережі в рамках наукової роботи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Первый шаг к аппаратным ускорителям нейронных сетей для программистов лежит через изучение основ HDL, RTL и лаб на FPGA [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/349750/>

2. Тимошук П.В. Основи проектування нейронних мереж. Навчальний посібник. / П.В. Тимошук, М.В. Лобур // Львів: Видавництво Львівської політехніки, 2007. –328 с.

УДК 004.05

Padalko B.O.¹, Polska O.V.², Kudermetov R.K.³

¹ student of KNTz-615 gr. ZNTU

² senior teacher of ZNTU

³ PhD, assistant professor of ZNTU

STATE OF THE ART OF APPLICATION PERFORMANCE MANAGEMENT

In the systems development life cycle, the maintenance and operations phase is one of the most important for business and one of the longest. Depending on the source, it could be from 50% to 80% [1, 2, 3] of the overall timeline. It is a hard requirement to guarantee seamless application operation which implies detecting

anomalies rapidly, avoiding false-positive errors in feedback loops and proving a good overview of the system operation status. Having a monitoring aspect unaccounted dramatically increases deprivation chances and tends to higher outages number with irreparable damages outcome.

This document provides an overview of the state application performance management nowadays, modern problems it faces and the direction it is developing.

Application performance management (APM) is a set of software and hardware components to monitor and manage performance and availability of software applications. Modern systems based on the following older concepts:

- monitoring underlying infrastructure and resources, such as network status, bandwidth utilization, CPU usage, memory consumption, disk i/o, disks free space, etc.;
- analyzing application logs and logs of services that main application is depending on, such as data persistence systems (network file systems, databases, FTP servers), runtime caches, web servers, access control system (LDAP, SSH), but not limited to;
- tagging business transactions and collecting metrics on them;
- simulating request to an application or its components to check system status;
- managing change events such as application deployment, configuration changes, other maintenance and operations activities.

APM suites emerged as a natural response to software complexity raise in attempts to reduce operations cost and improve reliability. Compared to previous generation tools effectiveness, it was an outstanding shift forward.

The fundamental difference between APM suits and standalone tools that satisfy separate requirements of APM (which are also known as dimensions) is that APM as a methodology enables contextual analyze and background of an issue to provide a comprehensive overview of a problem or system state. Contrary, standalone instruments detect a particular problem in a chain of events at a specific application ecosystem without sufficient connection to the rest of them. The practical benefit of APM use is the reduction of the mean time to identify/detect (MTTI/MTTD) which has a direct positive impact on the mean time to resolve/recover/repair (MTTR) because of the synergy of integrated APM tools suite. The lack of interoperability [4] could make it difficult or expensive to accomplish the same with standalone tools.

Estimated that about 80% of the APM value in terms of application visibility for the business comes from Real User Monitoring (RUM) [5]. Despite many advantages, RUM has its drawbacks, most important are traffic and user interaction requirements which are barely available for non-production environments. The inconsistency of traffic volume and user behavior intricate systematic metrics

yielding with an acceptable credible interval and makes RUM to lack benchmarking capability. To eliminate these problems Synthetic Monitoring (SM or STM where T stands for Transactions) is introduced to emulate user transactions using scripted scenarios. Using real user behavior snapshots could countervail their predictability and variations limit which makes STM a cheap and affordable alternative to RUM. The best result achieved when both RUM and STM used alongside. Monitoring user transactions is the most valuable part of APM and it solves an end-user experience management problem.

Despite modern approaches, application and services log entries analysis is still the most widely used technique. It is also the oldest and cheapest approach. Logs have very poor initial visibility due to limited visualization capability without prior processing. With developing software complexity and data volume increase, they may not provide an immediate advantage at the time of an incident, though logging is a fundamental ground for historical analysis and tracing. Combined with other historical metrics further logs analytics becomes possible.

Infrastructure monitoring is often generating a large volume of metrics which with ongoing modern systems complexity increase makes it hard to analyze them without full system context. During past years relying just on infrastructure metrics has been proven to be not sufficient enough to build high-confidence automation.

In a modern complex application, performing root cause analysis is still pretty manual work even while using various APM tools at the same time. One of the possible ways to address this issue is to extend instruments to match a particular application needs and integrate better with each other. Another potential solution is to develop better classic and ML algorithms [4, 6].

With recent advances in machine learning (ML) and artificial intelligence (AI) studies utilizing APM suits enable event noise reduction, predictive event alerting, probable cause analysis, intelligent automation. This approach is known as AIOps - Artificial Intelligence for IT operations and is developing rapidly in recent years since it emerging. AIOps solves a wide range of immediate problems of immense application complexity and big data and enables further IT operations transformation. It addresses a set of applied problems, such as large logs and metrics volume processing, matching different tools output, detecting relations in tools output, alert conditions cluttering and enables building feedback loops with low false-positive errors.

We can also notice that APM vendors start offering more dimensions as a service for end users and rapidly adapt cloud-first monitoring integrating with cloud providers directly requiring little or no efforts. Consolidating more of these dimensions enables them to address interoperability problems and prepare metrics dataset for further ML analysis.

APM solved a wide range of problems, such as contextual monitoring and traceability which was very challenging to achieve with segregated tools. Further

AIOps development addresses imminent software complexity increase and brings big data management for IT operations, reduces MTTI and MTTR and optimize the costs of the maintenance and operations phase even more, compared to APM impact.

REFERENCES

1. Schumaker, Carl W., Life cycle cost management: The long term view / Carl W. Schumaker, Roland D. Kankey // IEEE Proceedings of the National Aerospace and Electronics Conference, 1989, – vol. 3, – pp. 1221-1225.
2. Zarnekow, R., Distribution of cost over the application lifecycle -A multi-case study / R. Zarnekow, W. Brenner // Proceedings of the 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy, ECIS 2005.
3. Koskinen, J. Software Maintenance Costs / J. Koskinen // 2015. Available at <https://wiki.uef.fi/display/tktWiki/Jussi+Koskinen>
4. Heger, C. Application performance management: State of the art and challenges for the future / C. Heger, A. Van Hoorn, M. Mann, D. Okanović // Proceedings of the 2017 ACM/SPEC International Conference on Performance Engineering ICPE 2017, 2017, – pp. 429-432.
5. Dragich, L. Prioritizing Gartner's APM Model. The APM Conceptual Framework / L. Dragich, – 2012. Available at <https://www.apmdigest.com/prioritizing-gartners-apm-model>
6. Ahmed T.M. Studying the effectiveness of Application Performance Management (APM) tools for detecting performance regressions for web applications: An experience report / T.M. Ahmed, C.-P. Bezemer, T.-H. Chen, A.E. Hassan, W. Shang // Proceedings - 13th Working Conference on Mining Software Repositories, MSR 2016, 2016 – pp.1-12.

УДК 004.932

Потапов С.О.¹, Касьян К.М.²

¹ студ. гр. КНТ-514м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ІНТЕР'ЄРУ ПО ФОТОГРАФІЇ З ПОДАЛЬШИМ ЇХ ТРАНСФОРМУВАННЯ В КРЕСЛЕННЯ ДЛЯ ПОБУДОВИ ІОТ СИСТЕМ

Інтернет речей (IoT) - це середовище, в якому до мережі підключено практично все, що може представляти хоча б найменший інтерес для того, хто користується «речами». Інтерес цей може виражатися в можливості

управляти «річчю» віддалено і дізнаватися її стан. Причому, робити це можна в будь-якому місці і в будь-який час. Особливе становище в концепції IoT займають і наші оселі. Будинок - це центр життя людини і цілком природно те, що йому потрібно постійне відчуття впевненості в тому, що там все добре і безпечно. Сучасні технології відіграють в забезпеченні такої впевненості дуже важливу роль. Раніше, до IoT, про таке можна було тільки мріяти [1].

Побудова IoT систем зазвичай розбивають на 6 фаз. Перші три фази - це дослідження, вони зосереджені на формуванні ідей і оцінці можливостей проекту за рішенням поставленої перед ним завдання. За великим рахунком, все це - підготовка до створення успішного комерційного продукту. На трьох фінальних етапах роботи переходять до використання серійних компонентів, наприклад - датчиків. Те ж саме стосується таких частин системи, як шлюзи для інтернету речей. Тут же проводиться остаточне доопрацювання документації, в ній фіксується підсумковий набір характеристик продукту [1].

При створенні складних, високонавантажених систем з розпізнавання об'єктів, може виникнути проблема швидкості розпізнавання даних. Тому актуальним є розробка поліпшеного методу розпізнавання для підвищення продуктивності при обробці великих масивів даних. Завдання розпізнавання часто зустрічаються в різних прикладних задачах [2].

Метод Адам - який допомагає в оптимізації розпізнавання, застряганням в локальному мінімумі або з вибором занадто великого (малого) кроку навчання.

Адам (Adaptive Moment Estimation) - це алгоритм оптимізації адаптивної швидкості навчання, розроблений спеціально для навчання глибоких нейронних мереж [3].

Реалізація цього методу використовується в системі за допомогою бібліотеки TensorFlow.

TensorFlow відкрита програмна бібліотека для машинного навчання, розроблена компанією Google. Навчається вирішувати завдання шляхом позитивного посилення і обробляє дані на різних рівнях (вузлах), що допомагає знаходити вірний результат. Дана бібліотека пропонує інтерфейси API для мов Python і C / C ++, що дозволяють підключатися до програми розробника [4].

Для того щоб прискорити побудову IoT систем розумного будинку, була розроблена дана система, яка шляхом розпізнавання об'єктів інтер'єру за допомогою нейронної мережі, що перетворює всі розпізнані об'єкти в координати, за якими в подальшому йде формування креслення.

Структура системи розпізнавання об'єктів інтер'єру по фотографії включає в себе такі компоненти:

— клієнт;

- мобільний додаток;
- веб-додаток;
- нейронна мережа;
- конструктор креслень;
- база даних.

Система складається з декількох сервісів і працює в двох видах клієнтів. Робота можлива як через web-додаток, так і через мобільний додаток. Перший сервіс - це нейронна мережа яка розпізнає об'єкти і формує координати, які передає в другий сервіс - конструктор креслень. У цьому сервісі за отриманими координатами будується креслення, яким у подальшому можна маніпулювати, тобто можна пересувати об'єкти, додавати нові, проводити кабельну систему або ж бездротову, додавати необхідні датчики. Також система пропонує вибір датчиків з популярних інтернет-магазинів, з інформацією про ціну, рейтингу та відгуками.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. IoT Reference Implementation: Smart Home | In-tel® Software [Electronic resource]. – Available at: <https://software.intel.com/en-us/articles/iot-reference-implementation-smart-home>
2. Kasian, K., Bratchykov, V., Shkarupylo, V. Development of modified method for text recognition in standardized picture // Eastern-European Journal of Enterprise Technologies – 2015. –No. 3/2 (75). – pp. 11–17.
3. Diederik, P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9
4. Dean, Jeff; Monga, Rajat; et al. TensorFlow: Large-scale machine learning on heterogeneous systems [Electronic resource]. – Available at: <https://cse.buffalo.edu/~chandola/teaching/mlseminardocs/TensorFlow.pdf>

УДК 004.6:004.312.26

Предко В.Ю.¹, Паромова Т.О.²

¹ студ. гр. КНТ-516 ЗНТУ

² старш. викл. ЗНТУ

НАЛАГОДЖЕННЯ ТА УСУНЕННЯ ПРОБЕМИ НЕКОРЕКТНОГО ВІДОБРАЖЕННЯ ДАНИХ В СУБД

В даний час людство накопичило великий обсяг інформації в електронному вигляді. Всі ці документи мають різні формати, тобто різні форми подання інформації. Для обробки цього масиву інформації використовуються різні СУБД.

Кодування - це спосіб передачі інформації в комп'ютері, в якому описується зв'язок певного символу і числа йому відповідного, візуально представляє собою таблицю. Тобто в ній показано, як і будь-що будуть перетворені символи для подальшої взаємодії з системою. Значення даних, що базуються на заданому наборі символів, інтерпретуються з урахуванням заданої кодування, що визначає допустимі використовувані символи.

Історично першою була кодування ASCII, яка містила великі та малі літери англійського алфавіту, 10 цифр, пробіл і елементарні арифметичні дії. Так з'явилася ASCII, яка має 128 позицій коду (7 біт), виділених графічним символам і керуючим символам (керуючі коди), які розміщуються в перших 32-х кодах базової таблиці, починаючи з нульового [1, 2].

З проблемою зіткнулися пізніше: коли комп'ютери поширилися на НЕ англомовні країни, програмістам довелося додавати восьмий біт, який розширював таблицю до 256 місць без втрати сумісності і латиниці [2]. Труднощі полягали в тому, що для кожного готельного національної мови повинна бути створена своя таблиця, що ускладнювало роботу розробникам операційних систем і програмного забезпечення. В результаті виникало безліч інших типів кодувань, найпопулярніші з яких: Windows-1251 (cp1251), CP866 і KOI8-R [3].

Причини, за якими виникають проблеми, будуть розглянуті на прикладі популярної СУБД MySQL. Найбільш частою проблемою для російськомовного сегмента є перенесення кириличних символів, що в свою чергу веде до появи нечитаною інформації. До версії 4.1 в MySQL не передбачалася можливість явно оголошувати кодування. Строкові константи і значення стовпців інтерпретувалися з урахуванням кодування, прийнятою за замовчуванням при створенні сервера (зазвичай це latin1). Вбудовані значення можна було скасувати при виконанні параметром - default-character-set. Однак це може обмежити свободу дій з іншими таблицями, в яких стовпці розраховані на роботу з іншими кодуваннями. База даних, яка була налаштована на роботу в одному кодуванні, при зміні кодування сервера може так само привести до ряду проблем з індексами вже створених таблиць і завантаженими в них даними. Це виникає тому, що значення індексів зберігаються в порядку сортування з відповідною кодуванням, яка була встановлена в момент створення записів. Часто при переході з одного кодування на іншу, цієї проблеми може і не виникнути, тому що деякі кодування мають однаковий порядок сортування [1].

Для роботи з базою даних MySQL використовується клієнт MySQL Command Line Client, в якому проблеми з кодуванням виникають рідше, проте якщо використовувати командний рядок Windows, то труднощів може виникнути більше, так як її кодування CP-866 (DOS) і вона не буде сприймати іншу.

Методик вирішення даної проблеми може бути декілька:

- заархівувати таблицю за допомогою програми `mysqldump`, видалити, і відновити знову з архіву. Це призводить до перебудови індексів в процесі перевантаження таблиці;

- видалення індексів і відтворення їх за допомогою команд.

Не виключається зміна кодувань в самому середовищі MySQL і її файли конфігурації `my.ini` (в деяких версіях - `my.cnf`), прописавши в першому випадку в командному рядку команду «`SET NAMES <кодування>`», а в другому дописати в файл рядок «`default-character-set = <кодування>` » [4].

У майбутньому планується тестування швидкості виконання запитів у середовищі MySQL за допомогою прикладних мов програмування та вплив кодування на відображення інформації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дюбуа П. MySQL: Полное и исчерпывающее руководство по применению и администрированию баз данных MySQL 4, а также программированию приложений [Текст] / П. Дюбуа; пер. с англ. и науч. ред. Н.В. Воронина. – М. : Вильямс, 2017. – 1056 с.

2. Альошин А.. Кодування символів. Види кодувань. Таблиці ASCII. UNICODE [Електронний ресурс]. - Режим доступу: <http://vestikinc.narod.ru/AB/coding.htm>

3. Пушкарев П. Работа MySQL со строками. [Электронный ресурс]. - Режим доступа: <https://sqlinfo.ru/articles/info/2.html>

4. Гапон В. Как изменить кодировку MySQL базы данных и её таблиц [Электронный ресурс]. - Режим доступа: <https://ixnfo.com/kak-izmenit-kodirovku-mysql-bazyi-dannyyh-i-eyo-tablits.html>

УДК 621.38

Тягунова М.Ю.¹, Семенова К.В.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-514м ЗНТУ

МАТЕМАТИЧНА МОДЕЛЬ ОПТИМІЗАЦІЇ СИСТЕМИ ЗБОРУ ПРЕДМЕТІВ У ПРИМІЩЕННІ

В сучасному світі для вирішення рутинних завдань у промисловій сфері та у домашньому вжитку все частіше використовують роботів-помічників, оснащених великою кількістю датчиків [1-3]. Однією з таких систем є розроблена система для автоматичного збору предметів.

З метою оптимізації функціонування системи збору малогабаритних предметів пропонується застосувати метод, який мінімізує витрачання енергії за рахунок скорочення часу виконання завдання, а також зменшення числа підключених пристроїв.

Мінімальний час виконання завдання досягається при проходженні роботом шляху за найкоротшою траєкторією. Вважаючи, що система здійснює рівномірний рух, а час захвату предмета не залежить від шляху переміщення робота і є незмінним, визначення відстані до перешкоди (предмету) проводиться згідно наступного виразу:

$$S = v \cdot t, \quad (1)$$

де t – час, за який здійснюється переміщення при заданій швидкості v .

Звідси швидкість руху робота v є відношенням пройденого шляху до часу. Енергія руху E визначається як половина добутку маси робота на квадрат його швидкості. Враховуючи вищезазначене, отримаємо, що

$$E = \frac{m \left(\frac{S}{t} \right)^2}{2} = \frac{m S^2}{2 t^2}. \quad (2)$$

Пройдений шлях є прямо пропорційним величині затраченої енергії. А це доводить, що чим менше шлях долає робот, тим менше енергії буде витрачено системою.

Якщо система має N підключених пристроїв, які живляться від одного джерела, то енергія, яка витратиться роботом за весь пройдений шлях, визначатиметься як сума енергій E_i кожного i -го пристрою:

$$E_{\text{пр}} = \sum_1^N E_i. \quad (3)$$

Оскільки значення E_i не може бути від'ємним, то виходячи з математичного закону адитивності, чим менше значення енергії E_i i -тих пристроїв, тим менше значення енергії $E_{\text{пр}}$, яка витратиметься роботом під час проходження повної траєкторії шляху. Тобто, чим менше підключених до системи пристроїв, тим менше енергії буде витрачено роботом за певний час роботи.

Розглянута математична модель дозволяє довести, що даний метод мінімізації енергоспоживання робить систему збору предметів у приміщенні більш ефективною у використанні, ніж аналогічні роботи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шваб К. Четвертая промышленная революция / К. Шваб. — М. : «Эксмо», 2016. — 137 с.
2. Форд М. Роботы наступают: Развитие технологий и будущее без работы / М. Форд ; пер. с англ. — М. : Альпина нон-фикшн, 2016. — 430 с.
3. Бостром Н. Искусственный интеллект. Этапы. Угрозы. Стратегии / Н. Бостром ; пер. с англ. С. Филина. — М. : Манн, Иванов и Фербер, 2016. — 329 с.

УДК 004.932

Касьян К.М.¹, Сичов В.В.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-514м ЗНТУ

АВТОМАТИЗАЦІЯ ТЕПЛИЦІ З ДИСТАНЦІЙНИМ КЕРУВАННЯМ НА ОСНОВІ ІОТ СИСТЕМ

Інтернет речей (Internet of Things – IoT) перетворює звичні для нас речі у нові пристрої, створюючи як розумні годинники, так і розумні міста. Він під'єднує далекі від Інтернету засоби до мережі та надає їм нові функції. Це концепція комунікації об'єктів (“речей”), які використовують технології для взаємодії між собою та з навколишнім середовищем. Також ця концепція передбачає виконання пристроями певних дій без втручання людини.

Теплиця – це хобі або аграрна промисловість і цілком природно те, що потрібне постійне відчуття впевненості в тому, що там все добре і безпечно [1]. Сучасні технології відіграють в забезпеченні такої впевненості дуже важливу роль. Раніше, до IoT, про таке можна було тільки мріяти.

Розробка IoT системи розділяється на 2 етапи. Перший - це дослідження, вони зосереджені на формуванні ідей і оцінці можливостей проекту опираючись на поставлене перед ним завдання. За великим рахунком, все це - підготовка до створення успішного проекту. Наступний крок – це перехід до використання серійних компонентів, наприклад - датчиків. Те ж саме стосується таких системи, як шлюзи для інтернету речей. У цей крок також входить облаштування самої теплиці.

Основними напрямками буде встановлення та налагодження керування для датчиків:

- температури за для обігріву (грунту/повітря);
- температури за для своєчасної вентиляції повітря;
- освітлення (досвітлення);
- виміру вологості ґрунту для автоматичного поливу;

Платформа Blynk [2]- розроблена для Інтернету речей. Вона може контролювати апаратне забезпечення віддалено, може відображати дані датчиків, може зберігати дані, візуалізувати їх і робити багато інших цікавих речей.

У платформі є три основні компоненти:

- Blynk App - дозволяє створювати дивовижні інтерфейси для ваших проєктів за допомогою різних віджетів;

- Blynk Server - відповідальний за всі комунікації між смартфоном і апаратним забезпеченням. Можливо скористатись Хмарою Blynk або запустити приватний Blynk server локально. Це відкрите джерело, воно може легко обробляти тисячі пристроїв і навіть може бути запущений на Raspberry Pi;

- Blynk Libraries - для всіх популярних апаратних платформ - дозволити спілкування з сервером і обробляти всі вхідні та вихідні команди

Протокол MQTT (Message Queuing Telemetry Transport) [3]- це асинхронний протокол передачі телеметричних повідомлень за принципом видавець / передплатник з використанням брокера. Особливістю протоколу є його компактність, тому що величина заголовка пакета даних становить всього лише кілька байт. Він працює поверх стандартного протоколу TCP і, для забезпечення безпеки, передбачає використання механізмів аутентифікації на рівні «username / password» і використання шифрування транспортного протоколу TLS / SSL. На даний час, протокол MQTT став стандартом де-факто для побудови рішень Інтернету речей.

В ході розробки теплиця буде облаштована різними датчиками для стеження за навколишнім середовищем. Відштовхуючись від отриманих даних з цих датчиків і будуть виконуватись певні дії, такі як: увімкнення поливу, світла, обігріву, або відчинення/зачинення вікон. Все це буде виконано з використанням протоколу MQTT для передачі даних з пристроїв на сервер і навпаки. Також буде використана платформа Blynk для створення зручного інтерфейсу керування системою, та отримання інформації від датчиків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Что такое умная теплица и как сделать автоматическое управление своими руками [Электронный ресурс]. – Режим доступа: <http://teplicno.ru/obustr/umnaya-teplica.html>

2. We make Internet of Things simple for you [Electronic resource]. – Available at: <https://blynk.io>

3. Платформа ARM и брокер MQTT, как современная основа решений для Интернета вещей [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/unet/blog/407867>

VPN ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ НА РІВНІ МЕРЕЖІ

У інтернеті досі мало матеріалу про таку стару і просту, але зручну, безпечну і особливо актуальну в зв'язку з розвитком Інтернету речей технологію, як мобільна VPN (віртуальна приватна мережа). У цій статті роздивимось, як і навіщо можна налаштувати доступ у приватну мережу будь-якого пристрою з сім-картою без необхідності налаштування на ньому спеціалізованого ПЗ.

VPN як технологія використовується для вирішення найрізноманітніших мережевих завдань, об'єднаних загальною ознакою - ізольованою передачею даних між двома пристроями крізь велику кількість проміжних вузлів. На базі цього вже будуються більш складні рішення і вирішуються ті самі різні завдання. У звичному всім випадку для побудови VPN використовується мережа оператора фіксованого зв'язку або безліч різних мережевих протоколів (GRE, IPSec, L2TP і інші) і програмні продукти, що працюють з ними (Cisco AnyConnect, OpenVPN, TOR), але їх використання на конкретному крайовому пристрої відразу ж висуває до нього ряд вимог, невиконання яких веде до певних обмежень.

Перше серйозне обмеження - пристрій повинен на апаратному та програмному рівнях вміти працювати хоча б з одним з цих протоколів. Найчастіше це визначається програмним забезпеченням, яке легко знайти для ноутбука або смартфона, але існують випадки, коли завдання стоїть перед дуже простим з апаратної точки зору пристроєм, або у його ПЗ є обмеження.

Інше важливе обмеження - необхідність налаштування. Воно працює як для «дурних» пристроїв з першого пункту, так і для класичних смартфонів і комп'ютерів, яким попереднє обмеження невідомо. І якщо з першими все відносно просто і впирається в кількість часу, витраченого на налаштування, то з другими є варіанти. Найчастіше організації використовують VPN з метою безпеки, щоб уберегти службовий термінал від виходу в публічну мережу без належного корпоративного захисту або від передачі службових даних по публічному каналі. Кінцеві ж користувачі можуть з якихось своїх причин відключити або забути включити VPN, в результаті чого багато систем безпеки компанії можуть опинитися «за бортом».

Обидва ці обмеження легко знімаються, якщо доступ до VPN забезпечується на мережевому рівні. У випадку з мобільним зв'язком це може

бути реалізовано за допомогою «мобільного VPN». Пристрій будь-якої складності здатний передавати дані, буде передавати їх в правильну мережу. Неважливо, які налаштування зроблені на пристрої, при правильно конфігурованій мережі він в будь-якому випадку буде передавати їх туди, куди потрібно, і нікуди більше.

Пристрій буде отримувати адресу з внутрішньої мережі, що налаштовується віддалено, і доступ до нього буде можливо отримати тільки зсередини цієї мережі (або фізично). Для певного класу пристроїв це дуже важливо.

Здавалося б, VPN - класична послуга всіх операторів зв'язку для B2B сегмента, і навіщо в такому разі загострювати на цьому увагу? Вся справа в тому, як влаштована мережа передачі даних для пристроїв, що підключаються через GPRS, HSPA, LTE або іншу технологію мобільного зв'язку. Тут немає звичних всім мережевим адміністраторам vlan, немає свічів, немає навіть маршрутизаторів в звичному їх значенні. Зате є мережа радіодоступу (RAN) і пакетне ядро (PS Core).

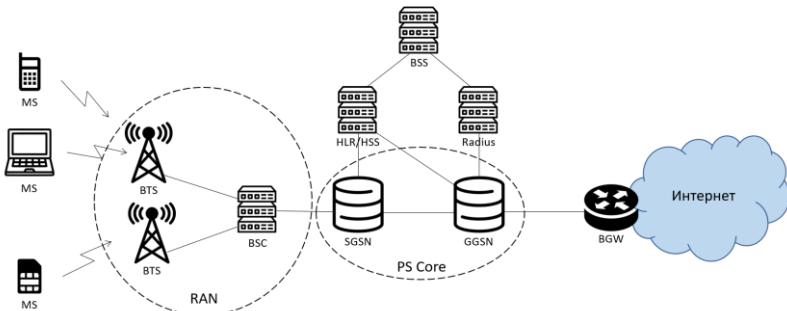


Рисунок 1 – Спрощена схема пакетної мережі оператора мобільного зв'язку.

У загальному випадку, кожен пристрій з сім-картою, зареєстрований в пакетній мережі (що пройшла процедуру GPRS attach або аналогічну), перед тим як почати передавати дані куди-небудь має ініціювати створення сесії передачі даних (PDP context) на маршрутизаторі ядра пакетної мережі, GGSN. Важливо: при ініціації сесії в запиті до GGSN в числі інших значаться параметри наступні параметри: APN, логін і пароль. APN (точка доступу) - дуже важлива сутність в логіці роботи GGSN: в залежності від того, з якою APN ініційована сесія, GGSN діє по-різному. В результаті успішної обробки запиту користувача GGSN повинен активувати у себе сесію передачі даних і повідомити пристрою її параметри, зокрема, видану пристрою IP адресу і адреси DNS. Тут є ряд важливих дуже важливих особливостей:

У запиті на ініціацію сесії пристрій ніколи не запитує, яку IP адресу він хотів б отримати;

Крім полів, що задаються в налаштуваннях пристрою «APN», «логін» і «пароль», в запиті до GGSN передається також номер телефону (MSISDN) абонента (тут і далі «абонент» - це кінцевий користувач, один пристрій з сім-картою, а «клієнт» - організація-замовник сервісу, до складу якої входять абоненти);

При активації сесії GGSN створює в своїй таблиці маршрутизації запис про нову IP адресу. Всі абоненти на GGSN позначаються записами в таблиці маршрутизації з префіксом / 32, тобто 1 абонент - 1 запис в таблиці. GGSN - це дуже продуктивний маршрутизатор;

Мережа оператора може на різних етапах (і на SGSN, і на GGSN) з різних причин змінювати поле APN в запиті на ініціацію сесії. Це дозволяє в деяких випадках зменшити, а в деяких взагалі виключити налаштування параметрів мережі на пристроях з сім-картою.

За перших трьох пунктів відразу виникає питання: що за IP адреса видається абоненту? Це визначається налаштуваннями тієї APN, з якої прийшов запит на активацію сесії. Порядку 99% користувачів передачі даних в мобільних мережах використовують звичайний доступ в Інтернет. Це відомі всім точки доступу internet.mts.ua, internet.lifacell.ua і так далі. У випадку з доступом в Інтернет GGSN видає адреси за принципом класичного DHCP із прописаних в налаштуваннях сірих підмереж. При виході в публічну мережу вони закриваються класичним NAT'ом (а точніше тієї його версією, яка PAT).

Але GGSN здатний на більше. Для вибору IP адреси він може зробити AAA запит до сервера авторизацій (Radius, наприклад). Така логіка налаштовується для окремих APN в залежності від їх призначення. Найпростіший випадок - послуга з надання постійної публічної IP адреси. Такі адреси, як правило, закріплюються за абонентами у біллінзі (BSS) оператора, і в залежності від IT архітектури потрапляють в ту чи іншу базу даних, до якої і звертається із запитом GGSN. Завдяки тому, що йому відомий MSISDN (номер телефону) абонента, який і буде міститися в запиті, така БД буде досить простою і може містити лише зв'язку номера та адреси. Додатково, в разі, якщо клієнт планує використовувати одну сім-карту для підключення декількох пристроїв (якщо сім-карта знаходиться в WiFi-роутері віддаленого офісу, наприклад), ця таблиця може містити також так званий «framed route» - префікс мережі, що знаходиться « за »сім-картою, який буде анонсований всім пристроям в мережі по протоколам динамічної маршрутизації.

Крім видачі адрес потрібно ще доставити трафік абонентів до клієнтських мереж, кожного до своєї. Тут все працює вже куди більш традиційно. На GGSN трафік спеціалізованих під роботу з VPN APN

маршрутизується на окремий маршрутизатор мережі оператора (він може називатися по-різному, іноді - VPN роутер), який в свою чергу виконує функцію класичного PE в схемі L3VPN. Він додає необхідні мітки, заголовки і це все і відправляє весь цей потік трафіку через маршрутизатори транспортної мережі в попередньо налаштовані стики або тунелі до мережі клієнта.

З урахуванням всіх цих деталей, способів організації мобільного VPN може бути кілька, і вони будуть відрізнятися один від одного комбінацією таких особливостей:

IP адреси, як вже було описано, можуть видаватися динамічно (кожен раз різна адреса з заданою підмережі) і статично (кожен раз одна і та ж адреса у конкретного абонента), що визначається і / або налаштуваннями APN, і / або налаштуваннями Radius-сервера;

IP адреси можуть видаватися Radius-сервером під управлінням оператора або під керуванням клієнта;

Пристрої, підключені до мобільного VPN, можуть взаємодіяти або тільки один з одним, або мати доступ до звичайної L3VPN мережі клієнта через прямий стик (VPN порт) з оператором або через тунелювання поверх мережі Інтернет;

У деяких випадках використання логіна і пароля для успішної активації сесії може бути обов'язково, а іноді не потрібно навіть заповнювати поле «APN».

Таких комбінацій з різними типами тунелювання, балансування трафіку між каналами доступу до «основного» VPN клієнта і принципом видачі адрес налічується кілька десятків. Для більшості випадків загальна схема виглядає наступним чином.

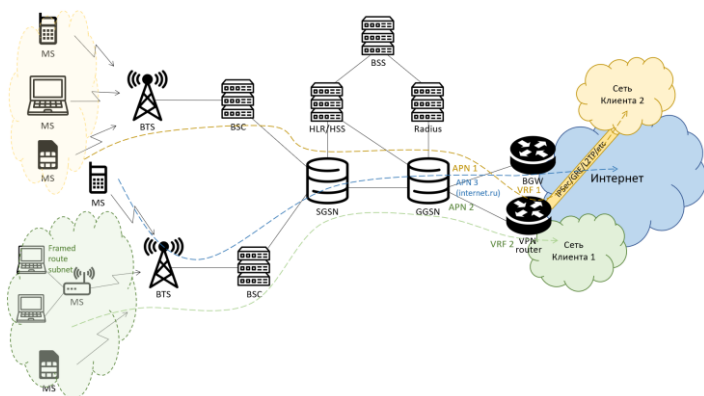


Рисунок 2 – Загальна схема організації мобільного VPN.

У підсумку після досить швидкого процесу реєстрації в мережі і отримання IP адреси пристрій отримує доступ в мережу клієнта, а мережа клієнта отримує доступ до пристрою. При цьому абонент знаходиться в ізоляції від усіх інших абонентів оператора, що не мають відношення до конкретного клієнта, йому не потрібно ніяких додаткових налаштувань, і весь трафік безальтернативно направляється в мережу клієнта, де обробляється вже відповідно до внутрішніх політик клієнта.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Брайан Клифтон. Google Analytics. Профессиональный анализ посещаемости веб-сайтов / Клифтон Брайан – М.: Вильямс, 2009. – 400 с.

УДК 004.65

Тімохін А.А.¹, Паромова Т.О.²

¹ студент гр. КНТ-516 ЗНТУ

² старш.викл. ЗНТУ

АНАЛІЗ РЕЙТИНГУ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

На сьогоднішній день величезна кількість інформаційних систем використовують системи управління базами даних (далі СУБД), які надають надійний і безпечний спосіб зберігання інформації, обробки і аналізу важливих даних організацій та підприємств. Правильний вибір СУБД дозволяє уникнути помилок при проектуванні системи, спрощує її експлуатацію і забезпечує можливість подальшого розвитку системи.

Рейтинг, що складається австрійською консалтинговою компанією Solid IT [1], на сьогоднішній день охоплює більш ніж 340 СУБД та оновлюється щомісяця (останнє оновлення рейтингу відбувалося в лютому 2019 році, що на момент написання статті є найбільш актуальним) і присуджує щорічну нагороду системі, яка придбала найбільшу популярність протягом року.

Перші п'ять позицій рейтингу займають такі СУБД: Oracle (31.65% від всієї частки ринку СУБД), MySQL (20.25%), Microsoft SQL Server (15.11%), PostgreSQL (4.16%) і MongoDB (3.94%).

Oracle Database - це об'єктно-реляційна СУБД перш за все орієнтована на потреби великого бізнесу, корпорацій, банківських систем, наукових центрів та інших сфер для зберігання і обробки великих обсягів даних [2]. Ця СУБД підтримує велику кількість користувачів, що одночасно працюють з різними додатками, забезпечує високу ступінь готовності, швидкість та цілісність даних. Також СУБД Oracle має відмінну переносимість як бази

даних, так і додатків між різними операційними системами. Дана СУБД є однією з найдорожчих на ринку.

СУБД MySQL на відміну від Oracle є Open Source програмним забезпеченням з відкритим вихідним кодом і розповсюджується під ліцензією GNU GPL (Відкрита ліцензійна угода) основною метою якої є надання прав копіювання, модифікації і поширення (в тому числі комерційних програм) користувачеві, а також гарантування користувачам дочірніх програм отримання тих же прав що і користувачам батьківських [3]. Це є однією з головних причин вибору розробників продукти яких, орієнтовані на потреби малого та середнього бізнесу для створення інтернет магазину або ж, наприклад, блогу. Не можна не відзначити той факт, що 30% сайтів світу використовують CMS WordPress (Content Management System - система управління контентом) [4] яка в свою чергу використовує MySQL в якості СУБД, також є безкоштовною. MySQL це реляційна СУБД яка володіє відмінною переносимістю та високою продуктивністю. Основними мінусами СУБД MySQL є погана масштабованість та підтримка великих обсягів даних [5].

Microsoft SQL Server - продукт компанії Microsoft, реляційна СУБД, орієнтована на ті ж сфери застосування, що і СУБД Oracle. Однією з відмінностей від інших систем управління є використання власної різновиди мови запитів SQL - T-SQL (Transact-SQL), який має свої особливості. Головною перевагою Microsoft SQL Server є простота установки і особливо оновлення, які встановлюються СУБД автоматично [6]. Це дозволяє значно знизити витрати на її обслуговування. Вона поставляється з інструментами, які допоможуть заощадити час розробників і адміністраторів БД в розробці і усунення несправностей. До недавнього часу головним недоліком SQL Server була відсутність переносимості БД на іншу ОС. У листопаді 2016 р Microsoft опублікувала версію яка працює на ОС Linux, але це не дуже покращило її становище на ринку, адже інші корпоративні СУБД мають кращу переносимість. СУБД не потребує систем з дорогим апаратним забезпеченням, має досить високу ціну, порівнянну з СУБД від компанії Oracle.

PostgreSQL є об'єктно-реляційною СУБД корпоративного класу має відкриті вихідні коди. Вона поширюється за ліцензією BSD (Berkeley Software Distribution license - Програмна ліцензія університету Берклі), яка допускає вбудовування в комерційні продукти, при цьому не відкриваючи вихідні коди отриманого додатку [7]. PostgreSQL підтримує розширені функції, які доступні тільки в дорогих комерційних СУБД. Вона має високу відмовостійкість та надійність. СУБД можна розгорнути на всіх популярних ОС по типу UNIX, Windows, macOS тощо. Здавалося б, PostgreSQL ідеальний варіант для великих корпорацій, але і вона має ряд мінусів.

Наприклад, в разі, коли необхідна висока швидкість, в особливості читання, ця СУБД буде програвати конкурентам виробникам комерційного ПЗ [8]. Коли не потрібна абсолютна цілісність даних або складні конструкції, вона може бути занадто складною для простих налаштувань. Мінусом для деяких користувачів може виявитися і те, що PostgreSQL не належить жодної з компаній, а це значить відповідальність за її функціональність ніхто не несе.

MongoDB - документно-орієнтована СУБД, призначена для зберігання ієрархічних структур даних (NoSQL). Розробники, перш ніж зробити вибір в сторону MongoDB, повинні розуміти вимоги до розроблюваного додатку. NoSQL реалізація містить в собі ряд переваг та недоліків перед реляційними базами даних. СУБД такого типу показують кращі результати в масштабованості бази [9]. Також така архітектура дозволяє зберігати великі обсяги даних при збереженні відмінних показників швидкості, і не вимагає великого досвіду в адмініструванні, так як «з коробки» вже має функції автоматичного оновлення СУБД та відновлення даних. MongoDB підтримується на всіх сучасних операційних системах. Основними недоліками NoSQL баз, зокрема MongoDB, є надання менших аналітичних функцій, ніж реляційні СУБД через те що орієнтовані на потреби веб-додатків в більшості випадків та не мають підтримки транзакцій. MongoDB є продуктом з відкритим вихідним кодом. Її можна розгортати практично на будь-якій машині, вона не вимоглива до апаратного забезпечення.

Розглянуті СУБД є найбільш затребуваними на біржі праці ІТ та займають близько 75% ринку СУБД отже забезпечують більшість потреб бізнесу, який потребує СУБД. На даний момент протистояння між СУБД немає. Замість того щоб шукати альтернативу тій чи іншій системі управління варто говорити про їх спільне використання для вирішення завдань, на яких той чи інший підхід показує себе найкраще. До того ж, все сильніше спостерігається інтеграція технологій різних компаній однієї в іншу. Займаючись вибором СУБД, можна вибрати одну технологію, а при зміні вимог до інформаційної системи, можна переключитися на іншу. Однак, розумне планування дозволить заощадити як кошти замовника, так і час розробників.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Knowledge Base of Relational and NoSQL Database Management Systems [Electronic resource]. - Available at: <https://db-engines.com/en/ranking>
2. Introduction to the Oracle Database [Electronic resource]. - Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
3. Использование ПО MySQL под коммерческой лицензией [Електронний ресурс]. - Режим доступа:

http://www.mysql.ru/docs/man/Using_the_MySQL_software_under_a_commercial_license.html

4. Популярность WordPress [Електронний ресурс]. – Режим доступа: <https://www.segodnya.ua/lifestyle/science/bolee-30-vseh-saytov-v-mire-teper-rabotayut-na-wordpress-1119881.html>

5. The advantages and disadvantages of MySQL [Electronic resource]. - Available at: <http://makble.com/the-advantages-and-disadvantages-of-mysql>

6. MS SQL Server history and advantages [Electronic resource]. - Available at: <https://bytescout.com/blog/2014/09/ms-sql-server-history-and-advantages.html>

7. Лицензия BSD [Електронний ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Лицензия_BSD

8. What is PostgreSQL? Introduction, History, Features, Advantages [Electronic resource]. - Available at: <https://www.guru99.com/introduction-postgresql.html>.

9. Pros and Cons of NoSQL [Electronic resource]. - Available at: <https://greengarageblog.org/7-pros-and-cons-of-nosql>.

УДК 004

Тіменко А.В.¹, Романюта М.М.²

¹ асист. ЗНТУ

² студ. гр. КНТ-514м ЗНТУ

З ЧОГО СКЛАДАЄТЬСЯ ІОТ

Інтернет речей (англ. Internet of Things, IoT) - концепція обчислювальної мережі фізичних предметів («речей»), оснащених вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем, яка розглядає організацію таких мереж як явище, здатне перебудувати економічні та суспільні процеси, виключає з частини дій і операцій необхідність участі людини.

У даній статті розглянуто те, за допомогою яких саме "речей" може бути реалізована в нашому світі ця ідея і те, якими способами вони можуть взаємодіяти один з одним або з зовнішнім середовищем.

Базові елементи діляться на кілька типів: сенсори, актуатори і гейти.

Сенсори і призначення цього типу елементів нічим не відрізняється від стандартних: різноманітні термометри, мікрофони, камери і десятки інших, менш поширених пристроїв. Деякі з них можна побачити на зображенні Sensors Starter Kit для Arduino.

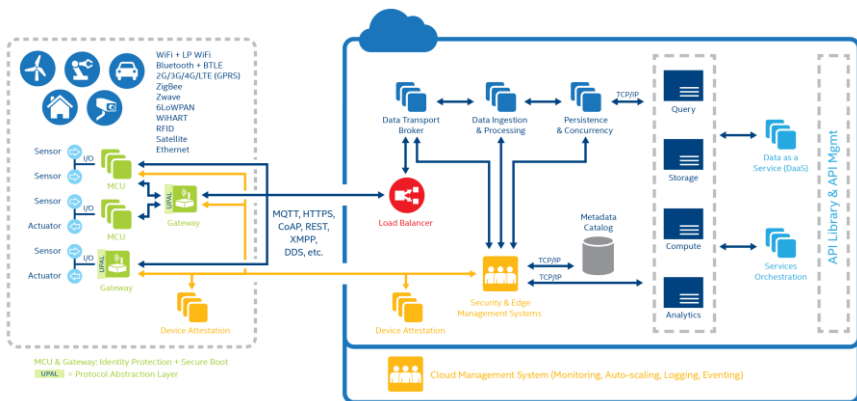


Рисунок 1 – Інтернет речей.

Даний тип елементів призначається для того, щоб впливати на навколишнє середовище, або на певний об'єкт в ньому. Цю роль можуть виконувати найрізноманітніші пристрої: від сервоприводів і динаміків до замків (звичайно, електронних) з освітлювальними приладами.

Ці пристрої, на які зазвичай покладають логіку поверхневого аналізу інформації, що надходить від підключених до них сенсорів. У певних ситуаціях, аналіз даних може вимагати малої кількості обчислювальних ресурсів, так що гейти цілком здатні приймати деякі рішення самостійно. Приймаючи такі рішення, вони відправляють певні команди управління на актуатори, які, в свою чергу, виконують вже свої функції. Якщо ж обробка інформації вимагає великих витрат, або ця інформація підлягає збору, гейти відправляють її на сервери, де з нею і проводиться подальша робота. Цілком собі ймовірно використання в ролі гейтів мікрокомп'ютерів або мікропроцесорів.

Для того, щоб побудувати моніторингову систему, досить буде використання лише сенсорів і деякого сервера, який буде виступати в ролі гейта. Наприклад, завдяки сенсору руху і умовній Raspberry PI, можна без особливих зусиль організувати облік кількості людей, що проходять через якусь прохідну.

Додавши в раніше сконструйовану модель актуатор в особі динаміка, можна домогтися того, щоб прохід кожного n-ного робітника був супроводжений фанфарами.

Так, ускладнювати конструкцію подібної комірки можна досить довго. Однак в певний момент неминуче з'явиться необхідність в довгостроковому зберіганні зібраної статистики, її аналізі, візуалізації та інше. Тут знадобляться вже повноцінні сервери, яким можна буде делегувати дані

обов'язках. Такі сервери в сукупності утворюють хмари, до яких і підключаються гейти.



Розглянемо те, якими засобами ці пристрої один з одним взаємодіють. Як видно на першому рисунку, є 2 умовні групи - хмара та периферія.

Осередки, що складаються з перерахованих вище типів пристроїв, як можна помітити, знаходяться в периферії і для комунікації використовують спеціальні протоколи взаємодії. Найбільше поширені LoRa і ZigBee. Обидві ці мережі є дуже повільними в порівнянні, наприклад, з 4G або навіть з 3G, проте мають і свої переваги.

Однією з головних є їх енергоефективність. Справа в тому, що ідея інтернету речей полягає в створенні середовища пристроїв, що комунікують між собою без участі людини. Варто зауважити, що в деяких випадках повністю уникнути втручання людини уникнути не вдасться. Наприклад, в системі підрахунку кількості людей, що пройшли, є сенсор руху. Йому, як і будь-якому іншому електричному пристрою, необхідне живлення. Проводити дроти з живленням до кожного такого сенсора (якщо їх більше 5 і вони сильно розкидані в просторі) здається не найкращою ідеєю. Відповідно, працювати вони будуть від батарейок або акумуляторів. Якщо споживання заряду буде надмірним, елементи живлення їм потрібно буде міняти досить часто. А це призведе до того, від чого прагне піти інтернет речей - потрібно буде комусь замінювати ці батарейки. А ось якщо сенсори будуть енергоефективні, то досить буде просто вставити батарейку і забути про це на рік, два, п'ять.

Ще однією перевагою цих мереж є висока стійкість. Кожен біт інформації в цих мережах відправляється окремим радіосигналом, тому його досить просто виділити на тлі ефірного шуму.

Таблиця 1 – Порівняння LoRa і ZigBee

Основні характеристики, що порівнюються		
Топологія	Зірка	прості і mesh
Частотний діапазон (залежить від країни)	2,4 ГГц, 868/915 МГц, 433 МГц, 169 МГц	2.4 ГГц, 915 МГц, 868 МГц
Ноди мережі	Базова станція Хост	Роутер Координатор (один із роутерів) Хост
Дальність на відкритому просторі	10 – 15 км	~ 500 м (залежить від потужності передавача)
Швидкість	0.3 - 50 кбіт/с	5 - 250 кбіт/с

А ось між периферією і хмарою, а так само і всередині хмари, використовуються, зазвичай, знайомі і звичні всім wi-fi з ethernet, стільникові та супутникові мережі і т. д.

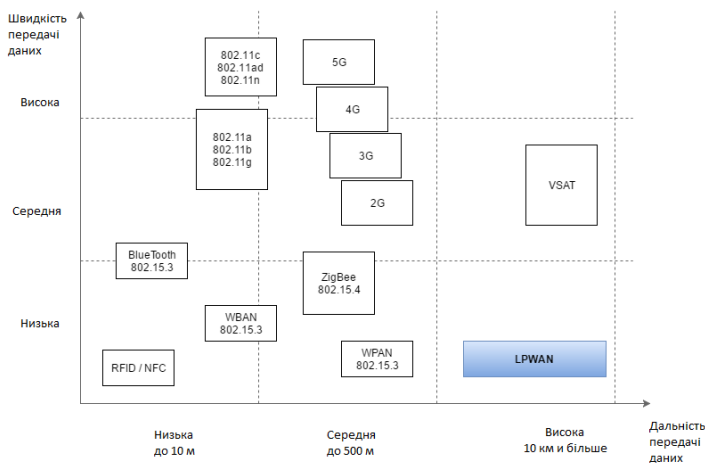


Рисунок 2 – Порівняння різних видів мереж на основі швидкості і дальності

Розглянувши влаштування мереж інтернету речей, можна точно сказати, що в плані апаратної частини немає нічого загадкового і складного. Зробити просту IoT-мережу може будь-хто, здатний купити досить дешеві на сьогоднішній день компоненти і написати код з пари рядків. Однак для того, щоб розробити і втілити в життя серйозні проекти як, наприклад, реалізацію концепції розумного будинку або навіть розумного міста, потрібно докласти величезну кількість зусиль. Адже для того, щоб всі ці пристрої працювали між собою потрібна платформа, здатна контролювати всі процеси, що протікають.

Так само не варто забувати, що в хмарах інтернету речей можуть використовуватися і інші технології, які допомагають розкрити його потенціал в більшій мірі. Такими можуть виступати і BigData, і BlockChain, і нейромережі з машинним навчанням. Але ж кожна з останніх перерахованих технологій являє собою окрему велику область комп'ютерних наук.

УДК 004.42

Точилін С.Д.¹, Демченко М.М.²

¹ канд. фіз.-мат. наук, доц. ЗНТУ

² студ. гр. КНТ-615 ЗНТУ

КРОСПЛАТФОРМНИЙ ВІРТУАЛЬНИЙ АМПЕРМЕТР-РЕЄСТРАТОР

У наш час модернізація обладнання та технологій у виробництві, системі освіти і наукових дослідженнях є актуальною задачею. Одним з шляхів її рішення є впровадження сучасного апаратно-програмного забезпечення, зокрема, віртуальних приладів.

При роботі віртуального вимірювального приладу аналогові сигнали від датчиків представляються в цифровому вигляді за допомогою систем збору даних на основі аналого-цифрових перетворювачів (АЦП).

На сучасному ринку електронних компонентів доступні мікроконтролерні плати ArduinoTM. Їх можна запрограмувати як системи збору даних для віртуальних приладів.

У даній роботі був розроблений кросплатформний віртуальний амперметр-реєстратор. Він складався з плати Arduino Uno, яка була запрограмована в якості системи збору даних з датчика струму GY-MAX471 та комп'ютерної програми, що була створена за допомогою мови програмування JavaTM.

Програма керувала роботою віртуального приладу та відображала результати вимірів. Вона мала графічний інтерфейс користувача, який дозволяв періодично реєструвати дані по вхідному струму в діапазоні 0-3 Ампера з 10-бітним вирішенням.

Період реєстрації установлювався користувачем в інтервалі від 0,2 до 3600 с. При цьому здійснювалося відображення чисельного значення останнього виміру струму та час його проведення, а також подання всіх зареєстрованих даних у табличному вигляді. Крім того, користувач мав можливість зупиняти та відновлювати реєстрацію, а також видаляти табличні значення вимірів або зберігати їх в *.txt або *.csv файлах.

Розроблений віртуальний амперметр-реєстратор може використовуватися для модернізації виробничих, лабораторних та експериментальних систем.

Надалі передбачається поліпшити функціональні можливості амперметра-реєстратора, зокрема, забезпечити його роботу в режимі з більш високим ступенем швидкодії та вирішення.

СИСТЕМА НА БАЗІ LARAVEL, ОПТИМІЗОВАНА ЗА ДОПОМОГОЮ FRONT END ТЕХНОЛОГІЙ

З приходом потужних комп'ютерів необхідність у зручності користувацького інтерфейсу зростає, оскільки зростає конкуренція у сфері інформаційних технологій. Невпинно відбувається розвиток інтерактивності та спрощення веб систем для користування ними. Саме це стало причиною створення нових методів оптимізації, розробки зручного та інформативного дизайну систем. Не менш важливим є забезпечення інтуїтивно зрозумілих інтерфейсів, що дає змогу використовувати систему більш ефективно та знижувати поріг для початку взаємодії користувача з системою.

З кожним роком все більше розробників створюють складні системи, які існують у веб середовищі та доступні віддалено користувачам. Створюються технології, які дозволяють розробити веб сервіси та системи якомога швидше, забезпечуючи надійність системи та зручність для користувача. Окремо необхідно зазначити ріст популярності мобільних пристроїв та їх використання для доступу в мережу інтернет. Зважаючи на це, розробники приділяють увагу проектуванню веб систем, орієнтованих також на мобільні пристрої, такі як телефони та планшети.

Для створення веб систем прийнято розподіляти розробку на Front End – розробку користувацького інтерфейсу та Back End – розробку серверної частини. Для взаємодії цих двох частин використовується спеціальний інтерфейс, який, незалежно від технологій і мов, на яких написані сервер та клієнтський інтерфейс, дозволяє передавати дані з клієнтської частини до серверної та навпаки. Найбільш популярними прикладами даних «мостів» є технології JSON та XML. Перша технологія є текстовим рядком, який містить необхідні дані. У другому випадку – це багаторівнева розмітка, яка схожа на розмітку html [1].

Не менш важливим фактором для створення зручного інтерфейсу є час очікування даних, з якими взаємодіє користувач. Чим менше час очікування, тим більш ефективною є система та її зручність для користування.

Нажаль, неможливо при роботі з веб системою зовсім прибрати затримку між взаємодією клієнтом та сервером, але значно зменшити час виконання запиту можливо та потрібно. Для цього необхідно, наскільки це є можливим, перенести функціонал системи на клієнтську частину, з метою

зменшення кількості запитів до сервера, та забезпечити відсутність повторних перезавантажень там, де вони не є необхідними.

Лідером з розробки клієнтських частин в мова програмування JavaScript. Довгий час вона була у версії ES5, що не давало змогу популяризувати об'єктно-орієнтовне програмування на JavaScript, оскільки версія ES5 не підтримує зручне використання класів та створення об'єктів, як в інших мовах програмування. Після переходу на нову версію ES6, у JavaScript з'явилась можливість створювати класи та об'єкти класичним способом, як в Java, PHP, Python та інших класичних мовах програмування. Перехід на нову версію JS надав можливість створювати бібліотеки та фреймворки для розробки клієнтських частин та популяризувати мову JavaScript для розробки веб систем.

Незважаючи на те, що постійно створюються нові бібліотеки та фреймворки для розробки інтерфейсів користувача, найбільш популярними технологіями для розробки клієнтських інтерфейсів є бібліотека React, фреймворк Angular, бібліотека VueJS.

Бібліотека React розроблена компанією Facebook і впевнено займає лідируюче положення у світі інформаційних технологій. Дана бібліотека має невеликий розмір, використовує віртуальний DOM для розробки системи, що дозволяє оптимізувати систему та підвищити швидкість її роботи [2].

VueJS, як і React використовує віртуальний DOM, представляє реактивність та компонентну структуру, фокусується на кореневій бібліотеці, в той час, як інші функції такі як маршрутизація і управління глобальним станом належать до інших бібліотек. В React не використовуються HTML теги для написання коду але, для зручності розробки, React має свій JSX синтаксис, який допомагає полегшити візуальне розділення коду на елементи структури та функціональні можливості. VueJS дає вибір у використанні JSX синтаксису або HTML тегів. VueJS та React для написання повноцінних веб систем потребують додаткових технологій, такі як Flux або Redux, які допомагають управляти станом даних [3].

Аналогом описаних бібліотек для написання клієнтської частини веб системи є фреймворк Angular. Даний фреймворк розроблений компанією Google та постійно має велику спільноту. Angular, починаючи з другої версії, є основним конкурентом React та VueJS. Даний фреймворк використовує мову Type Script, що в свою чергу, підвищує поріг для вступу у розробку на даному фреймворку. Використання Type Script та робота з реальним DOM є однією із ключових відмінностей даного фреймворку від попередньо розглянутих бібліотек [3].

Для оптимізації системи, розробленої на Laravel, необхідно;

- забезпечити зменшення часу очікування при виконанні типових задач, які не потребують перезавантаження сторінки;

- оптимізувати перше завантаження сторінки браузером користувача;
- забезпечити розподілення клієнтської частини (Front End) та серверної (Back End);
- оптимізувати клієнтську частину, яка використовує DOM дерево;
- підвищити зручність використання системи клієнтом.

За основу беремо торговельну систему, яка розроблена на фреймворку Laravel, з використанням шаблону проектування MVC (модель, вид, контролер). Дана система має клієнтську частину, яка складається з файлів шаблонізатору BLADE. При переході на сторінки, фреймворк створює запит до сервера, який крім необхідних даних надає розмітку сторінки, що повторює попередню завантажену розмітку HTML. Крім того при виконанні дій, таких як створення замовлення, додавання товару до кошику знову відбувається запит до сервера, який повторює попередню розмітку. Дана система використовує бібліотеку JQuery, яка працює з DOM деревом, тим самим збільшує час очікування роботи системи.

Для вирішення поставлених завдань обрана бібліотека React, за допомогою якої розроблено новий інтерфейс користувача, що забезпечує відсутність перезавантажень сторінок при переході по сайту. Також частина важливих функцій, які виконувалися фреймворком Laravel, виконується за допомогою React. Торговельна система використовує віртуальний DOM для фонового виконання задач перед тим, як вивести результати на сторінку сайту. Бібліотека JQuery вилучена, а функціонал за який вона відповідала, розроблено на Javascript ES6. Для забезпечення підтримки старішими браузерами клієнтської частини, ES6 переведено у попередню версію ES5 та мінімізовано для більш швидкого завантаження.

Оптимізація торговельної системи, розробленої на Laravel, за допомогою сучасних Front End технологій не торкнулася функціональних можливостей самої системи та її цілісності. Торговельна система так само виконує поставлені перед нею задачі, використовує шаблон проектування MVC, Dependency Injection, Fasad та має ті самі маршрути, що використовувалися системою до оптимізації. Тим паче, оптимізація системи принесла додатковий шаблон проектування MVVM, забезпечила більш зручний інтерфейс та знизила навантаження на сервер.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Алексеев, Г.В. Современные методы разработки компьютерных систем / Г.В. Алексеев. - СПб.: ГИОРД, 756. - 2012 с.
2. Ершов А.В. Интерактивные системы // А.В. Ершов, Ю.В. Гуляев, Н.А. Карпов, А.Ю. Пахомов. – М.: ЛИПС, 2014. – 596 с.
3. Сравнение с другими фреймворками. [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org/v2/guide/comparison.html>

УДК 658:004

Федько А.О.¹, Зеленцова І.Я.², Грушко С.С.³

¹ студ. гр. КНТ-514м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

³ канд. техн. наук, старш. викл. ЗНТУ

АНАЛІЗ АПАРАТУРНО-ТЕМПЕРАТУРНИХ ХАРАКТЕРИСТИК ПРИ РІЗНИХ ПІДХОДАХ ОПИСУ КЕРУЮЧОГО АЛГОРИТМУ НА МІКРОСХЕМАХ XILINX FPGA

Програмовані логічні інтегральні схеми (ПЛІС) є одними з найперспективніших елементів цифрової схемотехніки. ПЛІС являє собою кристал, на якому розташована велика кількість простих логічних елементів. Спочатку ці елементи не з'єднані між собою. З'єднання елементів (перетворення розрізнених елементів в електричну схему) здійснюється за допомогою електронних ключів, розташованих в цьому ж кристалі. Електронні ключі управляються спеціальною пам'яттю, в осередки якої заноситься код конфігурації цифрової схеми. Таким чином, записавши в пам'ять ПЛІС певні коди, можна зібрати цифровий пристрій будь-якого ступеня складності (це залежить від кількості елементів на кристалі і параметрів ПЛІС).

На відміну від мікропроцесорів, в ПЛІС можна організувати алгоритми цифрової обробки на апаратному (схемному) рівні. При цьому швидкодія цифрової обробки різко зростає.

Xilinx – американська компанія, що є одним з найбільших виробників програмованих логічних пристроїв. Компанія займається проектуванням напівпровідникових пристроїв програмованої логіки: FPGA, CPLD, ASIC. Мікросхеми Xilinx широко застосовуються в самих різних галузях промисловості, виробництва і сфери обслуговування – наприклад, у диференційованих системах і мережах сучасного покоління, включаючи серверні платформи хмарних обчислень, SDN / NFV, відео обробки, машинного зору, робототехніки, 5G Wireless і багатьох інших областях.

Мови опису апаратури (Hardware Description Language), служать для формального опису дискретних пристроїв обчислювальної техніки і можуть бути використані на всіх етапах розробки цифрових електронних систем. VHDL може використовуватися на етапах проектування, верифікації, синтезу і тестування апаратури так само, як і для передачі даних про проект, модифікації і супроводу.

VHDL підтримує три різних стилі для опису апаратних архітектур.

Перший з них - структурний опис (structural description), в якому архітектура представляється у вигляді ієрархії пов'язаних компонентів.

Другий - потоковий опис (data-flow description), в якому архітектура представляється у вигляді множини паралельних операцій мови, кожна з яких може управлятися логічними сигналами. Потоковий опис відповідає стилю опису, використовуваному в мовах реєстрових передач (RTL).

Третій - поведінковий опис (behavioral description), в якому логічні перетворення описуються послідовними програмними пропозиціями, схожими на наявні в будь-якій сучасній мові програмування високого рівня. Порівняння результатів імплементації схем автомата, описаних різними методами, в різні серії мікросхем фірми Xilinx, подані в табл.1.

Таблиця 1 – Апаратурно-температурні витрати схем, описаних різними методами

	Artix 7 (XC7A100T)		Spartan 3 (XA3S50)		Kintex 7 (XC7K70T)		Zynq (XC7Z010)	
	Структурний	Потоковий	Структурний	Потоковий	Структурний	Потоковий	Структурний	Потоковий
LUT	107	52	138	81	106	52	106	52
FF	6	46	16	52	6	46	6	46
Static power, mW	82.16	82.16	27.36	27.36	79.5	79.52	99.57	99.57
Junction temp., °C	25.4	25.4	26.2	26.2	25.2	25.2	25.5	25.5

Як показали результати досліджень, при різних підходах до опису керуючих автоматів, змінюються лише значення апаратних витрат мікросхеми, при цьому значення розсіюваної потужності та температури перемикачів залишаються незмінними, що вказує на те, що методи опису не впливають на температурні характеристики мікросхеми, а лише на витрати апаратури.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Офіційний сайт Xilinx All Programmable [Електронний ресурс] – Режим доступу: <https://www.xilinx.com>.
2. Бибило П.Н. Синтез логических схем с использованием языка VHDL. / П.Н. Бибило. – М.: Солон-Р, 2002. - 257 с.

3. Суворова Е.А. Проектирование цифровых систем на VHDL./ Е.А. Суворова, Ю.Е. Шейнин - СПб.: БХВ-Петербург. 2003. – 312 с.

УДК 004.021

Чиж С.Ю.¹, Скрупський С.Ю.²

¹ студ. гр. КНТ-514м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ПОБУДОВА N-АРНИХ ДЕРЕВ ІЗ БУЛЕВИХ ВИРАЗІВ

На даний час двійкова логіка дуже розповсюджена у світі, більшість усіх комп'ютерних систем у світі будуються за допомогою двійкової логіки. Майже будь-який цифровий пристрій або цифровий автомат побудований на двох станах 1 та 0.

Алгебра логіки – розділ математичної логіки, у якому логічні висловлювання і операції над ними досліджують алгебричними методами. Булевий вираз визначає значення істинності або хибності, значення задається за допомогою біт. Основними операціями такої алгебри є кон'юнкція операція І, диз'юнкція АБО, і заперечення НІ. Таким чином ця алгебра для описання логічних відношень має вигляд аналогічний тому, як описуються відношення у числовій алгебрі [1].

Деревом називають зв'язний граф без простих циклів. Орієнтоване дерево - це такий ациклічний орграф (орієнтований граф), у якого одна вершина, називана коренем, має напівступінь заходу, рівний 0, а інші - напівступені заходу рівні. Вершина з нульовою ступінню входу називається коренем дерева, вершини з нульовим напівступенем виходу називаються кінцевими вершинами або листками. n-арне дерево у свою чергу означає, що кожна некінцева вершина може мати n вихідних ребер [2].

Нажаль, на даний час не існує зручних засобів візуалізації булевих виразів. Тому метою даної роботи є розробка алгоритму для побудови зручної структури даних із булевих виразів, що значно спростить вирішення проблеми візуалізації для програмістів та наукових співробітників.

У випадку перетворення булевого виразу у n-арне дерево визначається, що будуть відображати кінцеві та некінцеві вершини, ребра, корінь дерева. Некінцеві вершини у такому дереві показують операцію над виразом, кінцеві вершини показують змінні з якими відбуваються операції, ребра показують над чим буде проводитися та чи інша операція, коренем дерева виступає загальна операція над усім виразом.

Основна логіка перетворення полягає у правильному розборі булевого виразу. Через те, що вирази можуть бути дуже великими та містити багато

різноманітних операцій, мати глибоку вкладеність, то зазвичай, для перетворення використовуються рекурсивні алгоритми. Звісно, рекурсивні алгоритми зазвичай потребують більше пам'яті та менш безпечні, але вони набагато легші для розуміння людині. В даному алгоритмі фактором виходу із рекурсії є знаходження будь-якої однієї змінної.

Під час проходження крізь вираз не потрібно забувати про постійну перевірку справності виразу. Вираз має дотримуватися чітких правил булевої алгебри, серед доступних символів обирається змінна над якою будуть проводитися операції, набір можливих логічних операцій, і також можуть бути доступні дужки. При знаходженні будь-якого іншого символу, або будь-якої логічної помилки, наприклад, закриту дужку без попередньої відкритої, вираз перестає бути справним, і його вже не можливо перетворити на n-арне дерево.

Проходження починається із першого символу виразу, і в залежності від нього алгоритм визначає свої наступні дії. Якщо символ відкрита дужка, то алгоритм рекурсивно «спускається» глибше до меншого виразу. Якщо ж була знайдена закрита дужка, то це означає, що менший вираз був опрацьований, і потрібно повернутися на рівень вище.

При знаходженні знаку будь-якої операції створюється вершина, яка є батьківським елементом для усіх задіяних у цій операції змінних. Потрібно зазначити, що у булевих виразах деякі операції не відділяються дужками, і навіть сам знак не пишеться, як у операції множення, тому алгоритм також відстежує, що якщо дві змінні знаходяться одна за одною то це означає, що була знайдена операція множення.

Якщо була знайдена змінна – це означає, що алгоритм дійшов до максимальної глибини. Для змінної створюється кінцева вершина, і алгоритм починає повертатися нагору.

Коли алгоритм опрацював усі підвирази, то батьківський елемент стає теперішнім коренем, а усі підвирази його дочірніми елементами. У кінці алгоритму повертається корінь дерева.

Правильна реалізація цього алгоритму дозволить переводити булеві вирази до вигляду n-арного дерева, це у свою чергу допоможе програмістам та науковцям легше працювати із різними завданнями пов'язаними із роботою з булевими рівняннями.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Алгебра логіки – Енциклопедія сучасної України [Електронний ресурс]. – Режим доступу: http://esu.com.ua/search_articles.php?id=43596
2. Дискретна математика - Деревя [Електронний ресурс]. – Режим доступу: http://evgavrilenko.ucoz.ru/DS/LEKCIYA_4.pdf

СЕКЦІЯ «ПРОГРАМНІ ЗАСОБИ»

УДК 004.3

Туленков А.В.¹, Пархоменко А.В.², Калініна М.В.³, Колпаков Д.А.⁴

¹ асп. ЗНТУ

² канд. техн. наук, доц. ЗНТУ

³ асист. ЗНТУ

⁴ інженер-проектувальник ПКФ «МОТОР»

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ LEGRAND ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМ РОЗУМНИЙ БУДИНОК

На теперішній час лідируючі позиції в Запорізькому регіоні у сфері домашньої автоматизації, а також розробки електричних та інформаційних систем будівель займає компанія ПКФ “Мотор”, офіційний дистриб’ютор італійської компанії Legrand, яка пропонує повний комплекс рішень для автоматизації готелів, лікарень, котеджів та квартир.

Для створення системи домашньої автоматизації дана компанія пропонує платформу MyHome. Головними перевагами MyHome є безпека та екологічність, простота у використанні, модульність, можливість розширення без проведення додаткових будівельних робіт, наявність можливостей для людей з особливими потребами, зручне дистанційне керування.

Як показали проведені дослідження, головною технічною особливістю платформи є спрощене дводротове неполярне з’єднання SCS (Simplified Cable Solution), яке дозволяє реалізацію мережі з різними топологіями: зірка, шина та змішана. Через SCS можлива передача чотирьох типів сигналів з частотною модуляцією: живлення 27 В, дані на частоті 9600 Гц, звук, відео. Протоколом передачі даних є мережевий протокол CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance), що надає множинний доступ з виявленням несучої частоти та запобіганням колізій до початку передачі даних. Всі керуючі пристрої та актуатори виконані на базі високопродуктивних контролерів STM32. Керуючими пристроями є клавіші, сенсорні панелі, блоки термоконтролю, стереопідсилювачі. Пропонується також рішення керуючих пристроїв з використання поширеного протоколу ZigBee.

Аналіз показав, що розробники використовують датчики контролю якості повітря, витоку води, але не використовують датчики вологості. Також особливістю є те, що деякі керуючі пристрої можливо використовувати тільки з відповідними актуаторами, що призначені для певної підсистеми, до якої вони входять (наприклад, підсистема термоконтролю).

Платформа дозволяє розробити підсистеми термоконтролю, домофонії, освітлення, енергоконтролю, охорони, звукового контролю.

Для програмування роботи системи домашньої автоматизації використовуються пристрої для розробки простих та складних сценаріїв. Блок простих сценаріїв дозволяє зберігати 16 сценаріїв для застосунків автоматизації, звукової підсистеми та підсистеми освітлення. Блок складних сценаріїв дозволяє зберігати до 300 сценаріїв, виконувати розгорнуті сценарії, пов'язані з певним часом і датою, з включенням або виключенням охоронної сигналізації, дозволяє симулювати присутність в помешканні за допомогою автоматичної активації ролетів або освітлення під час відсутності користувачів вдома та в попередньо визначений час.

Перевагами комплексу рішень MyHome є: можливість самостійного програмування та сканування пристроїв через спеціалізоване програмне забезпечення MyHomeSuite, яке може використовувати сам користувач; широкий набір мобільних додатків для управління автоматизацією; можливість підключення стороннього програмного забезпечення для управління (iRidium та OpenHab).

На сьогоднішній день, за кошти міжнародного освітнього проекту ALIOT «Інтернет речей: нова навчальна програма для потреб промисловості та суспільства» для ЗНТУ придбано обладнання від компанії Legrand та проводяться роботи зі складання навчально-демонстраційного стенду, що містить блок складних та простих сценаріїв, блок живлення шини, шлюз доступу до системи OpenWebNet, систему термоконтролю, керуючі пристрої, захисні автомати та ін. Стенд призначено для навчання студентів основним принципам створення систем домашньої автоматизації, принципів побудови силових шаф, принципам розміщення обладнання, практичних принципів шинної організації, програмування систем домашньої автоматизації, розвитку розуміння алгоритмів роботи систем домашньої автоматизації та використання можливості розширення системи за допомогою стороннього програмного забезпечення та апаратних засобів систем автоматизації різноманітних виробників.

Використання даного обладнання в навчальному процесі дозволить студентам отримати знання та практичні навички складання та проектування малих силових шаф, роботи з CAD системами, реалізації бездротових та дротових технологій передачі даних, а також програмування сценаріїв роботи систем домашньої автоматизації.

ХМАРНІ РІШЕННЯ SAP ДЛЯ РЕАЛІЗАЦІЇ ТЕХНОЛОГІЙ ІНТЕРНЕТУ РЕЧЕЙ

Німецька компанія SAP є одним з відомих постачальників комплексних рішень в Україні. Зокрема, вона пропонує цілу лінійку рішень для Інтернету речей.

SAP IoT Bridge - це цифровий командний центр, який дозволить операційним менеджерам забезпечити прозорість бізнес-процесів і дасть можливість діяти в режимі реального часу. SAP Global Track and Trace - хмарний додаток для уніфікованого трекінгу, моніторингу та отримання звітів про об'єкти та бізнес-процеси в мережевих каналах поставок. SAP Leonardo IoT Edge - хмарне програмне забезпечення для обчислень, зберігання даних і бізнес-семантики, за допомогою якого пристрої без доступу до дата-центру компанії можуть продовжувати ефективно брати участь у бізнес-процесах практично в режимі реального часу. SAP Digital Manufacturing Insights - хмарне рішення для централізованого управління виробництвом та оптимізації робочих процесів. SAP Asset Manager - хмарний мобільний додаток, що дозволяє відслідковувати стан здоров'я співробітників, обладнання, управляти техобслуговуванням і системою безпеки. Зокрема, платформа SAP Leonardo об'єднує різні програмні можливості: машинне навчання, Інтернет речей, Big Data, аналітику і блокчейн на SAP Cloud Platform разом з досвідом SAP, глибоким знанням індустрії і просунутою методологією design thinking.

Для навчання студентів сучасним інформаційним технологіям на основі використання продуктів компанії, ведуться дослідження за двома напрямками. Перший напрям «Хмарна платформа як служба для розробки програмного забезпечення»: новітні технології для цифрових перетворень бізнес-процесів (design thinking, машинне навчання, великі дані, Інтернет речей, блокчейн, інтелектуальні дані та аналітика даних); додаток платформи SAP Cloud platform; база даних SAP HANA In-Memory для розробки бізнес-додатків. Другий напрям «Розробка програмного забезпечення в системі SAP»: ABAP-словник (структура даних у системах SAP); робота з ABAP Workbench; розробка інтерфейсу користувача на SAPUI5 (HTML5).

УДК 004.93

Каврін Д. А.¹, Субботін С. О.²

¹ асп. ЗНТУ

² д-р техн. наук, проф., зав. каф. ЗНТУ

ВІДНОВЛЕННЯ ПРОПУСКІВ У ВИБІРКАХ З ВИКОРИСТАННЯМ МЕТОДІВ ЕВОЛЮЦІЙНОГО ПОШУКУ

Реальні вибірки даних часто містять в собі пропущені значення. Причини виникнення пропусків пов'язані з помилками в роботі обладнання, програмного забезпечення і т.д. Ігнорування подібних дефектів в даних може призводити до зниження продуктивності діагностичних систем і, можливо, до прийняття невірних рішень.

Існують різні методи заповнення (відновлення) пропусків в даних, які відрізняються за своєю природою, галуззю застосування та обчислювальною складністю. Умовно всі методи відновлення можна поділити на глобальні, в яких передбачається оцінка значень пропусків по всіх об'єктах вибірки, і локальні, які оцінюють значення пропусків в деякій локальній близькості [1].

В даній роботі пропонується модифікація методу локального відновлення пропущених даних ZET [1]. В основі методу ZET лежать припущення про надмірність даних у вибірці, їх лінійну залежність і локальну компактність. Одним з основних етапів методу ZET є підбір параметрів компетентності рядків (екземплярів) та стовпців (ознак), для пошуку, яких необхідно вирішити задачу оптимізації. У запропонованій модифікації, задача оптимізації вирішується методами еволюційного пошуку.

Для оцінки та порівняння якості відновлення пропущених значень, базовий метод ZET, в якому застосовується симплекс метод оптимізації, порівнювався з запропонованою модифікацією. В якості вихідної вибірки використовувався набір даних випробувань газотурбінних авіаційних двигунів [2]. Даний набір не мав природних пропусків, тому пропущені значення створювалися штучно, випадковим чином. Такий підхід дозволив оцінити роботу методу відновлення пропущених значень безпосередньо порівнянням відновлених даних з природними значеннями вихідної вибірки і розрахувати середньоквадратичну похибку (RMSE). Також проводилась оцінка швидкості роботи методу для різних обсягів пропущених значень. В якості міри близькості стовпців розглядалися значення кореляції та евклідової метрики. Таким чином, проводився аналіз трьох модифікацій методу, в залежності від способу оптимізації та міри близькості (рис. 1).

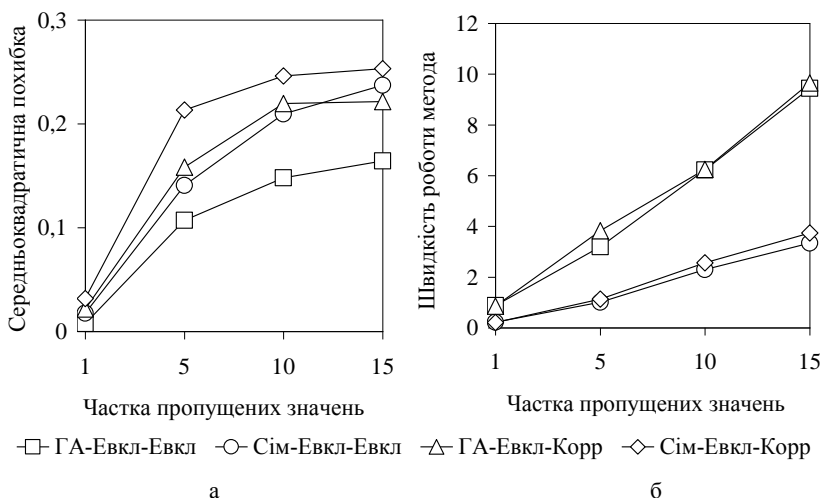


Рисунок 1 – Візуалізація залежностей середньоквадратичної похибки (а) і швидкості роботи модифікацій методу ZET (б) від частки пропущених значень у вибірці

Запропонована модифікація методу ZET була програмно реалізована при проведенні обчислювальних експериментів з відновлення пропущених значень у вибірках даних. Експерименти показали, що використання оптимізаційних методів еволюційного пошуку дозволяє більш точно відновлювати пропущені значення при невеликій втраті швидкості, в порівнянні з оптимізацією методом симплекса.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Загоруйко Н. Г. Прикладные методы анализа данных и знаний [Текст] / Н. Г. Загоруйко. – Новосибирск : ИИМ, 1999. – 270 с.
2. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиационных двигателей : [монография] / [Богуслаев А. В., Олейник Ал. А., Олейник Ан. А. и др.] ; под ред. Д. В. Павленко, С. А. Субботина. – Запорожье : ОАО «Мотор Сич», 2009. – 468 с.

УДК 004.93

Субботін С.О.¹, Корнієнко О.В.², Наривський О.Е.³

¹ д-р техн. наук, зав. каф. ЗНТУ

² асп. ЗНТУ

³ д-р техн. наук, проф., технічний директор ТОВ "Укрспецмаш"

НЕЙРОМЕРЕЖА ДЛЯ МОДЕЛЮВАННЯ КОРОЗІЇ СТАЛІ

У виробництві теплообмінного обладнання часто використовують корозійностійкі сталі і сплави, леговані хромом, нікелем і молібденом. Ці конструкційні матеріали мають високу корозійну стійкість в багатьох агресивних середовищах, але можуть піддаватися виразковій корозії в оборотних водах, що містять хлориди.

Тому актуальною проблемою є побудова моделей залежності критичних температур виразкоутворення (КТВ) від характеристик використовуваної сталі та оборотної води.

Перспективним засобом побудови таких моделей є штучні нейронні мережі. Серед відомих архітектур нейронних мереж найбільш потужними і одночасно доступними для подальшого аналізу та сприйняття людиною є багатопарові нейронні мережі прямого поширення сигналу, які обрано як базис у даній роботі.

Проте недоліками моделей, побудованих на основі нейронних мереж прямого поширення сигналу, є їхня надлишковість, що також негативно впливає на інтерпретабельність побудованих моделей.

Отже, актуальним завданням є удосконалення методів побудови моделей на основі нейромереж прямого поширення сигналу.

Метою роботи було удосконалення математичного забезпечення побудови нейромоделей та їхнє застосування для вирішення задачі моделювання критичних температур виразкоутворення сталі.

Нейронна мережа прямого поширення має шарувату архітектуру: у таких моделях вхідні ознаки розпізнаваного екземпляра подаються на входи мережі та далі поширюються від вхідного шару до вихідного. Навчання нейронних мереж прямого поширення, як правило, проводиться на основі градієнтного методу та техніки зворотного поширення помилки. У процесі навчання мінімізується функція помилки мережі E , яка визначає різницю між виходами мережі і цільовими значеннями. В якості функції помилки доцільно використовувати середньоквадратичну помилку E .

Одним з методів оптимізації процесу навчання нейронних мереж є виключення нейронів. Він полягає у тому, що в процесі навчання деяка частина нейронів видаляється з мережі. На кожній ітерації навчання виключаються випадкові нейрони. Кожен нейрон може бути виключеним з

ймовірністю P . Ймовірність виключення може бути різною для кожного шару нейронної мережі. При тестуванні та під час роботи моделі нейрони не виключаються. При цьому вихідні значення нейронів на шарах, де використовувалося виключення помножуються на коефіцієнт $Q = 1 - P$, P – ймовірність виключення нейронів на шарі.

Виключення нейронів дозволяє уникнути перенавчання моделі, що в більшості випадків приводить до збільшення точності моделі на тестових даних.

Для покращення узагальнюючих властивостей моделі пропонується визначати ймовірність виключення нейронів за їх впливом на помилку моделі:

$$P_n = \frac{P_b}{2} + \frac{P_b}{1 + e^{\frac{2Ck}{N} - k}},$$

де P_b – базова ймовірність виключення нейронів на шарі, N – кількість нейронів на шарі, C – кількість нейронів, для яких абсолютні значення градієнтів більші, ніж для n -го нейрона, k – коефіцієнт, що впливає на нелінійність залежності. Рекомендується обирати значення k з інтервалу $[1, 10]$.

Таким чином ймовірність виключення нейронів зростає зі збільшенням їх впливу на помилку.

При навчанні нейромережі з модифікованим методом виключення нейронів необхідно двічі розраховувати вихідні значення нейронів та градієнти. У перший раз вихідні значення та градієнти розраховуються з використанням всіх нейронів. Після цього визначаються нейрони, що повинні бути виключеними на ітерації. Далі проводяться етапи розрахування вихідних значень, розрахування градієнтів та етап корекції вагових коефіцієнтів з врахуванням виключених нейронів. При тестуванні та під час роботи нейромережі коефіцієнт Q , на який помножується функція активації, визначається за формулою $Q = 1 - P_b$, де P_b – базова ймовірність виключення нейронів.

Розроблено програмне забезпечення, яке реалізує запропонований метод, та використано його для визначення критичних температур виразкоутворення. Використання модифікованого метода виключення нейронів дозволяє зменшити помилку моделі в середньому на 4% в порівнянні зі звичайним методом виключення.

МОДЕЛЮВАННЯ ДИСКУРС-ПРОЦЕСУ ПРИ ПРОЕКТУВАННІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Сучасні інтелектуальні системи (ІС), що спілкуються природною мовою, застосовують поняття дискурсу у двох значеннях: дискурс, як процес, і дискурс, як текст.

Дискурс - покроковий процес спілкування ІС та людини природною мовою, з метою обговорення певної проблеми та досягнення узгодження їх цілей. Дискурс є гнучким, динамічним, складним процесом, цілеспрямованим за певною стратегією, але стратегія і навіть ціль можуть бути змінені у разі неможливості досягнути узгоджених цілей [1]. Спрямованість дискурсу визначається початковою ситуацією та ціллю.

Весь процес обговорення проблеми фіксується в дискурсі-тексті. На кожному кроці спілкування при створенні текстів учасників враховується зміст вже сформованого тексту за всіма кроками.

Для тексту дискурсу, як результату обговорення, характерні змістова цілісність, композиційна завершеність, структурованість, цілеспрямованість, змістова зв'язність. Останні три властивості характерні також для тексту окремих кроків, їх фрагментів, висловлювань.

Текст є статичним об'єктом, який у згорнутому вигляді містить процес його створення. Для розуміння змісту тексту ІС повинна розгорнути за текстом процес розуміння тексту. Процес реалізують методами дискурсивного аналізу, нового розділу практичної лінгвістики.

Сформулюємо які основні аспекти об'єднує в собі дискурс: прагматичний (спілкування виконується для застосування у певній діяльності); комунікативний (спілкування ведеться природною мовою); когнітивний (розуміння та формування текстів здійснюється на основі когнітивних структур – бази знань); «ментальний» пошук понять у когнітивних структурах ведеться за принципом асоціативного мислення) [2].

Проектування та дослідження складних процесів та систем полегшується, якщо застосовувати принцип моделювання системотехніки. Подання складного процесу дискурсу лінгвістичними моделями спрощує дослідження кожної групи властивостей дискурсу. В процесі дослідження дискурс-процесу були виявлені наступні лінгвістичні моделі:

організації концептуальної структури знань; виявлення концептів асоціаційним методом; виявлення відповідності між словами, реченнями і концептами та концептуальними структурами; умовиводу; підтримки

стратегій учасників дискурсу та їх зміни; виявлення структури тексту; створення тексту; контролю тексту на зв'язність, цілеспрямованість, цілісність, композиційну завершеність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Практичне програмування мовою Visual Prolog: навч. посіб. / [Дейнега Л.Ю., Камінська Ж.К., Левада І.В., Сердюк С.К.]. – Запоріжжя: ЗНТУ, 2016. – С. 118-150.

2. Спенсер Г. Ассоциативная психология [Текст] / Г. Спенсер, Т. Циген. – М.: Москва, 1998. – 560 с.

УДК 004.3

Ткачук Є.М.¹, Пархоменко А.В.², Каплієнко Т.І.²

¹ студ. гр. КНТ-124м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

РОЗРОБКА КЕРУЮЧОГО ПРИСТРОЮ СИСТЕМИ «РОЗУМНИЙ БУДИНОК» ДЛЯ КОРИСТУВАЧІВ З ОСОБЛИВИМИ ПОТРЕБАМИ

«Розумний будинок» - це система, створена за допомогою автоматизації та високотехнологічних пристроїв, що забезпечує, комфорт, ресурсозбереження та безпеку для користувачів. У найпростішому випадку вона повинна вміти розпізнавати конкретні ситуації, що відбуваються в будинку, і відповідним чином на них реагувати. При цьому, одна з підсистем може управляти поведінкою інших за заздалегідь розробленими алгоритмами. Принципи керування роботою розумного будинку традиційно базуються на використанні сучасних платформ домашньої автоматизації (наприклад, OpenHAB), а також смартфона, планшета, або комп'ютеру для організації взаємодії користувача з системою. Проте, користувачі з особливими потребами не завжди можуть виконувати певні дії, що пов'язані з використанням клавіатури, сенсорних панелей або голосового управління через фізичні вади. Тому, дослідження та організація методів взаємодії таких користувачів з системою домашньої автоматизації є актуальною задачею.

Метою даного проекту є створення пристрою, що кріпиться до руки, і за допомогою рухів надає можливість надсилати відповідний сигнал для керування підсистемами розумного будинку. Проект створюється на основі плати Arduino Nano, Bluetooth-модуля HC-06, 3-вісьового акселерометру GY-291 ADXL345 та додатку для смартфона під керуванням ОС Android 5.0 Lollipop та вище. Основні етапи роботи: програмування плати Arduino та підключених модулів для отримання даних з акселерометру та передачі їх за

допомогою Bluetooth; створення Android-додатку для отримання даних за допомогою Bluetooth та їх обробки з подальшою передачею необхідних даних в систему розумного будинку (на сервер). З акселерометру GY-291 ADXL345 отримуються дані щодо положення (X, Y, Z), поворот (roll), напрямок (heading), уклін (pitch) та формуються у вигляді x/y/z/r/p/h і передаються за допомогою Bluetooth.

На даний час триває створення Android-додатку для отримання даних за допомогою Bluetooth та відлагодження програмного коду Arduino з модулями.

УДК 004.3

Коломоєць Р.Р.¹, Пархоменко А.В.², Гладкова О.М.³

¹ студ. гр. КНТ-226сп ЗНТУ

² канд. техн. наук, доц. ЗНТУ

³ канд. техн. наук, старш. викл. ЗНТУ

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАЛИВКИ ГРАФІЧНИХ ЗОБРАЖЕНЬ

Заливка зображень – це процедура заповнення деякої ділянки зображення, обмеженої контуром, заданим кольором. Ця процедура широко використовується в графічних редакторах та дизайнерських програмах. На сьогоднішній день новою сферою застосування даної процедури є ігри, що дозволяють користувачам розмальовувати зображення та, таким чином, розвивати свої творчі здібності, або просто відпочивати. Існує багато алгоритмів заливки, але здебільшого вони вимагають оптимізації для застосування на мобільних платформах. Наприклад, проведені дослідження чотири-зв'язного рекурсивного алгоритму заливки дозволили виявити дуже повільне зафарбовування області та швидке переповнення стеку через відсутність контролю даного процесу. Тому, актуальною є задача реалізації оптимізованого алгоритму для швидкого зафарбування зображень та розробка програмного забезпечення графічної гри для мобільних платформ.

Для реалізації проекту використано ігровий движок Unity та мову програмування C#. Завдяки використанню технології Coroutines було додане багатопоточне обчислення, крім того, на зображення, які надходять до движку додається шар, який збільшує контури зображення, завдяки чому не потрібно глибоко аналізувати зображення для досягнення повністю залитої області. Інтерфейс програми представлено на рис.1. На даний час триває процес адаптації програмного продукту під мобільні платформи Android та iOS. Майбутні дослідження в цій галузі будуть спрямовані на подальшу

оптимізацію швидкості алгоритму та реалізацію анімації заливки, зокрема на високих розширеннях екрану.

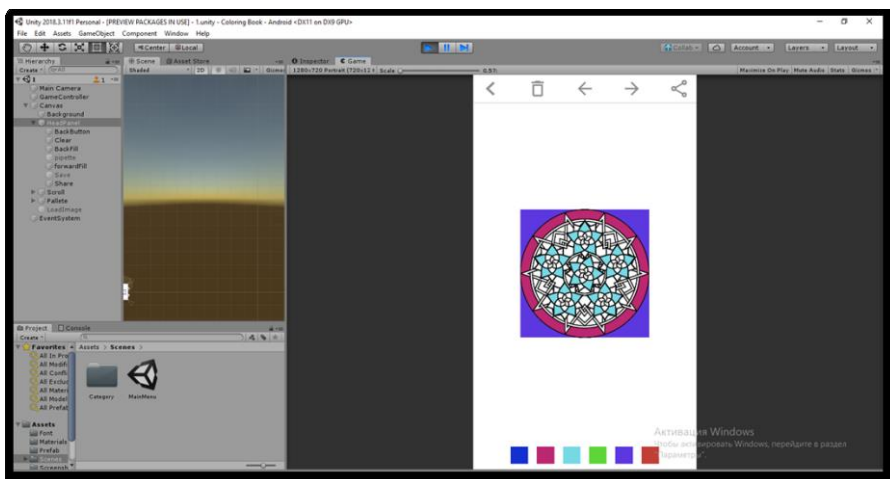


Рисунок 1 – Реалізація алгоритму заливки у ігровому движку Unity

УДК 004.946

Білов О.І.¹, Пархоменко А.В.², Соколянський О.В.³

¹ студ. гр. КНТ-715 ЗНТУ

² канд. техн. наук, доц. ЗНТУ

³ асп. ЗНТУ

РОЗРОБКА ВІРТУАЛЬНОЇ МОДЕЛІ ДЛЯ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ РОЗУМНИЙ БУДИНОК

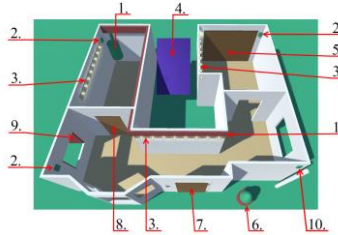
При роботі з віддаленими лабораторіями, наприклад з лабораторією Smart House & IoT, користувач працює у інтерактивному режимі з реально існуючим демонстраційним стендом розумного будинку, за яким можна спостерігати у режимі онлайн з веб-камери. Завдяки системі організації черги, кілька користувачів можуть одночасно користуватися однією лабораторією, але при цьому виконувати лише окремі, не пов'язані між собою експерименти. В той же час, для наочного ознайомлення з загальною структурою та використанням обладнання демонстраційного стенду лабораторії Smart House & IoT доцільно створити віртуальну модель розумного будинку. Тому, метою даної роботи є розробка віртуальної моделі, яка дозволить користувачу зробити віртуальний тур навколо та всередині

розумного будинку, дослідити розташування компонентів автоматизації, ознайомитися з їх призначенням та описом.

Віртуальні моделі використовуються для проведення турів по приміщенням, наприклад при бронюванні номерів готелю або при купівлі нерухомості. Такий засіб як Google Street View може допомогти користувачу зорієнтуватися на місцевості при плануванні подорожей, адже він дозволяє віртуально оглянути вулиці майже будь-якого кутка світу. Також, використовуючи віртуальні моделі, у Інтернет магазинах можна демонструвати продукцію та її функціонал, що значно спрощує засвоєння інформації. Віртуальні моделі є популярним інструментом сучасних віддалених лабораторій. Вони дозволяють ефективно проводити експерименти у разі, коли є проблеми з роботою реального обладнання або передачею зображення з веб-камер. Завдяки використанню віртуальної складової, віддалені лабораторії переходять в категорію гібридних онлайн лабораторій. Яскравим прикладом такої лабораторії є GOLDi (Grid of Online Lab Devices), створена в Технічному університеті Льменау, Німеччина.

Як показали проведені дослідження, розробка віртуальної моделі можлива з допомогою технології Java, з використанням графічної бібліотеки Swing, але ця технологія не підтримується багатьма сучасними браузерами та не дозволяє створювати 3D графіку. Саме тому оптимальним варіантом стало використання сучасної графічної бібліотеки WebGL у комбінації з HTML5 та JavaScript. В той же час, засобів, що можуть використовувати WebGL, досить мало. Основними конкурентами є Unreal Engine 4 та Unity 5. Хоча Unreal Engine 4 має офіційну підтримку експорту проектів на WebGL, у технічній документації вказано, що стабільність кінцевого продукту дуже низька. Тому, був обраний Unity 5, оскільки веб проект, розроблений з його допомогою, більш стабільний та продуктивний. Для створення 3D моделей вибір стояв між Blender та Autodesk 3DsMax. Autodesk 3DsMax – платна програма, але існує її спеціальна безкоштовна версія для навчання студентів. Blender – повністю безкоштовна програма, але вона гірше пристосована для моделювання інтер'єрів, а саме це має бути реалізовано для віртуального макету розумного будинку. Тому був вибраний Autodesk 3DsMax.

У віддаленій лабораторії Smart House & IoT реалізовано набір експериментів, що дозволяють підвищити комфорт та забезпечити зберігання ресурсів: Клімат контроль, Контроль освітлення, Сонячна станція, Контроль доступу, Контроль присутності, Контроль зони, Контроль вентиляції та інші. На рис. 1 наведено вигляд віртуального макету та основні види сенсорів та актуаторів, що встановлені на реальному стенді віддаленої лабораторії.



1 – датчик температури та вологості; 2 – датчики руху; 3 – світлодіодні стрічки; 4 – сонячна панель; 5 – сервопривід гаражних воріт; 6 – датчик вологості ґрунту; 7 – вхідні двері з RFID ключем; 8 – сервопривід міжкімнатних дверей; 9 – сервопривід вікна; 10 – датчик контролю периметру.

Рисунок 1 – Схема розташування елементів системи розумний будинок

Досвід роботи з віртуальним макетом розумного будинку може стати певною мотивацією для отримання усіх необхідних теоретичних знань та практичних навичок для користувачів, які на основі отриманого досвіду роботи з віддаленою лабораторією у майбутньому зможуть реалізувати власну систему домашньої автоматизації.

УДК 004.3

Левченко О.А.¹, Пархоменко А.В.²

¹ студ. гр. КНТ-214м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОЗРОБКИ ВБУДОВАНИХ СИСТЕМ НА ОСНОВІ ПРОГРАМОВАНИХ КРИСТАЛІВ

Системи на кристалі (PSoC) - це наступний крок в еволюції розвитку мікроконтролерних систем. Найважливішою їх відмінністю є те що PSoC не має фіксованої цифрової (UART, I2C, SPI і тд) та аналогової (компаратори, операційні підсилювачі, мультиплексори тощо) периферії. При цьому, розробник має можливість самостійно створювати (програмувати) потрібний функціонал, підключаючи необхідні модулі та системи до проекту.

Одним з відомих виробників PSoC є компанія Cypress Semiconductor. Вся продукція компанії поділяється на набори сімейств контролерів. Різні сімейства мають різне процесорне ядро (для PSoC 3 це MCS-51, для PSoC 4 - Cortex M0, для PSoC 5LP - Cortex M3, а для PSoC 6 - Cortex M0 + і Cortex M4), а також набір програмованої логіки. Логіка не тільки цифрова, але й аналогові блоки (АЦП, аналогові комутатори). На платах розміщені порти

GPIO. PSoC надає також операційні підсилювачі, які можна програмно комутувати.

Метою даної роботи є розробка тестових застосунків на основі плати CY8CKIT-145-40XX 4000S (рис.1). В ході досліджень були розглянуті апаратні та програмні особливості реалізації плати та її керуючих органів, типові помилки при роботі, а також проблеми захисту від зовнішніх впливів (особливо вологи). Створений проект - це сенсорна панель, яка дозволяє реалізувати ємнісне керування різноманітними електронними пристроями, як за допомогою дискретних сигналів (кнопки) так і через плавне регулювання (слайдер).

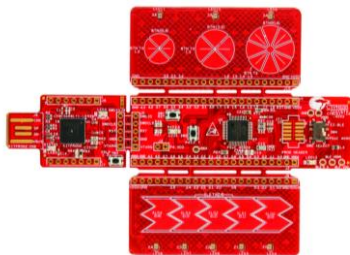


Рисунок 1 - Демонстраційна плата CY8CKIT-145-40XX 4000S

УДК 004.794

Льовкін В.М.¹, Ананченко Є.М.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-413м ЗНТУ

РОЗПІЗНАВАННЯ ЕМОЦІЙНОЇ ЗАБАРВЛЕНOSTІ КОМЕНТАРІВ КОРИСТУВАЧІВ ВІДЕОХОСТИНГУ

Важливим виявлення емоційної складової, яка закладена в певному повідомленні, є не тільки для вивчення особливостей сприйняття конкретної людини, але й для накопичення даних про сприйняття деяких ресурсів різними людьми. Зокрема такими ресурсами є медіаресурси, доступ до яких можуть отримувати користувачі відеохостингів. Тоді текстова інформація, отримана від користувачів, представляється у вигляді коментарів.

Використання засобів розпізнавання емоційної забарвленості коментарів повинно дозволити відслідковувати:

- відгуки про окремо взятий відео канал компанії (аналіз в межах однієї компанії);
- про відео канали різних компаній (аналіз різних компаній на ринку);

– про окремі відеоролики, а значить окремі продукти компанії (динамічний аналіз).

Розв’язати проблему розпізнавання емоційної забарвленості коментарів якісно з використанням словників, у яких виділено терміни, що характеризують емоційне забарвлення, неможливо, тому що будь-який термін, навіть той, що безсумнівно несе в собі емоційне забарвлення, може містити протилежні відтінки емоційного забарвлення. Наприклад, слово «неймовірний» може відображати позитивний відтінок («неймовірний світ») або негативний («неймовірний невдаха»). Тому дуже важливим є врахування контексту, тобто послідовності слів у коментарі користувача відеохостингу та їх зв’язку між собою.

В якості основної моделі для розпізнавання емоційної забарвленості коментарів користувачів відеохостингу Youtube використано нейронну мережу на основі архітектури довгої короткострокової пам’яті [1], оскільки важливим під час процесу розпізнавання є врахування контексту, в якому використовуються певні слова

Виконано проектування і реалізацію програмного забезпечення мовою програмування Python, що складається з програмної бібліотеки та веб-застосунку, створеного на основі використання фреймворку Flask. Програмний застосунок розроблено на основі використання бібліотеки для машинного навчання Tensorflow.

Розроблене програмне забезпечення (рис. 1) дозволяє визначати емоційне забарвлення заданого коментаря, визначати емоційне забарвлення кожного коментаря для заданого відеоресурсу відеохостингу Youtube, а також для заданого каналу вцілому, узгоджувати статистику за всіма коментарями відеоресурсу вцілому або цілого каналу.

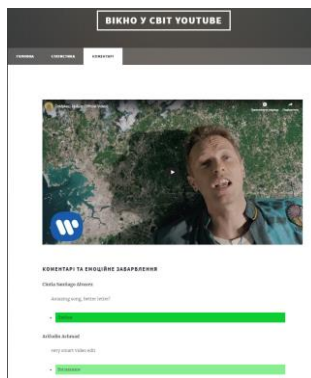


Рисунок 1 – Інтерфейс програмного забезпечення розпізнавання емоційної забарвленості коментарів користувачів відеохостингу

Наукова новизна роботи полягає в тому, що в ній запропоновано метод розпізнавання емоційної забарвленості коментарів користувачів відеохостингу, що передбачає попереднє оброблення вибірки даних, використання моделі на основі довгої короткострокової пам'яті для розпізнавання, її навчання та прийняття рішень за її допомогою.

Таким чином, у роботі запропоновано методу розпізнавання емоційної забарвленості коментарів та виконано його програмну реалізацію у вигляді відповідного веб-застосунку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Николенко, С. Глубокое обучение [Текст] / С. Николенко, А. Кадурын, Е. Архангельская. – СПб.: Питер, 2018. – 480 с.

УДК 004.794

Льовкін В.М.¹, Головізніна А.О.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТз-113м ЗНТУ

МЕТОД ВИЗНАЧЕННЯ РЕКОМЕНДАЦІЙ КІНОФІЛЬМІВ НА ОСНОВІ РОЗПІЗНАВАННЯ ОБРАЗІВ

Одним з напрямків, для яких дуже активно використовуються інформаційні технології, є медіаресурси, одним з найпопулярніших прикладів яких є кінофільми. Відповідно створення інструментаря, який дозволить користувачу отримувати персональні рекомендації та обирати для перегляду фільм, який швидше за все зацікавить користувача, є актуальним і важливим завданням.

Під час визначення рекомендованих кінофільмів важливо враховувати не тільки те, що користувач оцінив деякий кінофільм на певну оцінку, в результаті чого кінофільм було позначено як вподобаний. Важливо враховувати також порядок відповідних дій. Тобто на уподобання певного кінофільму впливає не тільки те, що його вподобала певна кількість людей і відповідна його середня оцінка, але і те в який саме момент і з яким саме досвідом це було зроблено.

Загалом користувач не завжди прагне передивитися фільм, який більшість користувачів вважає найкращим. Він може прагнути передивитися певний фільм у певний момент. Тому дуже важливо враховувати послідовність уподобань користувачів, а не тільки сам факт уподобання.

Для врахування послідовностей зручним інструментом є рекурентні нейронні мережі [1]. Вони, на відміну від багатозарових перцептронів,

дозволяють використання послідовності змінної довжини, що критично важливо для розглядаємої задачі, адже різні користувачі будуть мати різний попередній досвід: яке б обмеження на довжину послідовності не встановити, користувач, що нещодавно зареєструвався, може не мати такої кількості попередніх уподобань, що унеможливить будь-які рекомендації для нього.

Відповідно в даній роботі запропоновано метод визначення рекомендацій кінофільмів, який ґрунтується на основі використання вентильних рекурентних вузлів.

Розроблено програмне забезпечення на основі використання мови програмування Python, фреймворку Django та бібліотеки для машинного навчання Theano.

Розроблене програмне забезпечення (рис. 1) реалізує наступні функціональні можливості:

- рекомендування кінофільмів для перегляду;
- вподобання або невподобання доступних кінофільмів;
- перегляд додаткової інформації за кінофільмом;
- перегляд переліку новинок;
- перегляд переліку всіх кінофільмів;
- перегляд переліку переглянутих кінофільмів;
- пошук кінофільмів;
- реєстрація та авторизація облікових записів користувачів;
- керування обліковими записами користувачів.

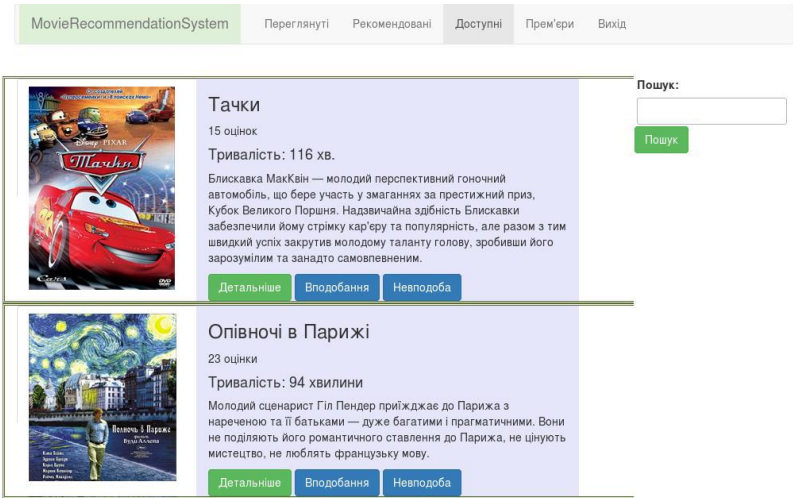


Рисунок 1 – Інтерфейс програмного забезпечення визначення рекомендованих кінофільмів

Наукова новизна роботи полягає в тому, що в ній запропоновано метод визначення рекомендацій кінофільмів, який застосовує модель на основі вентильних рекурентних вузлів, у якій в якості входів використовується послідовність уподобань користувачів, що дозволяє отримувати персональні рекомендації, які динамічно змінюються з часом.

Таким чином, виконана робота має наукову і практичну цінність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Николенко, С. Глубокое обучение [Текст] / С. Николенко, А. Кадурын, Е. Архангельская. – СПб.: Питер, 2018. – 480 с.

УДК 004.794

Льовкін В.М.¹, Лукашенко Ю.О.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-115 ЗНТУ

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОШУКУ ДРУЗІВ ДЛЯ СПІЛЬНОГО ВІДВІДУВАННЯ ПОДІЙ

Значне різноманіття напрямів активного відпочинку у сучасному світі, як-от концертів, фестивалів, форумів, спортивних змагань, та їхніх підвидів призводить до виникнення потреби пошуку людей, які розділятимуть ваші вподобання. Адже спільне відвідування подій, коли кожен з учасників виявляє максимальну зацікавленість процесом, приносить більше задоволення, ніж відвідування подій поодиночі.

Ідея пошуку друзів для спільного відвідування подій була реалізована у програмному забезпеченні «Co-Visiting», що представляє собою тематичний сайт, який надає наступну функціональність:

- реєстрація, авторизація користувачів;
- перегляд категорій подій;
- створення власної події із зазначенням назви, типу, опису, місця та часу її проведення, а також міста відправлення і способу пересування до місця події за умови перебування за межами міста, де відбуватиметься подія;
- коментування подій з можливістю налаштування приватності;
- перегляд профілю користувача;
- можливість вподобати події, збільшуючи рейтинг події та користувача, який її опублікував;
- глобальний пошук подій за ключовими словами та пошук у межах певних категорій.

Реалізація зазначених функцій передбачала створення класів, які інкапсулюють у собі бізнес-логіку програми, а також взаємодію з базою даних. Тож для розроблення програмного забезпечення було використано такий інструментарій:

- об'єктно-орієнтована мова програмування C#;
- платформа ASP.NET Core від Microsoft, призначена для створення різного роду веб-додатків[1]. Фактично вона є продовженням розвитку ASP.NET Framework, але надає можливість кросплатформності;
- Razor – спосіб рендерингу представлень, що має певний синтаксис переходу від даних об'єктів класів, написаних на C# до HTML-розмітки;
- система керування базами даних MSSQL з використанням Entity Framework для легкої взаємодії між об'єктами та таблицями і зв'язками бази даних шляхом застосування підходу «Code First»;
- мова стилів CSS з використанням бібліотеки Bootstrap.

У роботі використано архітектурний шаблон модель-представлення-контролер (MVC). Цей шаблон застосовується для відокремлення даних (моделі) від інтерфейсу користувача (представлення) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача[2].

Зокрема, інтерфейс домашньої сторінки (рис. 1) є представленням, дані для якого за замовченням витягуються з відповідної моделі, а взаємодія з інтерфейсом викликає спрацювання методів з відповідних контроллерів, – наприклад, пошуку і переходу на сторінку з обраною подією.

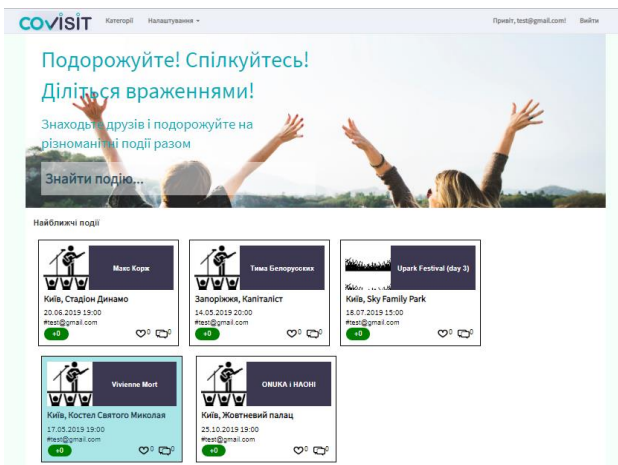


Рисунок 1 – Представлення домашньої сторінки програмного забезпечення пошуку друзів для спільного відвідування подій

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ASP.NET Core – новая эпоха в развитии ASP.NET [Електронний ресурс]. – Режим доступу : <https://metanit.com/sharp/aspnet5/1.1.php>
2. Freeman, A. Pro ASP.NET MVC 5 [Текст] / A. Freeman. – Apress, 2013. – 812 р.

УДК 004.794

Льовкін В.М.¹, Радєв П.А.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-136сп ЗНТУ

РИЗИК-ОРІЄНТОВАНИЙ ПІДХІД ДО РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ КЕРУВАННЯ ТЕПЛИЧНИМИ КОМПЛЕКСАМИ

Одним з очевидних світових трендів останніх років є «інтернет речей» (IoT), за якого безліч різноманітних пристроїв різного розміру та призначення продукують величезну кількість інформації про навколишнє середовище, що при доцільному використанні може приносити велику користь та автоматизувати процеси, задля яких раніше вимагалися значні витрати робочої сили, таким чином приносячи економічний ефект [1]. Однією з багатьох сфер людської діяльності, у яких доцільно застосовувати подібні технології, є сільське господарство. Наприклад, детальний моніторинг параметрів оточуючого середовища може попередити велику кількість людських помилок, що стаються через брак інформації.

Новизна розробленої системи полягає в тому, що на відміну від відомих агрегаторів даних IoT вона спрямована не тільки на збір даних, а й їх зворотнє відправлення (наприклад, керуючих сигналів), а також в тому, що на відміну від вже згаданих агрегаторів мікрокомп'ютери (МК) мають змогу самостійно реєструвати себе у системі, без необхідності втручання користувача у цей процес.

Користувальська частина додатку є браузерно-орієнтованою, отже може бути однаково успішно запущена на будь-якому пристрої. Інтерфейс панелі керування є адаптивним відносно розмірів екрану та інтуїтивно зрозумілим (рис. 1, 2). Головними принципами розробки були наступні:

- серверний додаток повинен працювати з різними видами МК (за умови, що інформація, яка від них надходить, відповідає певним вимогам);
- серверний додаток повинен працювати з великою кількістю пристроїв та запитів без впливу на загальну швидкість роботи;

– серверний додаток та база даних повинні бути легко переносними між пристроями та платформами без змін в архітектурі, щоб у випадку відмови сервера мати змогу перейти на резервний без значних втрат у часі та швидкості;

– додаток для МК має бути відмовостійким настільки, щоб у випадках відсутності інтернету не втрачати дані, а у випадку відсутності енергопостачання мати змогу самостійно розпочати роботу;

– додаток для МК мусить мати легко розширювану архітектуру, яка б дозволяла без складностей додавати нові датчики чи керуємі елементи;

– веб-сайт мусить мати інтуїтивно зрозумілий інтерфейс і надавати персоналу доступ до потрібної інформації без попереднього навчання.

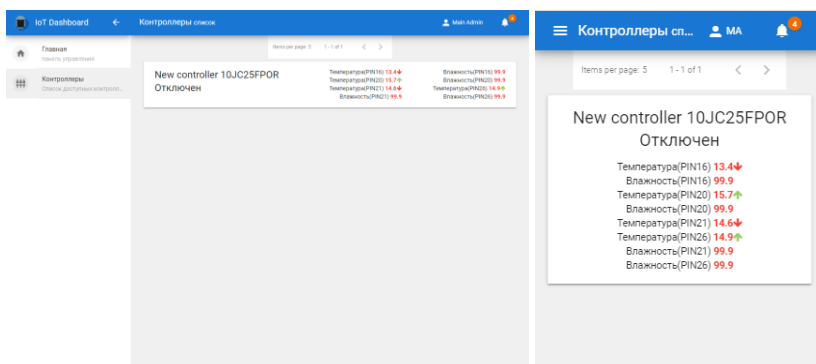


Рисунок 1 – Демонстрація інтерфейсу списку МК

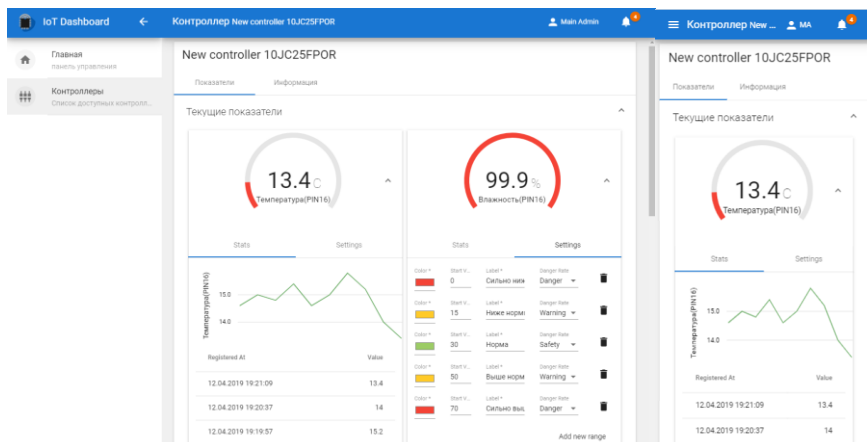


Рисунок 2 – Демонстрація інтерфейсу деталей МК

У різних частинах система використовує такі технології як WebApi2 (C#, сервер), UWP (C#, МК), Prism (C#, МК), SignalR (C#/JS, сервер/МК/клієнт), Entity Framework Core (C#, сервер/МК), Angular (JS, клієнт).

Прийняті протягом всього життєвого циклу рішення дозволили управляти ризиками проекту та досягти запланованого результату в короткі строки з обмеженими ресурсами, а отримана у результаті система виявилася придатною на практиці за результатами тестування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kranz, M. Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry [Текст] / M. Kranz. – Wiley, 2016. – 272 p.

УДК 004.794

Льовкін В.М.¹, Федічкін І.О.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-215 ЗНТУ

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ ДІЯЛЬНІСТЮ СТАНЦІЇ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

Під час ведення сучасного бізнесу повинна виконуватись велика кількість процесів, деякі з яких можливо автоматизувати. У роботі станції технічного обслуговування серед таких можна виділити наступні процеси:

- ведення складського обліку, що включає в себе забезпечення складу новими запчастинами та товарами для продажу, контакт з постачальниками товарів, моніторинг цін та курсу валют та інші;
- облік послуг станції, продаж товарів, відносини з клієнтами, запис клієнтської інформації, ведення відповідної документації;
- аналіз діяльності станції в цілому.

Для забезпечення автоматизації даних процесів було розроблено програмне забезпечення керування діяльністю станції технічного обслуговування. Серед аналогів даного програмного забезпечення можна виділити системи 1С, IT-Enterprise тощо, проте розроблений продукт має низку переваг, серед яких:

а) програмне забезпечення розроблено у відповідності з вимогами українського ринку, відповідно всі документи налаштовані для клієнтів з України;

б) інтерфейс програми представляє собою веб-додаток, тобто немає необхідності завантажувати його на пристрої, для роботи системи потрібен

лише браузер, наприклад, Chrome, а також немає залежності від самого пристрою, це може бути, як домашній комп'ютер, ноутбук, так і смартфон;

в) дані системи будуть знаходитись на віддаленому сервері, а доступ до них буде мати тільки клієнт через логін та пароль, для захисту каналів буде використовуватись протокол HTTPS, який шифрує дані під час передачі від клієнту до серверу і навпаки, що забезпечить належний захист;

г) програмне забезпечення розроблено цілеспрямовано для станцій технічного обслуговування, через це налаштування системи під клієнта є тривіальним процесом, а інтерфейс ергономічно підходить для виконання задач станції.

Для створення серверної частини програмного забезпечення використовувалась мова програмування PHP версії 7.2, яка має достатньо високу швидкодію та придатну архітектуру. Операційна система серверу – Ubuntu 16.04, заснована на Debian GNU/Linux, вона має вільну ліцензію та зарекомендувала себе серед інших Linux дистрибутивів. Для зберігання інформації використовується реляційна система керування базами даних MySQL [1], файли конфігурації ini, а також файли формату JSON.

В розробці використовувався об'єктно-орієнтований підхід [2], архітектурний шаблон модель-вигляд-контролер (MVC) та шаблон проектування Singleton.

Для створення клієнтської частини були застосовані: мова розмітки HTML, каскадні таблиці стилів CSS та мова програмування JavaScript.

Клієнтська частина створена у відповідності з останніми технологіями в веб-програмуванні, через це залежна від останніх версій браузерів, де ці технології підтримуються, вона має адаптивний веб-дизайн, тобто відображається коректно незалежно від розмірів екрану. Дизайн програми зроблений в мінімалістичному стилі (рис. 1).

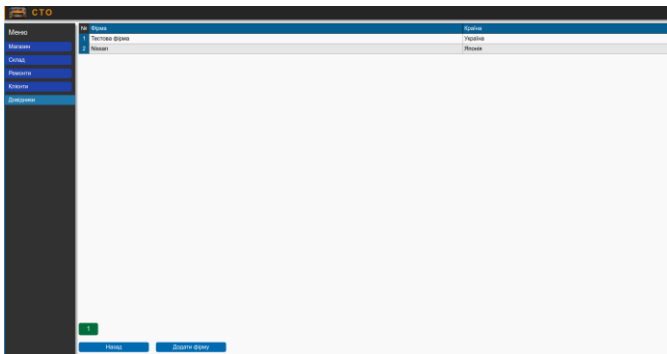


Рисунок 1 – Вікно перегляду довідника “виробники”

Таким чином, розроблене програмне забезпечення дозволяє виконувати основні функції, пов'язані з діяльністю станції технічного обслуговування, спрощуючи даний процес за допомогою використання пристроїв будь-якого типу та застосування сучасних технологій веб-програмування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MySQL 5.5 Reference Manual [Електронний ресурс]. – Режим доступу : <https://dev.mysql.com/doc/refman/5.5/en/>

2. Meyer, B. Touch of Class: Learning to Program Well with Objects and Contracts [Текст] / Bertrand Meyer. – Springer, 2009. – 920 p.

УДК 004.031

Гунько Е.В.¹, Гладкова О.М.²

¹ студ. гр. КНТ-115 ЗНТУ

² канд. техн. наук, старш. викл. ЗНТУ

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ СВІТЛОФОРМ

Перехрестя представляють одне з найнебезпечніших місць на дорозі, а також ще одна з багатьох причина заторів, затримки та ускладненого руху в містах, на дорогах. З причини об'єктивного збільшення кількості автотранспорту на дорогах, в багатьох розвинених країнах залишилося дуже мало безсвітлофорних перехресть. Некоректно складені алгоритми роботи світлофорних об'єктів на деяких вулицях часто заважають або створюють незручності та небезпеку як, в першу чергу, водіям так і пішоходам. Інколи буває що не оптимізоване перемикання сигналів світлофору надає можливість проїхати не всьому потокові транспортних засобів які цього потребують, а лише деякій його частині, в той же час з інших напрямків руху в цей час може не бути автомобілів зовсім.

Оскільки останнім часом гострота цієї проблеми назріває, а подекуди і перезріває в деяких містах нашої країни. Наприклад, у Києві, Львові, Одесі, Харкові, Дніпрі, це тільки загально відомі міста з транспортними проблемами, цей список не є вичерпний. У Львові та Києві поступово впроваджується автоматизована система керування дорожнім рухом. Таким чином в нашій країні треба звернути увагу на вирішення цих проблем та дослідити кращий закордонний досвід.

Посилаючись на [1] можна виокремити пункти, які показують алгоритми взаємодії авто зі світлофором. Таким чином ми повинні дотримуватися умов експлуатації.

Результати дослідження J'son & Partners Consulting ринку M2M/IoT в області інтелектуальної транспортної інфраструктури [2] показали, що у екосистему «розумних доріг» включають рішення для збору та обробки даних про транспортні засоби та дорожній інфраструктурі з метою прийняття рішень, а саме:

- детектори транспортного потоку;
- адаптивні світлофори;
- засоби автоматичної фіксації порушень дорожнього руху;
- електронні засоби невинної оплати проїзду;
- підключені інформаційні табло;
- системи автоматизованого управління освітленням;
- інші підключені об'єкти (наприклад, дорожні метеостанції);
- системи GPS/ГЛОНАСС.

В якості детекторів транспортного потоку можуть бути пристрої: індукційні датчики; датчика руху; датчики ваги; відеокамери.

Індуктивний датчик є розповсюдженим при будівництві інтелектуальних доріг, оскільки є простішими та дешевшими порівняно з відеокамерами. Індуктивний 3-провідний NPN датчик приближення призначений для визначення наближення до датчика металевих предметів. Працює датчик від напруги постійного струму від 6В до 36В. Вихід - нормально розімкнутий. Максимальний комутований датчиком струм становить 300 мА. [3]

Керуючим ядром системи керування світлофором була обрана платформа Arduino Mega 2560, оскільки вона швидко розробити проект та має достатню кількість пінів для підключення всього необхідного обладнання. Єдиним недоліком платформи Arduino є те що вона не підтримує багатозадачність, таким чином треба було обмежувати час виводу інформації на дисплей. Для виводу інформації про роботу логіки програми було прийнято рішення використовувати LCD дисплей LM5SP4. Дисплей працює за підтримки бібліотеки LM5SP4.h.

Принцип роботи розробленої інтелектуальної системи заснований на індуктивності звичайної котушки, яку і представляє з себе ця петля - є якась індуктивність, яка наводиться в петлі в режимі «очікування». При під'їзді автомобіля індуктивність змінюється, це зміна обробляється електронікою і посилається команда блоку управління, що необхідно перемкнути сигнал світлофора. На даний час система знаходиться на етапі тестування.

Таким чином була розроблена інтелектуальна система керування світлофором, яка може бути впроваджена на ділянках доріг, для оптимізації дорожнього руху на складних ділянках транспортного руху.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ДСТУ 4092-2002. Світлофори дорожні. Загальні технічні вимоги, правила застосовування та вимоги безпеки. – Введ. 2003-01-01. – Національний стандарт України, 2002. – 27с.
2. Cnews. J&P: Интеллектуальная транспортная инфраструктура становятся одним из самых значимых элементов интернета вещей. [Електронний ресурс]. – Режим доступу : http://www.cnews.ru/news/line/2016-03-21_jp_intellektualnaya_transportnaya_infrastruktura#
3. Arduino. Индуктивный датчик приближения. [Електронний ресурс]. – Режим доступу: <https://arduino.ua/prod1426-indyktivnii-datchik-priblijeniya-lj12a3-4-zbx>.

УДК 004.891

Куш А.В.¹, Пархоменко А.В.², Гладкова О.М.³

¹ студ. гр. КНТ-715 ЗНТУ

² канд. техн. наук, доц. ЗНТУ

³ канд. техн. наук, старш. викл. ЗНТУ

РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ БАЗИ ЗНАНЬ МЕДИЧНОГО ОБЛАДНАННЯ

Використання рекомендаційних систем в медицині сприяє її стрімкому розвитку та дозволить спростити ряд робочих процесів, підвищити їхню ефективність при наданні медичної допомоги. На сьогодні рекомендаційні алгоритми набули широкого розповсюдження та використовуються в інтернет-магазинах з продажу, різних хостингах з книгами, фільмами та музикою.

Впровадження рекомендаційних систем для вибору медичного обладнання дозволить швидко та ефективно знаходити необхідне обладнання не витрачаючи багато часу на пошуки та перегляд тисячі існуючих медичних виробів. Етапи розробки рекомендаційної системи умовно можна розділити на: проектування та створення сайту магазину, розробка бази знань (БЗ) медичного обладнання та створення рекомендаційного алгоритму.

На сьогоднішній день розроблено простий та лаконічний дизайн сайту. Реалізовано можливість авторизації та реєстрації. Обрано рекомендаційний алгоритм оснований на знаннях, в основі якого є робота з базою знань, яка призначена для керування знаннями. В даному випадку під знаннями будемо розуміти закономірності предметної області, принципи чи зв'язки, висновки, що отримані в результаті практичної діяльності та власного досвіду. Перед розробкою БЗ була створена семантична мережа, як основа моделі

представлення знань, з метою виділення основних критеріїв-правил, що є спільними для всіх виробів медичного призначення.

Особливість семантичної мережі полягає в єдиній базі знань та механізмі виводу нових фактів. Так на основі запиту будується семантична мережа, що відображає структуру питання. Таким чином відповідь формується на основі зіставлення загальної мережі бази знань та мережі отриманої на основі питання.

Використання семантичної мережі для побудови бази знань медичного обладнання дозволило досягти зрозумілого способу представлення знань на основі відношень між вершинами та дугами мережі, дуже близькому до природної мови. Ще однією перевагою при використанні цієї мережі була можливість виділення з усієї мережі, що представляє знання, невеликої ділянки семантичної мережі, що є важливим для конкретного запиту.

Таким чином, використання семантичної мережі дозволило відокремити головні критерії медичних виробів для побудови БЗ та наглядно структурувати та представити існуючі знання про медичне обладнання.

УДК 004.42

Попович В.В.¹, Каплієнко Т.І.²

¹ студ. гр. КНТ-136сп ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ТА АДМІНІСТРАТИВНОЇ ЧАСТИН ВЕБ-САЙТУ ДЛЯ ПРОДАЖУ РЕКВІЗИТІВ ДЛЯ ФОКУСІВ

На сьогоднішній день розробка інтернет-магазину та його використання є актуальним питанням, адже дуже багато людей кожного дня в Україні та й в цілому у всьому світі, не виходячи з домівки, купують товари через мережу інтернет. Більшість людей користуються мережею інтернет та щодня їх кількість лише росте, а отже і росте кількість користувачів, що купують в інтернет-магазинах.

Інтернет-магазин зменшує витрати продавця, заощаджуючи на утриманні звичайного магазину, також значно збільшуються межі продажу товару. Значними плюсами використання інтернет-магазину для покупця є можливість замовити товар в будь-який зручний час, цілодобово, без вихідних. Немає прив'язаності до місця замовлення, адже покупець може знаходитись в будь-якій точці планети. В цьому і є значні переваги інтернет-магазинів від звичайних.

Для організації продажу реквізитів для фокусів через мережу інтернет необхідно розглянути основні вимоги до інтернет-магазину. Основною

метою розробки даного інтернет-магазину є необхідність збільшити швидкість продажу товарів, а саме реквізитів для фокусів, а також зменшити час для придбання товарів покупцем.

Для досягнення цієї мети пропонується розробити клієнтську та адміністративну частини.

Клієнтська частина повинна мати такі функції: пошук товарів по інтернет-магазину, перегляд товарів по категоріям, реалізована «хлібна крихта», перегляд товарів в табличному та лінійному вигляді, фільтрація товарів по параметрам, сортування товарів за параметрами, перегляд повної інформації про кожний товар, авторизація покупця, реєстрація покупця, додавання товарів до кошика, редагування кошика, видалення товарів з кошику, оформлення замовлення, відображення інформерів, відображення новин, сповіщення покупця по електронній адресі про вдале замовлення з повним описом замовлення, сповіщення продавця по електронній адресі про нове замовлення з повним його описом, посторінкова навігація.

Адміністративна частина повинна мати такі функції: авторизація за правильним логіном та паролем для входу до адміністративної панелі лише адміністраторів, відображення нових замовлень, перегляд повного списку замовлень, перегляд детальної інформації замовлень, підтвердження, або видалення замовлення, посторінкова навігація, відображення товарів, переміщення товарів, додавання нових товарів, видалення товарів, відображення основних сторінок інтернет-магазину, переміщення основних сторінок, додавання нових основних сторінок та заповнення за допомогою влаштованого редактору, редагування основних сторінок, видалення не потрібних основних сторінок, відображення інформерів, переміщення інформерів, додавання нових інформерів, редагування інформерів, видалення непотрібних інформерів, відображення категорій та підкатегорій, переміщення категорій та підкатегорій, додавання нових категорій та підкатегорій, видалення не потрібних категорій та підкатегорій, відображення новин, додавання нових новин, редагування новин, видалення не актуальних чи не потрібних новин, відображення користувачів та адміністраторів інтернет-магазину, додавання нових клієнтів та адміністраторів, видалення клієнтів та адміністраторів інтернет магазину, редагування даних користувачів.

Необхідно приділити увагу й інтерфейсу, який повинен бути зручним у використанні та інтуїтивно зрозумілим для користувача. Також необхідно подумати й про безпеку і більш всього про те, щоб до адміністративної частини могли потрапити лише адміністратори за своїм логіном та паролем.

Інтернет-магазин вирішено розробити за допомогою мови програмування Hypertext Preprocessor, мови структурних запитів Structured query language, мультипарадигменої мови програмування JavaScript,

каскадних таблиць стилів Cascading Style Sheets, мови гіпертекстової розмітки Hyper Text Markup Language та технології AJAX.

Розроблений, ґрунтуючись на описаних вище вимогах, інтернет-магазин є дуже зручним у використанні та зрозумілим кожному, хто користується мережею інтернет. Інтернет-магазин дає можливість здійснити купівлю реквізитів для фокусів, не виходячи з дому, в будь-який зручний час цілодобово, без вихідних.

В роботі було розглянуто основні вимоги для організації продажу реквізитів для фокусів через мережу інтернет в інтернет-магазині. Було запропоновано рішення щодо автоматизації продажу реквізитів для фокусів через мережу інтернет. На даний момент інтернет-магазин знаходиться в процесі розробки та найближчим часом може бути впроваджено в експлуатацію.

УДК 004.42

Симоненко О.С.¹, Каплієнко Т.І.²

¹ студ. гр. КНТ-136сп ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-САЙТУ ДЛЯ ПРОДАЖУ ГОДИННИКІВ НА ОСНОВІ ВЛАСНОГО ФРЕЙМВОРКУ

На сьогоднішній день комп'ютеризація розвивається дуже швидко та відіграє значну роль у суспільстві. Комп'ютерні технології дозволяють автоматизувати багато процесів, які в недавньому минулому покладалися на людину. Інформаційні технології використовуються у різних сферах життєдіяльності.

З розвитком Інтернету популярність здобули сайти. Використання сайтів – це швидкий спосіб здобуття необхідної інформації. На даний час існує багато розроблених веб-сайтів, але технології не стоять на місці, вони дуже швидко розвиваються. Після порівняння існуючих рішень, було зроблено висновок, що актуальним завданням є розробка веб-сайту для продажу годинників з долученням нових технологій, а саме програмна реалізація власного фреймворку.

Для досягнення поставленої мети система повинна забезпечувати:

- відображення даних про товари з їх зображенням;
- додавання товарів у корзину, задля їх покупки;
- видалення товару з корзини;
- зручний пошук товарів;
- обчислення вартості продажу;

- можливість вибору валюти, та перерахунок вартості покупки за обраною валютою;
- можливість створення облікового запису для користувача та адміністратора магазину;
- для створення облікових записів використовувати логін та пароль, дані паролю повинні бути захищені, тому використовувати шифрування;
- можливість перегляду основних даних адміністратором магазину з адміністративної панелі;
- можливість адміністратору магазину в адміністративній панелі додавати та видаляти необхідні данні, а саме: продукти, категорії, користувачів, кеш данні, валюти та фільтри;
- можливість адміністратору магазину в адміністративній панелі керування замовленням, а саме видалення чи виконання.

Для розробки інтернет-магазину вирішено використовувати технології HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, Hypertext Preprocessor (PHP), Bootstrap, JavaScript, jQuery, AJAX.

Завдяки використанню адаптивної верстки веб-сторінки будуть добре відтворюватись на всіх пристроях.

Графічний інтерфейс інтернет-магазину інтуїтивно-зрозумілий та легкий у використанні (рис. 1). Усі покупки відображаються у корзині (рис.2). В ній можна побачити придбані товари, вартість кожного товару та загальну вартість, а також виконати такі функції: продовжити покупки, оформити замовлення, очистити корзину.

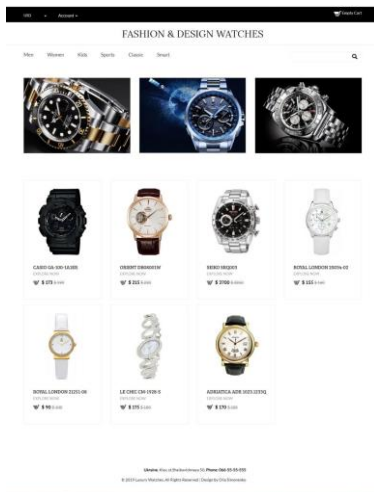


Рисунок 1 – Головна сторінка інтернет-магазину

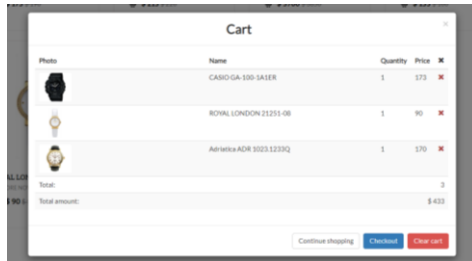


Рисунок 2 – Вигляд корзини покупок

В роботі було розглянуто актуальність розробки інтернет-сайту для продажу годинників. Було запропоновано рішення щодо розробки інтернет-магазину. На даний момент програмний продукт знаходиться в процесі розробки та найближчим часом буде доступний для використання.

УДК 004.414

Волинська А.В.¹, Зайко Т.А.²

¹ студ. гр. КНТ-114м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ВИКОРИСТАННЯ КОНЦЕПЦІЇ ЕКСТРАГУВАННЯ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка якісних програмних продуктів стримується якістю роботи програмістів і команд програмістів, що беруть участь в створенні цих продуктів. Одним з інструментів вдосконалення створення і розвитку програмних продуктів на всіх етапах життєвого циклу можуть бути методи, розвинені в теорії вирішення винахідницьких завдань (ТВВЗ).

Одним із методів, який можна використовувати при розробці програмного забезпечення і який заслуговує на увагу є метод екстрагування. Метод "Екстрагування" вперше введений в практику в Модерн ТВВЗ і призначений для прискорення вивчення моделей трансформації (прийомів).

Екстрагування - це виявлення (всіх) моделей трансформації, об'єктивно присутніх в артефакті в порівнянні з реальним прототипом, або з об'єктом, що обирається в якості такого прототипу, оскільки реальний прототип може бути не відомий, забутий (в історії) або неточно описаний.

Загальний опис процесу екстрагування полягає в наступному:

1. Береться будь-який артефакт і підбирається реальний або уявний (віртуальний) прототип для цього артефакту, тобто передбачається, що прототип був попередником артефакту в історії його конструктивного

розвитку; в ідеалі прототип має ту ж головну корисну функцію (призначення, мета застосування), що і артефакт.

2. Потім розглядаються позитивні зміни, наявні в артефакті, в порівнянні з прототипом.

3. Виявлені зміни мають, зрозуміло, конструкційний характер. Однак сутність екстрагування полягає в тому, щоб в кожному виявленому конструкційному зміні визначити креативну модель трансформації, об'єктивно присутню в цій зміні.

Розрізняють два основні варіанти екстрагування:

1. Екстрагування-1 – виявлення всіх об'єктивних моделей трансформації, повністю або частково присутніх в переході від прототипу до артефакту.

2. Екстрагування-2 – виявлення домінуючих трансформацій, необхідних для переходу від прототипу до артефакту і пов'язуються з певними усуненими протиріччями.

Розробка програмного забезпечення завжди пов'язана із винахідництвом, тому методи ТВВЗ можуть стати у нагоді будь-якому програмісту. Метод ТВВЗ широко використовується для створення і розвитку програмних продуктів на всіх етапах життєвого циклу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Орлов М. Азбука современной ТРИЗ. Настольная книга для изобретательного мышления [Текст] / М. Орлов. – Москва : Издательство АСТ, 2017. – 495 с.

2. Орлов М. Практическое руководство для изобретательного мышления [Текст] / М. Орлов. – Москва: Солон-Пресс, 2006. – 432 с.

УДК 517.977.58

Трубічина Д. І.¹, Зайко Т.А.²

¹ студ. гр. КНТ-114м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

АЛГОРИТМИ ТА ПРИНЦИПИ ВИРІШЕННЯ ЗАДАЧ ЗА ДОПОМОГОЮ ТВВЗ

Часто перед нами постають інтелектуальні задачі, які надзвичайно важко або взагалі, на перший погляд, неможливо вирішити. Тому, для розв'язання таких інтелектуальних завдань може бути застосована теорія вирішення винахідницьких задач, яка створена у 1946 р. ученим-дослідником, письменником-фантастом Генріхом Альтшуллером.

Автор зумів виявити і запропонувати для практичного використання алгоритми вирішення завдань, конкретні прийоми дозволу технічних протиріч, завдяки яким стало можливим економити час і кошти, розглядаючи проблеми через призму вже накопичених знань і відкриттів.

Теорія вирішення винахідницьких задач (ТВВЗ) – наука, яка дозволяє не тільки виявляти і вирішувати творчі завдання у будь-якій області знань, а й розвивати творче (винахідницьке) мислення та розвивати якості творчої особистості.

Алгоритм рішення винахідницьких задач (АРВЗ) є, мабуть, найпопулярнішим і дієвим елементом теорії Альтшуллера. Алгоритми являють собою докладний і досить трудомісткий опис послідовності винахідницького процесу, які може взяти на озброєння кожна людина, чия діяльність пов'язана з творчістю.

Довгий час єдиним інструментом вирішення творчих завдань – задач, які не мають ефективних механізмів вирішення, – був "метод спроб і помилок". На початку 20 століття різко зросла потреба в регулярному вирішенні таких творчих завдань, що призвело до появи численних модифікацій існуючого методу. Найбільш відомими з них стали: "мозковий штурм", "синектика", "морфологічний аналіз", "метод контрольних питань", "метод каталогу".

Суть всіх цих методів – підвищення інтенсивності генерації ідей і перебору варіантів. Але існує і протиріччя – можна заощадити час на генерацію ідей, але витратити його ще більше на аналіз отриманих варіантів і вибір найкращого.

Справитися з цим завданням дозволяють три принципи, що лежать в основі ТРВЗ.

1. Принцип об'єктивності законів розвитку систем – будова, функціонування і зміна поколінь систем, що підкоряються об'єктивним законам.

2. Принцип суперечності – під впливом зовнішніх і внутрішніх факторів виникають загострюються і вирішуються протиріччя. Проблема важка тому, що існує система суперечностей – прихованих або явних. Системи еволюціонують, долаючи протиріччя.

3. Принцип конкретності – кожен клас систем, як і окремі представники всередині цього класу, мають особливості, які полегшують або ускладнюють зміну конкретної системи.

АРВЗ включає три основні компоненти:

– Програму – послідовність операцій по виявленню і вирішенню протиріч, аналізу вихідної ситуації і вибору завдання для вирішення, синтезу, розвитку отриманих рішень, накопичення найкращих рішень і узагальнення матеріалів;

– Інформаційне забезпечення базується на інформаційному фонді, який включає: систему стандартів на рішення, технологічні ефекти, прийоми усунення протиріч та способи застосування ресурсів природи і техніки;

– Методи управління психологічними факторами необхідні внаслідок того, що програма АРВЗ призначена не для машин, а завдання вирішуються людиною, тому може виникати психологічна інерція, якою необхідно вміти управляти.

Висновок: за допомогою ТРВЗ вирішуються 2 типи завдань. Відомі (стандартні) типи винахідницьких завдань вирішуються з використанням інформаційного фонду, а невідомі (нестандартні) – з застосуванням АРВЗ.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Альтшуллер Г.С. Найти идею. Введение в ТРИЗ - теорию решения изобретательских задач [Текст] / Г. С. Альтшуллер. – Альпина Паблишер, 2019. – 402 с.

2. Меерович М.И. Теории решения изобретательских задач [Текст] / М.И. Меерович, Л.И. Шрагина. – Минск: Харвест, 2003. – 428 с.

УДК 303.7

Суха К.С.¹, Зайко Т.А.²

¹ студ. гр. КНТ-114м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ПРИНЦИПИ БОРОТЬБИ З ПРОТИРІЧЧЯМИ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЙ ТВВЗ

При створенні програмного забезпечення (ПЗ) розробники стикаються зустрічаються з безліччю проблем, завдань та протиріч, які необхідно швидко і ефективно вирішити. На кожному етапі життєвого циклу програмних продуктів необхідна якісна робота програмістів, що беруть участь в створенні цих продуктів. Не завжди виходить знайти швидке рішення проблем розробки ПЗ, складності перенесення та непомічені помилки досить сильно уповільнюють прогрес, в свою чергу це породжує відставання від графіка, перевищення кошторису тощо.

Одним із способів вдосконалення процесів створення та розвитку програмних продуктів можуть бути методи, розвинені в теорії вирішення винахідницьких задач (ТВВЗ). Технології ТВВЗ створено для знаходження ідеального кінцевого результату при виникненні технічних протиріч, коли спроба поліпшити одну характеристику технічної системи провокує погіршення іншої. Для програмних продуктів інструменти ТВВЗ також

можуть бути застосовані, тому що вони є технічними системами передачі й обробки інформації [1].

На думку Генріха Альтшулера, основоположника ТБВЗ, можливо було виявити з досвіду попередників часто повторювані прийоми успішних винаходів. Для цього було проведено аналіз близько сорока тисяч авторських свідоцтв та патентів, і за допомогою виявлених закономірностей розвитку технічних систем і прийомів винахідництва розробили ТБВЗ. Згодом Альтшулер довів, що в сутності більшості винахідницьких задач лежить відносно мала кількість протиріч між універсалізованими характеристиками технічних систем. Тобто існують типові технічні протиріччя, такі як «вага – міцність», «точність – продуктивність» [2].

Вивчивши багато винаходів вищого класу, Альтшулер зауважив, що протиріччя в таких завданнях усувалися особливими прийомами – способами зміни вихідної системи. На основі отриманих результатів Г. С. Альтшулер склав таблицю застосування цих прийомів, що складається з сорока прийомів, які залежать від типу протиріч, в якій по вертикалі розташовуються характеристики технічних систем, які за умовами завдання необхідно поліпшити, а по горизонталі – характеристики, які при цьому неприпустимо погіршуються. На перетині граф таблиці вказані номери прийомів, які з найбільшою ймовірністю можуть усунути виникле технічне протиріччя [3]. І хоча адаптація таблиці протиріч вимагає і уточнення формулювань принципів таким чином, щоб вони легко сприймалися програмістами, це зовсім не заважає її активному використанню вже сьогодні. Так, «принцип посередника» для використання у сфері розробки ПЗ слід переформулювати як «принцип віртуалізації», «принцип дроблення» – як «принцип підвищення модульності», а «принцип винесення» – як «принцип виділення функціональності». Розглянемо декілька технічних протиріч, які виникають у сфері розробки ПЗ і відшукаємо способи їх усунення, використовуючи таблицю протиріч.

Одне з типових протиріч, при використанні в якості мови розробки JavaScript, можна визначити наступним чином: ЯКЩО ізолювати контекст виконання додатків, ТО можна істотно підвищити безпеку корпоративних середовищ, АЛЕ при цьому виникнуть втрати в продуктивності. Згідно с таблицею застосування прийомів вирішення протиріч визначаємо відповідну пару: зміна (30, «Шкідливі фактори, що діють на об'єкт») – погіршення (39, Продуктивність). З набору прийомів, які пропонуються в якості варіантів рішення (22, 35, 13, 24) може бути обрано 24, «Принцип посередника», згідно з яким, слід на час приєднати до об'єкту інший (легко видаляємий) об'єкт. У даному випадку таким легко видаляємим об'єктом буде слугувати комплекс апаратних засобів підтримки віртуалізації, впровадження якого призведе до усунення протиріччя.

Інше протиріччя, що виникає при тестуванні інструменту розробки веб-інтерфейсу, полягає в тому, що ЯКІЩО тестована програма знає структуру веб-сторінки, ТО можна виконати перевірку, АЛЕ при цьому тестування стає залежним від веб-інтерфейсу, який постійно змінюється. Згідно з таблицею застосування прийомів вирішення протиріч можна визначити відповідну пару: зміна (35, Адаптація, універсальність) – погіршення (30, Шкідливі фактори, що діють на об'єкт). З набору прийомів (35, 11, 32, 31) можна обрати прийом 11, «Принцип заздалегідь підкладеної подушки», реалізація якого буде полягати в використанні заздалегідь ідентифікаційних полів для різних елементів сторінки.

Таким чином, можна зробити висновок, що використання принципів ТВВЗ для усунення технічних протиріч у сфері розробки ПЗ є ефективним та активно запровадженим рішенням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Підвищення ефективності розробки програмних продуктів на основі методів ТВВЗ [Текст]: матеріали наук.-метод. конф. «ТВВЗ-фест 2009», Санкт-Петербург, 27-29 липня 2009 р.: збірник праць / [редкол.: А.В. Єфімов (відпов. ред.) та ін.]. – СПб.: СПбПУ, 2009. – 302 с.
2. Одинцов І.О. Професійне програмування. Системний підхід [Текст] / І.О. Одинцов ; 2-е вид. перероб. і доп. – СПб.: БХВ, 2004. – 624 с.
3. Альтшулер Г.С. Алгоритм винаходу [Текст] / Г.С. Альтшулер ; 2-е вид. перероб. і доп. – М.: Московський робочий, 1973. – 296 с.

УДК 001.894.04

Гринченко П.В.¹, Зайко Т.А.²

¹ студ. гр. КНТ-114 ЗНТУ

² канд. техн. наук, доцент ЗНТУ

ФУНКЦІОНАЛЬНЕ РОЗГОРТАННЯ ТЕХНІЧНОЇ СИСТЕМИ НА ОСНОВІ ТВВЗ

Необхідність збільшення числа функцій програмного забезпечення (ПЗ) є актуальним питанням під час конкурентної боротьби ПЗ що мають подібні один одному головні функції. У таких випадках сукупність ідей може бути отримана методами фокальних об'єктів, гірлянд асоціацій, морфологічного ящика і т.д. Подібні методи створюють випадковим чином сформовані безлічі ідей і ініціюють простий їх перебір для виділення варіантів, що мають сенс і є доцільними для кінцевого користувача.

У своєму первісному вигляді перераховані вище методики не підходять для глибокого аналізу технічних систем (ТС), оскільки орієнтовані на вузькі і спеціалізовані дослідження. Кожна з них являє собою певну модифікацію морфологічного аналізу, але при цьому має дві головні переваги – системний підхід і повноту охоплення аналізованої області, які разом мають бути перенесені на розроблювану методику, основною метою якої є відмова від операцій простого перебору і порівняння варіантів.

Розроблений метод вирішує завдання синтезу нових ТЗ на базі вже існуючої. Під синтезом мається на увазі побудова структури і функціональної моделі ТЗ по заданій головній корисній функції (з урахуванням конкретних ресурсів і обмежень).

Алгоритм роботи розробленої методики включає 7 основних етапів:

1. Генерація задач. Основний етап методики, який задає правила пошуку нових ідей на основі розгортання вихідних уявлень про об'єкт по трьох координатах: перша координата розгортає набір властивостей об'єкта аналізу від відомих властивостей через їх узагальнення до потенційних властивостям, які потім трансформуються в нові функції; друга координата розгортає властивості об'єкта від механічних властивостей до біологічних і інформаційних; третя координата доповнює другу аналізом кожного властивості в просторі і часі. Перетин осей координат будує осередки своєрідною морфологічної структури. Однак, на відміну від морфологічного ящика, елементи осей координат взаємодіють за певним алгоритмом: елементи першої осі представляють собою операнди, а елементи другої і третьої вісей служать операторами перетворення вже відомих властивостей об'єкта в потенційні властивості і нові функції. Ці функції, отримані на перетині зазначених координат, переформулюються надалі в нові задачі;

2. Виконання предметної системної і надсистемної фільтрації задач, яка враховує обмеження, що вносяться конкретним взаємозв'язком ТС, що розглядається і її найближчої надсистеми (НС). В результаті фільтрації частина знайдених завдань зі списку виключається, а частина - модифікується.

3. Етап пошуку принципів дії (ПД) для виявлення конкретних способів реалізації залишившихся задач.

4. Критеріальна фільтрація, що дозволяє знизити надмірність числа виявлених принципів дії. В результаті фільтрації формулюється набір пропозицій розробника. Фільтр є адаптивним (посилення або послаблення обмежуючої сили того чи іншого критерію) в залежності від особливостей конкретного завдання і поточних вимог замовника і ринку

5. Об'єднання пропозицій розробника з можливими пропозиціями замовника. Пропозиції, отримані в результаті об'єднання, оформлюються в концептуальні ідеї.

6. Інтегрування ідей, яке дозволяє виявити найліпші ефекти від спільного використання ідей, заснованих на різних ПД, і виключити подібні один одному ідеї з різними ПД, які мають більш низьку ефективність.

7. Обґрунтування і ранжування концепцій – завершальний етап формування концептуальних пропозицій. Задача обґрунтування – довести можливість реалізації пропозиції в заданих умовах і виправданість витрат. Ранжування, як правило, доцільно проводити за витратами (з урахуванням наявних ресурсів) і швидкістю можливої реалізації.

У багатьох випадках розроблений метод дозволить не тільки виявити конкретні нові функції об'єкта разом з конкретними способами його застосування та принципами дії, але і допоможе визначити перспективи подальшого розвитку об'єкта за виявленим функціональним напрямком.

Розроблена методика призводить до гарантованого результату у вигляді великого за кількістю набору модифікацій і концептуальних напрямів розвитку вихідного об'єкта (від приватних модифікацій вихідного об'єкта до об'єктів з іншим принципом дії і іншою головною функцією), що робить її ефективною як для продовження «життя», так і для створення нових.

УДК 004.414

Мамутов Д.Ю.¹, Зайко Т.А.²

¹ студ. гр. КНТ-114м ЗНТУ

² канд. техн. наук, доц. ЗНТУ

ПРОБЛЕМИ ПІДГОТОВКИ ТА РЕАЛІЗАЦІЇ STARTUP У СФЕРІ ІТ

Стартап як форма малого підприємництва стала популярною у світі в першу чергу через зменшення бар'єрів входу в ринок, і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Основною причиною неуспішності більшості стартапів є неефективна стратегія інноваційної діяльності підприємства, та неефективний підбір відповідних маркетингових стратегій пропагування та впровадження інновацій. Запобігти цього можна якщо чітко структурувати процеси підготовки та реалізації стартапу. Загалом необхідні етапи розробки стартапу можна подати таким чином:

1. SEED – «посівна» стадія – на даній стадії існує лише ідея чи проект, компанія знаходиться в процесі формування, відбувається створення команди, проводяться маркетингові дослідження, визначається концепція

бізнесу, відбувається початковий збір фінансів, виконується створення прототипу.

2. STARTUP («стартап») – компанія щойно утворена, намагається організувати виробництво і вихід продукції на ринок, створюється команда, проводиться аналіз конкурентів, здійснюється пошук і залучення перших клієнтів на ринку, виконується перехід від прототипу до масштабованості продукту.

3. EARLY STAGE, EARLY GROWTH («раннє зростання») – компанія здійснює випуск і комерційну реалізацію готової продукції, хоча поки не має стійкого прибутку, відбувається збільшення кількості клієнтів, виконується пошук коштів на розвиток компанії, комплектується штатний персонал компанії

4. EXPANSION («розширення») – компанія займає певні позиції на ринку, стає прибутковою, відбувається розширення виробництва і збуту, проводяться додаткові маркетингові дослідження, відбувається збільшення капіталу і основних активів. На цьому етапі відбувається переорієнтація продукту на масового користувача.

5. MEZZANINE – «проміжна» стадія – на даній стадії для поліпшення короткострокових показників компанії залучають додаткові інвестиції, що збільшує її загальну капіталізацію.

6. EXIT («вихід») – етап розвитку компанії, на якому відбувається створення публічної компанії, продаж частки інвестора іншому стратегічному інвестору, первинне розміщення на фондовому ринку (IPO) або викуп менеджментом.

Аргументом на користь даного підходу є один з яскравих прикладів стартапів, що проходив подібний цикл – соціальна мережа Facebook. На етапі SEED у засновника була лише ідея створити щось подібне до студентського «facebook». На стадії STARTUP коли Цукерберг написав код нового веб-сайту, він запустив «Thefacebook». Спочатку доступ до сайту мали лише студенти Гарвардського коледжу, і протягом першого місяця зареєструвалася більш ніж половина студентів Гарварду. На стадії EARLY STAGE, EARLY GROWTH до Цукерберга приєдналися Дастін Московіц (програміст), Едуардо Саверін (бізнес-менеджер), Ендрю Мак-Коллум (графічний дизайнер) та Кріс Хьюз, щоб допомогти у просуванні веб-сайту. Згодом він відкрився для студентів інших університетів Ліги Плюща, і поступово для більшості університетів США та Канади. На стадії EXPANSION влітку 2004 року було засновано компанію «Facebook». У червні 2004 року вона отримала першу інвестицію від співзасновника системи PayPal Пітера. В останню стадію EXIT соціальна мережа увійшла 18 травня 2012 розмістивши свої акції в рамках IPO.

На даному етапі розвитку підприємництва в сфері інформаційних технологій спостерігається значне піднесення у розвитку українських стартапів, які досить швидко розвиваються і мають багато користувачів. Однак мала зацікавленість держави в сучасних інноваційних проектах змушує молоді ІТ-компанії співпрацювати з іноземними інвесторами, які допомагають втілити проекти в життя та стати конкурентоспроможними. Саме тому, необхідно сприяти ефективному функціонуванню стартапів в Україні, створювати механізми для підвищення результативності їхньої діяльності на вітчизняному ринку та визнання значущості інноваційних технологій для успішного розвитку українських підприємств.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. От идеи к реализации: 12 этапов проектирования стартапа [Електронний ресурс]. – Режим доступу: <https://kontur.ru/articles/1493>.
2. Дрейпер. У. Стартапы: профессиональные игры Кремниевой долины [Текст] / У. Дрейпер; предисловие Э. Шмидта; пер. с англ. В. Егорова. – Москва: Эксмо, 2012. – 378 с.

УДК 004.89

Субботін С.О.¹, Хохлова В.С.²

¹ проф. ЗНТУ

² студ. гр. КНТ-415 ЗНТУ

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ МЕДИЧНОЇ ДІАГНОСТИКИ

Машинне навчання – великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. Для машинного навчання використовують різні технології та алгоритми. Зокрема, можуть застосовуватися дискримінантний аналіз, байєсовські класифікатори, штучні нейронні мережи та багато інших математичних методів [1].

Машинне навчання з кожним днем займає все більше місце в нашому житті з огляду на величезного спектру його застосувань. Починаючи від аналізу пробок і закінчуючи самоврядними автомобілями, все більше завдань перекладається на машини. Алгоритми машинного навчання можна вважати найпотужнішим інструментом, орієнтованим на використання великих обсягів даних для прогнозування та прийняття рішень [2].

Метою даної роботи є дослідження та порівняння алгоритмів машинного навчання для вирішення задачі медичної діагностики.

У процесі виконання роботи були розглянуті та реалізовані наступні методи: лінійна регресія, метод k-найближчих сусідів, метод опорних векторів, ядрова регуляризація Тихонова (Kernel ridge regression), багат шаровий перцептрон (MLP regressor), random forest (випадковий ліс), дерева рішень.

Методи порівнювалися за такими параметрами: час побудови моделі (мс), об'єм пам'яті, витрачений на побудову моделі (MiB), об'єм пам'яті, який займає модель (KiB), кількість параметрів моделі, які можна налаштувати (ваги), помилка моделі для навчальної вибірки (%), помилка моделі для тестової вибірки (%).

Усі методи були реалізовані на мові програмування Python [3].

Аналізуючи результати, можемо зробити висновок, що найгірші результати дали метод опорних векторів та ядрова регуляризація Тихонова. Хоча метод опорних векторів витрачає дуже малий об'єм пам'яті на побудову моделі, він дав найбільшу помилку при тестуванні (~10%). Ядрова регуляризація Тихонова показала помилки на окремих значеннях.

Велику кількість неправильних рішень під час використання методу опорних векторів спостерігалася на графіку залежності спрогнозованих значень від реальних.

Лінійна регресія та метод k-найближчих сусідів показують гарні та середні результати по усім параметрам, мають середній відсоток помилок (~5%). Метод k-найближчих сусідів показує відносно поганий результат при тестуванні на навчальній вибірці, але порівняння з іншими методами він дає однаковий відсоток помилок на обох вибірках, тож загалом відсоток помилок є задовільним. Графік цього методу є досить лінійними.

Дерева рішень показали гарні показники майже по усім параметрам, але зробили значний відсоток помилки на тестовій вибірці – один з найгірших (~8%).

Спінні результати показали багат шаровий перцептрон та random forest. По усім параметрам, окрім помилок моделі, вони поступаються іншим методам, але ці два метода показали найменший відсоток помилок як для навчальної вибірки, так і для тестової (~2,5%). Підтвердженням невеликої кількості помилок є лінійність графіку залежності спрогнозованих значень від реальних.

Таким чином, якщо ми маємо обмеження в часі побудови моделі та пам'яті, то найкраще буде обрати метод k-найближчих сусідів. Результати моделювання показуватимуть середній відсоток помилок (~5%).

У випадку достатніх ресурсів пам'яті та часу найкращим методом з протестованих для вирішення поставленої задачі є random forest, який показує найменший відсоток помилок серед усіх методів (~2,5%).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Machine Learning – Машинне навчання [Електронний ресурс]. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>.
2. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных [Текст] / П. Флах. – М.: ДМК Пресс, 2015. – 775с.
3. Gorelick M. High Performance Python / M. Gorelick, I. Ozsvald. – Севастополь : ДМК Пресс, 2014. – 351 с.

УДК 004.9

Табунщик Г.В.¹, Горелікова Т.О.²

¹канд. техн. наук, проф. ЗНТУ

²студ. гр. КНТ-726сп ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА ДЛЯ РОБОТИ З RASPBERRY PI

Сьогодні технології удосконалюються і поширюються по всьому світу стрімкими темпами. Варто звернути увагу на той факт, що в щоденному житті дітей оточують найрізноманітніші технічні пристосування і пристрої. Для дітей, як і для багатьох дорослих, всі ці пристрої є абсолютно невідомими об'єктами, тобто кожен знає для чого потрібно той або інший пристрій, а також як ним користуватися, але принцип роботи відомий небагатьом. Освоєння техніки дозволяє дітям ближче дізнатися про принципи роботи таких пристроїв. Але у більшості українських шкіл бракує матеріальної бази для навчання роботі на основі одноплатних комп'ютерів.

В умовах сучасної школи, коли центром освітньої діяльності є дитина, а головним завданням кожного вчителя формування всебічно розвинутої гармонічної особистості, все більш актуальним стає застосування сучасних технологій навчання, що враховують індивідуальність кожного учня[1].

Однією із таких форм є інтерактивні технології, що сприяють збагаченню і розвитку творчого потенціалу як вчителя так і учня, розкриттю здібностей дитини, бажання і вміння вчитися, творити, відчувати радість пізнання. Саме при використанні інтерактивних методів учні вчать шукати інформацію, систематизувати її та узагальнювати[2].

На даний час обчислювальні системи проникли в усі сфери життєдіяльності людини: високопродуктивні вбудовані системи використовуються в абсолютно різних областях, починаючи від повсякденного життя і закінчуючи масштабним виробництвом. Велике

значення для розвитку технологій і виробництва має одноплатний комп'ютер[3].

Навчання дітей роботі з одноплатним комп'ютером – один із кроків для розвитку технологій в майбутньому.

Основним завданням проекрованої системи є навчання учнів роботі з одноплатним комп'ютером Raspberry Pi, платою розширення та датчики.

Для створення інтерактивної навчальної системи було вивчено і досліджено роботу Raspberry Pi з зовнішніми пристроями.

Програмний продукт володіє наступними функціональними характеристиками:

- можливість ознайомитись з теоретичними відомостями;
- можливість обрати завдання;
- наочна схема з'єднання плати і датчиків;
- перегляд коду додатку для роботи з датчиками;
- демонстрація роботи датчиків;
- завдання для освоєння матеріалу;
- можливість запустити виконане завдання безпосередньо з програми.

Також для створення системи були розроблені програми для роботи Raspberry Pi с датчиками, які виступають зразками для навчання.

Були розроблені програми:

- для зміни по черзі від червоного до зеленого поперемінно кольору двоколірного світлодіода;
- для спалахування різними кольорами світлодіодного модуля RGB;
- для відображення напису і зміні кольору світлодіода при натискання на кнопку;
- для відображення напису і зміні кольору світлодіода при зміні нахилу модуля перемикача нахилу;
- для підрахунку кількості натисків кнопки на пульті;
- для посилання модулем лазерного випромінювача сигналів Морзе;
- для відображення напису і зміні кольору світлодіода при блокуванні надходження світла до світлоприймача;
- для зміни яскравості світлодіода і значення між 0 і 255, що друкуються на екрані, під час зміни положення валу потенціометра на PCF8591;
- для виявлення напрямку руху джойстика і супроводження руху написами на екрані;
- для зміни значення, яке друкується на екрані під час повернення валу потенціометра.

Розроблене інтерактивне середовище призначене збільшити ефективність навчання дітей роботі з одноплатним комп'ютером Raspberry Pi, платою розширення та датчиками.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Андриади И. П. Основы педагогического мастерства [Текст] / И. П. Андриади. – М.: НИЦ ИНФРА-М, 2016. – 209 с.
2. Коджаспирова Г.М. Педагогический словарь [Текст] / Г.М. Коджаспирова, А.Ю. Коджаспиров. – М.: Академия, 2000. – 174 с.
3. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования [Текст] /И. В. Петров, В. П. Дьяконов. – Москва : СОЛОН–Пресс, 2004. – 256 с.

УДК 004.9

Миронова Н.О.¹, Сергієнко О. С.²

¹канд. техн. наук, доц. ЗНТУ

²студ. гр. КНТ-415 ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ

На сьогодні, обробка медіа даних, що представляють собою статичні зображення, відео чи навіть онлайн відео трансляції у реальному часі, є дуже актуальною темою. Обсяги медіа ростуть, необхідність у їхній обробці також. Багато хто з нас використовує засоби обробки зображень та відео у своєму повсякденному житті, ведучи блог у Instagram, Twitter, Facebook чи навіть YouTube, де простота, швидкість та автоматичність процесу обробки медіа робить його зручним та легким але обмеженим.

На допомогу прийшло спеціалізоване професійне програмне забезпечення, як Adobe Photoshop, Sony Vegas Pro та інше ПЗ для обробки відео та зображень. Але використання цих програм потребує певних навичок та має під собою ручну обробку та редагування, що підвищує складність використання та витрачає багато часу.

Засобом автоматизації певних рутинних ручних дій користувача для обробки зображень і не тільки можуть виступати алгоритми машинного навчання, а саме – нейронні мережі та лінійні алгоритми. За їх допомогою можна автоматизувати такі процеси обробки зображень як комплексне покращення якості, корекція кольору, накладання стилів та їх трансформування, масок та багато іншого лише в декілька простих кроків, без необхідності ручного редагування медіа даних.

Розроблене програмне забезпечення має автоматизувати певний процес обробки медіа, а саме – глибоке накладання стилю на окреме зображення чи відео, в декілька кроків, без участі користувача. Таке ПЗ дає змогу перетворити звичайне відео – у мультфільм, ваше фото – у ваш портрет, ніби

намальований видатним художником. Треба лише обрати стиль та натиснути одну кнопку. Таке трансформування стилю може використовуватися у сфері художньої обробки фото чи відео і особливо – у сфері розваг. Наприклад, накладання стилю в історіях користувачів Instagram у режимі реального часу лише одним рухом.

Також, розроблене ПЗ має бути крос-платформним та незалежним від інтерфейсної частини, представляти собою ядро обробки медіа даних, абстрагуючись від конкретної реалізації графічного інтерфейсу користувача.

Виходячи із вимог, розроблений трансформатор стилю представляє собою програмне забезпечення, розроблене на Python, з певним API для користування, що реалізує алгоритм згортувальної нейронної мережі (CNN) – безпосередньо використовує натреновану мережу для кожного конкретного стилю а також може тренувати нові конфігурації мережі для нових стилів. В якості інструменту для розробки нейронної мережі використовується Google TensorFlow Framework, функція втрат – VGG19, бібліотеки NumPy, SciPy, MoviePy. Для оптимізації процесів обробки медіа та тренування мереж, існує підтримка GPU з пакетом Nvidia CUDA та cuDNN (за замовчуванням встановлено CPU з BLAS/MKL).

Так як трансформатор стилю представляє собою крос-платформна програма для нейронної мережі, вона має лише API у вигляді параметрів запуску, отримуючи необроблений медіа-файл на вході та генеруючи оброблений файл на виході. Для нього може бути реалізований будь-який GUI, як у вигляді веб-сервісу, так і у вигляді додатку для якоїсь платформи чи у вигляді модуля складного комплексного програмного забезпечення. Для простоти демонстрації прикладу реалізації програми на практиці, в якості фінального зразка рішення з GUI, було використано додаток для Windows, розроблений на C++/Qt.

УДК 004.9

Миронова Н.О.¹, Сиротенко І.О.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-415 ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕДІАСЕРВЕРУ ДЛЯ ПРИЙОМУ ТА ОБРОБКИ ВІДЕОДАНИХ В РЕАЛЬНОМУ ЧАСІ

Реалії сьогодення неможливо уявити без великих спортивних, політичних чи інших масових подій. Більшість заходів, які охоплюють велику групу людей, зазвичай проводять за підтримки медіа компаній або засобів масової інформації, але залучення підтримкою ЗМІ потребує

зацікавленості з їх сторони або ж великих фінансових вкладів, які можуть дозволити собі здебільшого титани індустрії чи то вже відомі організатори з певною репутацією.

Велике поширення доступу в інтернет та сучасних технологій сприяє тому, що навіть невеликі компанії чи то різного роду діячі намагаються розширити свою аудиторію за допомогою трансляцій та висвітлення подій в реальному часі. В той час як ніша поставників послуг для великих компаній та професіоналів вже сформувалась, різні платформи з цільовою аудиторією напівпрофесійного та побутового рівня ведуть активну боротьбу в своїй сфері.

Критичним фактором для даної сфери є дуже велике навантаження і, як наслідок, надзвичайна вартість підтримки інфраструктури. Зосереджені хмарні технології для збереження та перекодування відео змусять брати з користувачів велику плату, а покладання такого роду задач на користувачів може значно ускладнити процес і зменшити кількість потенційних юзерів.

Розроблена платформа має бути простою, доступною і в той же час надати можливість певного роду кастомізації потоку відео за власним бажанням. Саме через простоту і поширення було обрано платформою веб. Це дозволить дуже легко масштабувати та оновлювати інфраструктуру, зберегти простоту та, використовуючи можливості сучасних браузерів, розосередити обчислювальне навантаження. Інтерфейс програми має приваблювати користувача, зберігати просту і прозору логіку. Для початку трансляції необхідно виконати не більше двох-трьох дій.

Клієнтська частина буде використовувати апаратні засоби системи як джерело відео, WebGL для обробки потоку відео. WebSockets та Restful API буде використано для сигналінгу та обміну повідомленнями з веб сервером, WebRTC як протокол для обміну медіа даними. За рендер користувацького інтерфейсу буде відповідати бібліотека з відкритим кодом - React.

Серверна частина буде відповідати за авторизацію (OAuth2), збереження інформації(Hibernate, PostgreSQL), створення івентів, прийом та перекодування відео (Kurento, webRTC) (рис.1-2).



Рисунок 1 – Авторизація

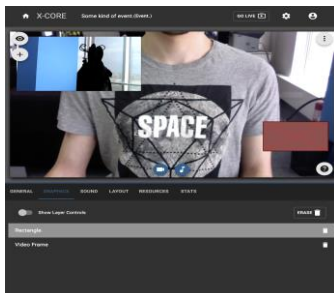


Рисунок 2 – Прев'ю модифікованого потоку та панель керування

УДК 004.9

Миронова Н.О.¹, Чадаєва А.К.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-415 ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН-СЕРВІСУ ДЛЯ ЗАМІТОК ТА ПЛАНУВАННЯ ЗАДАЧ

На сьогоднішній день розвиток технологій крок за кроком витісняє такі звичні нам речі, як щоденники і стікери. Та основна їх ідея з часом лише набуває актуальності. З моменту, коли інтернет став невід'ємною частиною нашого життя, змінився і обсяг інформації, який ми поглинаємо кожен день. Він виріс в десятки разів, і тримати в пам'яті всі потенційно важливі речі, в нас просто немає можливості.

Гаджети в свою чергу позбавили нас від паперової метушні. Тепер щоденники перейшли в електронний формат. У людей з'явилась можливість зберігати важливу інформацію у себе на телефоні, або на комп'ютері.

Та з нинішнім ритмом життя, нам дуже незручно бути прикутим до одного пристрою. В дорозі ми користуємося телефонами, на роботі та вдома планшетами й комп'ютерами. Наш щоденник може знадобитися нам у будь-який момент.

Завдяки тому, що мережа покриває вже більшу частину нашої планети, розміщення щоденника в інтернеті є найбільш оптимальним рішенням. Перебуваючи в будь-якій точці світу, маючи при собі лише девайс з можливістю виходу в інтернет та знаючи адресу і дані від аккаунта, ви в будь-яку секунду отримуєте доступ до свого "онлайн щоденника".

Розроблений онлайн-сервіс має бути простим, доступним та водночас багатофункціональним, наприклад: детальна структуризація заміток по

закладках та можливість зберігати масу різної інформації в тому числі сайти та мультимедійні файли. Інтерфейс має зберігати простоту блокнотів по принципу паперових стикерів.

Інтерфейс клієнтської частини виконаний за допомогою бібліотеки з відкритим кодом – React.

За збереження даних відповідає хмарна база даних від гугл – Firebase, що забезпечує можливість оновлення та читання даних в режимі реального часу на декількох пристроях одночасно.

УДК 004.9

Миронова Н.О.¹, Шемберко Ю.А.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-415 ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ДЕКОДУВАННЯ ВІДЕО

У житті кожної людини є моменти, які хотілось би запам'ятати на все життя. Таких моментів у кожного з нас більше, ніж достатньо та, на жаль, люди наразі фізично не здатні зберігати кожен з них в деталях. Але, так як ми живемо у вік інтенсивного розвитку технологій, сьогодні кожен має під рукою гаджет із вбудованою відеокамерою, що дає змогу записувати те, що відбувається довкола з можливістю подальшого відтворення.

Технології сьогодення окрім можливості записувати цікаві моменти з життя дають не менш прекрасну можливість ділитися такими моментами з іншими в різних соціальних мережах. Щоб мати змогу поринути в атмосферу того, що відбувається чи відбувалося з нами раніше, нашому віртуальному співрозмовнику достатньо мати пристрій, здатний відтворювати відео та, безпосередньо, сам відеоматеріал. Як же перетворити картинку перед очима та звук навколо за допомогою відеокамери в інформацію, яку можна буде відтворити у будь-який зручний нам момент? Для цього існує безліч програмних інструментів, що мають назву відеокодек. Відеокодек - це прилад або програмний засіб, здатний виконувати функції кодування та декодування цифрового відеопотоку. Кожен з них має такі параметри як швидкодію алгоритму, обсяг закодованих даних та їх якість. Деякі кодеки зберігають певний баланс між цими параметрами, інші - роблять акцент на користь одного з них. Зазвичай, дані характеристики і визначають сферу використання певного відеокодеку. Як приклад, можна навести один з найвідоміших та найпоширеніших кодеків – H264.

Якщо ж говорити про повноцінне програмне забезпечення (ПЗ), здатне відтворювати відео-контент, то недостатньо навчити його працювати

виключно з одним кодеком. Таке програмне забезпечення має бути максимально гнучким і, щоб охоплювати максимальну кількість користувачів, має вміти працювати з максимально можливою кількістю кодеків. Також, для максимізації кількості потенційних користувачів, таке ПЗ має бути кросплатформенним, тобто успішно розгортатися під контролем безлічі операційних систем (ОС). Інтерфейс користувача такого ПЗ має відповідати певним нормам: мати загальні елементи керування медіа-даними, бути максимально лаконічним, а результат взаємодії з ним має бути логічно очевидним.

Для реалізації даних вимог було обрано низько-рівневу мову програмування C++, потужний фреймворк для створення кросплатформенних UI-додатків Qt, а також відкриту бібліотеку для роботи з кодеками FFMPEG.

Схема роботи такої програми має виглядати приблизно таким чином (рис.1).

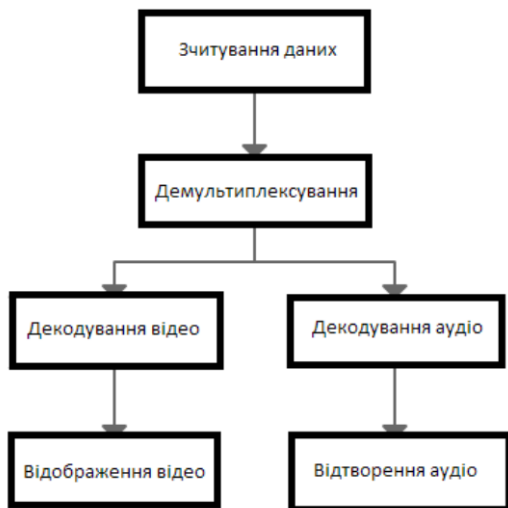


Рисунок 1 – Схема роботи програми-відео-відтворювача

В основному потоці програми здійснюється зчитування даних, визначення форматів даних та відповідного кодека. Після цього здійснюється демультимплексування – розділ медіа даних на окремі аудіо та відео потоки. Далі в двох окремих потоках здійснюємо паралельне декодування відео та аудіо даних. Відображення відео та відтворення аудіо слід виконувати із попередньою синхронізацією. Програма, збудована за даною схемою зможе успішно відтворювати медіаконтент різноманітного походження.

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ СПОРТИВНИХ ЗМАГАНЬ

Прогнозування спортивних змагань завжди викликало велику зацікавленість у фанатів та спортивної преси. З розвитком технологій, вдосконалення нових методів аналізу даних, використання нейро-мереж у прогнозуванні – результати стали точнішими, а зацікавленість до зменшення непередбачуваності у світі спорту виявили науковці.

Одними із найпопулярніших видів спорту є футбол. Для прогнозування результатів матчів було обрано Пуассонівський розподіл на основі кількості забитих голів.

Пуассонівський розподіл є одним із основних розподілів у теорії імовірностей і визначений наступною формулою.

$$P(x) = \frac{e^{-\lambda} \lambda^x}{x!}, \lambda > 0,$$

де λ – математичне очікування випадкової величини (середня кількість подій за фіксований проміжок часу), x – випадкова величина,

Відносно до футбольних метрик, λ – це показник середньої кількості голів в матчах чемпіонату. Аналізуючи футбольні метрики, було досліджено, що голи в чемпіонатах розподіляються за подібністю до графіка імовірностей Пуассона (рис. 1).

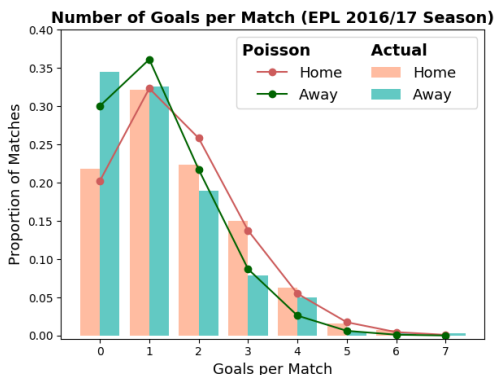


Рисунок 1 – Подібність графіків Пуассона до кількості голів в чемпіонаті

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ТА АНАЛІЗУ ВІДВІДУВАННЯ УЧНІЯМИ ШКОЛИ

Насьогодні розробка програмних засобів обліку та аналізу відвідування учнями школи є актуальною задачею. З розвитком мобільних пристроїв та мережі інтернет батькам учнів та адміністрації школи більш зручніше переглядати інформацію онлайн. Тому був розроблений модуль аналітики для веб-сайту школи, що дозволяє переглянути списки присутніх та відсутніх учнів, виконати контроль запізнень та прогулів учнів, переглянути щоденний моніторинг захворюваності, виконати формування фізичних груп залежно від стану здоров'я учнів та ін.

Для програмної реалізації модулю аналітики був обраний фреймворк Django – високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем.

Для розробки серверної частини динамічного сайту була обрана мова програмування Python.

Для розробки клієнтської частини динамічного сайту чудово підходить мова програмування JavaScript.

У фреймворку Django усі класи, що відповідають за логіку програми, успадковуються від основного базового класу View (рис.1). Були розроблені наступні класи:

- MainView – клас, який відповідає за контент, що виводиться на головній сторінці сайту: аналіз захворюваності, відсоток відсутніх школярів, відсутні по класам, побудова графіка хвороб, захворювань та прогулів;

- PupilsView — клас, який зв'язується з моделлю, отримує дані про усіх учнів та передає їх до шаблону.

- TeachersView — клас, який зв'язується з моделлю, отримує дані про усіх вчителів та передає їх до шаблону.

- AbsenceView — клас, який зв'язується з моделлю, отримує дані про усіх відсутніх та передає їх до шаблону.

- AddAbsenceView — клас, який отримує дані про наявні причини відсутності та зберігає дані, введені до форми заповнення відсутніх.

- AddGroupView — клас, який отримує дані про наявні групи по фізичній культурі та зберігає дані, введені до форми заповнення групи фізичної культури.

– AddDiscountView — клас, який отримує дані про наявні пільги на харчування та зберігає дані, введені до форми внесення школярів з пільгами.

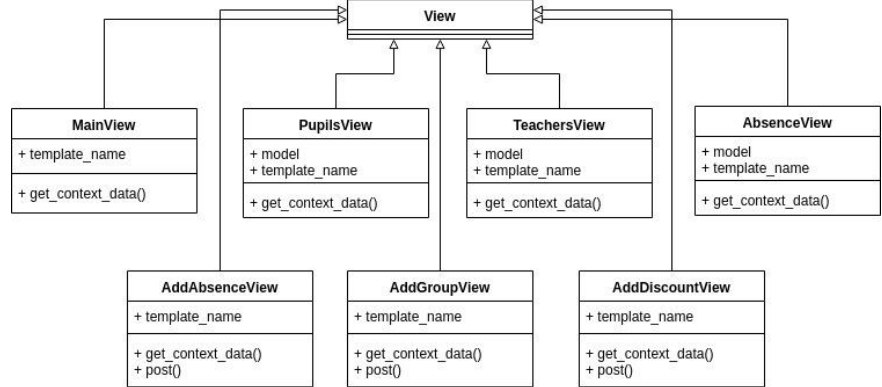


Рисунок 1 – Діаграма розроблених класів

Веб-додаток у Django організовується згідно шаблону проектування MVT(рис. 2): Модель (Model), Представлення (View), Шаблон (Template).

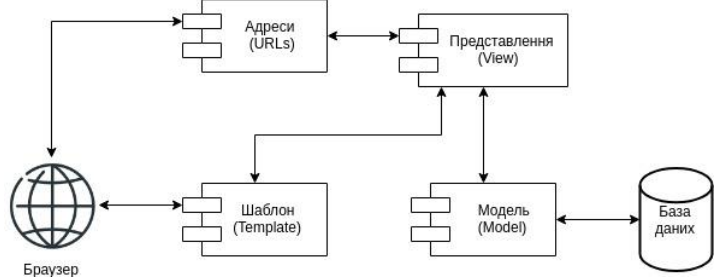


Рисунок 2 – Діаграма розроблених компонентів

Модель відповідає за роботу з даними. Вона взаємодіє з базою даних, перевіряє дані на коректність, створює розрахункові дані в разі потреби. Шаблон відповідає за те, як ці дані будуть виглядати, надає їм легку для читання форму.

Представлення є прошарком між моделлю, шаблоном і браузером. Коли відвідувач сайту запитує сторінку, цей запит передається представленню. Воно в свою чергу збирає всі моделі, які можуть стати в нагоді, і просить їх надати дані. Потім представлення вибирає шаблон, який найкраще відповідає запиту користувача і передає йому дані. Сформована сторінка (шаблон) передається як відповідь на запит.

У результаті роботи був створений модуль аналітики для веб-сайту школи на базі Web-технологій в середовищі розробки PyCharm на мові програмування Python, використовуючи Python фреймворк Django та базу даних SQLite3.

УДК 004.514.62

Сердюк С. М.¹, Басанець М. І.², Каменський Д. В.³, Діденко А. Є.³

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-117 ЗНТУ

³ студ. гр. КНТ-127 ЗНТУ

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ РЕСТОРАНОМ

На сьогоднішній день в Україні активно розвивається ринок закладів громадського харчування, а саме бізнес, пов'язаний із ресторанами, кав'ярнями і піцеріями. За статистикою, станом на 2016 рік в Україні нараховувалось близько 6000 ресторанів, 5000 кав'ярень та майже стільки ж піцерій [1]. Через велику конкуренцію власникам бізнесу громадського харчування необхідно постійно розвивати свою справу шляхом поліпшення вже існуючих та запровадженню нових якостей свого бізнесу. До таких якостей можна віднести: повний контроль постачання та витрати продуктів, зменшення паперової роботи, зменшення впливу людського фактору та як наслідок ймовірності помилок у зборі статистичних даних [2]. Через складність контролю цих характеристик, проблема підвищення якості управління рестораном є актуальною.

Метою роботи є розробка електронної системи управління рестораном «eMenu», що призначена для п'яти типів користувачів: відвідувач, офіціант, кухар, системний адміністратор, менеджер. Система повинна надавати менеджеру статистичні дані та можливість контролю постачання продуктів, системному адміністратору — можливість редагувати меню, кухарям — можливість розподіляти робочі задачі на кухні, офіціанту — можливість взаємодіяти з кухнею і користувачем, користувачу — можливість самостійно сформулювати замовлення. Мета всієї системи — централізація керування рестораном, аналіз статистичних даних та відстежування витрат продуктів у ресторани. Крім того, «eMenu» допоможе менеджеру сформулювати стратегію для подальшого розвитку ресторану шляхом аналізу статистичних даних.

Аналіз мети системи «eMenu» допоміг виявити основні типи користувачів та потрібний рівень підготовки, їх вимоги до інтерфейсів та основні функції кожного інтерфейсу.

Інтерфейс користувача встановлений на планшет з захистом від вологи. Планшет має механічну кнопку виклику офіціанту та налаштування яскравості. Формування замовлення здійснюється за допомогою інтерфейсу зображеного на рис. 1.

Через брак простору на екрані (екран планшета в десять дюймів), довелося зменшувати елементи керування, позбавлятися надписів та вказівок та замішувати їх найбільш точними піктограмами.

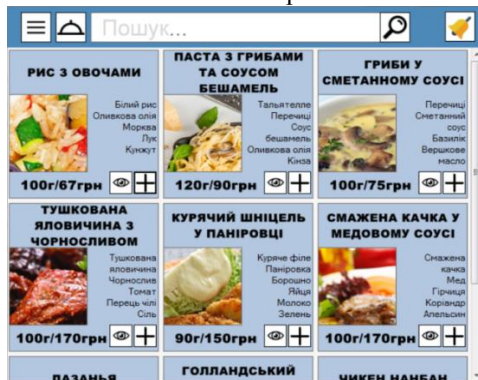


Рисунок 1 – Зовнішній вигляд інтерфейсу користувача

Таким чином, розроблений програмний продукт має наступні переваги: інтерактивність, зменшення впливу людського фактору та обсягів паперової роботи, надання статистичних даних, можливість швидко зробити замовлення та доступ до системи у будь-який час.

Для навчальних цілей було розроблено інтерфейси користувача та системного адміністратора. Планується подальший аналіз системи, розвиток та, можливо, програмна реалізація.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Казимирская В. А. Статистический анализ рынка ресторанных услуг Украины [Текст]/ Казимирская В. А., Тарасова К. И. – Одесса: Одесский национальный экономический университет, 2018. – 6 с.
2. Меркулов, М. Ю. Ресторан. 50 способов увеличить прибыль [Текст]/ М. Ю. Меркулов. – Санкт-Петербург: Питер, 2014. – 176с.

СИСТЕМНИЙ ПІДХІД ДО РОЗРОБКИ СИСТЕМИ МОНІТОРИНГУ ТУРИСТИЧНИХ МАРШРУТІВ

Сучасні туристичні застосунки надають малий обсяг інформації туристам, що захоплюються “зеленим” туризмом. Статистика травматизму та непередбачуваних ситуацій під час подорожі, вказує на необхідність підвищення безпеки туристів на маршрутах та забезпечення їх своєчасною допомогою. Отже розробка системи моніторингу туристичних маршрутів (СМТМ) є актуальною задачею.

Метою даної роботи є розробка орієнтованого на користувача підходу до проектування СМТМ, та програмна реалізація цієї системи.

Системний підхід до розробки базується на функціонально-структурній теорії А.І. Губінського [1]. Системний аналіз СМТМ складається з компонентного, морфологічного й еволюційного аналізів. Виходячи із задач даного дослідження, необхідності вводити еволюційні структури не було. Тому, була розроблена множина моделей, що формалізують знання про систему СМТМ

$$СМТМ = <KStr, MStr>,$$

де $KStr$ – множина компонентних структур; $MStr$ – множина морфологічних структур.

Для компонентного аналізу було розроблено компонентно-функціональну та функціонально-об’єктну структури. Компонентний аналіз дозволив визначити: тип користувачів; область застосування (“зелений” туризм) та вимоги до СМТМ, її функціональності та робочого середовища користувача.

Компонентно-функціональна структура для користувачів включає наступні функції системи:

надання користувачем інформації про особисті дані, час відправлення та час прибуття, маршрут, місце знаходження;

- отримання інформації про інших користувачів (маршрут, місце знаходження, особисті дані);

- редагування профілю зареєстрованого користувача, подання сигналу тривоги.

Морфологічний аналіз системи дозволив виявити різноманітні взаємозв'язки, що існують між компонентами системи, виділеними в

результати компонентного аналізу. В даній роботі було розроблено функціонально-часову структуру СМТМ.

СМТМ працює під керуванням операційної системи Android, з версією не нижче 4.4.2. Необхідна наявність Android-клавіатури, бажана наявність функції GPS. Необхідний обсяг оперативної пам'яті - 200-250Кб., обсяг пам'яті на пристрої - 6-7Мб. Мова інтерфейсу – українська.

Розроблений застосунок дає можливість користувачу зареєструватися перед подорожжю, залишивши свої особисті дані (П.І.П., номер телефону, домашня адреса, дата народження), номер телефону контактної особи (на випадок критичної ситуації), необхідний пароль для подальшої авторизації в особистому кабінеті, обрати/не обирати анонімність. Вирушаючи у подорож користувач має можливість авторизуватися в особистому кабінеті(авторизація за номером телефону та паролем) та вказати дати відправлення та прибуття і відповідно час відправлення та прибуття, вказати маршрут, яким вирушає та спостерігати його на карті. Під час подорожі користувач може редагувати дату та час прибуття, залишити повідомлення у своєму профілі, змінити налаштування анонімності, а також обрати інший маршрут. Існує можливість подати сигнал тривоги (у випадку наявності сигналу тривоги власнику телефону пропонується відправити смс-повідомлення про необхідність допомоги на номер телефону контактної особи профілю).

Також під час подорожі користувач має можливість вказувати своє поточне місце знаходження двома способами: після авторизації автоматично встановлюється поточне місце знаходження за GPS, або якщо це неможливо, користувач може вказати одну з точок маршруту. Користувач має можливість шукати за прізвищем та ім'ям, переглянути список тих, хто зараз потребує допомоги, переглянути всіх зареєстрованих туристів. Місце знаходження знайдених користувачів відображується на карті.

Практичне значення розробленої СМТМ, полягає в автоматизації діяльності туристів та рятувальників, надання їм релевантної інформації у реальному часі та покращенні безпеки екстремального туризму. Широкий спектр функцій, які виконує застосунок, дозволяє йому конкурувати на ринку аналогічних програмних продуктів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А. Н. Адаменко, А. Т. Ашеро, И. Л. Бердников и др.; под общ. ред. А. И. Губинского. – М.: Машиностроение, 1993. – 528 с.

РОЗРОБКА КОНЦЕПТУ «КАСА САМООБСЛУГОВУВАННЯ»

Сучасній роздрібній торгівлі для підтримки конкурентоспроможності вже недостатньо мати лише відповідний асортимент товарів і привабливий рівень цін. Конкуренція на ринку підводить ритейлерів до підвищення рівня сервісу та вибору найбільш економічно ефективних рішень, для задоволення потреб покупців.

Дослідження проведені компанією New Vision [1] виявили недоліки використання стандартного підходу до організації касових зон. Було з'ясовано, що 54% покупців стоячи біля каси втрачають від 30 хвилин до 4 годин на тиждень, а 64% покупців хоча б раз залишали магазин через великі черги.

З'являється необхідність збільшити пропускну спроможність касової зони, щоб скоротити черги та більш гнучко вибудувати графік роботи кас. Рішення даної проблеми – впровадження систем самообслуговування. У результаті підвищується якість обслуговування відвідувачів, отже збільшуються обсяги продажів.

Актуальність роботи – необхідність пошуку нового методу верифікації товару, що продається, для мінімізації можливості його крадіжки.

Мета роботи – створення концепту каси самообслуговування з використанням новаторських методів верифікації товару.

З існуючих систем самообслуговування на сьогоднішній день, можна виділити такі: Indigo компанії Chameleon Soft і NCR Self-Checkout компанії SoftMarket.

Міжнародне дослідження проведене IBM Corp Study, Global EPOS, MarketingCharts і NCR [2] виявило один, вагомий недолік використання кас самообслуговування – підвищення рівня крадіжок і шахрайства на 20%. Цей недолік з'являється через використання недосконалої системи верифікації товару: тільки за штрихкодом і ваговими характеристиками.

Верифікацію товарів було прийнято здійснювати за допомогою використання не тільки вагових характеристик і штрих-коду, а й аналізу даних, заснованого на машинному навчанні. Прикладом використання машинного навчання є програма зі штучним інтелектом AlphaZero компанії Google, яка шляхом використання штучних нейронних мереж за чотири години самоаналізу, перевершила рівень навичок найсильнішою на даний момент шахової програми Stockfish 8.

Серед значної кількості існуючих алгоритмів машинного навчання було обрано алгоритм згорткових нейронних мереж. Цей алгоритм розроблений для ефективної класифікації зображень і виявлення на них об'єктів, тому найкраще підходить для розв'язуваної задачі.

Робота згорткової нейронної мережі зазвичай інтерпретується як перехід від конкретних особливостей зображення до більш абстрактних деталей. При цьому мережа самонастроюється і сама виробляє необхідну ієрархію абстрактних ознак, фільтруючи незначні деталі і виділяючи істотні.

Пропонуємий алгоритм роботи машинного навчання, шляхом побудови і використання згорткових нейронних мереж, можна поділити на два етапи.

Перший етап – підготовчий. На цьому етапі проводиться збір даних (фотографій, ваги, розмірів) кожного товару. Отримані дані відправляються в систему, де шляхом аналізу і побудови нейронних мереж отримана інформація аналізується і перетворюється в модель товару, яка являє собою набір його характеристик. Отримана модель приймається за еталон даного товару, і записується в базу даних.

Другий етап – практичний. При здійсненні покупки клієнт сканує товар, у цей момент, камери розташовані в касі самообслуговування фіксують дані товару і відправляють їх на обробку. Система звіряє отримані дані з еталонами, і підтверджує або заперечує правильність зазначення товару. Це відбувається миттєво, оскільки системі не доводиться будувати нову модель для кожного товару, а достатньо лише звіритися з існуючими еталонами.

У результаті роботи було створено концепт системи самообслуговування, що базується на використанні вдосконалених методів верифікації товару з використанням вищезазначеного підходу.

Передбачено розвиток і вдосконалення концепту, а саме: досліджуються і аналізуються можливості поліпшення методу верифікації шляхом впровадження додаткових технологій: мікрочипів, теплових і ультразвукових датчиків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. SELF-CHECKOUTS ADD EFFICIENCY [Electronic resource]. – Access mode: <http://www.new-vision.com/en/solutions/customer-service/selfservice/self-service-checkout/>
2. Self-Checkout Usage Statistics [Electronic resource]. – Access mode: <http://www.statisticbrain.com/store-self-checkout-usage-statistics/>

РОЗРОБКА ІНТЕРФЕЙСУ БАНКОМАТА

Нині існує величезний досвід розробок різних інтерфейсів банкоматів. Наприклад, в Україні всім відомі такі банки як ПриватБанк, ОщадБанк. Але незважаючи на їх популярність та поширення серед користувачів, всі системи мають деякі недоліки. Враховуючи те, що загальновизначеного стандарту, концепту або методики проектування інтерфейсів банкоматів досі не існує, дана тема є актуальною. Тож метою цієї роботи є ретельний аналіз переваг та недоліків існуючих систем та розробка ергономічного та функціонального інтерфейсу, з яким клієнтам банків буде приємніше взаємодіяти, який приноситиме їм більше користі.

Банкоматами користуються люди різного віку, але у всі вони бажають працювати зі зручним і простим інтерфейсом, виконувати операції за невеликий проміжок часу, тобто інтерфейсом, орієнтованим на користувача.

В запропонованому інтерфейсі враховано те, які саме помилки може допустити людина під час роботи з банкоматом. Було проаналізовано діяльність людини при взаємодії з даною системою та розроблено максимально легкий інтерфейс задля спокою користувачів.

Проаналізувавши аналогічні програмні засоби було виявлено наступні недоліки:

- слабка система пошуку, недостатня локалізація;
- нелогічний інтерфейс;
- несвоєчасні повідомлення;
- напружуючий інтерфейс;
- наявність зайвих функцій (реклама), але відсутність деяких потрібних.

З урахуванням цих недоліків, було розроблено інтерфейс банкомата.

Розроблений програмний продукт дозволяє користувачам використовувати наступні функції: поповнити картку, отримати готівку, переказати кошти, використати обране, оплатити послуги, поповнити мобільний, отримати інформацію про баланс картки та курс валют, оформити кредит, змінити налаштування, зробити екстрений виклик.

Задля покращення інтерфейсу було використано візуальну ієрархію, візуальний потік інтерфейсу, групування та вирівнювання елементів, шаблони для компоновання сторінок (Візуальна схема, Центральна сцена, Діагональний баланс) [1]. Було забезпечено контроль над візуальним потоком інтерфейсу на сторінці, щоб користувачі проходили по потрібному шляху в

правильній послідовності. За допомогою порожнього простору були створені точки фокусування. Елементи управління розташовані вздовж простого візуального шляху. На кінці якого створено посилання, призначені для завершення завдання (назад на попередню сторінку, далі на наступну сторінку, завершення роботи з банкоматом).

Таким чином, було розроблено інтерфейс багатофункціонального та ергономічного банкомата. Серед його переваг можна виділити орієнтацію на користувача, зручний та інформативний інтерфейс (рис.1).



Рисунок 1 – Фрагменти інтерфейсу програми

Планується подальший розвиток даної програми та більш детальна реалізація деяких функцій (наприклад, спеціальний сценарій обслуговування для підтримки людей з обмеженими можливостями). Така інтелектуалізація системи значно підвищить її якість, ефективність та надійність, конкурентоспроможність на ринку аналогічних програмних продуктів та зробить її більш орієнтованою на конкретного користувача.

Отже, проведений системний аналіз та аналіз аналогів визначили основні функціональні вимоги до універсального інтерфейсу банкомата, підвищили ефективність діяльності користувачів, мінімізували зусилля та ризик помилок, зменшили психологічне навантаження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Головач В.В. Дизайн пользовательского интерфейса [Електронний ресурс] /В.В. Головач. – Режим доступу: <http://uibook2.usethics.ru/uibookll.pdf>

УДК 004.5

Сердюк С.М.¹, Полумієнко Д.О.², Дзандзава Г.М.²

¹ канд. техн. наук, доц. ЗНТУ

² студ. гр. КНТ-117 ЗНТУ

РОЗРОБКА КОНЦЕПТУ «СМАРТ-РАНЦЮ»

В сьогоденні існує безліч різновидів ранців. Кожен, хто носить цінні речі в рюкзаку регулярно замислюється про їхню схоронність. Мова йде не тільки про захист від вологи або падінь, але і від непомітного проникнення зловмисників.

Компанія XD Design створила рюкзак, який зможе захистити майно користувачів від усіх перерахованих вище неприємностей. Принаймні, так позиціонується цей продукт. Більшість компаній по виробництву смарт-ранців на ринку спрямовані на забезпеченні цілостності майна і не вирішують питання візуальної та технічної частин девайсу. Останнім часом на ринку присутні різноманітні смарт-ранці таких фірм як: Optima, Body Urban, Pacsafe, XD Design. Виробники цих ранців брали за основу: якість матеріалу з якого був виготовлений ранець, безпеку вмісту, або лише технічна складова ранцю. Виробники звичайно рухались в правильному напрямку, але жодна компанія ще не об'єднала усі ці компоненти. Тому розробка смарт-ранцю є актуальною задачею.

Метою даної роботи є розробка концепту смарт-ранця, оснований на методах системного підходу. Смарт-ранець має надавати інтуїтивно

зрозумілий інтерфейс та поєднувати у собі найсильніші складові ранців вироблених провідними компаніями.

Існує стандарт ДСТУ ГОСТ – 28631 [1], згідно з яким виготовляються ранці. Пропонується не тільки дотримуватись ГОСТу - 28631, але й використовувати стандарт IP-68, причому зробити основний акцент на технічній складовій.

Проведений системний аналіз дозволив виявити типи користувачів, їх вимоги до продукту та його основні функції.

До складу продукту входять такі компоненти: ранець з вбудованою сонячною батареєю, USB перехідник та мобільний додаток. Смарт-ранець обладнаний датчиком тиску, для запобігання випадкових натискань кнопок керування та сенсорного дисплею, датчиком наближення для контролю затягування лямок ранцю, чотирма кнопками керування, сенсорним дисплеєм та вбудованим діодним підсвічуванням, що керується користувачем за допомогою мобільного додатку.

Було проведено опитування, в ході якого респонденти висловлювали свою думку стосовно інтерфейсу, стилізації, зручності та практичності. Обробка результатів опитування дозволила усунути усі недоліки та удосконалити мобільний додаток.

Мобільний додаток надає інформацію (рис.1) про стан батареї, дає змогу користувачу самостійно обирати колір ранцю, знайти відповіді на запитання, або самому допомогти іншим користувачам в освоєнні, та зв'язатись з оператором. Додаток також має функцію оповіщення на мобільний телефон користувача при порушеннях роботи батареї або датчиків.

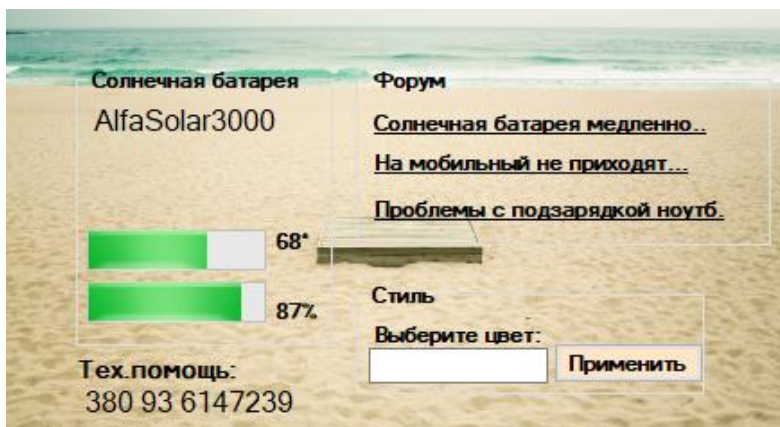


Рисунок 1 – Интерфейс мобильного додатку

Таким чином, було розроблено концепт «Смарт-ранцю». Серед його переваг: орієнтація на користувача, використання усіма відомих технологій по новому, таких як: сонячна батарея для портативної підзарядки пристроїв, діодне підсвічування для зміни зовнішнього виду ранця, зручний та інформативний інтерфейс, привертання уваги перехожих, доступність застосування в будь-який час доби.

Планується подальший розвиток концепту та його програмна реалізація.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сумки, валізи, портфелі, ранці, папки, вироби дрібної шкіргалантереї. Загальні технічні умови : ДСТУ ГОСТ 28631-2005, IDT. – [Чинний від 01.10.2005]. – Київ, Держспоживстандарт України, 2005. – 74 с. – (Національний стандарт України).

УДК 004.4

Скачко Л.П.¹, Чемерис К. М.²

¹асист. ЗНТУ

²студ. гр. КНТ-137 ЗНТУ

РОЗРОБКА ІНТЕРФЕЙСУ БОРТОВОГО КОМП'ЮТЕРА АВТОМОБІЛЯ

В даний час неможливо уявити собі сучасний автомобіль без такого досягнення науки і техніки, як бортовий комп'ютер (БК) – складної системи для збору і відображення відомостей про різні параметри функціонування автомобіля. Маючи бортовий комп'ютер, власник отримує безліч переваг, які допоможуть збільшити термін служби автівки та підвищити безпеку руху.

Метою роботи є розробка людино-машинного інтерфейсу бортового комп'ютера, призначеного для управління автомобілем.

Був проаналізований аналогічний інтерфейс такого відомого бренду як Tesla (Model S). Інтерфейс вказаного авто дуже вдалий, але бортовий комп'ютер дуже перевантажений інформацією, яка рідко використовується, до того ж комп'ютер занадто дорогий.

Тому, пропонується варіант інтерфейсу, який надає лише саму важливу інформацію для водія, і це дасть можливість не дуже вплинути на кінцеву ціну автівки.

Система може надавати водієві інформацію про стан автомобіля (заряд акумулятора, тиск в шинах, рівні рідин: моторне масло, гальмівна рідина, рідина гідропідсилювача керма і охолоджуюча рідина). Забезпечує

можливість управління бічними стеклами і люком, підігрівом сидінь і кондиціонером.

Система дозволяє користуватися картою і навігацією (з можливістю пошуку потрібного пункту, прокладання найкоротшого маршруту, часу його проходження при заданій швидкості) (рис. 1).

Комп'ютер показує пробіг автомобіля за день, місяць або рік.

Система надає також можливість підключати мобільні пристрої через Bluetooth, здійснювати пошук абонентів і здійснювати дзвінки зі списку або вручну за номером.

Через обмежений простір на 12-дюймовому екрані довелося зменшити елементи управління, розбити меню на 4 окремі блоки.

Мовою реалізації системи БК є мова C # в середовищі Visual Studio.

Вимоги до технічних засобів: процесор не нижче NVIDIA Tegra 3, 1.3 ГГц, RAM 1 ГБ, HDD 16 ГБ, бездротові технології Bluetooth, GPS. БК працює під управлінням ОС Ubuntu.

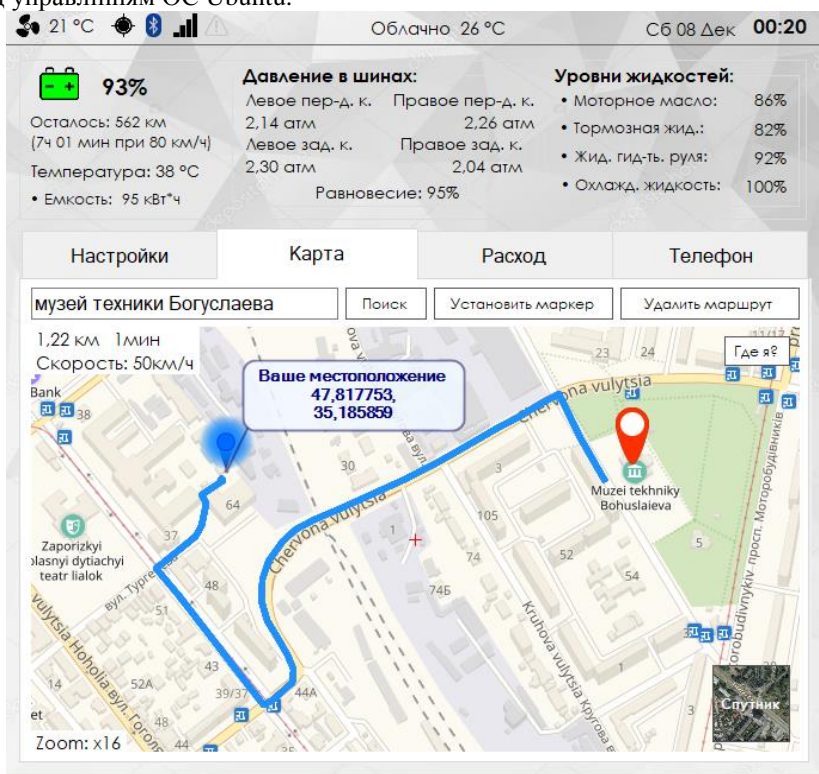


Рисунок 1 – Зовнішній вигляд інтерфейсу водія

Розроблена електронна система БК дозволяє водієві легше справлятися з управлінням авто, швидко отримувати всю необхідну інформацію про стан і безпеку транспортного засобу, менше відволікатися від дороги і впевненіше себе почувати за кермом.

Інтерфейс БК може при необхідності доповнюватися, поліпшуватися, отримувати іншу інтерактивну оболонку для зручності роботи з ним.

УДК 004.5

Камінська Ж.К.¹, Соколовський Д. В.²

¹асист. ЗНТУ

²студ. гр. КНТ-137 ЗНТУ

РОЗРОБКА КОНЦЕПТУ «РОБОТ-ПИЛОСОС»

Роботизована техніка поступово перетворюється з розкоші в необхідність. Компактні і розумні роботи-пилососи, для багатьох з нас, стали важливими помічниками в щоденному прибиранні [1]. Тому розробка концепту робота-пилососа являється актуальною на даний час.

Існує величезний досвід розробок різних проектів роботів-пилососів. Більшість з них спрямовані на певний вид прибирання але жодний з цих роботів не задовольняв усіх потреб користувачів. Не існує також загальновизнаного стандарту, концепту або методики проектування робота пилососа.

Останнім часом на ринку присутні різноманітні роботи-пилососи таких фірм як: LG, Electrolux, iRobot Corporations, Karcher, Samsung [2]. Роботи цих виробників оснащені: потужною системою всмоктування пилу, інтелектуальною системою прокладки маршруту для прибирання тощо. Але всі вище перераховані компоненти не були об'єднані в єдиному роботі.

У статті запропоновано концепт «Робота-пилососа», заснований на методах системного аналізу і загальних рекомендаціях з проведення функціонального аналізу.

Для використання розробленого програмного продукту користувач повинен мати телефон або планшет з операційною системою IOS починаючи з версії 9.0 або Android версією 6.0 або більш пізню.

Для синхронізації робота-пилососа з телефоном або планшетом слід використовувати Bluetooth підключення. В цьому випадку обмежуючим фактором являється дальність підключення, що має довжину не більше 50 метрів. Також передбачена можливість синхронізації через посередництво мережі Wifi. Даний спосіб більш універсальний і не має обмежень у довжині підключення.

Розроблений програмний продукт (рис. 1) надає інформацію про поточний стан компонентів робота-пилососа таких як: зарядка батареї, об'єм пилу в контейнері, кількість води в резервуарі. Реалізована можливість зміни мови інтерфейсу. Також програма надає змогу вибирати: режим прибирання (звичайне, вологе), режим ретельності (легке, середнє, ретельне), від якої залежить потужність всмоктування пилу та час прибирання, і можливість вибору маршруту прибирання, шляхом натискання на схему приміщення, яку робот-пилосос побудує при першому запуску даного програмного забезпечення.

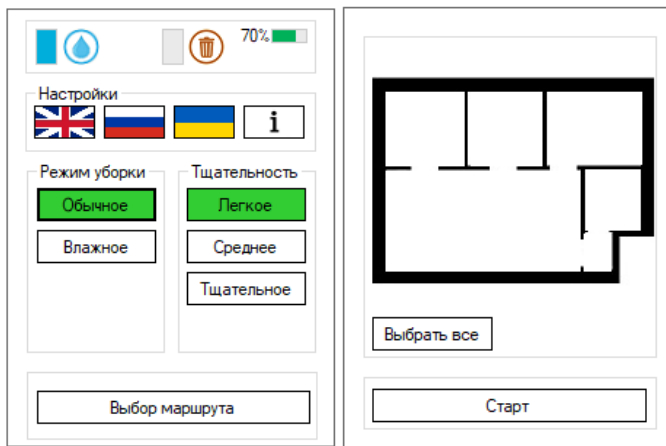


Рисунок 1 – Інтерфейс мобільного застосунку

Таким чином, було розроблено концепт «Робота-пилососа». Серед його переваг: зручний та інформаційний інтерфейс, змога налаштування режиму прибирання, доступність застосунку в будь-який час доби.

Планується подальший розвиток концепції та його програмна реалізація. Передбачається можливість керування та налаштування декількох роботів, а також можливість додавання схем планування приміщення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. A-class Technology. Як з'явилися роботи-пилососи [Електронний ресурс]. – Режим доступу: <https://a7.com.ua/public/kak-pojavilis-roboty-pylesosy> (дата звернення 05.12.2018).
2. Популярна електронніка. Історія відкриттів. Історія робота-пилососа [Електронний ресурс]. – Режим доступу: <http://scsiexplorer.com.ua/index.php/istoria-otkritiy/1904-istorija-robota-pylesosa.html> (дата звернення 05.12.2018).

СИСТЕМНИЙ ПІДХІД ДО РОЗРОБКИ СПОРТИВНОГО ЗАСТОСУНКУ

Сучасні спортивно-тренувальні застосунки під керування операційної системи Android надають вузький спектр можливостей, що часто призводить до встановлення користувачами декількох застосунків. Це вказує на потребу об'єднати найбільш актуальні для тренування функції в одному застосунку. Отже розробка спортивного застосунку (СЗ) під Android є актуальною задачею.

Метою даної роботи є розробка орієнтованого на користувача підходу до проектування спортивного застосунку для тренувань, та її програмна реалізація.

Системний підхід до розробки базується на функціонально-структурній теорії А.І. Губінського [1]. Системний аналіз СЗ складається з компонентного, морфологічного й еволюційного аналізів. Еволюційний аналіз проводиться у разі зміни у часі компонентних та морфологічних структур, за допомогою яких проводиться відповідний аналіз. Так як у застосунку є тільки один користувач, то необхідність у проведенні еволюційного аналізу була відсутня. Тому, була розроблена множина моделей, що формалізують знання про систему СЗ.

$$CЗ = <KStr, MStr>,$$

де *KStr* – множина компонентних структур; *MStr* – множина морфологічних структур.

Для компонентного аналізу було розроблено компонентно-функціональну та функціонально-об'єктну структури. Компонентний аналіз дозволив визначити: тип користувачів; область застосування (спорт) та вимоги до СЗ, її функціональності та робочого середовища користувача.

Компонентно-функціональна структура для користувачів включає наступні функції системи:

- надання користувачеві інформації, пов'язаної з обраним видом спорту та обраною темою, у відео форматі;
- надання можливості виконувати бажаний комплекс вправ із налагодженим секундоміром;
- надання можливості зберегти та переглядати результати виконання комплексів вправ;

– можливість використання спортивного секундоміру.

Морфологічний аналіз системи дозволив виявити різноманітні взаємозв'язки, що існують між компонентами системи, виділеними в результаті компонентного аналізу. В даній роботі було розроблено функціонально-часову структуру СЗ.

СЗ працює під керуванням операційної системи Android, з версією не нижче 5.0.0. Необхідна наявність Android-клавіатури, бажана наявність зв'язку з мережею Інтернет. Необхідний обсяг оперативної пам'яті – 1 Гб., обсяг пам'яті на пристрої – 55 Мб. Мова інтерфейсу – російська.

Розроблений додаток дає можливість переглядати відео матеріал, пов'язаний з видом спорту та окремої теми, що були обрані користувачем. Для перегляду використовується програвач відео-сервісу Youtube. Обов'язковим є зв'язок з мережею Інтернет.

Користувач має можливість обрати комплекс вправ для виконання, налаштувати секундомір з можливістю встановлення часу виконання кожної вправи, часу відпочинку між виконаннями вправ та кількості циклів виконання однієї вправи. По завершенню виконання є можливість збереження інформації про комплекс вправ та параметрів секундоміру.

Також є можливість використання спортивного секундоміру з власними налаштуваннями без зазначення комплексу вправ для виконання. Це реалізовано у ситуації коли бажаного комплексу вправ у застосунку не існує.

Функціональна частина застосунку реалізована у вигляді щоденника, який містить у собі статистичні дані особистих тренувань. Користувач може переглядати та аналізувати наступну інформацію: результати виконання комплексів вправ із зазначенням назви комплексу, що виконувався; час виконання кожної вправи; час відпочинку між вправами; кількість циклів виконання кожної вправи та загальний час виконання комплексу вправ.

Практичне значення розробленого СЗ полягає в наданні повної необхідної інформації користувачу у реальному часі для покращення якості та ефективності спортивних тренувань. Широкий набір функцій, що надає за стосунком, дозволяє йому конкурувати на ринку аналогічних програмних продуктів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А. Н. Адаменко, А. Т. Ашеро́в, И. Л. Бердни́ков и др.; под общ. ред. А. И. Губинского. – М.: Машиностроение, 1993 – 528 с.

РОЗРОБКА КОНЦЕПТУ ГРИ «НАВЧАЛЬНІ ШАХИ»

Існує незліченна кількість стратегій ведення гри у шахи. Ці стратегії, застосовані на практичному досвіді гравця, забезпечують йому перемогу або певну перевагу над суперником у грі.

Відомо багато програм для тренування навичок гри у шахи та, відповідно, напрацюванню досвіду: LiChess, Chessbase та ін. Більшість з них базується на принципі «унікальна картинка – запозичена логіка». За такого принципу індивідуальність програми лише в інтерфейсі, а стратегія ведення гри виконана шаховим двигуном (ШД) від сторонніх розробників: Stockfish, Komodo та ін. При застосуванні такого принципу розробки проекту за вибір складності гри відповідальний гравець, тобто в програмах побудованих за принципом «унікальна картинка – запозичена логіка» відсутня адаптивність програми до навичок гравця, що змінюються. Актуальність роботи - необхідність створення програм для навчання або підвищення рівня гравця у шахи. Ціллю роботи є створення гри, яка адаптується до існуючих навичок у гравця і допомагає підвищити його навички.

При аналізі програм з шахів було виявлено три закономірності:

1) практичні задачі, якщо наявні у програмі, розраховані на 1 - 4 дії користувача. Таке моделювання ситуації є «відірваним» від партії, яка переважно складається з сорока ходів;

2) після виконання практичної задачі програма ніколи не повторює задачу, тобто відсутнє закріплення вже отриманих гравцем знань;

3) найбільш поширені, тобто наявні майже у всіх програмах, два варіанти гри: гра з ШД і гра з іншим гравцем.

Базуючись на цих закономірностях, прийнято рішення розробки концепту з трьома режимами гри: штучне навчання, навчання у грі, вільна гра.

Штучне навчання включає в себе навчальні матеріали, після перегляду яких гравцю пропонують зіграти партію. Навчальні матеріали – це відтворення всесвітньо відомих дебютів, мітельшпільів, та ендшпільів на шаховому полі з коментарями до певних кроків від професійних гравців. Під час партії у цьому режимі ШД намагається максимально відтворити умови за яких доцільно застосовувати останню вивчену стратегію.

Навчання у грі – це, безпосередньо, гра у режимі «гравець – ШД», де ШД проводить стратегію, що ґрунтується на засвоєному гравцем матеріалі у

розділі «Штучне навчання». Тобто, ШД намагається під час партії моделювати вже вивчені гравцем ситуації з метою тренування та вдосконалення навичок гри й пам'яті гравця.

Вільна гра – варіант за яким гравець може грати або з ШД, або з іншим гравцем(за допомогою інтернету, чи спільного доступу до пристрою).

В усіх режимах гри між гравцем та ШД постійно працює програма-аналізатор (ПА), яка веде статистику прогресу гравця у грі. Статистика охоплює наступні аспекти: якісні показники гри – кількість перемог, програшів, нічиїх; прогрес у навчанні – % від вивчених навчальних матеріалів; застосування отриманих знань – кількість випадків застосування вже засвоєних стратегій; успішність застосування отриманих знань; найчастіші помилки гравця – помилки, які повторювались у різних партіях.

ПА, контролюючи статистику, за необхідністю видає підказки гравцю: на що треба звернути увагу, як краще було б зробити хід або привітання з успіхом, бо позитивний відгук стимулює гравця до самовдосконалення. Гравець може у будь-який момент гри переглянути статистику власноруч (рис. 1).



Рисунок 1 – Інтерфейс гри у режимі «Вільна гра»

За результатами роботи було створено концепт навчально-адаптивної програми, найбільшими перевагами якого є: орієнтованість на розвиток навичок гравця; адаптивність до рівня гравця, інтуїтивно зрозумілий інтерфейс; наявність як навчальних, так і загальноприйнятих режимів.

Передбачено розвиток і вдосконалення концепту, а саме: розглядається варіант інтегрування до гри месенджера, з метою поліпшення взаємодії та спілкування між гравцями, що підвищить якість навчання.

УДК 004.9

Степаненко О.О.¹, Федорченко Є.М.², Качан О.І.², Борисенко В.О.³, Шевчук М.С.⁴

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТз-114 ЗНТУ

⁴ студ. гр. КНТ-527 ЗНТУ

АВТОМАТИЗОВАНА СИСТЕМА КЕРУВАННЯ ТА МОНІТОРИНГУ УСТАТКУВАННЯ ОЧИЩЕННЯ ВОДИ УСТАНОВКОЮ ЗВОРОТНЬОГО ОСМОСУ РО/ДАТ - 2/0,4

Проблема забезпечення населення України доброякісною питною водою належить до найбільш соціально значущих, оскільки вода безпосередньо впливає на стан здоров'я громадян і кардинальним чином визначає ступінь екологічної та епідеміологічної безпеки цілих регіонів. Аналіз стану водопостачання населення в більшості регіонів України показав, що якість питної води в ряді випадків залишається незадовільним, а в деяких населених пунктах ця проблема набула кризового характеру. Реальна обстановка в джерелах водопостачання свідчить про погіршення якості води, пов'язаному зі скиданням у водойми антропогенних забруднюючих речовин, що є найбільш серйозною проблемою. В результаті ряд водних об'єктів став практично непридатним для здійснення питного водопостачання, так як їх якісні показники не задовольняють нормативним вимогам підготовки питної води на діючих станціях.

Автоматична система керування (АСК) установкою є комплексною системою, яка охоплює всі елементи технологічної схеми підготовки очищеної води, і призначена для автоматизованого контролю, управління та аналізу роботи установки.

Основними перевагами використання АСК є:

- значне скорочення частки ручних операцій при реалізації технологічного процесу знесолення води;
- підвищення надійності роботи установки за рахунок зменшення ймовірності помилкових дій персоналу;
- підвищення економічності та якості роботи установки за рахунок автоматично контролюваного дотримання технологічних режимів.

Регулювання технологічних процесів в водопідготовці, що відбуваються при експериментальному тиску і швидкостях, вимагає апаратури високого класу, що володіє швидкістю і точністю дії. Тому тут є актуальним використання вільно програмованих логічних контролерів (ПЛК). ПЛК служать заміною релейно-контактних схем управління, які збираються з

окремих частин (таймери, лічильники, реле), тим самим забезпечуючи надійний і економічний режим роботи енергообладнання при невеликому числі обслуговуючого персоналу. В даному проєкті розроблена автоматизована система управління (АСУ) установки зворотнього осмосу на базі контролера компанії Siemens SIMATIC S7-1414.

У даних контролерів є своє програмне забезпечення (ПЗ), яке називається Totally Integrated Automation Portal (TIA Portal), призначене для розробки систем автоматизації технологічних процесів від рівня контролера до рівня людино-машинного інтерфейсу. У TIA Portal інтегровані два програмних пакета, це Simatic Step 7 і Simatic WinCC, що робить рішення задач автоматизації найбільш простим і ефективним, а передові технології забезпечує зручний користувальницький інтерфейс.

Новизною проєкта є розробка АСК установки зворотнього осмосу РО/ДАТ - 2/0,4 з використанням інтегрованого середовища програмування TIA Portal v13 і вирішені наступні завдання:

- розроблена функціональна схема автоматизації установки;
- розроблено схему рівня перміату в накопичувальній ємності ;
- розроблена схема відкриття електромагнітного клапану промивки при перевищенні уставки за командою датчика електропровідності;
- розроблена схема відкриття електромагнітного клапану промивки при перевищенні тиску в установці.

УДК 004.927

Олійник А.О.¹, Федорченко Є.М.², Рудь М.С.³

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-413м ЗНТУ

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЕВОЛЮЦІЙНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА ПРИ РОЗПІЗНАВАННІ ОБРАЗІВ

Проблема швидкого надання клієнту будь-якої кількості ліків породила проблему автоматизації аптечного господарства, були придумані й розроблені різного роду засоби для полегшення обробки замовлень медпрепаратів клієнтів та їх швидкої видачі.

Ця задача являє собою класичну оптимізаційну проблему що важко вирішується традиційними техніками. Її ціль – знайти найкоротший шлях для комівояжера який повинен відвідати N точок. Цей тип задач існує у багатьох формах, з кількома інженерними застосуваннями що включають оптимальне

розташування газового трубопроводу, дизайн антени системи подачі, конфігурація транзисторів на платах, або сортування об'єктів для отримання найбільш відповідної конфігурації. Ейлер сформулював задачу комівояжера у 1759 році, і вона була формально названа й представлена корпорацією “Rand” у 1948 році [1, 2].

Стандартним шляхом рішення даної задачі є метод брутфорсу, або грубої сили, коли ми розраховуємо кожен з можливих шляхів та обираємо найкращий. Для n точок існує $(n-1)!$ можливих шляхів. Наприклад, для 10 точок кількість шляхів дорівнює 3628800, а для 20 дорівнює 2432902008176640000.

Існують методи що дозволяють знайти близькі до оптимуму рішення, наприклад, алгоритм найближчого сусіда та метод рою часток. Ці методи дозволяють знайти “непогане” рішення задачі комівояжера з довільною швидкістю.

Як засіб оптимізації, у розробленій системі був використаний генетичний пошук, розроблений Джоном Холандом у 1970-х роках та популяризований його студентом Девідом Голдбергом [1-3].

Розроблена система дозволяє вирішити основну задачу та має додаткові функції як то можливість задавати початкові параметри генетичного алгоритму, графічне відображення процесу вирішення задачі комівояжера, графічне та текстове представлення результатів роботи системи.

У розробленій системі було запропоновано модифікацію генетичного алгоритму (ГА) – а саме 3 методи генерації початкової популяції. У таблиці 1 наведено характеристику розроблених методів.

Таблиця 1 – Розроблені методи генерації початкової популяції генетичного алгоритму

Назва методу	Опис
Стандартний метод (Default)	При використанні цього методу при генерації початкової популяції кожен індивід буде складатися з однакової послідовності генів – номерів відвідуваних точок розташованих у порядку зростання від першого до останнього
Змішаний метод (Random)	При використанні цього методу при генерації початкової популяції кожен індивід буде складатися з різних послідовностей генів – номерів відвідуваних точок розташованих у випадковому порядку
Змішаний оптимальний (Random optimal)	При використанні цього методу при генерації початкової популяції кожен індивід буде складатися з однакових послідовностей генів, кожна з яких є оптимальним шляхом серед вказаної користувачем кількості згенерованих змішаних шляхів

На відміну від існуючих методів, модифікована версія ГА дозволяє обирати метод ініціалізації початкової популяції при вирішенні задачі комівояжера, що, у свою чергу, дозволяє на етапі ініціалізації генерувати більш пристосовані хромосоми (хромосоми з кращими значеннями функції пристосованості) та таким чином покращити результати роботи алгоритму.

Запропоновані модифікації дозволяють суттєво підвищити швидкість за рахунок того що при використанні стандартного методу ініціалізації початкової популяції, початкове значення розрахованої функції пристосованості та відповідно дистанції буде меншим ніж при використанні інших методів ініціалізації, що дозволить більш точно визначити оптимум задачі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Олійник А. О. Еволюційні обчислення та програмування : навч. пос. / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2010. – 324 с.
2. Haupt, R. Practical genetic algorithms / Randy L. Haupt, Sue Ellen Haupt. – 2004. – 261 p.
3. Gen M. Genetic algorithms and engineering design / M. Gen, R. Cheng. – New Jersey: John Wiley & Sons, 1997. – 352 p.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Харченко А.С.³, Гончаренко Д.А.⁴,
Косміна О.В.⁵

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-116 ЗНТУ

⁴ студ. гр. КНТ-216 ЗНТУ

⁵ студ. гр. КНТз-113м ЗНТУ

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я КУРЦІВ

Куріння - одна з найважчих проблем нашого часу, причина багатьох хвороб і передчасної смертності людини в сучасному світі. Щорічно в світі куріння стає причиною прямої або опосередкованої смерті 3 мільйонів чоловік. У зв'язку з цим все більшої актуальності набувають новітні методи, які можуть допомогти курцеві впоратися зі своєю небезпечною пристрастю. Відмовитися від тютюну неймовірно важко – ми боїмося стандартних побічних ефектів (різкий набір ваги, депресії, головні болі та ін.). Але це не

повинно бути виправданням для того, щоб махнути на себе рукою. М'який, поступовий, прогресивний метод відмови від куріння існує [1]. Розроблено мобільний додаток Cigarette Analytics, що аналізує інформацію, пов'язану з палінням користувача, та допомагає, завдяки низці методів, позбутися нікотинової залежності.

У додатку використовується поняття Змінної середньої. У статистиці змінна середня або ковзної середнє (англ. Moving Average) являє собою послідовну серію середніх значень з певним періодом згладжування і розраховується з метою визнання тенденції зміни випадкової величини. При цьому чим більше буде період згладжування, тим більше плавним буде графік. Цей показник не тільки використовується в якості самостійної методики при проведенні технічного аналізу, але і лежить в основі ряду інших індикаторів.

У статистиці і при проведенні технічного аналізу широко використовують різні різновиди ковзної середньої: проста, експоненціальна і зважена [2]. У додатку використовується виключення аномальних значень [3]. Рішення задач контролю, прогнозування і моделювання зазвичай передбачає обов'язкову обробку статистичної інформації про спостережуваному процесі. Зазвичай під терміном статистична обробка даних розуміють (виключення аномальних значень, статистичне згладжування, перевірку стаціонарності, формування інтервального оцінки поточного середнього). Виняток аномальних значень в статистичних даних зазвичай виконують із використанням користування наступних алгоритмів (алгоритму на основі статистики Діксона, алгоритму на основі t -статистики, алгоритму на основі "стандартної статистики"). Кожен з цих алгоритмів в певних умовах самодостатній, однак послідовне їх застосування виключає можливість помилкового віднесення того чи іншого значення параметрів спостереження в категорію "викиду". Віднесення даного в категорію «викиду» здійснюється лише в разі, коли не менше двох з трьох алгоритмів свідчать про це. Пропоновані алгоритми інваріантні як для мінімальних, так і для максимальних значень вибірки.

Розроблений додаток Cigarette Analytics, що допомагає курцям кинути палити і забезпечує виконання нижче перерахованих функцій, коректне завантаження інформації введеної користувачем до бази даних що до кількості випалених цигарок, часу затраченого на паління, марок цигарок та іншої корисної інформації, можливість змін та видалення інформації у базі даних, можливість детальної конфігурації додатку, що буде враховувати індивідуальні особливості кожного користувача, надання користувачу можливості користування віджетами додатку, можливість задання цілей що будуть допомагати кинути паління, відображення детальної статистики щодо пагубної звички, відображення мотивуючої інформації у режимі "Не палю",

можливість експорту та імпорту інформації у файл та з файлу, можливість користуватись додатком на кількох мовах.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено мобільний додаток Cigarette Analytics, що аналізує інформацію, пов'язану з палінням користувача, та допомагає, завдяки низці методів, позбутися нікотинової залежності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Андреева, Т.И. Табак и здоровье [Текст] / Т. И. Андреева, К. С. Красовский. – К., 2004. – 224 с.

2. Грешилов, А.А. Математические методы построения прогнозов [Текст] / А.А. Грешилов, В.А. Стакун, А.А Стакун. – М.: Радио и связь, 1997. – 112 с.

3. Зайдель, А.Н. Элементарные оценки ошибок измерений [Текст] / А.Н. Зайдель. – М.: Наука, 1965. – 120 с.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Харченко А.С.³, Гончаренко Д.А.⁴, Кожушаний С.І.⁵

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-116 ЗНТУ

⁴ студ. гр. КНТ-216 ЗНТУ

⁵ студ. гр. КНТ-413м ЗНТУ

ПРОГНОЗУВАННЯ ПОГОДИ

Відомий вислів «про погоду поговорити кожен любить...» відображає той великий інтерес, який проявляє людина до навколишнього середовища. Цей інтерес зачіпає і окремих людей, і їх спільноти, і нації, і людське суспільство в цілому. Щодобове зведення погоди, наприклад, примушує людину приймати різні рішення: чи брати йому парасольку, чи займатися пошуком притулку від урагану, що насувається та інше. У розвинутих країнах погода та клімат давно стали категоріями економічними. Кожного року у світі стихійні нещастя забирають близько 250 000 людських життів, розмір збитків, які наносяться майну, лежить у межах 50-100 млрд. американських доларів. Але світова статистика показує: якщо довіряти гідрометеорологічній інформації та адекватно на неї реагувати, то можна відвернути від 30 до 40% втрат і повністю уникнути людських жертв [1]. Особливо помітний

економічний ефект дає використання метеорологічної інформації в авіації, енергетиці, будівництві, риболовстві та судноплаванні, сільському господарстві.

Прогнози бувають синоптичні, чисельні, статистичні (фізико-статистичні) та інші. Методи дослідження прогнозу погоди бувають візуальні, статистичний та спектральний аналіз даних, метод неправильних сусідів визначення аргументів модельних функцій, методи апроксимації функцій багатьох змінних за допомогою узагальнених багаточленів, радіальних базисних функцій, штучних нейронних мереж тощо. Не дивлячись на безмежну кількість ситуацій, об'єктів та цілей, які вносять у процес свою специфіку, можна виділити основні етапи моделювання прогнозу погоди [2-4].

На першому етапі здійснюється отримання даних та їх систематизація. Далі застосовується один або декілька методів аналізу наявних тимчасових рядів метеовеличин, що спостерігаються. Це, наприклад, візуальний аналіз у вигляді графіків залежності змінної від часу, відновлення фазової траєкторії, спектральний та статистичний аналіз тощо.

На другому етапі формується структура моделі. Першочергово обирається тип рівнянь, далі задається вид вхідних у них функцій, після чого встановлюється зв'язок динамічних змінних (компонент вектору x) із величинами a . В якості змінних можуть виступати власні спостережувані змінні, але у більш загальному випадку цей зв'язок задають у вигляді $a = h(x)$, де h називають вимірювальною функцією.

Новизною проекту є розробка програми, метою якої є візуальне відображення погодних умов у вашому регіоні. Можливо відображення погоди на поточний день, на тиждень. Підтримується більше тисячі міст, інформація про погодні умови автоматично оновлюється кілька разів на добу.

Програмний продукт вирішує наступні завдання:

- програмний продукт вміє прогнозувати погоду самостійно, використовуючи спеціальні розрахунки та отримувати дані від метеорологічних станцій або опираючись на дані, отримані від сторонніх сервісів (веб-сервіси, тощо);

- програмний продукт відображає прогноз погоди у зручному форматі для користувача;

- програмний продукт повинен відображати вищеописані методи прогнозу погоди (короткостроковий та довгостроковий);

- програмний продукт надає можливість вибору прогнозування погоди у різних регіонах;

- програмний продукт відображає графік зміни прогнозу погоди за вибраний період.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кому і навіщо потрібен прогноз? [Електронний ресурс]. – Режим доступу: http://www.primpogoda.ru/articles/prosto_o_pogode/komu_i_zachem_nuzhen_prognoz/
2. Немного истории о метеорологии. [Электронный ресурс]. – Режим доступу: http://www.primpogoda.ru/articles/prosto_o_pogode/o_meteorologii/
3. Дымников В.П. Основы математической теории климата [Текст] / В.П. Дымников, А.Н. Филатов. – Москва, 1994. – 254 с.
4. Гордин В.А. Математика, компьютер, прогноз погоды [Текст] / В.А. Гордин. – Л.: Гидрометеиздат, 1991. – 132 с.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Гончаренко Д.А.³, Харченко А.С.⁴, Малашок Н.М.⁵

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-216 ЗНТУ

⁴ студ. гр. КНТ-116 ЗНТУ

⁵ студ. гр. КНТ-213м ЗНТУ

ЕКСПЕРТНА СИСТЕМА СПІВВІДНОШЕННЯ ЦІНА-ЯКІСТЬ ЛІКАРСЬКИХ ПРЕПАРАТІВ

Однією з найважливіших тенденцій сфери охорони здоров'я є підвищення рівня медичної допомоги і обслуговування. Лікарські засоби (ЛЗ) і вироби медичного призначення (ВМП) – це не тільки соціально значущий, але і життєво важливий товар.

Ціна, якість медичних препаратів, їх своєчасне замовлення, постачання - одні з найважливіших економічних характеристик в сегменті охорони здоров'я. Порушення товарообігу (нестача товарних запасів, завищена ціна, обмежений асортимент, низька якість) може привести до зменшення прав і законних інтересів громадян, зокрема конституційного права на охорону здоров'я та якісну медичну допомогу [1].

Адже, невід'ємною складовою медичної сфери є своєчасне і якісне лікарське забезпечення, яке залежить від наявності необхідних лікарських засобів ЛЗ і ВМП в лікувально-профілактичних закладах і аптечних пунктах. Важливим залишається і те, яка якість та ціна препаратів, чи наявні вони в необхідній кількості на даний момент.

Наукова новизна даної роботи полягає в тому, що завдяки комплексній оцінці фармацевтичного ринку була розроблена організаційно-економічна модель забезпечення ЛЗ і ВМП. За допомогою методичного підходу встановлені закономірні особливості сучасного сегменту ринку фармацевтичних препаратів, зокрема тенденція до збільшуючого поширення взаємозамінних препаратів, перевагою цінового показника над якістю продукту, невідповідністю між потребою споживача і представленим асортиментом в медичних закладах (аптеках) [2].

Розроблена організаційно-функціональна модель закупок дозволить: упорядкувати процес, підвищити відповідальність за якість і забезпечити економію коштів і часу, підвищити ефективність та результативність роботи лікувально-профілактичних закладів і аптекних пунктів, налагодити партнерські відносини з надійними споживачами ЛЗ, мінімізувати простой обладнання і працівників за рахунок своєчасної реалізації ЛЗ, досягти незначних термінів збереження на складах препаратів.

В процесі дослідження використовувалися методи системного, економіко-математичного, логістичного, статистичного, ретроспективний і перспективний аналізів. А також засоби стратегічного, сучасного маркетингу, документального дослідження. Обробка отриманих результатів здійснювалася з використанням сучасних математичних методів і комп'ютерних технологій.

В результаті виконання роботи було досліджено процес автоматизації замовлення лікарських засобів і виробів медичного призначення для фармацевтичних відділень – аптек.

Створений програмний продукт може бути використана для підвищення ефективності роботи фармацевтичних закладів (аптекних пунктів, магазинів спеціального медичного обладнання, оптики, діагностичних центрів), зокрема: переналаштувати асортимент препаратів згідно потреб цільової аудиторії (задовольнити потреби споживачів відповідно до ціни і якості ЛЗ), скласти правильну цінову політику, що залучає відвідувачів, вибірково формувати маркетинговий бюджет в відповідності з цінністю того чи іншої групи. (перехід від стратегії продаж «від категорій і брендів» до праці «зі споживачем»), продуктивно проводити політики завоювання лояльності споживачів (наприклад, ціленаправлена праця зі споживачами в окремих групах, наприклад – товари для дітей чи лікувальна косметика), звільнити оборотні кошти, що витрачаються на замовлення неліквідного асортименту та збільшити дохідну частини.

Практичною складовою є використання класичних методів економічних і маркетингових досліджень у поєднанні з методом глибинного аналізу даних (Data Mining), як більш якісного виявлення і задоволення потреб споживачів в лікарських засобах (життєво важливих і необхідних).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Законодавство України [Електронний ресурс]. – Режим доступу: <http://zakon.rada.gov.ua/laws/show/179-98-%D0%BF>

2. Аналіз цінової політики підприємства [Електронний ресурс]. – Режим доступу: <http://studentbooks.com.ua/content/view/172/39/1/1/>

УДК 004.9

Рисіков В.П.¹, Степаненко О.О.¹, Федорченко Є. М.², Скачко Л. П.³, Харченко А.С.⁴, Гончаренко Д.А.⁵, Плешко П.В.⁶

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ асист. ЗНТУ

⁴ студ. гр. КНТ-116 ЗНТУ

⁵ студ. гр. КНТ-216 ЗНТУ

⁶ студ. гр. КНТ-713м ЗНТУ

ПРОГРАМНА МОДЕЛЬ ВИЖИВАННЯ АГЕНТІВ У ШТУЧНОМУ СЕРЕДОВИЩІ

Сьогодні існує досить багато сучасних моделей штучного середовища, розміри, можливості та призначення яких є досить різноманітним. Одні орієнтовані на пошук оптимальних рішень, інші створені для передбачення виняткових ситуацій у межах побудованої моделі.

Наукова новизна роботи полягає у:

– визначенні функціонального мінімуму агента, необхідного для повноцінного його розвитку в умовах даної штучної системи;

– в моделі агента застосований механізм побудови поведінки схожий із сильно спрощеною формою механізму поведінки живих істот, у системі геном – це, набір послідовних команд (генів), а у живих істот геном – це набір спадкового матеріалу, який згодом буде диктувати правила поведінки істоти;

– застосування модифікацій до геному агента, подібних до модифікацій які можуть відбутися у геномі живих істот, що в результаті може привести як до зниження популяції так і для її підвищення;

– структура створеної програми дозволяє шляхом модифікації коду розширювати здатності штучної системи та її агентів до будь-яких розмірів, що в перспективі може її зробити більш спеціалізованою або мультифункціональною.

Програмна модель штучного світу призначена для користувачів, які бажають дізнатися вірне рішення для проблеми, на яку впливають різні фактори ззовні, дана модель побудована на базі моделі виживання і пристосування найпростіших агентів до умов зовнішнього середовища де рішенням задачі буде момент коли тривалість життя агентів перетне встановлений поріг. Основою для нашої програмної моделі штучного середовища виступає концепція природного відбору, яка у свою чергу базується на механізмах використовуваних в «еволюційних алгоритмах» та механізмах більш відомих під назвою «аналіз виживання» [1]. Основним призначенням розробленого програмного засобу є автономний пошук моделлю правильного рішення в межах спеціально створених умов, від яких це рішення повинне залежати. За приклад було взято модель виживання простих організмів в умовах штучно створеного середовища. Інтерфейс програми має бути простим та зрозумілим.

Розроблена програма відповідає усім основним вимогам для ефективної роботи програмної моделі штучного світу. Модель визначає функціональний мінімум агента, якого достатньо для повноцінного розвитку в умовах даної штучної системи. Модель побудована із застосуванням механізмів поведінки, які схожі на сильно спрощену форму механізму поведінки живих істот, у системі геном - це, набір послідовних команд(генів), а у живих істот геном - це набір спадкового матеріалу, який згодом буде диктувати правила поведінки істоти. Також в моделі було застосовано модифікації до геному агента, подібні до модифікацій які можуть відбутися у геномі живих істот, що в результаті може привести як до зниження популяції так і до її підвищення. Структура же програми дозволяє шляхом модифікації коду розширювати здатності штучної системи та її агентів до будь-яких розмірів, що в перспективі може її зробити більш спеціалізованою або мультифункціональною.

Отримані результати демонструють працездатність моделі та ефективність методу пошуку рішень, запропонованого посередництвом використання цієї моделі.

Розроблена програма виконує такі функції: автономно функціонує допоки необхідне рішення не буде знайдене, за необхідністю підтримує можливість втручання оператором у процес пошуку рішення, спираючись на результати видає оператору детальну інформацію.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Lohn, J. An Evolved antenna for deployment of NASA's space technology 5 mission [Electronic resource] / [Jason D. Lohn, Gregory S. Hornby, Derek S. Linden]. – Access mode: <http://www.human-competitive.org/sites/default/files/lohn-paper.pdf> (дата звернення: 10.12.2018).

2. Шварц Б. MySQL. Оптимизация производительности [Текст] / Б. Шварц. – Москва : «Символ-Плюс», 2010. – 723 с.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Гончаренко Д.А.³, Харченко А.С.⁴,
Короткий О.В.⁵

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-216 ЗНТУ

⁴ студ. гр. КНТ-116 ЗНТУ

⁵ студ. гр. КНТ-113м ЗНТУ

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ПОБУДОВИ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ МЕДИЧНОЇ СТАТИСТИКИ

Сучасні ІТ розробки позитивно впливають на розвиток нових способів організації медичної допомоги населенню. Проведення онлайн-консультацій для пацієнтів і персоналу, обмін інформацією про хворих між різними установами, дистанційне фіксування фізіологічних параметрів, контроль за проведенням операцій в реальному часі - всі ці можливості дає впровадження інформаційних технологій в медицину. Це виводить інформатизацію охорони здоров'я на новий рівень розвитку, позитивно позначаючись на всіх аспектах його діяльності.

Під час виконання роботи було розроблено програмний комплекс призначений для синтезу та навчання НМ MLP, за допомогою алгоритмів зворотного поширення помилки, Delta Bar Delta, пакетного зворотного поширення помилки, пружного зворотного поширення помилки (Resilient Back-propagation) та алгоритму спряжених градієнтів. Також , було виконано дослідження та програмна реалізація методів створення та навчання нейронних мереж з метою їх подальшого використання для роботи з медичною статистикою.

В процесі виконання роботи були вирішені наступні задачі:

- проведено аналіз предметної області;
- з'ясовано вимоги до програмного забезпечення;
- розроблено систему, що реалізує роботу з медичною статистикою, а також програмне рішення для неї.

Функціями програмного засобу є створення, модифікація, навчання та використання нейронних мереж для роботи з медичною статистикою.

Ідея використання нейронних мереж для задач діагностики та прогнозування помітно спрощує процес оцінки технічного стану об'єктів. Під штучними нейронними мережами (НМ) передбачають обчислювальні структури, що складаються з великої кількості однотипних елементів, кожен з яких виконує прості функції.

Причини, що послужили застосуванню НМ в задачах діагностики та прогнозування:

- для реалізації нейромережових алгоритмів необхідна мінімальна інформація про об'єкт;

- при реалізації НМ можлива паралельна обробка інформації, що, по-перше, значно збільшує швидкість роботи системи і, по-друге, підвищує надійність системи [1].

Практична значимість даної роботи полягає в можливості створення нейронної мережі для роботи з різноманітною медичною статистикою, яка значно розширить можливості з аналізу та прогнозування даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Штучна нейронна мережа [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Харченко А.С.³, Гончаренко Д.А.⁴,
Бережняк О.О.⁵

¹ канд. техн. наук, доц. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-116 ЗНТУ

⁴ студ. гр. КНТ-216 ЗНТУ

⁵ студ. гр. КНТ-413м ЗНТУ

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ МОВ ПРОГРАМУВАННЯ СЕРВЕРНИХ СКРИПТІВ ТА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ РЕАЛЬНОГО ЧАСУ З ВИКОРИСТАННЯМ NODEJS ТА WEBSOCKET

Ні для кого не секрет, що за останні 10 років веб-розробка сягнула захмарних висот, включно з її методологіями, неймовірно широким інструментарієм та стеком технологій на будь-який смак. Якщо подивитися уважно - сьогодні саме розробка веб та мобільних додатків і сервісів складає найбільший процент за поширенням і популярністю попиту, як серед замовників продуктів, так і серед спеціалістів на ринку праці - ця галузь не

настільки нова, щоб про неї мало хто знав, або було небагато матеріалів і поширеності, щоб її було складно опанувати, але при цьому вона досі дуже актуальна і поки не планує зменшувати оберти.

Як було вказано раніше кількість та варіаційність технологій дозволяють зробити один і той самий додаток багатьма способами, що зумовлено різноманітністю самого стеку. Нині, як в Україні, так і в будь-якій іншій країні, дуже велику популярність мають реал-тайм додатки. Найпоширенішим варіантом реалізації такого додатку – є чат. Раніше, до появи такого протоколу як WebSocket, програмісти мусили використовувати long polling – це посилання запросу на сервер, та очікування певної дії, у цей час з'єднання з сервером є неперервним. Це призводить до затрати великих ресурсів, як технічних так і фінансових щоб сервер працював стабільно, бо він повинен тримати велике навантаження. Також, основну роль у цьому відіграє вибір мови програмування, яка буде використана на стороні API серверу.

На це впливає багато факторів, такі як: швидкість розробки, ціна розробки, кількість потенціальних користувачів та які функції матиме додаток.

Правильно підібраний стек технологій для реалізації додатку, дозволить знизити кількість часу на вирішення проблем під час розробки, унаслідок чого зменшити собівартість додатку для потенціальних замовників, а так дозволить помітно збільшити ефективність такого додатку за рахунок використання особливостей вибраної мови програмування [1, 2]. Проводячи аналіз та дослідження мов програмування, ми побачили, що JavaScript має найбільшу швидкість розробки що допомагає заощадити кошти, проте, не підходить для складних розрахунків на стороні серверу. Також, один з найбільших недоліків NodeJS (JavaScript) є те, що він являється однопоточним, що не дозволяє правильно виконувати певні дії. Java – являється більш ресурсоемною для розробки додатку. Потребує більше часу, що, в свою чергу, змушує витратити більше коштів. Проте, це одне з найефективніших рішень для високонавантажених додатків. Python – одна з найпопулярніших мов програмування для нейромереж, швидкість розробки більша ніж у Java, однак менша ніж у JavaScript. Використовується для реалізації складних додатків. Розроблена програма представляє собою клієнт-серверне застосування у вигляді веб-проекту, який складається з трьох частин: VueJS клієнту, NodeJS та WebSocket серверу. Програма призначена для демонстрації можливостей WebSocket-серверу, який використовується в додатках реального часу.

Функції програмного продукту: реєстрація користувачів, авторизація користувачів, перегляд списку повідомлень, відправлення повідомлення, перегляд списку користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Схема клієнт-серверної архітектури. [Електронний ресурс]: Режим доступу: [//ru.wikipedia.org/wiki Клиент_—_сервер#/media/File:Two-tier_architecture.ru.svg](http://ru.wikipedia.org/wiki/Клиент_—_сервер#/media/File:Two-tier_architecture.ru.svg)
2. NodeJS. [Електронний ресурс]: Режим доступу: <https://nodejs.org/uk/>

УДК 336.221

Дубровін В.І.¹, Дейнега Л.Ю.², Джрагацпанян Н.В.³, Соколовський Д.В.⁴

¹ канд. техн. наук, проф. ЗНТУ

² старш. викл. ЗНТУ

³ студ. гр. КНТ-117 ЗНТУ

⁴ студ. гр. КНТ-137 ЗНТУ

ОПТИМІЗАЦІЯ ОПОДАТКУВАННЯ

Політика оподаткування є частиною економічної політики, що реалізується в сфері податкових відносин. Податкове планування й оптимізація оподаткування грають суттєву роль в діяльності всіх типів підприємств. Для платника податків природно прагнути до максимального зменшення податкового зобов'язання.

Для вибору варіанту системи оподаткування найбільш відповідної до інтересів і платника податків, і держави пропонується використовувати апарат теорії ігор. Теорія ігор - це математична теорія конфліктних ситуацій, що розробляє рекомендації найбільш раціональним способом дій кожного з учасників, тобто таких дій, які забезпечили б йому найкращий результат. У грі можуть стикатися інтереси двох або декількох супротивників. Завданням теорії ігор є вироблення рекомендацій для гравців, тобто визначення оптимальної стратегії.

Держава, проводячи податкову політику, прагне отримати максимальний дохід в бюджетну систему. З іншого боку, платник податків прагне мінімізувати податкові платежі.

Нехай є надходження сум (a_1, a_2, \dots, a_n) податкових платежів за роками (t_1, t_2, \dots, t_n) при існуючій системі оподаткування і варіанти пропонованої системи оподаткування (b_1, b_2, \dots, b_n) .

Варіанти пропонованої та чинної системи оподаткування характеризуються розміром надходжень до бюджетної системи. Наприклад, ситуація характеризується тим, що держава діє за варіантом b_1 , а надходження сум дорівнює a_2 . Тоді умовний дохід держави складе: $b_1 - a_2 = ab_{12}$.

Поступаючи аналогічно з усіма можливими варіантами дій конфліктуючих сторін, отримаємо матрицю гри (табл. 1).

Таблиця 1 – Матриця гри

		Платник податків				
		1	2	3	...	n
Держава	1	ab_{11}	ab_{12}	ab_{13}	...	ab_{1n}
	2	ab_{21}	ab_{22}	ab_{23}	...	ab_{2n}
	3	ab_{31}	ab_{32}	ab_{33}	...	ab_{3n}

	n	ab_{n1}	ab_{n2}	ab_{n3}	...	ab_{nn}

Додамо до кожного елементу значення максимального елементу матриці $\max ab_{ij}$, для того щоб всі елементи матриці можна було звести до двох зв'язаних задач лінійного програмування. При цьому ціна гри збільшується на:

$$G_{ij} = ab_{ij} + \max ab_{ij}$$

Для отримання оптимальної змішаної стратегії для платника податків треба скласти дві задачі лінійного програмування, використовуючи отриману матрицю: задачу мінімізації середнього програшу платника податків та задачу максимізації отриманого податку.

Ціну гри можна визначити за формулою: $1/F_{\min} \leq 1/F_{\max}$.

А істинна ціна гри знаходиться на інтервалі: $[1/F_{\min}; 1/F_{\max}]$.

Оптимальною стратегією платника податків и держави є той варіант, який входить до інтервалу істинної ціни.

Застосування на практиці елементів теорії гри дозволяє вибрати оптимальні моделі оподаткування, які найбільшою мірою відповідають інтересам платника податків і держави.

Отже, застосування на практиці елементів теорії ігор дозволяє вибрати оптимальні моделі оподаткування, які найбільшою мірою відповідають інтересам платника податків і держави.

FLUTTER – НОВИЙ ПІДХІД У РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ

Flutter – це мобільний додаток GoogleSDK для створення високоякісних власних інтерфейсів на iOS і Android за дуже короткий час, а також це єдиний спосіб розробки додатків під Google Fuchsia. Flutter працює з існуючим кодом, використовується розробниками та організаціями по всьому світу, і є вільним і відкритим вихідним кодом [1].

Flutter – SDK ((від англ. software Development Kit) — набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи (програмної або програмно-апаратної)), дозволяє створювати гібридні мобільні додатки. Зокрема, це означає можливість використання однієї і тієї ж кодової бази на платформах Android і iOS. Код пишуть на Dart, мовою розробленою Google [2].

Як React Native, Flutter також має реактивний стиль. Flutter використовує інший підхід, щоб уникнути проблем з продуктивністю, викликаних необхідністю моста JavaScript. Dart компілюється «заздалегідь» (AOT) у нативний код для декількох платформ. Це дозволяє Flutter спілкуватися з платформою, не переходячи через мост JavaScript. Компіляція до нативного коду також покращує час запуску програми [3].

Flutter революційно реалізує віджети. Віджети - це елементи, які впливають і керують переглядом і інтерфейсом до програми. Віджети є однією з найважливіших частин мобільної програми на Flutter. Фактично, вони самі можуть зробити програму [4].

Flutter має нову архітектуру, яка включає віджети, які добре працюють і виглядають, є швидкими, налаштовуються і розширюються. Flutter не використовує віджети платформи (або DOM WebViews), він надає свої власні віджети.

Flutter піднімає віджети і візуалізатор з платформи в додаток, що дозволяє їм бути налаштованим і розширюваним. Все, що вимагає Flutter від платформи, – це полотно, в якому відображаються віджети, щоб вони могли з'являтися на екрані пристрою, і мати доступ до дій (дотиків, таймерів тощо) і сервісів (розташування, камера і т.д.) [5].

Існує ще інтерфейс між програмою Dart і власним нативним кодом платформи (iOS або Android), який кодує та декодує дані, але це може бути на порядок швидше, ніж мост JavaScript [6].

У Dart є сховище програмних пакетів, що дають змогу розширити можливості програм. Наприклад, існує цілий ряд пакетів, які полегшують доступ до Firebase для створення додатка без сервера. Існують також пакети під назвою "плагіни", які полегшують доступ до платформних послуг та апаратних засобів, таких як акселерометр або камера, в ОС-незалежний спосіб [7].

Завдяки такому підходу створення користувацького інтерфейсу полягає у збірці макета з безлічі різних невеликих елементів [8].

Розглянутий принцип поєднується з ще однією базовою ідеєю, на якій заснований Flutter – віддається перевага композиції перед успадкуванням. Це означає, що якщо вам потрібно створити якийсь новий віджет, то, замість того, щоб розширювати клас Widget (в Android це робиться шляхом розширення класу View), новий віджет збирають з декількох існуючих [9].

Дуже особлива можливість Flutter – гаряче перезавантаження. Гаряче перезавантаження – це механізм з високою швидкістю роботи. До того ж він дає можливість змінити код, що описує інтерфейс, в процесі виконання програми зі збереженням раніше введених даних [9].

Ще одна властивість Flutter полягає в тому, що в цій бібліотеці є вражаючий набір вбудованих компонентів для інтерфейсу користувача. Насправді, тут присутні два набори віджетів – віджети в стилі Material Design (для Android) і в стилі Cupertino (для iOS). З них можна вибирати те, що вам потрібною Найприємніше – це крос-платформність. Скажімо, якщо в проєкт доданий який-небудь віджет в стилі Material Design або Cupertino, він буде виглядати однаково на будь-якому пристрої на платформі Android або iOS. В результаті розробник може не турбуватися про зовнішній вигляд створеного ним інтерфейсу на різних пристроях [10].

Зазвичай потрібно, щоб додатки для Android виглядали не так, як додатки для iOS. Відмінності полягають не тільки в кольорах елементів інтерфейсу, але і в розмірах, і в стилі віджетів. Забезпечити правильне відображення інтерфейсів додатків на різних платформах можна за допомогою тем, якими володіє Flutter.

Отже, основні переваги Flutter: [11]

- переваги реактивних переглядів, без мосту JavaScript;
- швидкий, плавний і передбачуваний; код компілює AOT на нативний код;
- розробник має повний контроль над віджетами та макетом;
- містить віджети, що налаштовуються;
- багато інструментів для розробників, з гарячим перезавантаженням;
- більш продуктивний, більш сумісний, більш комфортний у використанні;

– створення програми для декількох платформ з однієї кодової бази, не відмовляючись від продуктивності або потужності.

Хоча Flutter все ще перебуває в альфа-версії, ця бібліотека вже зібрала навколо себе дуже велику і активну аудиторію. Завдяки цьому Flutter підтримує безліч пакетів (бібліотек, наприклад залежність Gradle в Android). Серед доступних пакетів є такі, які допомагають працювати з зображеннями, виконувати HTTP-запити, ділитися даними, зберігати налаштування, працювати з сенсорами, користуватися можливостями Firebase. І все це можна застосовувати і при розробці для Android, і при розробці для iOS.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What's Revolutionary about Flutter [Електронний ресурс]. – Режим доступу: <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514> (дата звернення: 11.03.2019).

2. Пять причин проникнуться симпатией к Flutter [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/ruvds/blog/349622/> (дата звернення: 11.03.2019).

3. Flutter (SDK) [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Flutter_\(SDK\)](https://ru.wikipedia.org/wiki/Flutter_(SDK)) (дата звернення: 11.03.2019).

4. Build beautiful native apps in record time [Електронний ресурс]. – Режим доступу: <https://github.com/flutter/flutter> (дата звернення: 11.03.2019).

5. Introduction to widgets [Електронний ресурс]. – Режим доступу: <https://flutter.dev/docs/development/ui/widgets-intro> (дата звернення: 11.03.2019).

6. Flutter [Електронний ресурс]. – Режим доступу: <http://developer.alexanderklimov.ru/android/theory/flutter.php> (дата звернення: 11.03.2019).

УДК 004.89

Короткий О.В.¹, Горбачов В.С.²

¹ викл. ЗЕТК ЗНТУ

² студ. гр. РПЗ 16 2/9 ЗЕТК ЗНТУ

ЗАСОБИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗШИРЕННЯ МОЖЛИВОСТІ ЛЮДЕЙ

У сучасних вчених є гарна новина для винахідників. Протягом наступних 20-ти років. Те, як ми працюємо зміниться більше, ніж за останні 2 тисячі років. Більш того, є припущення, що людство знаходиться на порозі

нової ери в історії. Відомо 4 основні історичні епохи, що розрізняються за характером нашої праці:

- перша епоха: мисливців-збирачів тривала кілька мільйонів років;
- потім кілька тисяч років припали на епоху землеробства;
- промислова епоха тривала кілька століть;
- сучасна інформаційна епоха почалася всього кілька десятиліть тому;

І зараз починається нова епоха розвитку людства Епоха вдосконалення. В цю еру природні здібності людини будуть розширені обчислювальними системами, що допомагають людям думати, роботизованими системами, що допомагають будувати і цифровий нервовою системою, яка виводить нас далеко за межі почуттів.

Останні пару мільйонів років всі інструменти, які люди використовували до цього були абсолютно пасивні. Вони виконують тільки те, що від них вимагають. Не більше. Наш найперший інструмент різав тільки те, куди ми його направляли. Звичайний різак висікає тільки там, куди його направляє майстер. Навіть найсучасніші інструменти не працюють без чіткого керівництва. До сьогоднішнього дня люди зобов'язані були вручну вкладати свої задумки в наші інструменти. Найсучасніші програми працюють тільки за рахунок того, що програмісти прописують усі можливі варіанти розвитку подій. Однак, вже зараз найсучасніші вчені і програмісти працюють над тим, щоб зробити інструменти «активними». Основною задачею стає взаємодія з технікою, коли людина задає рамки та напрямки для роботи програми, а вона вже займається реалізацією.

В 2016 році вчені з США створили перший квадрокоптер за допомогою штучного інтелекту. В програму для моделювання вони вбудували штучний інтелект. Вказали конкретні умови: у квадрокоптера повинно бути 4 винта, максимально легка маса, гарний аеродинамічний КПД. І програма моделює мільйони можливих варіантів моделей, які задовольняють заданим умовам. І в результаті цього експерименту було створено неймовірний корпус для квадрокоптера з надзвичайно дивною формою, до якої людський мозок ніколи би не здогадався. На стадії розробки комп'ютери використовують алгоритми для синтезу та обробки інформації. Що б створити свій дизайн. Все, що від нас вимагається - визначити, в який бік потрібно рухатися. Програмістам вже не потрібно робити мільйон ескізів корпусу.

Як цей підхід буде застосовуватись в житті? Авіакомпанія AIRBUS кілька років працює над моделлю літака майбутнього. Нещодавно, що б впоратися з деякими труднощами вони стали використовувати штучний інтелект. Необхідно було створити перегородку кабіни літака і для цього був використаний той же принцип, що і з шасі квадрокоптера. Створена перегородка була набагато міцніше і вдвічі легше оригіналу. Їй буде оснащени нові літаки компанії Airbus. І тепер ми переходимо до того, що

комп'ютери можуть створювати. Вони можуть пропонувати свої варіанти вирішення наших конкретних завдань. Такі, що наш мозок ніколи б не зміг дійти до цього. Довелося б прорахувати мільйони різних комбінацій, враховуючи різні параметри. Але комп'ютерів все-ще не вистачає одного - інтуїції. Вони змушені щоразу починати з нуля і все тому, що вони не здатні вчитися.

1952 рік – тоді створили ПК, який міг грати в хрестики – нулики.

Потім, 45 років по тому, в 1997 році ПК під назвою Deep Blue обігрує в шахи у Каспарова.

У 2011 році ПК під назвою Watson обігрує двох чоловіків в грі, яку вони придумали з нуля. Що для ПК набагато складніше, ніж гра в шахи. Для перемоги Watson повинен був застосовувати логіку. А чи не працювати за задалегідь заданими схемами.

І ось, пару тижнів тому програма Alpha Go компанії Deep Mind обігрує світового чемпіона в гру Go - найскладнішу з сучасних ігор. У грі можливих ходів більше, ніж атомів в галактиці. Тому, що б перемогти, Alpha Go потрібно було самостійно розвивати інтуїцію і логіку. У деяких моментах творці Alpha Go не розумів, чому програма робила те, що робила. За цей короткий проміжок існування комп'ютерів вони пройшли шлях від дитячих ігор до кульмінації стратегічного мислення.

Роботи допоможуть нам створювати щось нове або вдосконалювати те, що ми вже придумали. І тут може стати в нагоді наша нервова система, що б розуміти і контролювати те, що відбувається. І що буде, якщо наділити техніку нервовою системою. У Лос-Анжелісі є така компанія Bandito Brothers. Її основна спеціалізація – створення шалених автомобілів, здатних на абсолютно нереальні речі. Вони об'єдналися з командою, яка займається розвитком штучного інтелекту. І так, вони взяли традиційні частини від гоночних балідів і зробили для автомобіля нервову систему. Тобто, вони забезпечили свій автомобіль десятками датчиків. За кермо посадили першокласного гонщика. Пригнали її в пустелю і весь тиждень їздили з шаленою швидкістю. Мозок автомобіля помітив все, що з ним відбувалося. Він зафіксував 4 мільярди базових координат, а так само всі сили, які діяли на автомобіль. Потім вони зробили дещо нереальне – помістили всі ці дані в штучний інтелект під назвою Dream Catcher (ловець снів). І вони попросили програму створити надмісний автомобіль. Людина цього ніколи б не створила сама. Створений корпус має ідеальні аеродинамічні показники та міцність і так само має місця під класичний двигун і сидіння для водія і штурмана.

Як результат, після всього почутого, можна зробити висновок, що в майбутньому людство чекає неймовірне розвиток технологій, де здатності людського мозку будуть розширені обчислювальними здібностями техніки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Wikipedia [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Штучний_інтелект (дата звернення: 13.03.2019).
2. Всемирная конференция TED [Електронний ресурс]. – Режим доступу: https://www.ted.com/talks/maurice_conti_the_incredible_inventions_of_intuitive_ai#t-278792 (дата звернення: 13.03.2019)
3. everest.ua [Електронний ресурс]. – Режим доступу: <https://www.everest.ua/ai-platform/analytics/shtuchnij-intelekt-ai-shho-ce-take-i-chomu-ce-v/>
4. ai.lviv.ua [Електронний ресурс] . – Режим доступу: <http://ai.lviv.ua/ais/>
5. aiconference [Електронний ресурс]. – Режим доступу: <https://aiconference.com.ua/uk/news/printsipi-raboti-iskusstvennogo-intellekta-i-perspektiva-ego-ispolzovaniya-92238>

УДК 004.422

Зубко Е. О.¹, Колпакова Т.О.²

¹ студ. гр. КНТ-415 ЗНТУ

² канд. техн. наук, доцент ЗНТУ

ПЕРСПЕКТИВИ ВИКОРИСТАННЯ СУЧАСНИХ ФРЕЙМВОРКІВ ДЛЯ ШВИДКОЇ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ

Фреймворк – це програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

Фреймворк відрізняється від бібліотеки тим, що бібліотека може бути використана в програмному продукті просто як набір підсистем близької функціональності, не впливаючи на архітектуру основного програмного продукту і не накладаючи на неї ніяких обмежень. Фреймворк встановлює правила побудови архітектури додатку, задаючи на початковому етапі розробки поведінку за замовчуванням, формуючи каркас, який можливо розширювати і змінювати відповідно до зазначених вимог. Фреймворк може включати допоміжні програми, бібліотеки коду, мови сценаріїв і інше програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

Одним з головних переваг при використанні фреймворків є те, що веб-додатки часто використовують стандартизовану структуру організації компонентів. Створення структури при розробці на фреймворках значно спрощується. По суті, фреймворк – це безліч конкретних і абстрактних класів, а також визначень способів їх взаємодії. Конкретні класи зазвичай

реалізують зв'язки між класами, а абстрактні класи являють собою точки розширення, в яких каркаси можуть бути використані або адаптовані. Для забезпечення розширення можливостей зазвичай використовуються техніки об'єктно-орієнтованого програмування, наприклад, частини програми можуть успадковуватися від базових класів фреймворку.

На відміну від самописних рішень чи вже існуючих CMS розробка на основі фреймворку забезпечує простоту підтримки проекту та його подальшого розвитку.

За допомогою фреймворку можлива реалізація будь-яких бізнес-процесів, як наприклад оформлення замовлень за допомогою кошика інтернет-магазину, а не тільки тих, які спочатку закладені в систему.

Проекти на базі фреймворків легко масштабуються та модернізуються, а також, як правило, працюють значно швидше і витримують більше навантаження, ніж CMS чи самописні системи, а також мають значно більший рівень безпеки. Саме тому багато популярних сучасних інтернет-магазинів мають в своїй основі саме їх, а не готові CMS.

Наразі існує кілька широко використовуваних фреймворків, та їх популярність з часом тільки зростає (рис.1).

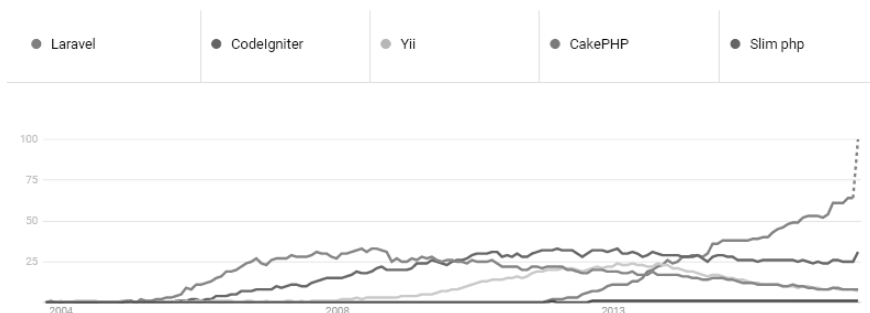


Рисунок 1 – Динаміка росту популярності використання фреймворків

Потенційним недоліком розробки проекту з використанням фреймворку є те, що терміни розробки типового функціоналу на фреймворках більше, ніж при використанні CMS. Крім того, фреймворки містять лише базові компоненти бізнес-логіки, тому багато функцій мають бути реалізовані вручну під певне завдання. Для розробки на фреймворку потрібне розуміння логіки бізнес-процесів, які потрібно реалізувати, наприклад, якщо в CMS вже є якийсь встановлений процес обробки замовлень, то фреймворки такого не пропонують.

Таким чином, незважаючи на певні недоліки, розробка інтернет-магазинів за допомогою сучасних фреймворків є достатньо перспективною і дозволяє отримати швидкий, функціональний та захищений інтернет ресурс.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Преимущества и особенности фреймворка при разработке сайтов[Електронний ресурс]. – Режим доступу: <https://axis.com.ua/articles/>.

2. Что выбрать для разработки сайта – CMS или Framework? [Електронний ресурс]. – Режим доступу: <https://redline.by/novosti/chto-vyibrat-cms-ili-framework.html>

3. Stauffer Matt. Laravel: Up and Running: A Framework for Building Modern PHP Apps / Matt Stauffer. – O'Reilly Media, 2016. – 795 p.

4. Pecoraro C. Mastering Laravel / C. Pecoraro. – Packt Publishing, 2015. – 232 p.

СЕКЦІЯ «СИСТЕМНОГО АНАЛІЗУ ТА ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ»

УДК 620

Пархоменко Л.О.

канд. фіз.-мат. наук, доц. ЗНТУ

АНАЛІЗ СТРУКТУРНОГО СТАНУ ПОРИСТИХ СИСТЕМ

Питання формування та еволюції мікроструктурного стану фазодисперсних матеріалів систем газ – тверда речовина мало вивчені. В більш вигідному становищі перебувають системи газ – метал, пов'язані з проблемою пористості, що виявляє істотний вплив на службові характеристики матеріалів, що застосовуються в різних галузях (металургії, атомній техніці, мікроелектроніці й приладобудуванні, енергомашинобудуванні, космічній та авіаційній техніці і багатьох інших). У зв'язку з цим важливе значення набуває вироблення найбільш сучасних наукових уявлень про пористий стан речовини, достовірної інформації про генезис, морфологію, механізми і кінетику формування та розвитку пор.

Статистично основною характеристикою пористих систем являється функція щільності розподілу пор за їх істинним розміром. У випадку пор сферичної форми – розподіл за їх радіусами або діаметрами. Ця розмірна величина суттєво пов'язана з іншими фізичними величинами (розчинністю, ступенем стійкості, міжфазною поверхневою енергією та іншими). Відповідні розподіли визначаються фізично змістовними характеристиками – значеннями середнього, модального і найбільшого розмірів пор, їх дисперсії, асиметрії та іншими. Їх явний вигляд в аналітичному та кумулятивному зображенні може бути отриманий теоретичним шляхом. Поряд з ними, для практичних цілей аналізу, можуть бути використані ефективні розподіли, що взаємно і однозначно впливають з теоретичних. При ідентифікації з подібними їм за своєю суттю та змістом експериментальними гістограмами виникає можливість дешифрування останніх з виявленням змістовної сторони пористості матеріалу, який аналізується.

Істинні розміри пор не завжди ідентичні значенням розмірів, отриманих за допомогою різних методів дослідження. Така невідповідність можлива й при оцінюванні експериментальних даних з використанням наступних методів: електронно – мікроскопічного, проекційного, рентгеновського, порометричного, адсорбційного, капілярно – конденсаційного, гідродинамічного т. ін. Відповідний висновок може бути віднесений і до дискретного розподілу пор за їх розмірами (гістограмами). Тривимірні розподіли сфероїдальних (глобулярних) пор неправильної форми за їх

ефективними розмірами можна отримати методами кількісної мікрографії (металографії, матеріалографії) шляхом перерахунку двовимірних розподілів перетинів на тривимірні за тими ж розмірами.

Основною характеристикою пористості в матеріалі є функція щільності розподілу пор за розмірами, яка надається у вигляді гістограми – образа. Підтвердження трансформації в залежності від передісторії та засобу отримання масиву пор в матеріалі, характеру їх саморозвитку така функція містить важливу інформацію про фізичну природу процесів в системі. Отже, існування різних механізмів формування пористості в твердому тілі (дифузійного, вакансійного, конденсаційного, деформаційного і радіаційного) повинно знаходити своє відображення за тими чи іншими ознаками, в будові експериментальних гістограм.

З іншого боку, виявлення природи внутрішньосистемних процесів в масиві пор шляхом зіставлення експериментального розподілу – образа з його теоретичною подобою, отриманим за припущенням дії тих же механізмів, – складає сутність і призначення ідентифікованого аналізу. Методологічною основою рішення поставленої задачі є наявність вихідних експериментальних гістограм, що відповідають кожному з механізмів або ж теоретично строго обумовлених положень їх існування.

Взаємодія радіаційного випромінювання з металами супроводжується формуванням системи газонаповнених пор. Останні при низьких температурах схильні до міграційної коалесценції, а при більш високих – до оствальдовської коагуляції, ускладненої низкою супутніх процесів. Міграційна коалесценція протікає шляхом переміщення по дислокаційним каналам найбільш дисперсних пор до найближчих більш крупних з наступним приєднанням до них. При оствальдівській коагуляції більші пори підростають за рахунок надходження до них речовин від пор меншого розміру, що розчиняються (заростають) в матричній фазі. Ці процеси і супутні їм в сукупності й визначають характерні особливості огрубління пористості в металах.

Таку картину підтверджує система мікропор, сформована в алюмінії після його опромінювання іонами ксенона. Зміст мікропор вказує на наявність міграційної коалесценції та її перехід до коагуляції пор зі збільшенням часу ізотермічного нагріву. В цьому зв'язку можна вважати, що формування пор відбувається багатостадійно.

ВИКОРИСТАННЯ МОДУЛЯ LIVELINK FOR MATLAB ДЛЯ ВИРІШЕННЯ ЗАДАЧ ОПТИМІЗАЦІЇ ГЕОМЕТРІЇ ТЕПЛООБМІННИХ ПРИСТРОЇВ

Використання простих балансових моделей для дослідження теплового обладнання дає досить наближені результати та не дозволяє визначити вплив різних факторів на ефективність теплообміну. Тому, аналіз процесів теплообміну з використанням сучасних чисельних методів на основі тривимірних математичних моделей є досить актуальним.

Розрахунок циліндричних та призматичних теплообмінників з турбулізаторами проводився за допомогою потужного програмного комплексу COMSOL Multiphysics методом скінчених елементів. Для моделювання руху газів використовувалася тривимірна система диференціальних рівнянь Нав'є-Стокса. Для опису температурного режиму металевих каналів і вставок використовувалося рівняння теплопровідності. Розглядалася спряжена постановка задачі. Наявність вбудованого модуля LiveLink for MATLAB дозволяє здійснити двосторонню інтеграцію COMSOL Multiphysics та MATLAB та суттєво розширити функціонал COMSOL Multiphysics. Модуль дозволяє використовувати всі можливості MATLAB і його інструментів для підготовки, перетворення і подальшого удосконалення моделі, в тому числі розширення можливостей коду MATLAB за рахунок потужних засобів мультифізического моделювання, моделювання геометрій на основі імовірнісних і графічних даних, довільний статистичний аналіз результатів моделювання та ін.

Використання скриптів MATLAB з циклічними алгоритмами дозволило повести серію чисельних експериментів, дослідити вплив геометричних параметрів теплообмінників на ефективність теплообміну та визначити оптимальну геометрію в залежності від зовнішніх умов функціонування. Досліджені також конструкції зі змінним кроком встановлення турбулізаторів або гвинтоподібних вставок нерегулярної структури. Нерівномірність кроків встановлення турбулізаторів регулюється двопараметричною степеневою залежністю. Застосування такого підходу дозволяє перерозподілити теплове навантаження по висоті теплообмінника та підвищити ефективність його роботи.

Запропонована модель дозволяє на стадії проектування опалювальних пристроїв визначати оптимальні геометричні параметри теплообмінних

елементів та уникнути необхідності проведення кошторисних натурних експериментів.

УДК 512.56

Терещенко Е.В.

канд. фіз.-мат. наук, доц. ЗНТУ

ГРАТКИ ФОРМАЛЬНИХ ПОНЬЯТЬ ДЛЯ АНАЛІЗУ БІОМЕТРИЧНИХ ДАНИХ

При проведенні селекційної робота зі створення великоплідних батьківських форм є актуальною задача по визначенню споріднених ліній соняшника за біометричними ознаками. Математичний апарат аналізу формальних понять дозволяє побудувати ґратки понять $B(G, M, I)$, що є множиною всіх понять контексту $K = (G, M, I)$, які упорядковані по вкладенню об'ємів [1]. Формальний контекст $K = (G, M, I)$ складається з множини об'єктів G та ознак M , та відношення I інцидентності між ними. Формальним поняттям контексту $K = (G, M, I)$ є пара виду (A, B) , де $A \subseteq B$, $B \subseteq M$, $A' = B$, $B' = A$. Запис $B(G, M, I)$ означає всю множину формальних понять контексту $K = (G, M, I)$. Багатозначний контекст (G, M, W, I) складається з множин об'єктів G , ознак M , значень ознак W та тернарного відношення $I \subseteq G \times M \times W$. Біометричні дані польового експерименту містяться в таблиці MS Excel, де лінія соняшнику розміщується у рядку та характеризується за 86 ознаками. Отже, множина об'єктів G -це лінії соняшників, множина ознак M - це ознаки, з якими ведеться спостереження, множина значень ознак W визначається можливими варіаціями ознак соняшника. Тернарне відношення $I \subseteq G \times M \times W$ задається протоколом польового експерименту. Значна потужність множини ознак M та значень ознак W обумовлює інтерес до побудови айсбергу ґратки понять, наприклад, що містить n найпоширеніших значень. Айсберг дозволить виокремити значущі та менш значущі ознаки для характеристики лінії соняшника.

Робота виконана в рамках договору про співробітництво між Інститутом олійних культур Національної академії аграрних наук України та ЗНТУ на базі кафедри системного аналізу та обчислювальної математики та ДБ05028 «Аналіз біометричної інформації» (2018-2021pp) [2].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ganter B. Formal Concept Analysis/ B. Ganter, R. Wille // Mathematical Foundations. – Berlin: Springer, 1999.
2. Буренко, К.С. Параметри крупноплідного сояшника у випробуваннях гібридів 2011-2012 [Текст]/ К.С. Буренко, К.В.Ведмедева // Науково-технічний бюлетень Інституту олійних культур НААН. – Запоріжжя, 2013. – Вип.18. – С.26-30. ISSN 2078-7316

УДК 519.816

Бакурова А.В.¹, Ропало Г.М.²

¹ д-р. екон. наук, проф., НУ «ЗП»

² асп., НУ «ЗП»

КОНЦЕПЦІЯ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ АПТЕЧНОЮ МЕРЕЖЕЮ

Ефективне управління сучасною компанією має забезпечуватися створенням такої управлінської інформаційної системи, що охопить всі об'єкти та рівні управління. Основними проблемами управління аптечною мережею є недостатність управлінського досвіду керівників підрозділів, відсутність інформаційної освіти та фінансових ресурсів для формування такої інформаційної системи. Проте інтеграція нових підходів до управління мережею та застосування інновацій у автоматизації її діяльності аптечній характеризується, перш за все, економічною ефективністю. Проблема побудови та впровадження інформаційної системи управління стоїть гостро перед будь-якою аптечною мережею, адже рівень інформаційного забезпечення управлінської діяльності у таких компаній, як правило, є досить низьким, а відповідно, інформаційна система управління не є повністю ефективною.

Інформаційна система – взаємопов'язана сукупність засобів, методів та персоналу, що використовуються для зберігання, обробки та видачі інформації в інтересах досягнення поставленої мети. Основою інформаційних систем є інформаційні моделі, що описують інформаційні потоки в управлінні за допомогою певних алгоритмів та процедур оброблення інформації.

Метою інформаційної системи є підготовка інформації для використання її управлінським апаратом. Вона забезпечує збереження, оброблення та узагальнення]. Основними принципами побудови ефективних інформаційних систем є: принцип інтеграції (багаторазове використання даних для вирішення великої кількості завдань), принцип системності

(обробка даних в різних аспектах для отримання інформації необхідну на всіх рівнях управління), принцип комплексності (механізація і автоматизація процедур перетворення даних на всіх етапах функціонування системи).

Відповідно, інформаційна система управління аптечною мережею – сукупність управлінської інформації та інформаційної підтримки системи управління та зв'язки між ними. Основними функціями інформаційних систем управління є збір, реєстрація та обробка даних, що отримані в місці збору інформації (наприклад, в аптечному закладі) та передані до інформаційного центру з метою подальшої її обробки.

Структуру інформаційної системи складає сукупність окремих її частин – підсистем. Підсистема – це частина системи, яка виділена за певною ознакою.

1. Прийняття рішень

Система підтримки прийняття рішення про розміщення аптек на основі нечіткої логіки. Для її побудови використовуються методи нечіткої математики та розв'язується задача лінгвістичної апроксимації. Вхідні змінні системи, що впливають на вибір місця для торгової точки представляються у вигляді лінгвістичних змінних. Такими змінними є трафік, якість пішохідного потоку, купівельна спроможність, тип району, конкурентне середовище, відстань до медичних установ. Для кожної лінгвістичної змінної визначаються терм-множини, що складаються з кількох термів.

Система підтримки прийняття рішень про управління потенційним попитом аптек мережі на основі побудови когнітивної моделі. Якісний аналіз ефективності управління мережею передбачає визначення тенденцій перебігу процесів, якісну оцінку цих тенденцій і вибір заходів, що сприяють їх розвитку в потрібному напрямку. Когнітивна модель оснований на формалізації причинно-наслідкових зв'язків, які мають місце між факторами, що характеризують досліджувану систему. Результатом формалізації є представлення системи у вигляді когнітивної карти.

Рекомендаційні системи забезпечують організацію та підтримку цифрового зв'язку з клієнтами мережі. Використання рекомендаційних систем ґрунтується на пропозиції нових елементів для конкретного користувача на основі попередніх вподобань користувачів або думок інших однодумців користувача.

СППР управління розвитком мережі аптек в умовах конкуренції. Такі СППР націлені на вирішення задач розширення мережі, управління ціновою політикою, управління логістикою.

2. Управління продажами.

Система управління взаємовідносин з клієнтами. Дозволяє вести діалог з клієнтом через представників аптеки та аптечної довідки, формувати клієнтську базу з різними групами клієнтів (вік, стать, історія покупок

клієнта тощо). Лояльність клієнта виражається в економічний ефекті для аптечної мережі та збільшенні її прибутків.

Система продажу. Створення ефективних збутових каналів, налагодження системи своєчасної доставки лікарських засобів в аптечні заклади, проведення акцій та надання знижок.

Система управління лояльністю персоналу. Система взаємовідносин між власниками аптечної мережі та працівниками аптечних закладів повинна базуватись на довірі та повазі, оскільки основне прагнення співробітника аптеки — це справедливе ставлення до себе та можливість професійно розвиватися в рамках своєї роботи.

3. Управління проектами. У складних проектах необхідно обробляти безліч інформації, для цієї мети використовуються інформаційні системи управління проектами. В основі цих систем незалежно від їх рівня і вартості закладені методи мережевого планування і управління. Інформаційна система управління проектами підвищують ефективність, обґрунтованість і швидкість прийняття управлінських рішень та дозволяють автоматизувати всі основні операції проектної діяльності.

4. Фінансовий аналіз. Важливу роль у реалізації завдань фінансового аналізу відіграє його інформаційне забезпечення. Основним джерелом інформації при проведенні фінансового аналізу є дані бухгалтерського обліку та фінансової звітності підприємства, мовлення користувача. Автоматизація фінансового аналізу дає змогу підвищити ефективність управління, оскільки буде налагоджений оперативний зв'язок між головним центром управління і підрозділами підприємства. Такі можливості зумовлені тим, що комп'ютерний спосіб обробки інформації вимагає формального та чіткого опису аналітичних процедур у вигляді алгоритмів, що призводить до покращення порядку виконання обов'язків працівників фінансової служби.

5. Аудиту. Економічна діяльність аптечної мережі пов'язана з виконанням певних функцій контролю. Процес контролю полягає у визначенні найбільш вразливих місць та негативних процесів у аптечній мережі для їх заміщення більш ефективними.

6. Бухгалтерський облік. Побудова інформаційної системи та автоматизація роботи бухгалтерського обліку, мережі дозволяє навести порядок і в складському обліку, в постачанні та реалізації продукції, товарів, допомагає відстежувати договори з контрагентами та підрядними організаціями, швидше розраховувати заробітну плату, своєчасно здавати офіційну статистичну звітність.

Використання описаних систем в структурі інформаційної системи сприятиме прийняттю обґрунтованих стратегічних управлінських рішень, спрямованих на зростання конкурентоспроможності і поліпшення якості управління аптечною мережею.

УДК 519.25

Вакар Д.С.¹, Терещенко Е.В.²

¹ студ. гр. КНТ-815 ЗНТУ

² канд. фіз.-мат. наук, доц. ЗНТУ

СТАТИСТИЧНА ОБРОБКА БІОМЕТРИЧНОЇ ІНФОРМАЦІЇ ПОЛЬОВОГО ЕКСПЕРИМЕНТУ

Робота виконана в рамках договору про співробітництво між Інститутом олійних культур Національної академії аграрних наук України та ЗНТУ на базі кафедри системного аналізу та обчислювальної математики та ДБ05028 «Аналіз біометричної інформації» (2018-2021pp) [1]. Одна з основних олійних культур, що вирощуються на Україні, є соняшник. На ринку збуту, вартість товарного соняшнику приблизно на 20-50% вище, ніж олійного. На сьогоднішній день в Реєстрі сортів рослин України є декілька сортів і гібридів з достатньо високою масою 1000 насінин (від 80 г. і більше), але на жаль, існуючі гібриди по крупності насіння істотно поступаються сортам. Тому селекційна робота зі створення великоплідних батьківських форм є актуальною. Для автоматизації статистичної обробки польового експерименту проведено аналіз структури бази даних польових досліджень лабораторії Інституту олійних культур, визначились з методикою обробки даних польових досліджень, розроблено структуру шаблону для розрахунку та еталонного листа початкових даних, розроблено макрос для автоматизації розрахунку статистичних оцінок. Дані польового експерименту заносяться в таблицю MS Excel, кожна лінія розміщується у рядку та характеризується за 86 ознаками. Порядок розміщення ліній довільний. Для кожної лінії, дані заносяться по 5 екземплярам з певної кількості рослин. Вибраковується занадто великі та занадто малі. Для оцінки стійкості лінії порівнюють дані за 5 років з таблиці [1]. Для кожної лінії, для кожної ознаки, за певний проміжок часу, оцінюються параметри статистичних оцінок: «середнє арифметичне», «похибка», «відхилення», «варіація», «мінливість». в залежності від типу даних. Створенно шаблон для автоматизації статистичної обробки даних в таблиці Excel засобами Visual Basic for Applications (VBA).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Буренко, К.С. Параметри крупноплідного соняшника у випробуваннях гібридів 2011-2012 [Текст]/ К.С. Буренко, К.В.Ведмедева // Науково-технічний бюлетень Інституту олійних культур НААН. – Запоріжжя, 2013. – Вип.18. – С.26-30. ISSN 2078-7316

СТВОРЕННЯ ФУНКЦІОНАЛЬНО-АНАЛІТИЧНОГО МОДУЛЯ PYTHON ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ СУБД SQLITE

На поточний момент Python є однією з найпопулярніших мов програмування загального призначення в світі. Він має досить простий синтаксис, в той час як його стандартна бібліотека включає безліч функцій. Python підтримує структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Спільнота програмістів Python постійно розширюють його можливості, створюючи спеціалізовані модулі для вирішення різних практичних задач. SQLite - це вбудована кроссплатформенная СУБД, яка підтримує досить повний набір команд SQL і доступна у вихідних кодах (на мові C).

Метою даної роботи було створення модуля Python, що містить набір функцій для аналізу даних, що зберігаються в електронних таблицях, наприклад Microsoft Excel або Google Sheets, передачі цих даних в бази даних SQLite, що передбачає автоматичне формування і виконання запитів на мові програмування SQL, контроль за ходом виконання і збір статистичної інформації, такої як обсяг сформованих і переданих даних, час виконання, розмір отриманих файлів і т.д.

Також серед функціоналу модуля є можливості формування запитів на вибірку даних з баз SQLite, створення і видалення таблиць і даних з них (запити SELECT, CREATE, DROP).

Передбачена можливість генерації електронних таблиць будь-якого розміру з випадковими числами для аналізу можливостей масштабування баз даних SQLite.

Модуль укомплектований довідкової документацією відповідно до стандарту PEP 257. Модуль протестований і готовий до практичного застосування.

УДК 004.6

Бакурова А.В.¹, Пасічник М.С.², Терещенко Е.В.³, Філей Ю.В.⁴

¹ д-р. екон. наук, проф. ЗНТУ

² асп. ЗНТУ

³ канд. фіз.-мат. наук, доц. ЗНТУ

⁴ канд. юр. наук, доц. ЗНТУ

БАГАТОФАКТОРНИЙ АНАЛІЗ ДИНАМІКИ ЗЛОЧИННОСТІ В УКРАЇНІ

Для проведення реформування в превентивній політиці і протидії росту злочинності та соціально-економічній напруженості необхідно змінити процес реагування на порушення від карального на упереджувальний. Для розробки упереджувальних стратегій проведемо аналіз динаміки злочинності в Україні.

У табл.1 наведено дані, що отримано з відкритих джерел інформації [1]. Прийmemo такі позначення: x_1 -чисельність активного населення, тис.осіб, x_2 - чисельність безробітних, тис.осіб, та x_3 - рівень зарплатні, грн.

Таблиця 1 – Вихідні дані багатofакторної моделі

Рік	Результуюча ознака Всього засуджено осіб	Фактори		
		x_1	x_2	x_3
1	2	3	4	5
2000	230903	22830,80	2655,80	230,00
2001	202609	22426,50	2455,00	311,00
2002	194212	22231,90	2140,70	376,00
2003	201081	22171,30	2008,00	462,00
2004	204794	22202,40	1906,70	590,00
2005	176934	22280,80	1600,80	806,00
2006	160865	22245,40	1515,00	1041,00
2007	152772	22322,30	1417,60	1351,00
2008	146858	22397,40	1425,10	1806,00
2009	146383	22150,30	1958,80	1906,00
2010	168774	20894,10	1713,90	2250,00
2011	154356	20893,00	1661,90	2648,00
2012	162881	20851,20	1589,80	3041,00
2013	122973	20824,60	1510,40	3282,00
2014	102170	19920,90	1847,60	3480,00

Продовження таблиці 1

1	2	3	4	5
2015	94798	18097,90	1654,70	4195,00
2016	76217	17955,10	1678,20	5183,00
2017	76804	17854,40	1698,00	7104,00
2018	немає даних	17939,50	1578,60	8865,00

Побудуємо багатофакторну лінійну модель F рівня злочинності в залежності від таких факторів впливу, як чисельність активного населення x_1 , чисельність безробітних x_2 та рівень зарплатні x_3 (за табл.1):

$$F = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3, \quad (1)$$

За результатами обчислень в Ms Excel отримано багатофакторну лінійну модель:

$$F = 585699,1 - 19,87566x_1 + 33,61224x_2 - 26,67692x_3, \quad (2)$$

з характеристиками: коефіцієнт детермінації $R^2=0,87$, $F_{\text{стат}}=22,67$. Високий рівень коефіцієнту детермінації $R^2=0,87$ є статистично значимим, бо табличне значення F-розподілу при рівні значимості 0,05 дорівнює 4,1, тобто розрахункове значення більше табличного: $22,67 > 4,1$. Результат моделювання графічно представлено на рис.1.

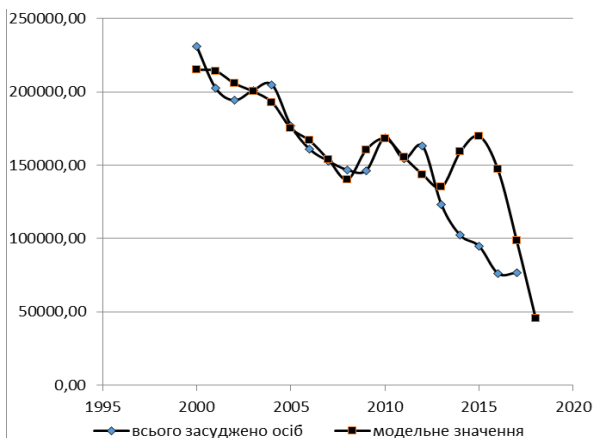


Рисунок 1 – Порівняння реальної та модельної кількості засуджених осіб у 2000-2018рр.

Аналіз моделі (2) підтверджує наявність тісного зв'язку між виокремленими соціальними факторами впливу та рівнем злочинності. Можна стверджувати, що підвищення рівня безробіття викликає підвищення рівня злочинності ($\alpha_2 = 33,61224$), а підвищення чисельності активного населення і рівня зарплатні знижує рівень злочинності ($\alpha_1 = -19,87566, \alpha_3 = -26,67692$). Отже, ці фактори можна застосовувати для зниження соціально-економічної напруженості у суспільстві.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Офіційний сайт Генеральної прокуратури України [Електронний ресурс]. – Режим доступу: <https://gp.gov.ua/>

Наукове електронне видання
комбінованого використання
Можна використовувати в локальному та
мережному режимах

ТИЖДЕНЬ НАУКИ-2019.

Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції серед студентів,
викладачів, науковців, молодих учених і аспірантів 15–19
квітня 2019 року

Один електронний оптичний диск (DVD-ROM);
супровідна документація.
Тираж 100 прим. Зам. № 695

Видавець і виготовлювач
Запорізький національний технічний університет
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64 Тел.:
(061) 769–82–96, 220–12–14

Свідоцтво суб'єкта видавничої справи ДК № 2394 від 27.12.2005.