

НПК МНІС ІП-2018

ЗБІРНИК НАУКОВИХ ПРАЦЬ
МОЛОДИХ
НАУКОВЦІВ І СТУДЕНТІВ

ЧАСТИНА

2



ПРИСВЯЧУЄТЬСЯ 30-РІЧЧЮ
ПІДГОТОВКИ ІТ-ФАХІВЦІВ В
ХМЕЛЬНИЦЬКОМУ
НАЦІОНАЛЬНОМУ
УНІВЕРСИТЕТІ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Хмельницький національний університет

Військовий інститут Київського національного університету
ім.Тараса Шевченка

ПВНЗ “Університет економіки і підприємництва”

Тернопільський інститут агропромислового виробництва

Інтелектуальний потенціал - 2018

збірник наукових праць молодих науковців і студентів

Присвячується 30-річчю підготовки ІТ- фахівців в Хмельницькому національному університеті

сформовано за матеріалами

Всеукраїнської науково-практичної конференції
молодих науковців і студентів «Інтелектуальний потенціал – 2018»

14-16 листопада 2018р.

Частина 2

Математичне моделювання та інженерія програмного забезпечення

Хмельницький
2018

ББК 74.480.278

С.88

«Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ- фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.2: Математичне моделювання та інженерія програмного забезпечення – 96 с.

Відповідальний редактор: Капітанець С.В.

Відповідальний за випуск: Чещун В.М.

Редакційна колегія:

Желавський О.Б.

Капітанець С.В.

Мясіщев О.А.

Чещун В.М.

Тимофєєва Л.В.

ЗМІСТ

Антонюк Р.О., Праворська Н.І. Модель підвищення ефективності керування підприємством громадського харчування з використанням програмного комплексу	5
Барабаш С.О., Огневий О.В. Метод структурного синтезу мобільних web-додатків	8
Бичко Д.В., Шендрик В.В. Методологія вибору оптимального постачальника підприємства в умовах багатокритеріальності та невизначеності	12
Бобилєва О.С., Дмитрієва О.А. Програмна підтримка паралельних обчислень	15
Бондар Д.В., Бедратюк Л. П. Аналіз та обробка великих об'ємів даних за допомогою in-методу технологій на прикладі APACHE SPARK	17
Гаврілова О. І., Кунгурцев О.Б. Виявлення міжфразових зв'язків при побудові словників предметної	20
Галкін М.С., Форкун Ю.В. Проблема вибору криптографічного алгоритму для шифрування даних користувача	23
Глушко Р.М., Муляр І.В. Метод розробки децентралізованих захищених програм на основі технології BLOCKCHAIN	28
Голодов Д.В. П'ятикоп О.Є. Інформаційна технологія розпізнавання та обліку помольних куль	32
Єремєєв Є.Ю., Панфьорова І.Ю. Розробка системи показників якості веб-контенту	34
Згара К.Г., П'ятикоп О.Є. Особливості розробки корпоративної соціальної мережі на прикладі університету	38
Карась В.Г., Яшина О. М. Модель програмного комплексу для підвищення ефективності набуття стійких позитивних звичок	39
Коваль В.А., Огневий О.В. Технологія програмної реалізації системи управління інформаційними ресурсами на основі автономних сценаріїв	44
Кушнірук С.Л., Муляр І.В. Адаптивний метод проектування програмного забезпечення	48
Левковська У.О., Кучерук О.Я. Модель оцінювання математичної компетентності студентів	51
Лисенко О.В., Матвієнко Ю.С. Розробка мобільного додатку гібридної реальності	54

Люба В.М., Григоров С.С. Моделювання динаміки відвідування сайту компанії ТОВ каса “Люкс-Рейзен”	56
Мацьков О.П., Бармак О.В. Аналіз технологій розпізнавання облич на зображеннях для вбудованих систем	60
Пишний М.В., Яшина О.М. Доцільність використання акторної моделі при побудові високонавантажених сервісів обробки замовлень	63
Пілецький П. Є., Чумаченко Д. І. Інтелектуальне мультиагентне моделювання епідемічного процесу іксодового кліщового бореліозу ...	67
Подоконний Д.В., Гурман І.В. Програмне забезпечення оптимізації роботи стоматологічного кабінету шляхом розв’язку задач лінійного програмування неграфічним методом	70
Сапужак О.М., Драч І.В. Чисельний підхід до знаходження наближеного розв’язку рівняння збурень Кортвега-де Вріса-Бургера для задачі динаміки рідинного автобалансира	75
Троянов Б.В., Григоров С.С. Моделювання тенденції динаміки валютного курсу гривні	79
Франчук М.Ю., Праворська Н.І. Програма-оптимізатор виробництва і реалізації продукції одягу ТОВ «Бембі»	82
Хітрін О.В., Бедратюк Л.П. Проблема забезпечення швидкого шифрування інформації для передачі даних між клієнтом та сервером	85
Шостак А.С., Ратушняк Т.В. Розробка навчального модуля «особові рахунки» для інформаційного забезпечення автоматизованого робочого місця касира банку	89
Ящук Р.В., Форкун Ю.В. Аналіз методу оптимізації згорткових нейронних мереж при розпізнаванні образів на мобільних пристроях	92

Модель підвищення ефективності керування підприємством громадського харчування з використанням програмного комплексу

Антонюк Р.О.

Науковий керівник: к.п.н. доц. Праворська Н.І.

Хмельницький національний університет

У світі коли «Інтернет» є невід'ємною частиною життя, майже, кожної людини, сервіси управління є необхідною частиною для запоруки успішного бізнесу. Сервіс – програмний комплекс, що включає в себе усі методи ведення та управління власним бізнесом. Від того наскільки сервіс є зручним та багатофункціональним залежить те, чи стане потенційний клієнт його використовувати. Для досліджуваної теми «Модель підвищення ефективності керування підприємством громадського харчування з використанням програмного комплексу» основна вимога залишається незмінною – підвищення результативності керування власним бізнесом. Для розвитку проекту і підтримки роботи сервісу було вирішено розробити його для малого бізнесу, а саме – підприємства громадського харчування. Цей вибір було зроблено через те, що саме в таких закладах необхідно підвищити ефективність керування, оскільки ведення цього бізнесу зараз проводиться «на папері». Також результати нашої роботи не повинні припадати пилом у просторах «Інтернет», а повинні бути використані.

Метою та призначенням цього проекту є реалізація цілого програмного комплексу, що зможе підняти рентабельність керування та ведення бізнесу підприємств громадського харчування. Комплекс включатиме в себе управління працівниками, торговими точками, постачальниками, товарами, складом та іншим.

Сформовані необхідні для проектованої системи функціональні вимоги. З огляду на попередні розділи система буде використовуватися декількома групами користувачів. Приведемо список цих груп:

- власники – група користувачів, яка налаштовує свій власний бізнес. Додає продавців, постачальників, торгові точки, продукти та інше;
- продавці – це група користувачів, яка буде вести продажі на торгових точках, прийматиме товар від постачальників та інше;
- гості – припускається, що це група потенційних користувачів програмного комплексу. Вони можуть здійснювати керування «віртуальним» підприємством громадського харчування.

Для кожної із груп користувачів існують свої функціональні вимоги. Приведемо список для власників: а) система повинна забезпечувати можливість самостійно реєструватися новим власникам; б) власники можуть брати під повний контроль торгові точки, продавців, постачальників, склади, продукти, продукцію, що реалізується на його торгових точках відповідно до рецептури; в) при замовленнях на торгових точках сервіс повинен також показувати ці дані власнику; д) проектований програмний продукт має надавати доступ до системи управління тільки після авторизації у свій власний кабінет за логіном і паролем.

Для продавців на торгових точках власника: а) система повинна надавати можливість створення нового замовлення; б) приймати товар від постачальників; в) відмінити існуюче замовлення; г) проєктований програмний комплекс забезпечує змогу доступу до власного кабінету тільки після запрошення власника.

Гості, або неавторизовані користувачі мають роль неавторизованого відвідувача. Дана роль надає відвідувачам сервісу можливість переглядати всі можливості даного сервісу за допомогою демонстраційних кабінетів.

Важливою для системи є також роль головного адміністратора. Головним адміністратором виступає особа, яка є власником даного сервісу. В цьому разі висуватимуться наступні вимоги: а) система повинна надавати можливість керування усім сервісом; б) адміністратор може переглядати дії інших користувачів системи; в) може блокувати інших користувачів; г) має змогу допомагати в разі необхідності користувачу системи (створити продукт, запросити продавця, тощо); д) проєктований програмний продукт повинен надавати доступ до даних системи адміністраторам тільки після авторизації за логіном і паролем.

Щодо нефункціональних вимог до системи, то вони є уніфіковані:

- система на даному етапі повинна мати веб інтерфейс;
- цей же етапі відзначається можливістю отримання програми системи для планшетів;
- система повинна бути готова до впровадження нової групи користувачів, такою групою є постачальники;
- існує готовність системи до розміщення на web-хостинг хмарного типу зберігання.

Говорячи про системний підхід, можна говорити про деякий спосіб організації наших дій, який буде охоплювати будь-який вид діяльності, виявляючи закономірності і взаємозв'язки з метою їх більш ефективного використання. При цьому системний підхід є не стільки методом вирішення завдань, скільки методом постановки задач. Вважають, що «правильно поставлене питання – це вже є половина відповіді». Це якісно більш високий, ніж просто предметний, спосіб пізнання. Розроблювальний програмний продукт повинен відповідати основним системним параметрам (простота, надійність, ефективність), принципам щодо інформації та основним принципам системного підходу:

- цілісність;
- ієрархічність;
- структуризація;
- системність;
- множинність.

Проєктування і розробка програмного комплексу включає: затвердження первинного технічного завдання розробки сервісу, дизайн – створення графічних елементів макету, стилів і елементів навігації, розробка програмного коду, модулів, бази даних і інших елементів необхідних в

проекті, тестування і розміщення сайту в мережі Інтернет.

Відповідно до описаних потоків даних основною задачею даної системи є організація ефективного ведення керування підприємством громадського харчування. Зважаючи на те, що користувач програмної системи може працювати з нею як із планшетів, так і з веб-інтерфейсу, потрібно створити додаток під платформи IOS та Android та веб-версію. Тому, структуру програми буде формувати архітектура «Клієнт-Сервер», де клієнт являє собою простий інтерфейс для роботи із сервером. На стороні сервера виконується вся обчислювальна логіка. Сервер буде взаємодіяти із спроектованою раніше базою даних. Отже, декомпозиція задач програми буде здійснюватися для створення серверу. Згідно моделі декомпозиції, основні варіанти роботи із програмою розділяються по ролях користувачів. Детально взаємозв'язок функціональної моделі для кожної групи користувачів, описаний у діаграмі варіантів використання, можна показати за допомогою діаграм взаємодії. За допомогою даних діаграм можна визначити класи, які потрібно створити, зв'язки між ними, а також операції та відповідальність кожного класу.

Згідно описаної стратегії розробки, розроблювальна програмна система має складатися із двох основних модулів:

- Сервер – написаний за допомогою PHP Framework Yii2;
- Веб-клієнт – веб-інтерфейс для роботи користувача;
- Мобільний клієнт – мобільний інтерфейс для роботи користувача;

Варто вказати, що на даному етапі ведеться розробка тільки веб- та android клієнтів, у майбутньому будуть додатися інші клієнти веб-сервісу – IOS або настільні додатки. Приклад взаємодії клієнта і сервера для реєстрації користувача можна побачити на рисунку 1.

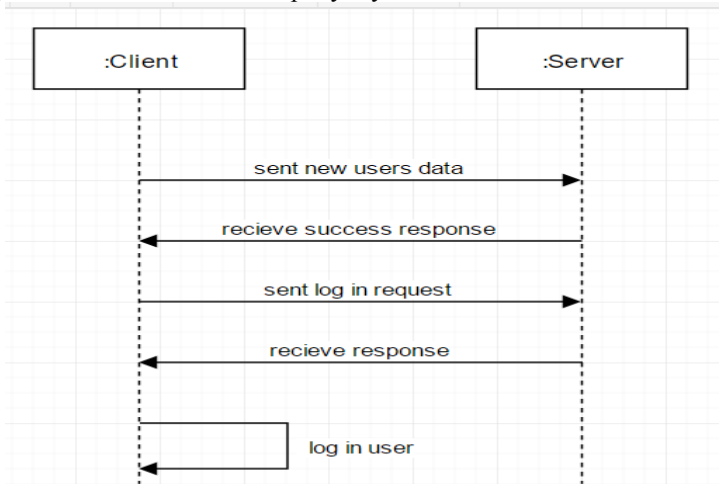


Рисунок 1 – Схема взаємодії клієнта та серверу

Отже, в даній роботі створено програмний комплекс для підвищення ефективності керування підприємством громадського харчування. В результаті розробки, на основі ряду досліджень та публікацій було проаналізовано питання впровадження та розвитку сервісу надання послуг даною сукупністю програм. Основні задачі, виконані спроектованим комплексом, здатні підвищити продуктивності керування та ведення бізнесу. Розробка даного продукту є обґрунтованою та економічно вигідною. Якщо розглянути актуальність проектування і виконання зі сторони існуючих рішень, то аналоги даного програмного комплексу є, але вони представляють собою досить громіздкі, дорогі та складні в обслуговуванні програмні продукти, або вже є давно застарілими. Одними із подібних програм вважаються ті, що написані на 1С. Також є такі системи, як Poster та FrontPad, недоліками яких вважається ціна та складність у використанні. У майбутньому запропонований сервіс можна розширити, додавши до нього низку клієнтів для різних мобільних та настільних платформ та впровадити платні послуги.

Література

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений: 3-е издание // Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон / – М.: ООО "И. Д. Вильямс", 2008. - 720 с.
2. Физерс М.К. Эффективная работа с унаследованным кодом // М.К. Физерс / – М.: Диалектика-Вильямс ", 2018. - 400 с.

Метод структурного синтезу мобільних web-додатків

Барабаш С.О.

Науковий керівник - ктн. доц. Огневий О.В.

Хмельницький національний університет

З метою покращення збірки мобільних web-додатків розробники веб-сайтів намагаються знайти рішення, яке дозволить швидко і ефективно використовувати існуючу компонентну базу, що збільшить якість збірки та полегшить підтримку та редагування проекту на пізніх етапах його створення.

Рішення структурного синтезу проектів потребує чіткого визначення, опису та категоризації компонентів і модулів за допомогою семантичних мереж [1]. Використання таких мереж дозволить провести чіткий зв'язок між поставленими задачами та оптимальними рішеннями. Головна ідея - це сприйняття кожного логічного процесу, як окрему задачу, а програму, як існуюче рішення. Дотримуючись такої схеми, можна прискорити пошук рішень для конкретних задач.

З врахуванням вищесказаного, розробка методу структурного синтезу мобільних веб-додатків є актуальним науково-технічним завданням, вирішенню якого присвячена дана магістерська робота.

На основі проведеного у першому розділі аналізу, сформулюємо нові нові цілі, у список вимог для майбутньої ІС:

1. Компоненти, що використовуються в прикладній системі, повинні мати сумісність із усіма вимогами базових стандартів.

2. Наявність комбінацій стандартів в проектах ІС, які є актуальними для користувачів систем, розробників систем та організацій, що їх використовують.

3. Чітке визначення класів і параметрів базових стандартів, для подальшого забезпечення взаємодії внутрішніх програмних елементів, в межах однієї системи. Забезпечення зв'язку між системами і зв'язку між сервісами, які виконують автоматизовані і не автоматизовані процеси предметної області.

У нашій роботі використовується модель, для профілю інформаційної системи (на основі онтологічного каркасу). Базовий шар каркасу буде формуватися за рахунок класифікаторів і специфікацій. На верхньому рівні каркасу буде знаходитися концепт Document, який став основою для подальших наслідувань специфікацій і стандартів. Все документи поділені на два рівні: рівень еталонних документів і рівень прикладних документів.

Перший рівень налічує декілька концептів: Classifier, ModelClassifier, InteroperabilityModelClassifier і Reference. Така модель може розширюватися на будь-якому рівні розробки, за допомогою додавання у її склад класифікаторів кожної групи, також можна додавати і самі групи, які не будуть вносити серйозних змін до структури [2].

На другому рівні з'являється група концептів, яка виконує формалізацію прикладних документів. Для визначення множини всіх документів використовується концепт AppDocument, його наслідує InterfaceDocument, який виділяє лише документи, що мають регламентовану поведінку для кожного інтерфейсу. Кожний документ має модульну структуру, яка задається за допомогою концепту InterfaceDocumentPart, кожний модуль може нести в собі певну кількість логічних груп параметрів ParameterSet (кожна група має власні параметри Parameters). Якщо частини документів починають класифікуватися хоча б за однією ознакою, то вони стають ClassifiedDocumentPart. Параметри, які регламентуються в нормативній документації, мають певну область значень, що задається доменом. Такий домен описується за допомогою концепту DomainMode. Виконуючи опис домену задається базовий системний тип SystemDataType, який буде визначати параметри і обмеження (Restriction). Один параметр може мати декілька визначених екземплярів DomainMode. Це необхідно для формування стандартизованих профілів ІС. Існує концепт (граничний) InterfaceProfile, який використовується для формування вибірки параметрів

нормативних документів і режиму їх використання. Даний концепт є основою для інших концептів, які моделюють роботу функцій входу-виходу і способи взаємодії із системою. В залежності від системи, інтерфейси визначаються форматами і структурами даних, які у них передаються [3].

Представлена загальна схема майбутньої автоматизованої системи компонентної збірки веб-додатків. В даній схемі комбінація систем представлена у вигляді підсистеми первинного аналізу і форматування даних, підсистеми, яка формує кінцевий склад веб-додатків та підсистема управління збереження даних про процеси та компоненти (рис. 1).

Запропоновано етапи інформаційних перетворень процесу компонентної збірки веб-додатків, що здійснюються системами: формалізації вхідного впливу S_F , формування семантичної мережі процесів і компонент (S_{SN}), формування компонентної моделі веб-додатки (S_M). Зовнішнє середовище формує наступні типи вхідних впливів: $P, P'czP$ – множина процесів, $C, C' \subset C$ - множина компонент, Q - множина запитів компонентного складу цільового веб-додатку.

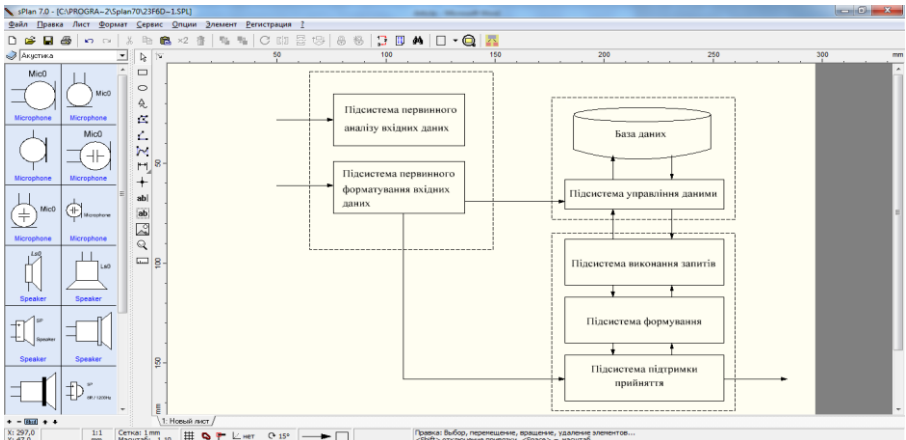


Рисунок 1 - Загальна схема автоматизованої системи компонентної збірки мобільних веб-додатків

Семантична модель, побудована на такій схемі, повинна вираховувати і оцінювати індивідуальні властивості компонентів і процесів, які входять до неї. Також потрібно слідувати ієрархії процесів та зберігати списки готових рішень, які успішно застосовуються.

Підбірка процесів буде відбуватися ітераційно і на кожному кроці сформується множина процесів, які будуть претендувати на потрапляння у структуру веб-додатку. Компоненти можна умовно розділити на зовнішні та внутрішні. Зовнішні компоненти надають системі гнучкості, оскільки вона може використовувати компоненти в режимі віддаленого доступу,

використовуючи онлайн ресурси. При наявності успішного підключення до мережі Інтернет, можна завантажувати компоненти, які знадобилися системі автоматичної збірки. Внутрішні компоненти потребують більшої уваги тому, що їх не можна завантажити онлайн. Такі компоненти додаються вручну під час формування збірки. Вони представлені у вигляді різноманітних класів та бібліотек.

В процесі дослідження виконаний аналіз процесу побудови стандартизованого профілю інформаційної системи. Відзначено, що для формування стандартизованого профілю ІС необхідно забезпечити можливість отримання такої інформації:

1. Множини інтерфейсів, що функціонують на одному або декількох рівнях інтероперабельності.

2. Повний профіль процесу, який визначається всіма його інтерфейсами.

3. Перелік сервісів, які можуть виступати в якості компонент сполучення для двох різних сервісів.

Виконано формалізацію семантичної моделі стандартизованого профілю ІС, а також на її основі формалізована модель бази знань онтологічного типу. Запропонована модель включає 3 шари: шар специфікацій, шар предметної області, шар інформаційної системи. Модель визначає загальний каркас, в рамках якого можуть бути побудовані прикладні семантичні моделі, які застосовуються безпосередньо на практиці.

Реалізована система запитів до бази знань, що дозволяє формувати стандартизований профіль ІС. Запропонований каркас визначає базові концепти і ролі, які використовуються при описі:

– автономних інформаційних систем, вбудованих програмних і апаратних компонент, представлених в сервіс-орієнтованому форматі, що володіють певними інтерфейсами, що дозволяють обмінюватися інформаційними потоками різних типів в різних напрямках,

– стандартизованих профілів, представлених сукупністю профілів окремих інтерфейсів з описом структури пропускаються інформаційних потоків в різних напрямках,

– нормативної бази, що включає специфікації і стандарти, що визначають системи класифікацій решті нормативної документації.

Запропоновано етапи інформаційних перетворень процесу компонентної збірки веб-додатків. Представлена загальна схема автоматизованої системи компонентної складання, яка включає підсистему первинного аналізу та форматування даних, підсистему управління сховищем даних про процеси і компонентах, а також підсистему, які формують компонентний склад веб-додатки.

Запропоновано семантична модель, яка описує:

- індивідуальні властивості процесів та компонент веб-додатків;
- ієрархії процесів ("ціле" - "частина");

- комбінації процесів, які були успішно застосовані при розробці веб-додатків;
- компоненти, що реалізують процеси і взаємодіють один з одним за допомогою множини інтерфейсів.

Література

1. Аристов А.В. Разработка и исследование алгоритмов компонентной сборки Web-приложений на основе семантических сетей: дис. д-ра техн. наук: 05.13.11, / Аристов Алексей Владиславович. - К., 2016. - 125 с.
2. Глазунов В.Н. Поиск принципов действия технических систем М.: 1990
3. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем: Учебник / А. М. Вендров. — М: “Финансы и статистика” – 2014. – 544 с
4. Грекул, В.И. Проектирование информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина – 2-е изд., испр. — Интернет-университет информационных технологий - ИНТУИТ.ру, 2010. – 200 с.

Методологія вибору оптимального постачальника підприємства в умовах багатокритеріальності та невизначеності

Бичко Д.В.

Науковий керівник – к.т.н., доц. Шендрик В.В.

Сумський державний університет

Стрімкі зміни економічної ситуації в країні примушують населення шукати товари високої якості за низькими цінами. Це підштовхує потенційного покупця витратити власний час на пошук оптимального постачальника, який задовольняє усім його потребам. Даний процес є складним, що знижує швидкість прийняття рішень.

Метою даної роботи є розробка методології вибору оптимального постачальника підприємства в умовах багатокритеріальності та невизначеності.

Пропонується алгоритм пошуку оптимального постачальника. Він складається з 5 етапів:

1. Визначення системи оцінювання.
2. Визначення критеріїв.
3. Нормалізація оцінювання критеріїв.
4. Нормалізації критеріїв.
5. Визначення оптимальної альтернативи методом аналізу ієрархій.

У роботі оцінювання важливості кожного критерію виконується за допомогою модифікованої психометричної шкали Сааті(таблиця 1).

Таблиця 1 – Шкала оцінювання критеріїв

№	Рівень важливості	Кількісне значення
1	Байдуже	1
2	Не важливо	2
3	Важливо	3
4	Дуже важливо	4
5	Абсолютно важливо	5

На другому етапі виділено вісім критеріїв, які оцінює потенційний покупець за допомогою шкали з таблиці 1:

1. Відповідність характеристик.
2. Зворотній зв'язок.
3. Швидкість відправки.
4. Нейтральні та негативні відгуки.
5. Позитивні відгуки за 1 місяць.
6. Позитивні відгуки за 6 місяців.
7. Кількість продажів за 1 місяць.
8. Кількість продажів за 6 місяців.

Кількісні значення кожного критерія отримуються завдяки парсингу даних з AliExpress. Нормалізація оцінок покупців та постачальників дозволяє привести отриману інформацію у зручний вигляд для розрахунків.

На третьому етапі виконується нормалізація оцінок (формула 1), які отримані від потенційного покупця та дають у сумі 1.

$$w_i = \frac{\overline{w}_i}{\sum_{i=1}^n \overline{w}_i} \quad (1)$$

де \overline{w}_i – нормалізоване значення оцінки i -го критерія; w_i – поточне значення оцінки i -го критерія.

На четвертому етапі відбувається нормалізація кожного критерію (формули 2 – 5).

Розрахунок якості відповідності характеристик/зворотнього зв'язку/швидкості відправки виконується відповідно до формули 2:

$$P_k = w_i \times \frac{Z_i}{5}, \quad (2)$$

де P_k – нормалізоване значення відповідності характеристики/зворотнього зв'язку/швидкості відправки; Z_i – оцінка відповідності характеристик/зворотнього зв'язку/швидкості відправки.

Розрахунок відсотку нейтральних та негативних відгуків виконується відповідно до формули 3:

$$P_1 = w_i \times \frac{N_{SMT} + N_{SMO}}{N_{SMF} + N_{SMT} + N_{SMO}} \quad (3)$$

де P_1 – нормалізоване значення відсотку нейтральних та негативних відгуків; N_{SMF} – кількість негативних відгуків за 6 місяців; N_{SMT} – кількість нейтральних

відгуків за 6 місяців; N_{SMO} – кількість позитивних відгуків за 6 місяців.

Розрахунок відсотку позитивних відгуків за 1 місяць/6 місяців виконується відповідно до формули 4:

$$P_2 = w_i \times \frac{N_{MF}}{N_{MF} + N_{MT} + N_{MO}} \quad (4)$$

де P_2 – нормалізоване значення відсотку позитивних відгуків за 1 місяць/6 місяців; N_{MF} – кількість негативних відгуків за 1 місяць/6 місяців; N_{MT} – кількість нейтральних відгуків за 1 місяць/6 місяців; N_{MO} – кількість позитивних відгуків за 1 місяць/6 місяців.

Розрахунок кількості продажів за 1 місяць/6 місяців виконується відповідно до формули 5:

$$P_3 = 0.001 \times w_i \times (N_{MF} + N_{MT} + N_{MO}) \quad (5)$$

де P_3 – нормалізоване значення кількості продажів за 1 місяць/6 місяців.

На п'ятому етапі рахується комплексний показник конкурентоспроможності кожного постачальника:

$$C_k = \sum_{i=1}^N P_i \quad (6)$$

де C_k – комплексний показник конкурентоспроможності k-ого постачальника; P_i – нормалізоване значення i-ого критерію.

Після підрахунку показника конкурентоспроможності кожного постачальника визначається найкраще значення, яке є оптимальним за заданими критеріями (рис. 1).

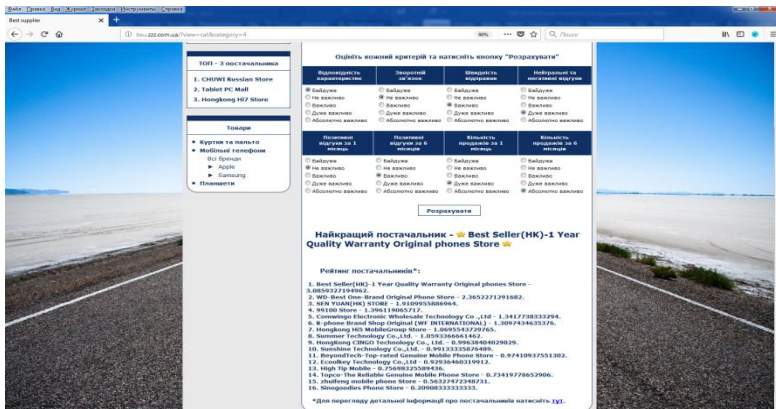


Рисунок 1 – Результат роботи алгоритму

Результатом роботи є розроблена методологія пошуку вибору оптимального постачальника підприємства в умовах багатокритеріальності та невизначеності.

Програмна підтримка паралельних обчислень

Бобилева О.С.

Науковий керівник – д.т.н., проф. Дмитрієва О.А.

Донецький національний технічний університет

Паралельні обчислення являють собою тип обчислювальної архітектури, в якій кілька процесорів виконують або обробляють додаток або обчислення одночасно. Паралельні обчислення допомагають у виконанні великих обчислень шляхом поділу навантаження між більш ніж одним процесором, всі з яких працюють через обчислення в той же самий час. Більшість суперкомп'ютерів використовують паралельні принципи обчислень у своїй роботі.

Паралельна обробка, як правило, здійснюється в операційних середовищах або сценаріях, які вимагають масивного обчислення або великої потужності. Основною метою паралельних обчислень є збільшення доступної потужності обчислень для прискорення обробки додатків або вирішення завдань. Як правило, інфраструктура паралельних обчислень розміщується в межах одного об'єкта, де в серверній стійці встановлено багато процесорів або окремі сервера з'єднані один з одним. Сервер додатків відправляє запит обчислень або обробки, який розповсюджується невеликими частинами або компонентами, які одночасно виконуються на кожному процесорі або сервері [1].

Існує безліч технологій паралельного програмування, які сприяють вирішенню задач на паралельному комп'ютері: FORTRAN, ARCH, A++/P++, ABCL, Ada, C/C++, Cilk, Cray MPP Fortran, CODE, DVM, F--, Fortran 90/95, Fortran M, Haskell, HPC++, mpC, MPI, OpenMP, ParJava, pC++.

Найпоширенішими можна назвати дві технології, а саме:

- MPI – Message Passing Interface;
- OpenMP – Open Multi-Processing.

В якості найбільш поширеної можна виділити технологію, що застосовується для програмування багатопроцесорних систем з розподіленою пам'яттю, з використанням паралельних процесів, що взаємодіють за допомогою передачі повідомлень. Така модель вважається природною для подібного роду систем, тому що передача повідомлень по мережі – можливо єдиний спосіб взаємодії процесів, що виконуються на різних процесорах, які не мають загальної пам'яті.

Основним засобом є Message Passing Interface або MPI – програмний інтерфейс, що входить у стандартний комплекс програмного забезпечення практично будь-якого багатопроцесорного обчислювального комплексу. Використовується для передачі даних, здатний відправляти та отримувати повідомленнями від процесорів, які працюють над деякою проблемою. Розроблено Вільямом Гроуппом та Евіном Ласки.

До складу даної середовища програмування входить бібліотека з інтерфейсом для однієї або декількох мов програмування, а також засіб для

виконання та зборки паралельного додатку. Його можна застосовувати також для кластерів і суперкомп'ютерів. Інтерфейс бібліотеки розроблено за стандартом версії 1.1, що було прийнято у 1995 році, але існують і більш пізні версії, наприклад версія 2.0 – 1997 року, 3.0 – 2012 рік 4.0 – 2016 рік. Усі документи можна знайти у спілці розробників стандарту MPI Forum [2].

Основним засобом програмування для середовищ з розподіленою пам'яттю є MPI. Довгий час для розробки програм у системах з загальною пам'яттю застосовувалась бібліотека MPI або бібліотека для роботи з потоками – POSIX Threads [3]. Програмування на MPI має той недолік, що застосування механізму обміну повідомленнями не дозволяє у повній мірі скористатися можливостями, що дає загальна пам'ять. Тобто, система може працювати неефективно. Застосування потоків хоч і дозволяє повністю задіяти ресурси системи, але є засобом достатньо низького рівня, що є більш характерним, для розробки системних додатків, аніж для паралельного програмування.

В один час почав поширюватися підхід, заснований на розширенні традиційних мов програмування конструкціями, що полегшують автоматичне виявлення паралельних операторів і функцій у програмі. Таких розширень було запропоновано багато. Необхідність уніфікації засобів розробки паралельного програмування призвела до створення у 1997 році єдиного стандарту, що отримав назву OpenMP (Open Multi-Processing) [4]. Стандарт розвивається і зараз і має версію 4.5.

З моменту першої версії стандарту було розроблено багато трансляторів, що підтримують це розширення [2-6].

Література

1. What is parallel computing [Електронний ресурс]. – Режим доступу: <https://www.techopedia.com/definition/8777/parallel-computing>
2. MPI Forum [Електронний ресурс]. – Режим доступу: <http://mpi-forum.org/>
3. POSIX Threads Programming [Електронний ресурс]. – Режим доступу: <https://computing.llnl.gov/tutorials/pthreads/>
4. OpenMP [Електронний ресурс]. – Режим доступу: <http://www.openmp.org/>
5. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл. В. Воеводин // СПб.: БХВ – Петербург, 2002. – 608с.
6. Лупин С.А. Технологии параллельного программирования / С.А. Лупин, М.А. Посыпкин // М.: ФОРУМ: ИНФРА-М, 2011. – 208с.

Аналіз та обробка великих об'ємів даних за допомогою in-memoгу технологій на прикладі APACHE SPARK

Бондар Д. В.

Науковий керівник - д.т.н. Бедратюк Л. П.

Хмельницький національний університет

Вступ

На сьогоднішній день все частіше головним критерієм будь якої системи стає час обробки даних. Найяскравішим прикладом даних систем є системи аналізу даних. Необхідність даних систем як основної компоненти будь якого сучасного програмного забезпечення з кожним роком стає більш очевидною. З кожним днем кількість даних для обробки стає все більше, а час за який їх треба обробити скорочується.

З щоденним розвитком обчислюваних потужностей виникла можливість використовувати велику кількість оперативної пам'яті для аналізу та обчислення даних. В результаті виник ряд способів вирішення задач на основі роботи із тільки оперативною пам'яттю обчислювальних машин замість використання класичних способів на основі регулярних звернень до постійних запам'ятовуючих пристроїв.

Запропоноване рішення дає змогу спростити вирішення цілого класу задач. Найчастіше це регулярні обчислення, які не вимагають зберігання вхідних даних для повторного використання.

Постановка проблеми

У попередніх розділах описані основні проблеми, через яке було виведено дане рішення, як одне з найбільш оптимізованих по часу виконання. Даний клас задач вимагає пошуку вирішення наступних проблем:

1. Як завантажити дані у пам'ять?
2. Як проаналізувати та обробити дані?
3. Як зберегти результати обробки?
4. Як перевірити швидкодію задачі?

Мета роботи: сформулювати рішення поставленої проблеми на основі описаного рішення що дасть змогу прискорити час обробки вхідних даних.

Аналіз існуючих рішень

Здійснивши аналіз інструментарію на основі IN-MEMORY PROCESSING можна виділити два класи реалізацій:

- на основі реляційних баз даних - Oracle, Microsoft SQL Server чи DB2;
- низка нереляційних рішень - Hazelcast, Kafka, Spark.

Перший клас рішень дає змогу робити певні IN-MEMORY обчислення у існуючих базах даних. Даний спосіб чудово підходить для обробки даних які уже знаходяться у одній із згаданих реляційних баз. Головним недоліком даного рішення є необхідність налаштування додаткових модулів до баз даних, використання який може вплинути на швидкодію бази даних для інших систем, які використовують її за прямим призначенням. Крім того

дані рішення вимагають збільшення обчислювальних потужностей серверу.

У випадку коли дані зберігаються не у реляційних базах даних, або задля зведення до мінімуму будь яких ризиків із базою даних даного програмного комплексу використовують сторонні нереляційні рішення. Вони мають наступні переваги:

- можливість завантажувати данні з будь яких джерел;
- можливість зберігати результати у будь які джерела;
- можливість працювати на окремих серверах незалежно від інших модулів програмного продукту.

Виклад основного матеріалу дослідження

Для проведення порівняльних дослідів було наступні рішення:

- класичне рішення на основі послідовної вичитки даних та збереження без використання додаткової пам'яті;
- рішення на основі Apache Spark для вичитки даних та їх обробки у оперативній пам'яті серверу.

Apache Spark був обраний, оскільки він є універсальним рішенням для роботи із Big Data та In memory processing. Він набув популярності за свою гнучкість та велику кількість компонентів для різних потреб в тому числі і In memory processing.

Основною задачею даного досліду є отримання відповідей на відкриті у попередньому розділі питання. Для реалізації обох рішень було обрана мова програмування Python версії 3.7. Перше рішення буде базуватися на стандартному наборі бібліотек вищезгаданої мови. Для іншого рішення будуть використовуватися вищезгадані бібліотеки та драйвер до Spark який називається PySpark.

Вхідними даними є база даних зареєстрованих авто в Україні за 2013-2017 роки під назвою «Відомості про транспортні засоби та їх власників» яка знаходиться у архівах формату Zip. Вихідними даними є результати обчислень збережених у файлах формату JSON.

Задачею є фільтрація даних за наступним критерієм – вік авто при реєстрації є від 2 до 4.

Для упередження будь яких похибок при аналізі даних було обрано середовище тестування Databricks community через наступні переваги:

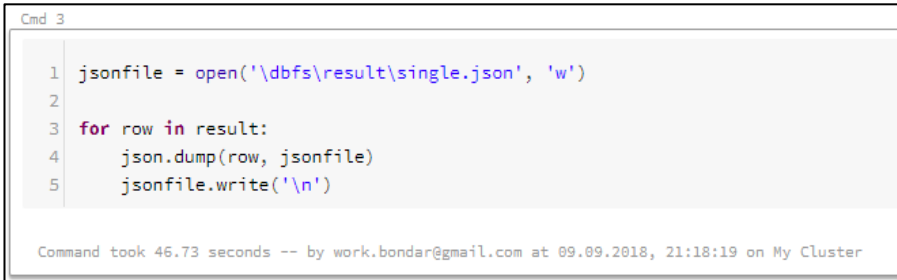
- можливість виконувати задачі в окремому сервері;
- можливість визначати час виконання та кількість використаних ресурсів одного блоку коду.

Приведемо кроки алгоритму який буде реалізовано у обох рішеннях:

- завантаження даних із відкритого реєстру і конвертація архівів у файли формату CSV;
- вичитка даних із CSV файлів;
- фільтрація отриманих даних;
- збереження результатів у формат JSON.

Результати

Після опрацювання вхідних даних в об'ємі 3.5 Гб ми отримали час виконання кожного кроку алгоритму як зображено на рисунку 1.



```
Cmd 3

1 jsonfile = open('\dbfs\result\single.json', 'w')
2
3 for row in result:
4     json.dump(row, jsonfile)
5     jsonfile.write('\n')

Command took 46.73 seconds -- by work.bondar@gmail.com at 09.09.2018, 21:18:19 on My Cluster
```

Рисунок 1 – Результати виконання блоку коду у Databricks

Приведемо результати виконання всіх кроків алгоритму за допомогою двох вищеописаних способів за допомогою таблиці.

Таблиця 1 – Час виконання кроків алгоритму

Назва рішення	Час завантаження	Час вичитки	Час обробки	Час збереження
Класичне	12 хв	4 хв	30 сек	2 хв
На основі Spark	12 хв	3 хв	10 сек	1 хв

Із даної таблиці стає зрозумілим що рішення на основі Apache Spark має кращу швидкодію на наступних кроках:

- вичитка даних;
- обробка або фільтрація даних;
- збереження результатів;

Проте необхідно врахувати, що дана швидкодія була реалізована за допомогою використання більш ніж 3.5 Гб оперативної пам'яті відведеного для обчислень серверу. Класичному рішення знадобилося менш ніж 600 Мб даного ресурсу.

Висновки

В даній роботі було виділено основні вимоги, що висуваються до систем на основі in-memoгу оброблень; визначено переваги та недоліки основних підходів до вирішення задач обробки; досліджено існуючі платформи та виокремлено їх особливості.

Здійснено аналіз отриманих результатів від двох програмних засобів, за допомогою якого можна дослідити переваги роботи системи оброки тестових даних на основі Apache Spark.

Виявилось, що обраний in-memoгу processing фреймворк у цілому справляється з поставленою задачею. Також, є кроки алгоритму де Apache

Spark у зв'язку із вбудованою інтеграцією з HDFS на порядок швидше виконує задачу.

Використання In memory processing парадигм для побудови розподіленої системи обробки даних дозволяє забезпечити автоматично швидке та паралельне обчислення і зберігання даних на внутрішніх дисках вузлів кластера.

Описаний підхід приховує від розробника деталі внутрішнього устрою Apache Spark і надає простий інтерфейс для мови Python, який дозволяє сконцентруватися на предметній стороні задачі, а не на її реалізації. Apache Spark дозволяє фахівцям з обробки та аналізу даних і інженерам прикладних програм працювати з будь-якими джерелами даних.

Література

1. James A. Scott. Getting Started with Apache Spark / James A. Scott. – USA: MapR technologies Inc, 2015. – С. 88.

2. Jacek Laskowski. Mastering Apache Spark. – Режим доступу: <https://www.gitbook.com/book/jaceklaskowski/mastering-apache-spark/details>. – Дата доступу: 23.08.2018.

3. Kannan Prakasam. Beyond Apache Hadoop MapReduce Apache Tez and Apache Spark – Режим доступу: <http://www.sjsu.edu/people/robert.chun/courses/CS259Fall2013/s3/F.pdf>. – Дата доступу: 27.08.2018.

4. Офіційна документація Apache Spark. – Режим доступу: <https://spark.apache.org/>. – Дата доступу: 24.08.2018.

5. Офіційна документація Databricks. – Режим доступу: <https://docs.databricks.com/>. – Дата доступу: 25.08.2018.

6. О.А. Литвинов. Розподілена обробка інформації / О.А. Литвинов, В.С. Ханецький — Д.: ТОВ «Баланс-Клуб», 2013. – С. 314.

Виявлення міжфразових зв'язків при побудові словників предметної області

Гаврілова О. І.

Науковий керівник - к.т.н, проф. Кунгурцев О.Б.
Одеський національний політехнічний університет

Система побудови словників предметної області (СПСПО) використовує урахування частот появи терміну в тексті для вирішення питання доцільності включення того чи іншого терміну в словник. Робота направлена на вирішення проблеми анафор – міжфразових зв'язків (МЗ), які непрямо вказують на термін, але не враховуються СПСПО [1]. Розглядається розробка методу коригування частот появи термінів із автоматичним виявленням МЗ в текстах природною російською мовою, українською та англійською мовами.

Якщо визначити час на таке редагування СПО, як t , то він

складатиметься з часу читання експертом-лінгвістом однієї сторінки (t_r), часу виокремлення слів-термінів (t_t), часу виокремлення потенційних анафор (t_a), часу прийняття рішення про те чи є термін та анафора кореферентами (t_d), часу коригування частоти (t_f).

Формула 1 визначає необхідний час на коригування частот появи термінів в тексті:

$$t = n_p(t_r + t_t + t_a + t_d(n_t + n_a) + t_f), \quad (1)$$

де n_p – кількість сторінок в тексті, n_t - кількість термінів, n_a - кількість потенційних анафор.

Впровадження автоматизованого методу обробки текстів та файлів з частотами термінів на предмет МЗ, виправлення частот термінів дозволить скоротити t_r , t_t , t_a , t_f , та t_d . Ручним способом експерт-лінгвіст в середньому витрачає 4, 20, 30, 15 та 20 хвилин відповідно доперелічених часових характеристик.

Для того, щоб визначити, які саме види анафор найбільш впливають на зміни показника частоти згадування терміна, було проведено дослідження 50 текстів різних предметних областей. При цьому визначалося скільки разів на 1000 слів тексту зустрічається анафора, який її вид, чи можна її формально визначити. Середній показник зустрічаємості анафори по різних стилях текстів склав 54,3 на 1000 слів.

Згідно з лінгвістичними дослідженнями [2], випадки анафори підлягають класифікації; та не всі класи підлягають формалізації, з метою автоматизувати процес їх пошуку. В ході роботи були визначені типи анафоричних зв'язків, які можливо використати для вирішення поточного завдання.

Процес знаходження пари «анафора-референт» можна розбити на декілька етапів:

Етап 1. Виявлення речень тексту, що підлягають аналізу на предмет виявлення анафор.

Етап 2. Виявлення слів, граматично не пов'язаних з членами речення, тобто вставних слів, вставних речень і зворотів.

Етап 3. Визначення виду речення: просте, складне або речення з прямою мовою.

Етап 4. Визначення міжфразових займенникових анафоричних зв'язків.

Завдання визначення анафори в англійських текстах в цілому може бути вирішена запропонованим алгоритмом, але потрібно розглянути деякі особливості пошуку анафор [3]:

- 1) Проблема надмірного «it».
- 2) Перевизначення іменних фраз.
- 3) Важливість одухотвореного.

На основі методу визначення міжфразових зв'язків запропонована технологія побудови СПО (рис.1).

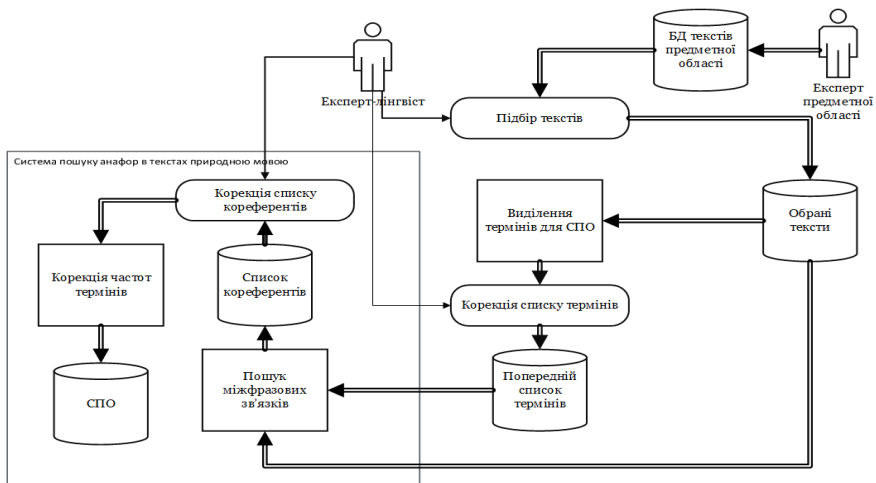


Рисунок 1 – Технологія побудови СПО

Для визначення, як автоматизований спосіб пошуку анафор впливає на час, затрачуваний на коригування частот термінів, порівняно з «ручним» способом було проведено дослідження експертом обох способів. Результати відображені в таблиці 1, в середньому час коригування автоматизованим способом приблизно в 30 разів менший, ніж затрачуваний «вручну».

В таблиці 2 описані результати таких експериментів.

Таблиця 1 – Оцінка часу складання словника (у хвиликах)

Метод побудови словника	t_r	t_t	t_a	t_f	t_d
«Ручний»	4	20	30	15	20
Автоматизований	10^{-3}	10^{-3}	10^{-3}	10	10

Таблиця 2 - Зміна характеристик частот (числа входжень) термінів

Тип текста	Розмір текстів (число слів)	Всього виявлено МЗ	Терміни	
			Всього (кількість)	Змінилось число входжень на (%)
Електротехніка	2500	69	61	11.6
Інформатика	2397	47	58	18.9
Енергетика	4700	175	139	26.9
Економіка	5971	197	148	32.8
Юридичні науки	3500	109	83	31.3

Експерименти допомогли встановити, що приблизно 5% міжфразових зв'язків, що виявлені експертом, не враховуються програмою. Ці помилки

пов'язані із складністю речень в наукових текстах, пунктуаційними помилками, зловживанням іншомовних слів та ін.

Експерименти також показали, що можливо із допомогою розробленої системи скоротити час на урахування міжфразових зв'язків з метою коригування списку термінів, що входять у текст російською та українською мовою, через застосування автоматизованого способу пошуку анафор та автоматизації корекції частот термінів. При цьому програмна реалізація порівнювалась з «ручними» діяльностями експерта-лінгвіста. Винайдені під час тестування неточності встановлення кореферентів є результатом умов, що не залежать від програмної реалізації, а є результатом дії фактору людини – допущення граматичних помилок.

Таким чином в ході дослідження проблеми анафори в текстах українською, російською та англійською для СПСПО:

- 1) Встановлено вплив анафор на частоти входження термінів;
- 2) Визначені типи анафор, що підлягають урахуванню;
- 3) Встановлено, що автоматичний метод урахування анафор покращує часові характеристики обробки текстів СПСПО;
- 4) Ураування анафор покращує правдивість результатів роботи СПСПО.

Література

1. Кунгурцев О.Б. Облік міжфразових зв'язків при автоматизованій побудові тлумачного словника предметної області/ О.Б. Кунгурцев, А.І. Гаврилова, А.С. Леонгард, Я.В. Поточняк// Інформатика та математичні методи в моделюванні – 2016. – №6(2) – С. 173-184.

2. Воронкова А.В. Стратегии когнитивной обработки дискурсивной анафоры пропозитивно-именного типа / Воронкова А.В. С. 27 – 40

Проблема вибору криптографічного алгоритму для шифрування даних користувача

Галкін М.С.

Науковий керівник – к.т.н., доц. Форкун Ю.В.

Хмельницький національний університет

Вступ. Проблема захисту інформації не є новою. Вона з'явилася ще задовго до появи комп'ютерів. Стрімке вдосконалювання комп'ютерних технологій позначилося й на принципах побудови захисту інформації.

На сьогодні криптографічні методи мають широке застосування. Виключенням не стала й область збереження персональних даних користувачів в системах, а також забезпечення надійних методів передачі цієї інформації з збереженням їх цілісності.

При цьому все ж актуальним залишається завдання вирішення проблеми вибору тієї чи іншої технології та методів шифрування щоб за конкретних умов забезпечили оптимальне використання ресурсів

обчислювальної машини та виконували криптографічні перетворень даних за відносно невеликий проміжок часу.

Аналіз літературних даних та постановка проблеми.

Основна класифікація криптографічних алгоритмів показана на рисунку 1. Багато авторів порівнювали ці алгоритми на основі складності по часу та використанню пам'яті[1]. У іншій статті[2] ці алгоритми порівнюються на основі таких параметрів як час виконання перетворень для одиниці даних, довжина ключа, захищеність та обмеження що стосуються кожного алгоритму.

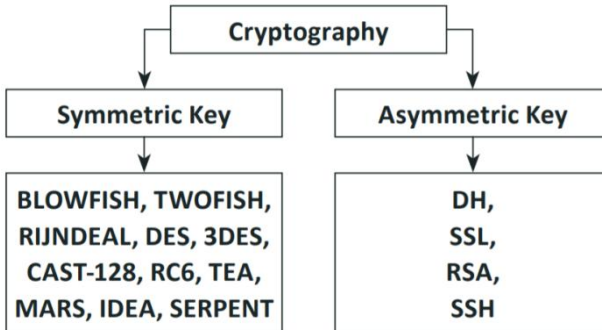


Рисунок 1 - Схема поділу криптографічних алгоритмів

Проаналізувавши знайдений матеріал, виникає проблема вибору та визначення рішення у вигляді найбільш оптимальних криптографічних алгоритмів чи сукупності криптографічних алгоритмів що формують криптографічну систему. Знайдене рішення має враховувати час на виконання операцій кодування/декодування, надійності захисту та часу передачі даних.

Мета та задачі дослідження. Метою роботи є оцінювання криптографічної стійкості методів асиметричного та симетричного шифрування інформації та швидкості виконання криптографічних перетворень для конкретного набору даних задля забезпечення максимальної ефективності та швидкості роботи системи.

Виходячи з мети роботи, задачею є дослідження методів шифрування, представлених в роботах [2] та [3], з точки зору їх криптографічної стійкості та порівняння їх швидкостей виконання.

Виклад основного матеріалу дослідження. Серед багатьох існуючих криптографічних алгоритмів вибрані наступні найбільш часто використовувані: DES, 3DES, CAST-128, BLOWFISH, IDEA, AES, RC6 та RSA. Всі криптографічні алгоритми порівнюються на основі структури, безпеки, гнучкості (що дозволяє розширити алгоритм у майбутньому), надійності (з точки зору відомих способів атак), та обмежень [1], [2]. Таблиця 1 ілюструє порівняльне дослідження вибраних алгоритмів.

Таблиця 1 – Порівняння алгоритмів за структурою, безпекою, гнучкістю та можливими способами атаки.

Algorithm	Structure	Flexibility and Modification	Known Attacks
DES	Feistel	NO	Brute Force Attack
3DES	Feistel	YES, Extended from 56 to 168 bits	Brute Force Attack, Chosen Plaintext, Known Plaintext
CAST-128	Feistel	YES, 128 and 256 bits	Chosen Plaintext Attack
BLOWFISH	Feistel	YES, 64-448 key length in multiples of 32	Dictionary Attack
IDEA	Substitution-Permutation	NO	Differential Timing Attack, Key-Schedule Attack
AES	Substitution-Permutation	YES, 256 key length in multiples of 64	Differential Timing Attack, Key-Schedule Attack
RC6	Feistel	YES, 128-2048 key length in multiples of 32	Brute Force Attack, Analytical Attack
RSA	Factorization	YES, Multi Prime RSA, Multi power RSA	Factoring the Public Key

Безпека в криптографії ґрунтується на тому, як захищений алгоритм від різних атак. Продуктивність цих криптографічних алгоритмів базується на структурі, довжині ключа, розмірі блоку, кількості використаних циклів та криптографічного часу. У кінцевому рахунку, це фактори, які впливають на безпеку певного алгоритму. Розмір блоку відіграє важливу роль у шифруванні та дешифруванні, він виступає базовою одиницею даних.

Більший розмір блоку забезпечує більшу безпеку. AES використовує розмір блоку 128 біт, що в два рази більше, ніж у всіх інших симетричних алгоритмах в обговоренні. Інша оцінка полягає у кількості циклів, що використовуються для процесу шифрування/дешифрування (рис. 2).

Підвищення кількості циклів трансформації, посилює безпеку, оскільки єдиний круг Feistel не забезпечує достатньої безпеки. DES і BLOWFISH мають 16 раундів(циклів) трансформації. 3DES алгоритм має в 3 рази більше циклів ніж DES, тобто 48 раундів. AES має різну кількість раундів залежно від розміру ключа. RC6 - найкращий кандидат за таким критерієм, оскільки має 20 раундів трансформації. Основною проблемою алгоритмів з симетричними ключами є атаки Brute Force, при яких перебираються всі можливі ключі до тих пір, поки не знайдеться точний ключ для розшифровки повідомлення. Довші довжини ключів зменшують доцільність таких атак, оскільки збільшується кількість можливих комбінацій ключа для перебору (рис. 3).

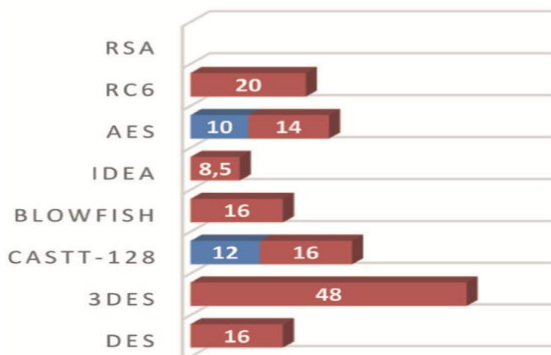


Рисунок 2 - Діаграма кількості циклів трансформації

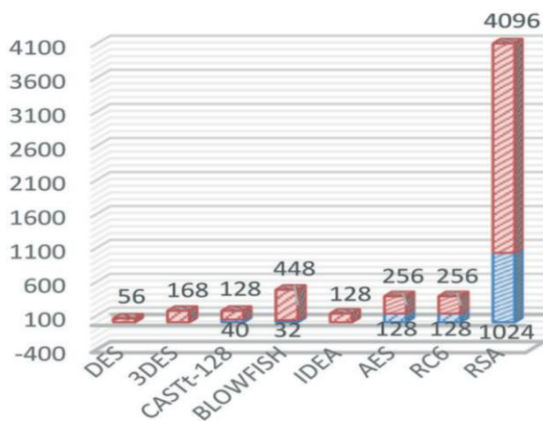


Рисунок 3 - Діаграма довжин ключів

DES має слабкий ключ 56 біт. CAST-128, IDEA використовує 128-бітний ключ, який вважається середнім по складності ключем. 3DES має ключ в 168 біт, що забезпечую гарну стійкість до атаки. RC6 та AES мають змінну довжину ключів 128, 192 і 256, які забезпечують більшу кількість комбінацій ключа. BLOWFISH використовує 448-бітні ключі, які вважаються найдовшими та сильними, наскільки це стосується атак грубої сили.

У асиметричній RSA ключ обміну не потрібен, що підвищує безпеку алгоритму. RSA використовує факторизацію для криптографічного процесу, що значно знижує швидкість алгоритму. Симетричні алгоритми, такі як AES, BLOWFISH та RC6, набагато швидше, ніж RSA. Безпека криптосистеми визначається схемою захищеного шифрування для захисту від нападів. Хоча CAST-128, IDEA, DES, 3DES швидші але вони менш захищені через слабкі ключі.

Результати досліджень. Дослідження показує, що у випадку з симетричними алгоритмами RC6, Blowfish та AES, вони вважаються

безпечними та ефективними за рахунок високої безпеки та менших обмежень. Розширення та гнучкість RC6, Blowfish та AES алгоритмів є вищими в порівнянні з іншими симетричними алгоритмами. Порівняння симетричних та асиметричних ключів показує, що RSA є більш безпечним, ніж будь-який симетричний криптографічний алгоритм.

У порівнянні чітко зазначено, що, хоча асиметричні алгоритми мають вищу безпеку, вони потребують більше часу для обробки та вимагають більшої пам'яті. Практично для ключа використовувати асиметричні алгоритми типу RSA, а для шифрування/дешифрування використовувати симетричні алгоритми. Ефективні криптосистеми можуть бути забезпечені шляхом застосування декількох алгоритмів як гібридної криптосистеми, яка забезпечує високу надійність та безпечне передання даних.

Висновок. В ході даного дослідження було сформовано основні завдання та вимоги до системи збереження особистої інформації користувача зі сторони застосування криптографічних алгоритмів.

В роботі проведено аналітичне дослідження різних алгоритмів симетричного шифрування, таких як DES, 3DES, CAST-128, BLOWFISH, IDEA, AES, RC6 та алгоритму асиметричного шифрування RSA. Аналіз базується на архітектурі алгоритмів, аспектах безпеки та обмеженнях, які вони мають.

З одержаних результатів зроблено висновок про те, що для побудови швидкої та безпечної системи збереження та обміну користувацької інформації слід використовувати комбіновані криптографічні системи. Найбільш вдалим визначено об'єднання таких криптографічних методів як симетричного алгоритму AES – для шифрування/дешифрування даних, та асиметричного RSA – для безпечної передачі ключа шифрування/дешифрування.

Література

1. Mandal, A. K., Parakash, C. and Tiwari, A. (2012). Performance evaluation of cryptographic algorithms: DES and AES. Electrical, Electronics and Computer Science (SCEECS), 2012 IEEE Students' Conference on. IEEE, 2012.
2. Nadeem, A. and Younus Javed, M. (2005). A performance comparison of data encryption algorithms. Information and communication technologies, 2005. ICICT 2005. First international conference on. IEEE. [8] Salama, D. et al. (2008). Performance Evaluation of Symmetric Encryption Algorithms.
3. William, S. (1999). Cryptography and network security: principles and practice, pp. 23-50, Prentice-Hall, Inc.
4. Apoorva, Y. K. (2013). Comparative study of different symmetric key cryptography algorithms. International Journal of Application or Innovation in Engineering and Management, vol. 2, no. 7, pp. 204-6.
5. Daemen, J. and Rijmen, V. (1999). AES Proposal: Rijndael. AES Algorithm Submission, September 3, <http://www.nist.gov/CryptoToolKit>

Метод розробки децентралізованих захищених програм на основі технології BLOCKCHAIN

Глушко Р.М.

Науковий керівник: ктн. доц. Муляр І. В.
Хмельницький національний університет

Біткойн - це новий тренд у світі технологій. Він був запущений у 2009 році, а статусом на 2017 рік здобув капіталізацію ринку на 200 млрд. доларів США [1] та став шостою валютою у світі за об'ємом капіталізації. У той же час його ціна сягала близько 12 тисяч доларів за один біткойн. Статистика показує, що ідею біткойна і криптовалют сучасний світ сприйняв позитивно, що дало змогу цій технології набрати великої популярності та довіри у світі. Чому біткойн став популярним?

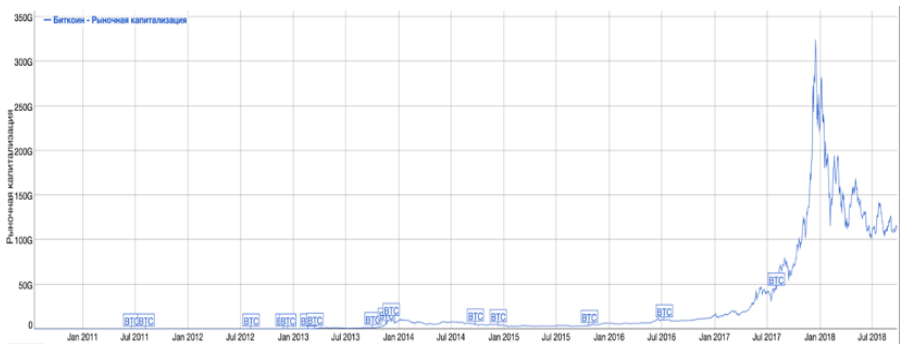


Рисунок 1 - Динаміка ринкової капіталізації біткойна за час існування

Разом із появою Біткойна почався розвиток та інтеграція ідей розподіленості та децентралізації у життя людей. З'явилося багато стартапів та ідей про можливі способи його використання. Біткойн, блокчейн, децентралізація – це ідеї та підходи, потенціал яких випередив свій час. Він змінив не тільки бачення економіки і фінансового сектору, але і змінив парадигму програмування. Змусив людей інвестувати гроші в ідеї, які потенційно змінять наш світ завтра. В сукупності вони створюються математичну модель, якій довіряють тисячі людей у світі та яку не вдалось взломити або знищити – ні одній хакерській організації і до сьогоднішнього дня. Це лише дайжест причин та можливостей, які привертають увагу та змушують заглибитись у світ Біткойна.

Над створенням електронно-валютних систем подібних до Біткойна працював Девід Шаума, котрий першим придумав концепцію електронно-валютної системи e-cash [4]. Цю ідею розвинув далі Увей Даи, запропонувавши вирішення проблеми консенсусу в мережі та використання складних обчислювальних задач для підтвердження роботи в мережі [2]. Хол

Фінні паралельно дійшов до практично таких самих думок з приводу організації електронно-валютних систем[3]. Незважаючи на це, тільки у 2009 році завдяки роботі Сатоші Накамото вдалось отримати першу практичну реалізацію подібної системи на прикладі платформи Біткойн.

Проте на цьому розвиток ідей, які стояли за платформою Біткойн, не закінчились. Над удосконаленням та реалізацією нової платформи, яка би розвинула можливості та заняла обмеження, які були у Біткойна, почали працювати Віталій Бутерін та його команда [5]. У 2013 році вона була запущена під назвою Ефіріум та дозволяє створювати розумні контракти, які би дозволяли розпоряджатись крипто-токенами та звільняти від потреби в арбітражних сторонах при заключенні контрактів в реальному житті. Смарт контракти були реалізовані, як акаунти, якими управляє код контракта. Вони запускаються безпосередньо на платформі Ефіріума, який вміщує своєрідне віртуальне середовище для запуску програм, які є повними по Тюрінгу.

Розвиток смарт контрактів та їхня інтеграція в повсякденне життя людей залежить від розвитку інтернету речей - популяризації побутових пристроїв із вбудованими датчиками та модулями зв'язку для передачі інформації. Ця інформація могла би бути доступна для контрактів через спеціальних провайдерів - оракулів, та братись до уваги при розробці розумних контрактів.

Розвиток децентралізованих програм, як напрямку, призвело до розвитку альтернативних підходів до проектування програм. Це було зумовлено децентралізованою природою нових програм та технологіями, які використовуються для зберігання даних такі, як блокчейн. Визначними в розробці подібних програм є розуміння архітектури, підходів до проектування децентралізованих систем, а також безпека, надійність та захист даних, оскільки часто річ іде про чужі активи та кошти. Аналізом та удосконаленням цих аспектів буде йти мова у цьому дослідженні.

Завданням дослідження є підвищення стійкості та надійності децентралізованих систем, на основі аналізу існуючих математичних моделей, на яких базуються децентралізовані програми та розробка методу, який підвищує стійкість систем в певних умовах, забезпечуючи надійний обмін крипто-токенами.

Для досягнення цього в магістерській роботі були вирішені наступні завдання:

1. Створення методу розробки децентралізованих програм, які базуються на технології Blockchain.
2. Удосконалено існуючу модель розподілу крипто-токенів для досягнення кращої стабільності роботи.
3. На основі існуючої моделі розроблено метод розподілу крипто-токенів у децентралізованих програмах на Ефіріумі.
4. Розроблено алгоритм реалізації методу розподілу крипто-токенів.
5. Розроблено програмне забезпечення, яке реалізує розроблену

модель та алгоритм для платформи Ефіріум.

Розвиток децентралізованих програм, як напрямку, призвело до розвитку альтернативних підходів до проектування програм. Це було зумовлено децентралізованою природою нових програм та технологіями, які використовуються для зберігання даних такі, як блокчейн. Визначними в розробці подібних програм є розуміння архітектури, підходів до проектування децентралізованих систем, а також безпека, надійність та захист даних, оскільки часто річ іде про чужі активи та кошти. Аналізом та удосконаленням цих аспектів буде йти мова у цій роботі. Аналізуючи досвід існуючих децентралізованих програм та можливості, які дає нам індустрія було створено метод розробки децентралізованих програм, на основі технології блокчейн. Метод можна розбити на такі етапи:

1. Створення Прототипу - прекрасною практикою є створення текстового опису суті сервісу чи програми, який би охоплював та проявив настільки ідею та функціонал, наскільки це можливо на етапі продумування логіки програми. Хорошо сформована ідея та концепція програми - повинні стати головною інформацією для наступних етапів та головним чинником для прийняття рішень.

2. Вибір реалізації Blockchain-а - так як розробка буде вестись із використанням технології блокчейн, потрібно буде вибрати підхід, який би задовольнив потреби прототипу та був оптимальним для бюджету проекту. Однією із найважливіших метрик тут є потреба у створенні власного протоколу. Протокол - це головне у децентралізованих програмах на блокчейні. Для програм, які повинні реалізовувати революційні та нові ідеї у світі блокчейна - потрібно буде реалізовувати власний протокол, так як вона покликана удосконалити існуючі, тому вони їй не будуть підходити. Так системи для децентралізованого управління файлами, такі як [] будуть потребувати власні реалізації протоколів, так як поки не існують такої платформи, яка би ідеально підходила для цієї унікальної задачі. Якщо задача і програма може використовувати існуючі протоколи, тоді це сильно удешевить її розробку. Платформи типу Ефіріума дозволяють швидко підняти власний блокчейн проект, буде можливість використовувати токени платформи, та кошти учасників, які уже провели капіталізації цієї платформи. Це означає, що система буде користуватись токенами платформи, яка уже здобула довіру. Також Ефіріум дозволяє створення власних токенів на основі ефірів. Дуже важливим плюсом є можливість використання віртуальної машини Ефіріум платформи, разом із смарт контрактами. У деяких випадках додатково стоїть питання приватності блокчейна, а сама можливість лише обмеженому колу осіб мати до нього доступ. Ці фактори і визначають, яким чином у проекті буде реалізовані блокчейн.

3. Вибір технологічного стеку - природа взаємодії із блокчейном - є асинхронною. Програма-клінт робить запит практично при будь-якій дії із блокчейном і далі чекає на відповідь. Це важливий фактор при виборі стека

технологій для реалізації програми клієнта. Зазвичай, є сенс реалізувати веб-клієнт для взаємодії із блокчейном. У веб-програмуванні JavaScript ідеально підходить для асинхронних інтеграцій, а також існує багато фреймворків здатних полегшити розробку. Це все, а також вибір способу комунікації між блокчейном - це вибір стека технологій. Якщо програма буде використовувати смарт контракти - тоді додатково потрібно підібрати зручний спосіб роботи та відлагодження смарт контрактів на стадії розробки. Не менш важливим на цьому етапі є продумування того, як буде виглядати система для тестування і розробки децентралізованої програми. Це забезпечить швидшу розробку та витратиться менше проблем на неполадки пов'язані із проблемами локального середовища розробки.

4. Реалізація - це етап фактичного створення децентралізованої програми, яка будується на основі рішень, прийнятих на попередніх етапах метода розробки. Фактично розробка зводиться до реалізації крипто-протокола або реалізації смарт контрактів для блокчейн платформ та розробки клієнта, котрий зробить зручним використання блокчейн сховища. Зазвичай зручно мати веб-програму, як клієнт для блокчейна, що робить доступним його для широкого кола людей, не потрібно буде встановлювати настільні програми для комп'ютера.

5. Реліз - запуск робочої версії програми готової для кінцевих користувачів та запуск маркетингових компанії для реклами продукту

6. Продаж початкових токенів - якщо програма працює із власною реалізацією токен-протокола, тоді потрібно залучити інвестиції та провести капіталізації перших токенів системи.

Метод, запропонований у дослідженні дозволяє оптимально розробляти децентралізовані системи основані на Блокчейн технології для програм практично будь-якої складності.

Література

1. Біткоїн побив новий рекорд – 12 тисяч доларів за монету [Електронний ресурс]. - Режим доступу: <https://www.epravda.com.ua/news/2017/12/5/631882/> (дата звернення 05.12.2017)

2. Ecash - Wikipedia [Електронний ресурс]. - Режим доступу: <https://en.wikipedia.org/wiki/Ecash> (дата звернення 27.09.2018)

3. Blind signatures for untraceable payments [Електронний ресурс]. - Режим доступу: <http://www.hit.bme.hu/~buttyan/courses/BMEVINIM219/2009/Chaum.BlindSigForPayment.1982.PDF> (дата звернення 21.09.2009)

4. Digital Cash [Електронний ресурс]. - Режим доступу: <http://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/DigitalCash.html>

5. What is Ethereum? — Ethereum Homestead 0.1 documentation [Електронний ресурс]. - Режим доступу: <http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html> (дата звернення 13.11.2016)

Інформаційна технологія розпізнавання та обліку помольних куль

Голодов Д. В.

Науковий керівник – к.т.н., доцент, П'ятикоп О. Є.

ДВНЗ «Приазовський державний технічний університет»

На сьогодні дуже актуально використання систем комп'ютерного зору (СКЗ) в процесах виробництва. Сучасні СКЗ дозволяють домогтися більш високої точності позиції, збільшити продуктивність, отримати адекватні дані в режимі реального часу порівняно з можливостями людини. Актуальним напрямком використання СКЗ є автоматизація розпізнавання об'єктів на зображенні.

У роботі розглядається задача автоматизації розпізнавання та обліку помольних куль, на зображенні завантажувального лотку гартівного барабану, який встановлено на пристрої для термічної обробки куль. Металеві кулі виробляються різного діаметру від 40 до 120 мм, а їх температура при обробці сягає 600°C. Виробництво куль передбачає такі технологічні операції: нагрів заготовки у секційній печі, формування шару, закалка шару у гартівному барабані, опускання куль у бункер складу. Перед спусканням куль їх необхідно підрахувати.

Навпроти лотку встановлюється камера, яка фіксує обертання барабану. З відеоряду необхідно вилучати кадри з зображенням лотку барабану з кулями. Проблемою є розташування куль у лотку: кулі можуть бути одна за однією, можуть бути зліплені. В залежності від освітлення кулі можуть давати віддзеркалення або тіні. Такі ситуації показані на рис. 1.

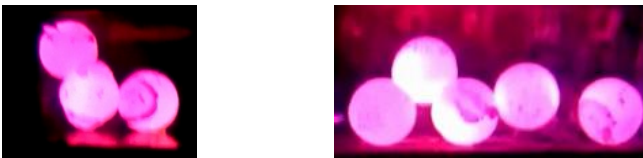


Рисунок 1 – Різновид розташування помольних куль у лотку

Також для наведеної задачі існує ще декілька особливостей. По перше, несприятливі умови роботи камери: можливе засвітлення, накопичення пари у повітрі. По друге, кулі у лотку можуть бути різної температури і необхідно розпізнавати та рахувати тільки гарячі, які відрізняються кольором. Усі ці проблеми потребують індивідуального підходу до вирішення задачі.

Після дослідження традиційних підходів до обробки зображення та розпізнавання об'єктів були визначені необхідні етапи:

- захват кадру з відео ряду;
- аналіз зображення на наявність об'єктів у кадрі -встановлення області появи куль;
- передобробка: перетворення у систему кольорів HSV – підбір границь;
- фільтрації за допомогою функції Гауса – дослідження параметрів;

- сегментація областей рожевого кольору – підбор параметрів порогів;
- виявлення помольних куль - ідентифікація кіл на основі перетворення Хафа.

Перші два етапи безпосередньо залежать від особливостей прикладної задачі та налаштування камери. Для їх реалізації програмними методами був перехоплений відеопотік та обрано кадри. Далі на кожному кадрі була виділена лише смужка області розміром 566x20 пікселів, де очікується поява рожевих куль. В цій області перевірається наявність рожевого (маджентового) кольору. Розташування смуги та % заповнення експериментально встановлено.

Перед подальшою програмною обробкою зображення було проведено моделювання у середовищі MATLAB процесу перетворення у систему кольорів HSV. Моделювання підтвердило, що використання простору HSV доцільне для наданої задачі, були встановлені параметри перетворення. Для проведення сегментації встановлені верхні та нижні границі відбору за кольором в просторі HSV, які дозволяють на цьому етапі визначити вірно помольні кулі з похибкою 0.2%.

Для реалізації інформаційної технології було розроблено автоматизовану інформаційну систему розпізнавання та обліку помольних куль із використанням засобів C# та СУБД MySQL.

Тестування системи було проведено на комп'ютері з характеристиками: процесор: Intel® Core i5-3350P CPU @ 3.10GHz × 4., ОЗП: 8 Гб, ОС: Ubuntu 16.04 LTS. Для експерименту використовувалось відео з роздільною здатністю 640x480 тривалістю 257 секунд, кадрів в секунду ≈30. В результаті роботи системи було відібрано 188 кадрів з кулями із 7772.

Система нарахувала 483 кулі із 485, що становить 99,5%. На відео лоток змінюється перед камерою кожні 0,7с. Експериментально встановлено, що середній час роботи методу визначення кольору: 0,000238 мс, а середній час роботи методу по визначенню куль: 0,002457 мс. Ці дані показують можливість системи працювати у реальному часі. Оскільки є запас часу, то подальші дослідження будуть направлені на підвищення якості обліку.

В результаті виконання роботи було:

- досліджено процес та проблеми обробки кадрів з помольними кулями;
- обрані методи обробки зображення помольних куль: фільтрація на основі функції Гауса, перетворення у систему кольорів HSV, сегментація за кольоровим порогом, градієнтний метод Хафа; досліджені їх параметри;
- побудовані моделі інформаційної технології за методологіями IDEF0, IDEF3, IDEF1X, DFD;
- розроблено автоматизовану інформаційну систему розпізнавання та обліку помольних куль із використанням засобів C# та СУБД MySQL;
- проведені експерименти, які підтвердили можливості роботи системи у реальному часі.

Розробка системи показників якості веб-контенту

Єремєєв Є.Ю.

Науковий керівник – к.т.н, доц. Панфьорова І.Ю.

Харківський національний університет радіоелектроніки

У кожної організації є свій веб-контент, тобто контент сайту. Грамотно розроблений сайт може стати засобом поширення інформації про організацію. Можливості сайту організації не обмежуються тільки наданням інформації для користувачів, з його допомогою можна також дуже добре прорекламувати пропонований товар або послугу. Хоча люди дуже критично, а іноді й з великим негативом ставляться до реклами, але ніхто не може заперечити те, що без реклами неможливо було б дізнатися про існування того чи іншого підприємства.

На сьогоднішній день найефективнішою рекламою є контекстна реклама. Контекстна реклама дає уявлення про ті товари і послуги, які потребує потенційний клієнт. Причина в тому, що цей вид реклами пов'язаний з ключовими запитами, які користувачі вводять на пошукових сайтах. Пошуковий сайт, видаючи необхідний результат на запит, також показує рекламу, яка знаходиться в тому ж контексті, що і введений запит. Необхідно відстежувати ефективність контекстної реклами, контролювати і аналізувати доходи і витрати [1].

Найефективнішим інструментом для просування сайту є контекстна реклама.

У науковій роботі поставлена загальна задача:

- розглянути актуальну проблему аналізу показників ефективності всередині сайту та ззовні сайту;
- проаналізувати, чому для бізнесу важливо знати показники веб-трафіку;
- виявити потреби у контекстній рекламі та проаналізувати показники її ефективності;
- оцінити взаємодію показників всередині та ззовні сайту, проаналізувати, у чому різниця між їх інструментами.

Необхідно обґрунтувати вибір контекстної реклами. Для даної моделі пропонується спосіб розрахунку параметрів цільової функції, що дозволить урахувати додаткові обмеження, які накладаються на розв'язання задачі в реальних умовах.

Передбачити поліпшення ефективності контекстної реклами завдяки направленості розв'язання задачі на мінімізацію основних коштів.

Запропонувати схему для розробки ефективної контекстної рекламної компанії. Від ефективності контекстної реклами залежить успішність веб-сайту, що призведе до підвищення успішності бізнесу.

Для бізнесу важливо знати показники веб-трафіку. Цю задачу вирішують за допомогою веб-аналітики.

Веб-аналітика – відстеження, збір та вимірювання кількісних і якісних даних про відвідуваність сайту з подальшим їх аналізом. Головне завдання веб-аналітики – оптимізація сайту та ініціатив веб-маркетингу.

Веб-аналітика охоплює багато сфер, які вимагають застосування різних методологій або методів збору даних.

Існує два види інструментів веб-аналітики:

- інструменти, які пов'язані з існуванням певного веб-сайту – «внутрішні сайтові»;
- інструменти, які не залежать від існування веб-сайту – «зовнішні сайтові» [2-4].

Внутрішні сайтові інструменти оцінюють переміщення користувачів по сайту та ефективність веб-сайту.

Зовнішні сайтові інструменти використовуються для визначення:

- розміру потенційної аудиторії;
- частки рекламного впливу;
- показників «шуму» (коментарі), які мають місце в Інтернеті в цілому.

Зазвичай припускаються помилки, порівнюючи внутрішні сайтові показники з зовнішніми сайтовими, оскільки значення можуть радикально відрізнятись навіть для основних показників сайту (наприклад, число щомісячних відвідувачів або перегляд сторінок). Це пов'язано з тим, що зовнішні сайтові інструменти вимірюють потенційну аудиторію веб-сайту, а внутрішні сайтові дозволяють вимірювати реальні показники веб-сайту.

Контекстна реклама – це реклама, зміст якої залежить від інтересів користувача. Тому однією з головних її переваг перед іншими видами реклами є її «ненав'язливість» потенційному клієнту.

Контекстна реклама буває пошуковою і тематичною. Пошукова реклама показується в результатах пошуку пошукових систем. Обов'язкова умова показу пошукової реклами – наявність в явній формі пошукового запиту, який задається користувачем.

Тематичні контекстні оголошення показуються на сторінках сайтів, що входять в рекламну мережу Google, якщо тематика реклами відповідає інтересам користувача. Тематична реклама показується як додаткова інформація до змісту сторінок, які переглядає користувач. Вона знаходиться в сфері його уваги.

Залежно від мети рекламної кампанії в контексті будуть змінюватися: місце розміщення самих оголошень; їх вартість; кількість оголошень та інші параметри.

Ефективність контекстної реклами можна оцінити за такими показниками:

- покази, які визначаються кількістю випадків, коли оголошення було показано;
- кількість кліків по оголошенню, які привели до переходів на сайт.

Не менш важливим показником є CTR – відношення числа кліків по оголошенню до числа його показів.

CPC – ціна за клік, тобто сума, яку платить рекламодавець контекстній системі за клік по оголошенню, зроблений користувачем. Важливим показником ефективності контекстної реклами є сеанс і його середня тривалість.

Показник відмов – це відсоток відвідувань, за які користувачі переглянули всього одну сторінку і не зробили інших дій.

Конверсія – один з найважливіших показників ефективності контекстної реклами. Це частка візитів, в ході яких відвідувачі вчинили цільову дію (наприклад, відправили реєстраційні дані, оформили замовлення та ін.).

CPA складає основу таких показників як CPL і CPO. CPL (Cost per Lead) – вартість отримання лід (контактних даних користувача.

CPO (Cost per Order) – вартість оформленого замовлення.

Найважливішим показником ефективності контекстної реклами є ROI (Return On Investment) – коефіцієнт повернення інвестицій, що відображає рентабельність вкладень.

Для розрахунку прибутку від виручки необхідно відняти всі витрати, які несе бізнес. Якщо ROI дорівнює 100%, то це означає, що інвестиції окупилися, але не принесли прибутку, тобто скільки інвестували – стільки й заробили.

Аналогічний показник – ROMI (Return on Marketing Investment) – коефіцієнт повернення інвестицій в маркетингу, оцінює рентабельність реклами, вимірюється у відсотках. Формула аналогічна ROI з тією різницею, що як прибуток використовуються доходи, які принесли рекламні кампанії, а як витрати – обсяг маркетингового бюджету

Збільшення показників ефективності контекстної рекламної кампанії дозволить їй стати ефективним засобом просування.

Існує ряд проблем, пов'язаних з контекстною рекламою, які необхідно вирішити для поліпшення показників.

Перша проблема – низький показник конверсії. Низька конверсія може бути з декількох причин:

- ігнорування «мінус-слів»;
- ігнорування географічного та часового таргетингу;
- ігнорування якості посадкової сторінки.

Ще однією проблемою є низький показник клікабельності. Причиною цього може бути погана якість оголошення або некоректний розрахунок бюджету.

Третя проблема – це занадто велика ціна за оголошення і цьому є кілька причин – коли аналітик робить акцент на високочастотні запити і занадто високі ставки на поганих майданчиках. Високочастотні запити коштовні, тому що серед них висока конкуренція, а ефект не завжди задовольняє, оскільки оголошення показується нецільовій аудиторії, яка по

ньому не буде клікати. В результаті маємо низький CTR, значну кількість витрачених коштів і відсутність результату. Для вирішення проблеми необхідно правильно будувати оголошення: мінімум загальних запитів і максимум точних запитів. Надалі неефективні запити можна прибрати і поставити на їх місце інші релевантні слова.

Ще однією проблемою неефективної контекстної реклами є низькі позиції оголошення, а причина цього – неправильна ціна і неправильна оптимізація оголошення. Оптимальне місце для розміщення оголошення – верхні блоки. Для попадання на верхній рівень в Google AdWords ключовим критерієм є не окремих параметр ставки, а загальний рейтинг оголошення (сукупність релевантності оголошення, величини його поточного і очікуваного CTR, якості цільової сторінки і ставки).

Для вирішення задачі розробки ефективної контекстної рекламної кампанії запропоновано використовувати схему із шести кроків:

- створення проекту рекламної кампанії;
- налаштування параметрів;
- підбір слів;
- складання оголошень;
- управління контекстною рекламою
- запуск проекту.

Для даної моделі запропонований спосіб розрахунку параметрів цільової функції, що дозволить урахувати додаткові обмеження, які накладаються на розв'язання задачі в реальних умовах. Передбачено значне поліпшення ефективності контекстної реклами завдяки направленості розв'язання задачі на мінімізацію основних коштів.

У роботі використаний автоматизований інструментарій для роботи з контекстною рекламою – «SeoPult». Програмний модуль дозволяє управляти контекстною рекламою таким чином, щоб управління бюджетом виконувалось в автоматичному або ручному режимах, при мінімізації бюджету на рекламну кампанію.

Література:

1. Контекстная реклама. Основы. URL: https://elama.ru/blog/kontekstnaya-reklama-snovy/?verify_authentication=false (дата звернення: 01.12.2017).
2. Создание веб-контента. URL: <https://texterra.ru/upload/img/kak-sozdavat-webcontent.pdf> (дата звернення: 01.12.2017).
3. Создание и структурирование веб-контента. URL: https://www.verisign.com/ru_RU/website-presence/website-optimization/web-content-development/index.xhtml (дата звернення: 07.12.2017).
4. Руководство по продвижению веб-контента. URL: <https://www.w3.org/Translations/WCAG20-ru/> (дата звернення: 07.12.2017)
5. Clifton B. Advanced Web Metrics with Google Analytics. Canada - 600 p.

6. Контекстная реклама. URL: <http://sotnik.biz.ua/what-is-google-adwords/> (дата звернення: 07.12.2017).

7. Эффективность контекстной рекламной кампании. URL: <https://www.icontext.ru/consulting/calculating-efficiency/> (дата звернення: 30.11.2017).

8. Показатели эффективности контекстной рекламы. URL: <https://elama.ru/blog/glossariy-15-klyuchevyh-pokazateley-effektivnosti-kontekstnoy-reklamy/> (дата звернення: 30.11.2017).

9. SeoPult – Автоматизированное продвижение сайтов (SEO), ведения контекстной рекламы, контент маркетинга и других рекламных каналов. URL: <https://seopult.ru> (дата звернення: 10.12.2017).

Особливості розробки корпоративної соціальної мережі на прикладі університету

Згара К. Г.

Науковий керівник – к.т.н., доцент, П'ятикоп О. Є.

ДВНЗ «Приазовський державний технічний університет»

Останні роки в світі сучасних інформаційних комунікаційних технологій відбувається активний розвиток технологій мобільного зв'язку та передачі даних. Комбінація цих технологій забезпечує мобільний доступ до ресурсів мережі Інтернет. Завдяки цьому у суспільстві все більше використовується мобільний (бездротовий) Інтернет. Різні служби мобільної передачі даних, засновані на технології мобільного доступу в Інтернет, надають користувачам широкий вибір послуг, розваг та засобів спілкування.

Людство активно користується Інтернетом вдома й на роботі – практично весь час можна перебувати on-line. Доступність мобільного Інтернету радикально змінила форми, зміст, механізми, функції соціальних комунікацій. Одним з таких проявів стали соціальні мережі, які набули на сьогодні статусу невід'ємного атрибуту нашого життя. Представити сучасну людину без соціальних мереж просто неможливо. Спілкування, пошук інформації і друзів, обмін новинами, можливість слухати музику, дивитися відео і фотографії.

В соціальних мережах можна найти однодумців, для цього користувачі об'єднуються в групи. Групи створюють не лише по інтересах та хобі, а й в залежності від належності до співтовариства, організації тощо. Можливо створити групу з колегами по роботі або навчанні для обговорення робочих питань. Але зазвичай поширені соціальні мережі першочергово не ставлять таке завдання. Тому актуально створити таке інформаційне середовище, в якому можна було б створювати корпоративні об'єднання. Наприклад, для спілкування бухгалтерів, або студентів в рамках одного вузу або вишів країни. Програмне забезпечення такої соціальної мережу повинно включати в себе:

– модуль роботи з даними (для користувачів): авторизація,

завантаження даних, створення та передача повідомлень;

- модуль для адміністрування: реєстрація нових користувачів; перегляд вмісту, який був поданий; створення широкомовних повідомлень користувачам; широкомовна передача з статусом для кожного користувача в університеті; відправка трансляції у вигляді push-повідомлень;

- веб інтерфейс для користувачів та мобільний додаток.

Звичайно соціальна мережа це - інформаційна система, тому передбачає використання бази даних, наприклад, можливо використання MySQL. Архітектура такої системи повинна також включати сервер. Для цих цілей автор віддає перевагу Node.js. Створення такого продукту дозволить інформувати студентів про життя університету або кафедри, обговорювати питання.

Модель програмного комплексу для підвищення ефективності набуття стійких позитивних звичок

Карась В. Г.

Науковий керівник – к.т.н, доцент Яшина О. М.

Хмельницький національний університет

Кожна особистість має безліч різних звичок, що можуть бути як позитивними так і негативними. Звичками, по суті, є багато особливостей характеру людини, її способи взаємодії з оточуючими і методи вирішення багатьох життєвих і професійних завдань. Особливості сприйняття і способи мислення – це теж звички. Всі вони виникли в процесі розвитку людини як певні інструменти, за допомогою яких людство намагалося пристосуватися до навколишнього світу.

За [3, 5] звичка – це певна дія (або послідовність дій), яка завдяки багаторазовим повторенням, стає автоматичною, тобто що не вимагає свідомого контролю і зусиль. Звичка – це утворена дорожка в мозку, по якій рухаються нервові імпульси. Звичка – це поведінкова програма, яка автоматично включається в голову у відповідь на певні події-стимули навколишнього світу.

Можливість самостійно формувати свої звички – це шанс набути максимально корисні і ефективні інструменти для успішного вирішення найрізноманітніших життєвих і професійних завдань, замінивши ними застарілі та неефективні. Це можливість зробити життя більш осмисленим, продуктивним.

Найголовніше правило у набутті корисної звички – регулярність. Якщо треба успішно сформувати звичку, треба виконувати одні і ті ж дії кожен день, приблизно в один і той же час протягом деякого тривалого відрізка часу. Тіло і розум повинні звикнути до нового алгоритму дій і думок, а в головному мозку повинні встигнути сформуватися нові стійкі нейронні зв'язки.

Термін, за який набувається звичка, є для кожного різним: комусь достатньо 3-х тижнів, а деяким особам знадобиться півроку. Але якщо брати якусь розумну цифру, то можна сказати, що 90 днів – це оптимальний час для вироблення практично будь-якої стійкої звички, за умови, що індивід буде це робити кожен день. Так що, якщо для формування корисної звички, краще “закладати” саме стільки часу. Один день в тиждень (але не більше) можна зробити вихідним.

Перед початком проектування моделі, потрібно проаналізувати існуючі рішення, щоб виявити необхідність проектованого продукту.

Сьогодні існує мала кількість програм, можливості яких дозволяють використовувати їх для набуття звичок. Проте і ті варіанти або володіють поганим дизайном, або є платними. Також вони не можуть повністю задовольнити усіх висунутих вимог.

Однією із програм для набуття звичок товарів є додаток із назвою «HabitHub». Розробником цієї програми виступає «Ranome Studios». Додаток володіє такими функціями:

- відстеження до п'яти звичок безкоштовно, більше п'яти – платно;
- встановлення щоденних цілей або весь на тиждень;
- наявність нагород, які можуть бути “зароблені” шляхом накопичення балів за виконання звички
- нагадування, коли треба виконати звички;
- перегляд графіків і діаграм, щоб відслідковувати прогрес протягом довгого часу.

Основний недолік додатку – це перенасиченість рекламою і недоступність багатьох функцій в безкоштовній версії і ціна – в платній.

Виходячи з актуальності даної теми, метою даної роботи є створення моделі мобільного додатку для набуття звичок особою, аналіз інформації, надання рекомендацій для користувача для того, щоб зробити вищезгадані 90 днів максимально корисними. Дуже важлива зручність модельованого додатку, його дизайн, синхронізація даних з віддаленим сервером для збереження даних.

Зважаючи на все вищесказане, можна сформулювати основні вимоги до розроблюваного ПЗ:

- можливість створити запис про звичку і відмічати її виконання;
- представлення усіх агрегованих даних у вигляді графіків;
- задавання необхідної частоти виконання для конкретної звички;
- нагадування про необхідність виконання деякої звички;
- можливість експорту даних у вигляді таблиці для Excel.

Попри це, існують стандарти, яких треба дотримуватись, щоб мобільний додаток мав шанси на популярність та успіх:

- хороший дизайн, що вписується в канони платформи і містить різноманітні анімації переходу, правильну палітру кольорів;
- синхронізація користувацьких даних з віддаленим сервером для

синхронізації між усіма пристроями користувача;

- хороша швидкість роботи і плавність;
- різні теми оформлення.

Також додаток має бути швидким, щоб досвід використання користувача був якомога кращий.

Необхідним для проекрованої системи функціональні вимоги. З огляду на вищесказане, система буде використовуватися однією групою людей. Це користувач – це основна і єдина група осіб, яка працюватиме зі своїми власними звичками. Попри те, що в нашій моделі існує тільки одна група осіб, процес її взаємодії з системою є досить складним. Головним робочим напрямком моделі є відслідковування виконання своїх звичок.

Також група користувачів системи має індивідуальні функціональні вимоги. Для цієї категорії, вимоги виглядають наступним чином:

- система повинна забезпечувати можливість додавати нові звички;
- також повинна додана зміна властивостей вже введених звичок;
- можливість формувати різні статистичні дані по кожній звичці;
- можливість представлення статистичних даних у вигляді графіків, кожен з яких висвітлює дані відносно деякого атрибуту;
- експорт даних.

Проектована система повинна відповідати нефункціональним вимогам, які перераховано нижче:

- синхронізація даних з віддаленим сервером;
- програмна архітектура бути масштабованою, тобто має бути побудована правильна архітектура для безболісної підтримки;
- система повинна бути готова до розміщення в магазині додатків.

Системний підхід передбачає деякий спосіб організації дій особистості, такий, який охоплює будь-який вид діяльності, виявляючи закономірності і взаємозв'язки з метою їх більш ефективного використання. При цьому системний підхід є не стільки методом вирішення завдань, скільки методом постановки задач. Вважають, що «правильно поставлене питання – це вже є половина відповіді». Це якісно вищий, ніж просто предметний, спосіб пізнання. Розроблювальний програмний продукт повинен відповідати основним системним параметрам (простота, надійність, ефективність), принципам щодо інформації та основним принципам системного підходу:

- цілісність;
- ієрархічність;
- структуризація;
- системність;
- множинність.

Проектування і розробка програмного комплексу включає: затвердження первинного технічного завдання розробки сервісу, дизайн - створення графічних елементів макету, стилів і елементів навігації, розробка

програмного коду, модулів, бази даних і інших елементів необхідних в проєкті, тестування і розміщення сайту в мережі Інтернет.

Відповідно до описаних потоків даних основною задачею даної системи є забезпечення всіх можливих засобів для набуття особистістю позитивних звичок.

Додаток повинен працювати на мобільних пристроях для більш тісної взаємодії з користувачем, бо відомо, що телефон – це пристрій, що перебуває з користувачем, фактично, завжди.

Сформувавши та проаналізувавши предметну область можна вказати, що основним інтерфейсом для роботи користувачів є смартфон. Дана модель проєктується з врахуванням того, що у майбутньому буде розроблятися версія під інші платформи. У результаті визначеної вище моделі, можемо сформуванати таку діаграму потоків даних, яка показує загальну поведінку моделі і додатку, що її реалізовує (рисунок 1). Представлена на цій схемі модель є основою проєктування інтерфейсу користувача і додатку в цілому.

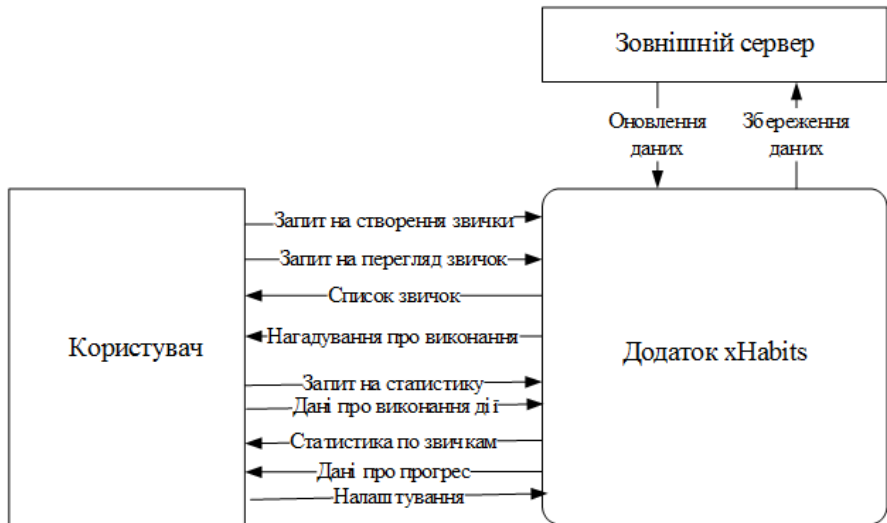


Рисунок 1 – Діаграма потоків даних

Перша платформа, на якій буде реалізовано додаток – Android. Ця платформа була вибрана, тому що вона популярна, поріг входження нижчий, ніж на iOS, і магазин додатків набагато лояльніший до розробника.

Після вибору платформи, на якій буде реалізовуватися розроблюваний продукт, важливо визначитися з мінімальною версією Android, яку може підтримувати наш додаток. Тут треба орієнтуватися на 2 фактори: розповсюдженість ОС і інструменти, що є доступні для кожної з версій.

Таблиця 1 – розповсюдження кожної з версій Android

Версія	Кодова назва	API	Розповсюдження
2.3.3 – 2.3.7	Gingerbread	10	0.2%
4.0.3 – 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%

Очевидно, що вказані вище фактори є двома протилежностями: чим менша підтримувана версія ОС, тим більша кількість користувачів може залучитися, з іншої сторони – підтримка старих платформ змушує використовувати старі інструменти і не дає доступ до нових, що суттєво знижує швидкість і зручність розробки. Кінець кінцем, було вирішено встановити мінімальною версією 4.4 Kitkat. Цей вибір є найбільш оптимальним, тому що додаток з такою мінімальною версією охоплює більшу частину ринку (96,5%) і всі сучасні інструменти розробки підтримуються даною версією.

Отже, в даній роботі створено програмний комплекс для допомоги при набутті звичок користувачем. В результаті розробки, на основі ряду досліджень та публікацій було проаналізовано питання формування звичок в психологічному аспекті. Основні задачі, виконані розробленим комплексом, здатні підвищити ефективність процесу набуття звички. Розробка даного продукту є обґрунтованою. У майбутньому сервіс можна розширити, додавши до нього додатки під різні платформи.

Література

1. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коаллен, Келли А. Хьюстон Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е издание. Вильямс 2008 год.
2. Майкл Физерс Эффективная работа с унаследованным кодом. Вильямс 2017 год.
3. Запорожченко В. Г. Образ жизни и вредные привычки.
4. Колесов Д. В. Предупреждение вредных привычек у школьников.
5. Mark L. Murphy Busy coder's guide to Android development.

Технологія програмної реалізації системи управління інформаційними ресурсами на основі автономних сценаріїв

Коваль В.А.

Науковий керівник к.т.н., доц. Огневий О.В.

Хмельницький національний університет

Для вивчення питань практичного застосування розроблених моделей і принципів створення і функціонування автономних сценаріїв необхідно виділити основні проблеми застосування технології автономних сценаріїв (АС) для управління інформаційними ресурсами корпоративної інформаційної системи (КІС) сучасного підприємства при реалізації системи управління інформаційними ресурсами (ІР) на основі АС.

Процес створення системи управління інформаційними ресурсами (СУІР) на основі АС, як і будь-якої іншої системи управління, повинен ґрунтуватися на відповідній технології розробки [2].

Стандартні технології створення програмних додатків включають послідовність етапів, що обмежені тимчасовими рамками і повинні закінчуватись випуском конкретного продукту (програмних модулів, моделей, документації тощо).

Етапи створення програмного продукту включають послідовність наступних процесів [3]:

- виявлення цілей і завдань відповідного етапу;
- визначення вимог;
- вибір методів вирішення завдань, технологій і платформ;
- реалізація завдань (модель інформаційних процесів підприємства, вимоги до системи, інформаційна модель, програмний модуль, проект системи і ін.);
- документування;
- аналіз результатів.

Створення будь-якої ІС, відповідно до сучасної методології, являє собою процес розробки і послідовного перетворення ряду узгоджених моделей на всіх етапах життєвого циклу (ЖЦ) системи. На кожному етапі ЖЦ системи УІР створюються специфічні моделі - вимоги до системи та підприємства, проект системи, вимоги до програмних модулів, інтерфейсу тощо. Моделі формуються робочими групами розробників та проєктувальників, зберігаються і акумулюються в репозиторії проєкту. Створення моделей, їх коригування, контроль і впровадження (надання в колективне користування) здійснюється з використанням спеціальних програмних CASE-засобів (AllFusion Modeling Suite, Borland Application Life Cycle Management, Oracle Developer Suite і ін.) [4].

Розглянемо більш ґрунтовно етапи проєктування і реалізації СУІР на основі автономних сценаріїв.

Кожна стадія (етап) має завершуватися в чітко визначеній контрольній точці. До цього часу повинні бути досягнуті певні визначенні результати і прийнято конкретне рішення про подальшу розробку.

На передпроектній стадії (етапі) проводиться вибір проекту до реалізації, виконується його початковий аналіз, тобто визначаються межі проекту, проводиться оцінка його розмірів, вартості та дохідності, вивчення всіх можливостей реалізації.

Далі проводиться аналіз предметної області, який включає:

- всебічне обстеження і детальний опис компанії і КІС;
- аналіз результатів обстеження та виявлення факторів, що впливають на якість реалізації завдань управління інформаційними ресурсами;
- вироблення рекомендацій за первинними напрямками удосконалення СУІР.

Результатом аналізу предметної області є формування вимог до СУІР, що повно і точно відображають цілі та завдання конкретної КІС. Множина моделей опису вимог потім перетвориться в систему моделей, що описують концептуальний проект (або модель) СУІР. Концептуальна модель це формалізований опис понять і відносин предметної області відповідно до універсальних правил і стратегій поведінки об'єкта. Закінченням цієї стадії (етапу) є затвердження Технічного завдання (або плану робіт), на підставі якого виконуються наступні етапи.

На етапі проектування розробляється детальна базова архітектура майбутньої системи, описуються всі проектні рішення, пов'язані з розробкою інформаційного забезпечення СУІР, вибором платформ і технологій, на основі яких буде функціонувати система, програмних засобів реалізації, визначаються вимоги до інтерфейсу користувача тощо. [3].

Розробка архітектури СУІР передбачає визначення її компонентів (інформаційних ресурсів, автономних сценаріїв, груп користувачів) і функцій, визначених на основі стану керованого інформаційного середовища, способів взаємодії між АС, небажаних (виняткових) ситуацій, які можуть виникнути при виконанні автономними сценаріями своїх функцій, і реакцій на ці ситуації. Результатами етапу проектування є:

- модель СУІР;
- схема бази даних (на підставі розробленої на етапі аналізу ER-моделі);
- набір специфікацій автономних сценаріїв (вони будуються на базі моделей функцій);
- опису всіх елементів динамічного і статичного аспектів СУІР (специфікації класів, об'єктів, атрибутів і операцій);
- колекція шаблонів класів автономних сценаріїв і заготовок текстів програм для кожного з них;

- колекція дій, виконуваних сценаріями;
- колекція інтерфейсів.

Етап проектування завершується затвердженням технічного і робочого проектів системи.

На етапі реалізації і тестування створюється та заповнюється даними БД, впроваджується інтерфейс системи, проводиться кодування вбудованих процедур і окремих модулів, доробка розроблених автономних сценаріїв, їх тестування, інтеграційне тестування системи УІР в цілому, перевірка її якості та відповідності вимогам користувачів. Створюються матеріали для навчання та керівництва користувачів, системна документація. Результатом стадії реалізації і тестування є продукт, готовий до експлуатації.

Традиційні підходи до побудови інформаційних систем базуються на ідеї того, що на початку проекту складно визначити весь обсяг даних та перелік всіх аналітичних задач, що будуть розв'язуватись кінцевими користувачами [3]. Наприклад, методологія Oracle DWM FT (Datawarehouse Method Fast Track - метод створення сховищ даних «високошвидкісна траса») виходить з припущення, що розробники впродовж цілого життєвого циклу інформаційної системи будуть визначати і аналізувати вимоги до сховища даних. Заснована на DSDM (Dynamic System Development Method - метод розробки динамічних систем) ця методологія реалізує підхід RAD (Rapid Application Development - швидка розробка аплікацій).

Методологія RAD отримала широке застосування у сфері створення автоматизованих інформаційних систем. Вона охоплює всі етапи життєвого циклу інформаційних систем. Основні принципи RAD полягають в наступному:

- використання спіральної моделі ЖЦ системи;
- повне завершення робіт на кожному етапі ЖЦ не обов'язкове;
- застосування CASE-засобів і засобів швидкої розробки додатків;
- тестування і розвиток системи здійснюється одночасно з її розробкою.

Залежно від масштабу створюваної СУІР, процес проектування може забезпечуватися «ручним» проектуванням або різними засобами його автоматизації, наприклад, CASE-засобами.

CASE - програмні засоби, що підтримують процеси створення і супроводу автоматизованої системи, включаючи аналіз і формулювання вимог, проектування прикладного ПЗ (додатків) і баз даних, генерацію коду, тестування, документування, забезпечення якості, конфігураційне керування і керування проектом тощо [4]. CASE-засоби, системне ПЗ і технічні засоби утворюють повне середовище розробки СУІР.

Системи підтримки процесів розробки і супроводу (CASE) дозволяють знизити складність процесів створення програмних додатків, збільшити їх ефективність, зменшити витрати на розробку і максимально використовувати потенціал інформаційних технологій. Для скорочення зусиль на

обслуговування додатків при їх адаптації до постійних змін в програмно-технічному середовищі та забезпечення єдиного, простого інтерфейсу з кінцевими користувачами фахівці рекомендують використовувати інструментальні засоби розробки прикладних систем, орієнтовані на архітектуру цих систем.

Серед методологій, орієнтованих на системну архітектуру програмних засобів можна віділити інтегровану програмну архітектуру ISA (Integrated Software Architecture), що розвивається фірмою Software AG, і CASE-технологію розробки систем фірми ORACLE, які доведені до промислового зразка. Для системи, що розвивається, та використовує нестандартні рішення, методи і технології, застосування тиражованих CASE-засобів є досить складним, тому вони можуть використовувати замовну або самостійну розробку аналогічного інструментарію.

Система управління інформаційними ресурсами КІС включає інструментальні і прикладні програмні компоненти. Перший компонент орієнтований на підтримку процесів аналізу і організації предметної області, проектування АС із заданою поведінкою. Він має засоби розробки специфікації архітектури системи і поведінки сценаріїв, засоби швидкого генерування програмних додатків і спостереження за поведінкою створених сценаріїв. Другий - забезпечує ефективне середовище для функціонування АС і СУІР в цілому. Для зручності користувачів і забезпечення гнучкого управління інформаційними ресурсами прикладний програмний комплекс повинен включати додаток для адміністратора, який дозволяє створювати АС, управляти ними, а також координувати дії АС, які виконують функції: фільтрації файлів користувачів, передачі файлів на сервер і їх архівування, контролю вільного дискового простору на сервері і робочих станціях, резервного копіювання, регулярного чищення дисків, забезпечення доступу користувачів до архівних даних, оповіщення користувачів про результати виконання завдань.

Література

1. Коновалов О.Ю. Агентні технології у розподілених обчислювальних системах / О.Ю. Коновалов // Наукові записки Українського науково-дослідного інституту зв'язку.- 2013. - № 2. - С. 63-68.

2. Примостка А.О. Особливості розробки та проектування мультиагентних систем / А.О. Примостка // Наукові записки Національного університету «Острозька академія». Серія «Економіка» : збірник наукових праць / ред. кол.: І.Д. Пасічник, О.І. Дем'янчук. – Острог : Видавництво НУ «Острозька академія», 2015. – Випуск 28. – С. 159–163.

3. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів: навч. посіб. / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук.-К.: ЦУЛ, 2012. - 296 с.

4. Циганов О.В. Основи проектування систем штучного інтелекту: навч. посіб. / О.В.Циганов.– Одеса: Наука і техніка, 2006. – 197с.

Адаптивний метод проектування програмного забезпечення

Кушнірук С.Л.

Науковий керівник: к.т.н. доц. Муляр І.В.

Хмельницький національний університет

В теперішніх умовах багато уваги приділяється проблемам, що пов'язані з проектуванням складних програмних систем. При цьому значний комплекс проектних завдань пов'язаний не лише з створенням нових програмних комплексів, але і з необхідністю структурної та параметричної перебудови діючих систем з метою їх реформування та модернізації. Динамічні зміни зовнішніх умов, потребують постійного технічного та алгоритмічного удосконалення [2]. Це призводить до необхідності розробки математичного та програмного забезпечення для комплексного рішення задач аналізу та синтезу складних програмних систем з урахуванням ієрархічності та можливості перебудови структури програмного забезпечення.

Процес структурного та параметричного синтезу програмних систем ускладнюється високою динамікою, нестабільністю та стохастичністю вимог до програмного забезпечення, що ускладнює використання аналітичних моделей для прийняття проектних рішень. Це призводить до необхідності застосування алгоритмічних оптимізаційних моделей, в яких відсутні явні аналітичні формулювання критеріїв оптимальності та обмежень, а наявна лише можливість визначення їх значень для кожного із варіантів з застосуванням різних моделюючих процедур. Складність оцінки властивостей таких моделей обмежує можливість використання стандартних оптимізаційних процедур, що в кінцевому результаті знижує ефективність процесу оптимального проектування [2].

Рішенням вказаної проблеми може бути досягнуто при використанні адаптивного підходу до проектування програмних систем. Даний підхід передбачає побудову комплексу алгоритмів оптимізації які забезпечують поєднання процесу більш повної формалізації задачі з її рішенням, та їх поєднання з процедурами багатоваріантного моделювання в інтелектуальну адаптивне середовище з можливістю його динамічного налаштування на різні класи вирішуваних задач оптимального проектування. При цьому, важливою вимогою до розроблюваного математичного забезпечення являється можливість ефективного рішення задач що описані складними алгоритмічними моделями.

Таким чином, актуальність магістерської роботи визначається необхідністю розробки теоретичних основ, математичного та програмного забезпечення для рішення задач пошуку оптимальних варіантів складних програмних систем при їх реформуванні та модернізації з можливістю врахування динамічних та стохастичних аспектів роботи.

На сучасному етапі розвитку ринкової економіки актуальними

проблемами залишаються проблеми створення нового програмного забезпечення та комп'ютерних систем а також, не менш важливими проблемами, залишається модернізація та реструктуризація вже створених та працюючих програм та систем .

Першим етапом розробки нової програмної системи або удосконалення вже існуючої, є комплексний всебічний аналіз всіх характеристик яким система повинна відповідати. Для того щоб сформувати загальне уявлення в цілому про програмну систему як про систему яка проектується, необхідно проаналізувати наступні характеристики [3]:

Група 1:

- 1) Алгоритмічна складність (логіка алгоритмів обробки інформації).
- 2) Склад та глибина роботи функцій обробки які будуть реалізовані.
- 3) Повнота та системність функцій обробки.
- 4) Об'єм файлів програми.
- 5) Вимоги до операційної системи та до технічних засобів обробки

програмними засобами.

- об'єм дискового простору.
- необхідна кількість оперативної пам'яті для запуску.
- тип процесора.
- версія операційної системи.

Наявність обчислювальної мережі.

Наступна ступінь диференціації характеристик виникає при переході до аналізу окремих компонентів програми. При цьому відбувається деталізація характеристик першої групи. Так, структура програми деталізується у вигляді схем основних технологічних взаємозв'язків в середині кожного з модулів [4].

До другої групи характеристик відносяться характеристики форм організації основних функціональних процесів програми:

Група 2:

- 1) Технології, мови програмування за допомогою яких буде проводитись розробка.
- 2) Організація зв'язків між компонентами програми.
- 3) Створення інтерфейсів для комунікації між компонентами створених з використанням різних технологій або мов програмування.
- 4) Специфікація даних з якими буде працювати той чи інший компонент системи.
- 5) Рівень завантаженості каналів зв'язку.
- 6) Характер передачі даних між компонентами програми.
- 7) Спосіб зберігання даних.
- 8) Швидкість обробки даних.

На основі аналізу цих характеристик визначаються напрямки створення програмної системи та визначаються можливі напрямки розвитку

при її подальшій модернізації.

На рисунку 1 показана схема процесу проектування програмної системи яка відповідає трьом рівням оптимізації.

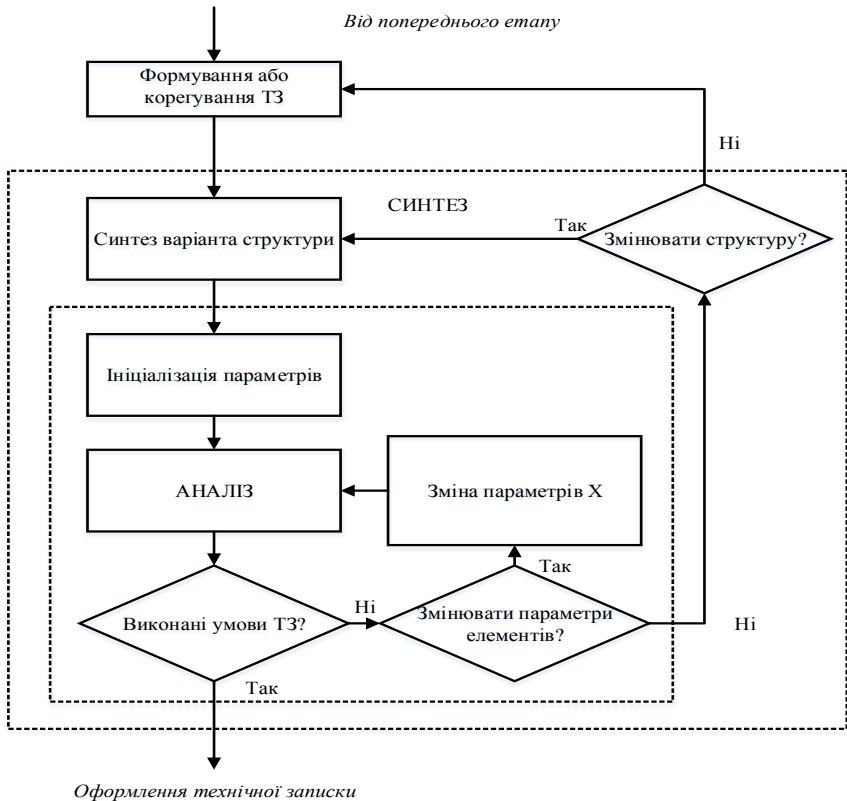


Рисунок 1 - Схема типового маршруту проектування

Перший рівень полягає у виборі найкращої ідеї або принципу дії об'єкта проектування; другий рівень – у пошуці найкращого структурного рішення або принципу дії; третій рівень – у визначенні найкращих параметрів для вибраної структури. Задачі першого рівня характерні для етапу зовнішнього проектування і вирішуються з використання експертних підходів і методів. Математичне забезпечення сучасних САПР орієнтовано на рішення задач другого та третього рівня.

Отримання оптимального рішення пов'язано з вибором найкращого варіанта з деякої допустимої множини рішень. Склад оптимізаційного процесу являє собою набір варіантів, які відповідають вимогам до

функціонування програмної системи, а також набору експлуатаційних і технічних обмежень.

Література

1. Информационные системы и технологии: Научное издание. / Под ред. Ю.Ф. Тельнова. - М.: ЮНИТИ, 2016. - 303 с.
2. Черников Б.В., Поклонов Б.Е. Оценка качества программного обеспечения практикум. - М.: ИД. "Форум" : ИНФРА - М. 2012-400с.
3. Хеффельфингер, Д. Разработка приложений Java EE 6 в NetBeans 7 / Д. Хеффельфингер. - М.: ДМК Пресс, 2013. - 330 с.
4. Максимов, Н.В. Современные информационные технологии / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 512 с.

Модель оцінювання математичної компетентності студентів

Левковська У. О.

Науковий керівник – к.п.н., доц. Кучерук О. Я.

Хмельницький національний університет

Однією з перспективних тенденцій реформування вищої освіти на сьогодні є впровадження компетентнісного підходу, який акцентує увагу на результатах навчання, причому в якості результатів розглядається не сума засвоєної інформації, а здатність застосовувати знання, вміння та особистісні якості для успішної діяльності в певній галузі.

На сьогоднішній день немає однозначної відповіді на питання про закономірності і механізми формування та розвитку компетентності, немає єдності поглядів на структуру компетентності, мають місце значні труднощі в поясненні механізмів формування компетентності. Важливим питанням є оцінка компетентності.

Питання вдосконалення методів вимірювання та оцінювання результатів навчання є одним з найбільш актуальних не тільки у педагогічній теорії й практиці. Оскільки його розв'язання дозволяє визначати ефективність та шляхи покращення навчального процесу, методів і організаційних форм навчання, забезпечення якості освіти а й сприяє якісному особистісному розвитку особистості спроможної: застосовувати набуті знання, бути конкурентоспроможним, компетентним, володіти необхідними професійними навичками, здійснювати самоаналіз та самоконтроль, бути лідером, прислуховуватись до думки оточуючих, вміти ставити цілі та їх досягати.

Важливою складовою професійної компетентності випускників, особливо інженерних спеціальностей, є математична компетентність.

Питаннями формування математичної компетентності займалися

А. Раков, Г. Луканкін, Ю. Колягін, Е. Зеєр, І. Зіненко, Л. Ляшенко, Л. Нізамієва, Я. Стельмах, А. Дахін, А. Маркова, Л. Тархан та інші.

Математична компетентність, за С. Раковим, – це вміння бачити та застосовувати математику в реальному житті, розуміти зміст і методи математичного моделювання, уміння будувати математичну модель, досліджувати її методами математики, інтерпретувати отримані результати, оцінювати похибку обчислень [4].

Оцінка рівня сформованості компетентності, зокрема математичної, – нове для вузівської системи завдання, яке неможливо вирішити лише за допомогою традиційних методів контролю та інструментів оцінювання. На даний момент відсутні єдині підходи, методи та алгоритми оцінки рівня сформованості компетентності.

Для оцінки компетентності важливо розуміти її структуру. Ми розглядаємо математичну компетентність, як 3-компонентну структуру: когнітивний, діяльнісний та особистісно-мотиваційний компоненти (рис.1).

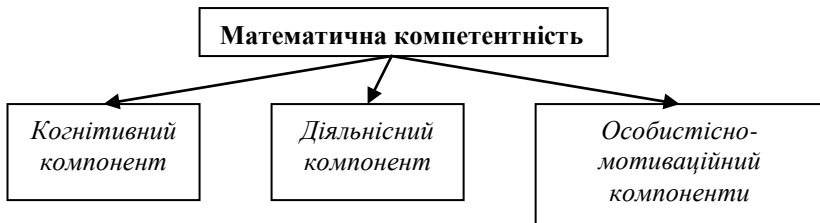


Рисунок 1. Структура математичної компетентності

Математична компетентність формується в результаті вивчення не однієї дисципліни, тому рівень її сформованості залежить від результатів вивчення всіх зазначених дисциплін. Схематично співвідношення між математичною компетентністю та результатами навчання можна виразити наступним чином:

$$MK \Leftrightarrow (P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_n),$$

де MK – математична компетентність; P_i – результати навчання по різним дисциплінам ($i = 1, 2, \dots, n$); знак \wedge означає логічну операцію кон'юнкцію.

Більшість науковців звертають свою увагу на оцінювання саме першого та другого компонентів: І. Ісаков – застосування двохступеневого статистичного методу; П. Мітрошин – метод аналізу ієрархій (найбільш поширений метод); М. Шут – оцінювання компетентності на основі тестової технології; С. Баркалов – модель оцінювання знань на основі оцінки латентних змінних за Рашем; С. Мойсєєв – оцінювання латентних змінних за допомогою методу найменших квадратів; Н. Кочерга – оцінювання за допомогою теорії парних матричних ігор; В. Васильєв – використання статистичного аналізу для оцінювання знань; М. Маляр – застосування

апарату нечіткої логіки та інші.

Важливо оцінювати всі компоненти в цілісності, тоді буде чітко видно всі моменти та етапи на яких виникають труднощі і що дійсно варто поліпшувати. Крім того, важливо показати як залежить результат оцінювання компетентності від всіх складових: отриманих знань під час навчання; те як людина потім використовує набуті знання, застосовуючи до вирішення життєвих задач; особистісний компонент. У свою чергу, кожен компонент може бути оцінений за декількома показниками. Їх кількість у різних компонентів може бути різною.

При створенні моделі не всі компоненти математичної компетентності можуть бути оцінені за допомогою строгого математичного апарату, наприклад поняття мотивації, тип мислення, рівень самоорганізації і т.д. Для їх аналізу доводиться вдаватися до таких методів як експертні. Але експертні оцінки найчастіше виражаються словами природної мови і носять нечіткий, невизначений характер. Внаслідок цього, для вирішення завдань аналізу і обробки інформації, що має нечіткий характер краще застосовувати нечіткі моделі, які спираються на теорію нечітких множин.

Аналіз проведених досліджень в галузі нечіткого моделювання в такій науці як педагогіка і, зокрема, в питаннях оцінки компетентності виявив, що хоча багато дослідників акцентують увагу на необхідності використання даного методу, конкретних розробок і апробованих методик дуже мало.

Література

1. Головань М. С. Компетенція та компетентність: порівняльний аналіз понять / М. С. Головань // Педагогічні науки: теорія, історія, інноваційні технології. – 2011. – № 8 (18). – С. 224 – 234.
2. Захарченко В. М. Національний освітній глосарій: вища освіта / 2-е вид. перероб. і доп./авт.-уклад.: В. М. Захарченко, С. А. Калашнікова, В. І. Луговий, А. В. Ставицький, Ю. М. Рашкевич, Ж. В. Таланова/ За ред. В. Г. Кременя. – К. :ТОВ Видавничий дім «Плеяди», 2014. –100 с.
3. Красильнікова Г. В. Моніторинг якості швейної галузі у ВНЗ. Теоретичні та методичні засади, Хмельницький. 2015.– 125с.
4. Раков С. Формування математичних компетентностей випускника школи як місія математичної освіти. / С. Раков // Математика в школі. – 2007. – №5 – 2 -7 с.
5. Хом'юк В. В. Математична компетентність майбутнього інженера: аналіз феномені .Педагогічні науки: теорія, історія, інноваційні технології,. Вінницький національний технічний університет, 2004, – № 3 (37).

Розробка мобільного додатку гібридної реальності

Лисенко О.В., Матвієнко Ю.С.

Полтавський національний педагогічний університет імені В.Г. Короленка

В статті проаналізовано поняття та характеристики гібридної реальності та доповненої реальності як її основної складової. Продемонстровано результати розробки мобільного додатку із впровадженням технологією гібридної реальності для туристичної галузі міста Полтави.

Гібридна (змішана) реальність (англ. Mixed reality, MR) є наслідком об'єднання реального і віртуальних світів для створення нових оточень і візуалізацій, де фізичний і цифровий об'єкти співіснують і взаємодіють в реальному часі. Існує не тільки в реальному або віртуальному вигляді, а як суміш реальної і віртуальної реальності, охоплює доповнену реальність і доповнену віртуальність.

Змішана реальність з двома її попередниками взагалі стала точкою спотикання для зміни нашої повсякденності назавжди. Відмінними характеристиками, за якими виокремлюються різні типи «реальностей», є рівень або глибина занурення у віртуальний простір, реальність відображення віртуальних об'єктів і своєрідний спосіб взаємодії з ними. Слід зазначити, що термінологічні межі розмиті, і ту ж змішану реальність іноді називають «гібридною реальністю», а є ще такі терміни як «програмувана реальність» або «віртуальна реальність з повним зануренням (immersive vr) тощо.

Віртуальна і доповнена реальність в наш час відчувають сильне взаємопроникнення на інформаційно-розважальному ринку як у сфері навчання, так і передачі інформації. Гібридна реальність є яскравим тому підтвердженням. Головною причиною такого проникнення можна вважати широкий спектр прикладних областей, в яких ці технології можуть застосовуватися [4,5].

Рональд Азума (Ronald Azuma) виділив три ознаки, якими має володіти гібридна реальність. В першу чергу це комбінування реального та віртуального світу. Так само, важлива інтерактивність і тривимірне представлення об'єктів. Існують деякі основні характеристики та вимоги, необхідні для нормального функціонування технологій гібридної реальності. Перш за все, в тому чи іншому вигляді необхідна наявність обчислювальної платформи, здатної створити умови для взаємодії з фізичним об'єктом. Іншим важливим елементом систем гібридної реальності є дисплей для відображення об'єктів доповненої реальності та при необхідності віртуальної сцени. Якщо раніше роль таких елементів виконували звичні для сприйняття монітори, то сьогодні їх роль виконують складні фізичні системи і різного роду спеціальні сенсори [3].

Наступною важливою умовою функціонування технологій, які лежать в основі гібридної реальності, є можливість інтерактивного введення, зміни існуючих віртуальних умов, можливість інтерпретації досліджуваного об'єкта

саме в тому вигляді, в якому це потрібно. При цьому система повинна бути зрозумілою для того, щоб нею в різних формах могли користуватися не тільки професіонали, але і широкий загал людей.

Технології гібридної реальності в сучасних умовах не є чимось фантастичним. Але по-справжньому бурхливий розвиток за прогнозами буде пов'язаний з появою великої кількості технологічно розвинених Hands free пристроїв. Це дозволить виконувати широкий спектр прикладних задач в різних сферах людської діяльності: навігація, складні медичні операції, складна картографія, нестандартні військові маневри. Одним з головних умов бурхливого розвитку Hand free пристроїв є технологічний розвиток цифрової галузі в цілому. Однак вважається, що до піку розвитку сучасної електроніки ще далеко. Продукти, засновані на гібридній реальності, дозволять виконувати багато стандартних завдання незвичайним способом, викликати у кінцевого споживача і користувача додатковий інтерес, залучати його в процес.

В результаті виконання наукової роботи було поставлено мету створити мобільний додаток гібридної реальності для туристичної галузі Полтави.

Спираючись на результати аналізу засобів розробки, було вирішено реалізовувати мобільний додаток із використанням фреймворка Vuforia, який імпортується до Unity 3D. Vuforia надає змогу швидко і легко оволодіти практикою і необхідними знаннями для роботи з доповненою реальністю, а Unity 3D надає змогу попрацювати із професійним і одночасно інтуїтивно зрозумілим ігровим рушієм. [1]

Розроблений таким чином додаток зможуть використовувати як гості Полтави, так і полтавці, які хочуть більше дізнатися про своє місто. Крім того додаток може використовуватися в загальноосвітніх навчальних закладах під час краєзнавчої роботи та при вивченні історії рідного краю.

Мобільний додаток передбачає встановлення на мобільний пристрій. Разом із програмним додатком у комплект входить роздрукована карта міста. При наведенні мобільного пристрою із встановленням на нього додатком на розгорнуту карту, на дисплеї з'являються тривимірні моделі пам'ятників Полтави із можливістю їх детального розгляду, отриманням довідкової інформації у супроводі екскурсії. При наявності шолому віртуальної реальності стає доступна віртуальна реальність. Користувач може буквально перенестися у місце на мапі і переглянути так, ніби він знаходиться там безпосередньо.

Програма дозволяє: відображати на екрані мобільного пристрою тривимірні моделі пам'яток Полтави; бачити їх позиціонованими на мапі, що дасть туристу можливість прокласти маршрут прогулянки містом; маніпулювати із об'єктами, обертаючи їх, збільшувати та зменшувати; переглядати супровідний інформаційний контент, в тому числі відео- та фото-матеріал; при наявності шолому віртуальної реальності переноситися віртуально у різні місця Полтави.

В ході наукового дослідження було охарактеризовано технологію

гібридної реальності, проведено аналіз засобів розробки мобільних додатків із використанням технології гібридної реальності, розроблено мобільний додаток з використанням технології гібридної реальності.

Розроблений додаток можна доповнюватися і застосовуватися у навчальному процесі. Він дає можливість користувачеві отримувати інформацію про відстежуваний об'єкт. Розроблений додаток може використовуватися в процесі екскурсійних прогулянок містом як солідними туристичними агенціями, так і окремими туристами. Крім того програма може застосовуватися в якості практичного прикладу впровадження технології гібридної реальності. В перспективі планується збільшити функціональність додатку.

Література

1. Developing with Vuforia [Електронний ресурс] URL: <https://www.slideshare.net/thanhitpro/ar-with-vuforia>
2. Hartley R. Multiple view geometry in computer vision. 2nd edition. / Hartley R., Zisserman A. // Cambridge University Press, 2004.
3. Sandor C. Immersive mixed-reality configuration of hybrid user interfaces. / C.Sandor, A. Olwal, B. Bell and S. Feiner. //In ISMAR '05, pp. 110–113, 2005.
4. Антон Чивчалов, «Через 5 лет рынок дополненной реальности увеличится более чем в 3 раза» // ARNext- все о дополненной реальности, 7 ноября 2013г.- URL: <http://arnext.ru/news/mobile-ar-2018-8799>
5. Яковлев Б. С., Пустов С. И., Классификация и перспективные направления использования технологии дополненной реальности // Известия Тульского государственного университета. Технические науки [Електронний ресурс] URL: <http://cyberleninka.ru/article/n/klassifikatsiya-i-perspektivnye-napravleniya-ispolzovaniya-tehnologii-dopolnennoy-realnosti>

Моделювання динаміки відвідування сайту компанії ТОВ Каса «Люкс-Рейзен»

Люба В.М.

Науковий керівник – к.п.н., доц. Григоруک С.С.

Хмельницький національний університет

Компанія ТОВ Каса «Lux-Reisen» - одна з наймасштабніших компаній України, що надає послуги по придбанню квитків на автобус і авіапереліт у будь-яку країну Європи та США.

Метою дослідження було знайти шляхи, за допомогою яких можна було збільшити кількість відвідувачів на сайті.

Дана тема є актуальна, тому що більшість фірм та компаній для продажу та рекламування своїх послуг використовують свій сайт, який у теперішній час, здебільшого, є одним з найпотужніших маркетингових прийомів. Наразі, чим більше користувачів відвідують сайт, тим швидше

розповсюджуватиметься інформація про діяльність компанії та зростатиме ймовірність того, що відвідувач сайту скористається послугами даної компанії, що, у свою чергу, сприятиме збільшенню її прибутку. У зв'язку з цим виникає необхідність відстежувати динаміку та структуру відвідування сайту компанії.

Завдання дослідження полягало у тому, щоб: проаналізувати динаміку відвідувань сайту компанії Каса «Люкс-Рейзен»; визначити слабкі та сильні сторони існуючого сайту; розробити практичні рекомендації щодо покращення структури та функціональності сайту компанії Каса «Люкс-Рейзен».

Показники користування сайтом відстежуються компанією як за допомогою модуля статистики, що розміщений безпосередньо на сайті, так і за допомогою системи «Google Analytics». Для нашого дослідження було взято дані з «Google Analytics» про відвідування користувачами інтернет ресурсу компанії ТОВ Каса «Люкс-Рейзен».[1]

Проведено аналіз таких показників як: відвідування, кількість користувачів, кількість переглянутих сторінок, унікальні перегляди сторінок, кількість переглянутих сторінок за одну сесію, час перебування на сторінці, показник відмов, нові відвідування, сеанси та активність користувачів. Аналіз здійснювався за період від 1 серпня 2016 по 12 жовтня 2018 р.

Побудувавши графік по даних отриманих з «Google Analytics», можемо побачити як змінювалась кількість сеансів протягом періоду з 1 серпня 2016 по 12 жовтня 2018 року. Також була побудована середня плинна з інтервалом через кожні 5 тижнів.

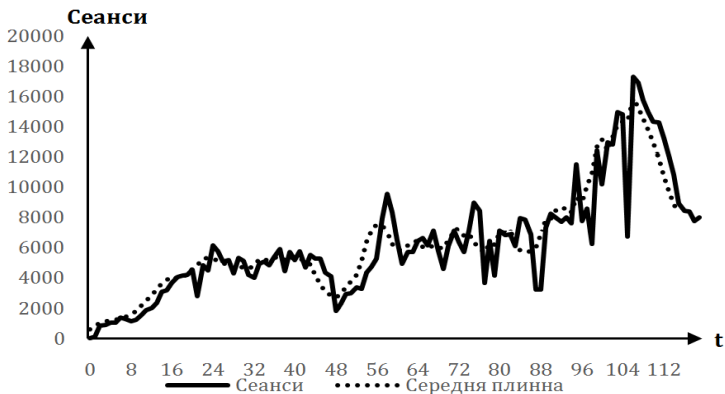


Рисунок 1 – Графік зміни сеансів

Для оцінки динаміки відвідувань сайту було обчислено: абсолютний приріст, відносний приріст, темп зростання, коефіцієнт росту, темп росту, середній рівень ряду, середній темп росту, середній темп приросту.

Таблиця 1 – Середні показники динаміки відвідування за період від створення сайту до 12 жовтня 2018 р.

	За весь час	з 12.10.2017-12.10.2018
Середній рівень ряду	6304,98319	19545,9230
Середній абсолютний приріст	67,15966387	-146,7692308
Середній темп росту	55,04890481	96,87007851
Середній темп приросту	-44,95109519	-3,129921489

Проаналізувавши дані, та обчисливши їх за часовими рядами, було виявлено що на даному етапі кількість відвідувачів на сайті є спадною. Також побудувавши графіки, можна побачити що дана тенденція з відвідувачами повторюється майже в той же самий період минулого року. Звичайно можна сказати що якщо відвідувачу потрібен квиток на автобус, він заходить і купує його, якщо не потрібен, то йому і заходити йому на сайт не має сенсу. Тому можна сказати що на відвідування сайту впливає і сезонний показник. Під час зимових свят, канікул, літнього періоду кількість відвідувачів різко збільшується. А в інший період року є більш низькою, але стабільною.

Продовжуючи попередній аналіз часового ряду в напрямі виявлення тренду, було проведено статистичну перевірку наявності його в динамічному ряду. Для цього було використано метод аналітичного вирівнювання. Під час вирівнювання використовувалась параболічна залежність. Дані були взяті за період від 12 жовтня 2017 по 12 жовтня 2018 року з місячним інтервалом.

У результаті було побудована трендова модель, вигляду:

$$y = -41,35x^4 + 1039,53x^3 - 8628,35x^2 + 28464,30x - 14370,29$$

Так як основним завданням було побудувати модель динаміки відвідування, та розробити план по привабленню користувачів, я склав ряд завдань які потрібно виконати.

1. Зробити редизайн сайту.

Якщо сайт має поганий зовнішній вигляд, шанс того що користувач заїде на нього ще раз різко зменшується до 0, він піде далі шукати той сайт, на якому йому буде приємно перебувати.

2. Створити блог.

Звичайно основною метою сайту компанії є продаж квитків, але користувачі - люди, їм буде цікаво читати остані новини про компанію, акційні знижки, створення нових маршрутів та цікавих дорожніх лайфхаків. Тому було найнято штатного копірайтера, який буде вести блог, та максимально зацікавити користувача вернутись і прочитати нову, цікаву статтю.

3. SEO-оптимізація сайту.

Потрібно оптимізувати сайт під пошукові системи, та вивести даний ресурс як мінімум в топ-10. Адже чим вище сайт знаходиться в пошуковій системі, тим більше людей його побачить та можливо скористаються його можливостями.

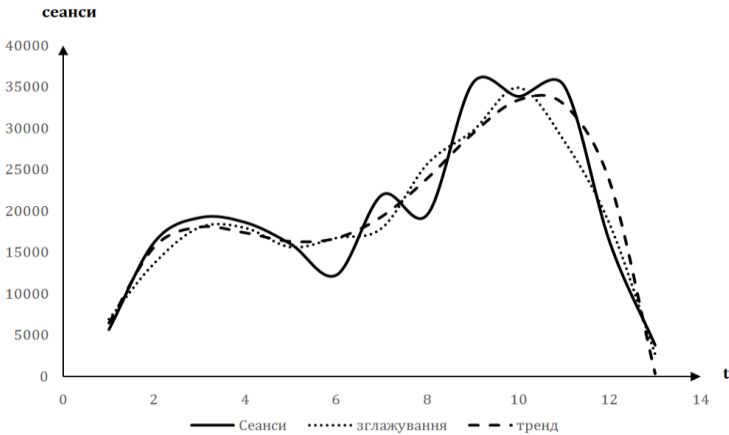


Рисунок 2 – Модель відвідування сайту

Зараз інтернет ресурс компанії знаходиться на першому етапі(редизай).

Висновки. Сайт, портал чи наукова сторінка в мережі інтернет при їх створенні та просуванні повинні бути інформативно насиченими та цікавими для користувачів. А це вимагає величезної роботи як над їх змістом та зовнішнім виглядом, так і показниками їх використання. Сучасні сайти, змагаючись за увагу до себе, пропонують інтернет-аудиторії дедалі більше інтерактивних можливостей та нових сервісів; забезпечують легкість і доступність у пошуку, обробці та використанні інформації. Означені тенденції збільшують очікування користувачів з приводу функціональності веб-сайтів та актуалізують застосування веб-аналітики, що дає змогу обирати стратегічні орієнтири розвитку науково-інформаційних порталів, контролювати ключові показники їх ефективності.

Література

1. Міжнародні автобусні перевезення Каса Люкс-Рейзен [електронний ресурс] / Веб-сайт Каса Люкс-Рейзен – Режим доступу: <https://kasa.lux-reisen.com/> – Назва з екрана.
2. Уманець Т. В. Статистика: навч. посібник/Т. В. Уманець, Ю. Б. Пігарев. - 2-ге вид. - К. : Вікар, 2003

Аналіз технологій розпізнавання облич на зображеннях для вбудованих систем

Мацьков О.П

Науковий керівник – д.т.н Бармак О.В
Хмельницький національний університет

Камери стали постійним супутниками людини в суспільстві. Вони є на вулицях, в супермаркетах та інших публічних місцях, де їх присутність пояснюється питаннями безпеки. Але це не єдині пристрої які їх використовують, наприклад усі смартфони та планшети комплектуються пристроями фіксації зображення. Якість фото цих камер значно покращились за останні роки і буде продовжувати прогресувати в майбутньому. Вони є майже скрізь, але їх зазвичай використовують лише для зйомки фотографій, відеозаписів або відеорозмов. Мета камер — фіксація важливих моментів життя. Однак їх можна використовувати для інших цілей.

Одною із таких цілей є комп'ютерний зір, який використовується для виявлення, стеження та класифікації об'єктів на цифрових зображеннях та відео. Прикладом успішного застосування такої технології є застосунки MSQRD та Snapchat. Основна функція цих додатків є додавання динамічних фільтрів на обличчя користувача після його виявлення.

Іншим прикладом використання комп'ютерного зору з яким кожного стикається пересічний користувач є сервіси Google та Facebook. Зокрема Google Photos дозволяє класифікувати фотографії осіб та типів об'єктів, що на них зображені. Facebook застосовує розпізнавання облич для кожної нової фотографії завантаженої користувачем, та автоматично пропонує відмітити особу зображену на фото.

Усі наведені вище приклади використовують комп'ютерний зір, але також вимагають машинного навчання. Дані необхідні щоб “навчити” машину розпізнавати об'єкти, більший об'єм даних дозволяє досягти кращого результату.

Сьогодні гостро постає проблема використання алгоритмів виявлення та розпізнавання облич у вбудованих системах. Необхідність в них є в середовищах, де не можна мати стабільне інтернет з'єднання через особливості розташування об'єкту або ж питань безпеки. Ще одним обмеженням може виступає розмір таких систем, тенденції сучасної електроніки дозволяють вмістити їх на плату, розмірами з кредитну карту. Однак при усіх цих умовах апаратний комплекс повинен забезпечувати безперебійне виконання задач із заданою точністю. Для цього можуть використовуватись рішення як на базі стандартизованих архітектур таких як: x86 та ARM, так і нестандартизованих як RISC-V. Комп'ютерний зір є важливою частиною дронів, як цивільних так і воєнних, систем асистування водію (ADAS) та робототехніки.

Процес розпізнавання обличчя потребує декілька зображень для

кожного об'єкта. Сам процес є комплексним і потребує значної кількості передуючих етапів обчислень.



Рисунок 1 – Процес розпізнавання обличчя

Алгоритми виявлення обличчя є важливою компонентом усього процесу, адже для розпізнавання використовується лише та частина оригінального зображення, що містить обличчя. Існує декілька інструментів, що використовуються для реалізації цієї функції.

Одним з них є ЗНМ (згорткові нейронні мережі), логіка роботи методу нагадує зорові біологічні процеси у живих організмах. Робота над методом розпочалась у 1950-х роках Девідом Г'юбелом та Торстен Візел з дослідями, що показували як візуальні кортекси тварин містять нейрони та індивідуально відповідають за рецептивні поля. Сусідні клітини мають подібні рецептивні поля, що можуть перетинатись. Розмір та розташування рецептивних полів змінюється систематично по всіх корі, щоб сформувати повне відображення зорового простору.

ЗНМ приймає 2d зображення як вхідні данні. ЗНМ складається із декількох шарів, які називають прихованими. Кожен такий шар містить декілька нейронів, кожен з яких має певну вагу та приймає вхідну інформацію. Для першого шару такою інформацією є 2d зображення, для останнього шару вихідною інформацією є перебачений клас, який є персоною на фото.

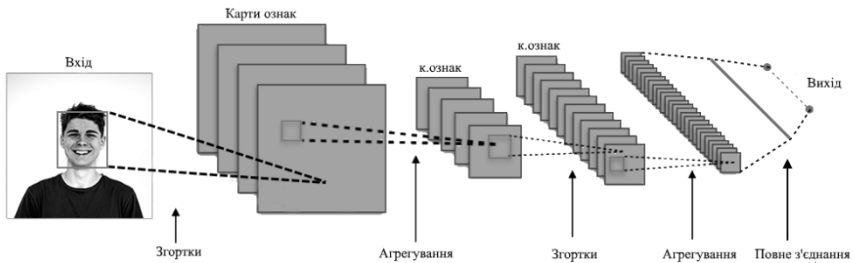


Рисунок 2 – Приклад архітектури ЗНМ мережі

Новий виток практичного застосування метод отримав у 2005 році, після публікації кількох статей, що описували більш ефективні засоби для тренування таких мереж із застосуванням графічних процесорів. Цьому сприяє паралелізм, який зумовлений архітектурою нейронних мереж. Сам алгоритм є найбільш швидким та точним у порівнянні з аналогами, але

потребує значних обчислювальних потужностей та складного налаштування тому рідко використовується у вбудованих системах.

Гістограма напрямлених градієнтів інший метод виявлення об'єктів на зображеннях який також можна використовувати для виявлення облич. Цей метод потребує чорно-білих зображень, кожен піксель представляє собою цілочислове значення градієнта. HOG метод порівнює кожен піксель із сусіднім, більшість випадків оточений вісьмома іншими пікселями, ціль знайти напрям де зображення стає темнішим.

Перевагою цього методу є відсутність залежності він від змін освітлення, якщо картинка темніша то і усі пікселі будуть також. Для виявлення обличчя зображення розбивається на квадрати 16 на 16 пікселів. Метод рахує скільки разів кожен напрямок був знайдений на попередньому кроці та залишає лише один, найбільш часто знаходжуваний. Усі ці кроки повторюються для кожного зображення в тестовому наборі даних.

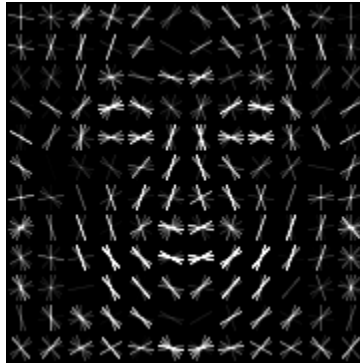


Рисунок 3 – Графічне представлення HOG методу в бібліотеці dlib

Менша кількість вхідних параметрів та краща можливість оптимізації у при роботі з даними дозволяють використовувати цей метод у вбудованих системах, за умови наявності в них достатніх ресурсів, та наявності спеціалістів здатних оптимізувати метод під конкретну задачу.

Метод Віоли-Джонса представлений в 2001 році, і став одразу став важливим інструментом в задачі розпізнавання облич.

Даний алгоритм використовує технологію динамічного вікна, тобто рамка розміром менша за оригінальне зображення, рухається з деяким кроком по зображенню за допомогою слабких класифікаторів та визначає чи є у розглянутому вікні обличчя.

Метод складається із двох алгоритмів: алгоритм навчання та алгоритм розпізнавання. На першому етапі на тестовому наборі даних проходить визначення ознак, паритетів та границь. Після завершення навчання створюється база даних алгоритму, що містить у собі слабкі класифікатори.

Алгоритм розпізнання сканує зображення в декількох масштабах починаючи із базової шкали. Вона зазвичай передбачає розмір вікна 24 на 24 пікселя, та 11 масштабів який збільшується на 1.25 при кожній наступній ітерації.

Даний алгоритм є найменш ресурсозатратним з усіх представлених, тому часто використовується у вбудованих системах. До його недоліків можна віднести довгий час навчання моделі для отримання довільних результатів, та підвищені вимоги до вхідних даних. Зокрема рекомендовано, щоб обличчя були прямо направлені на об'єкти камери.

Усі з представлених алгоритмів широко використовуються для вирішення задач з виявлення та розпізнавання облич. За допомогою комбінації методів та гнучкому налаштуванню уже сьогодні можливий рівень точності, що дозволяє запровадження комплексних систем, що використовуються як в комерційних цілях так і соціальних сферах з використання звичайних та вбудованих АІС.

Література

1. A comparison of facial recognition's algorithms – Nicolas Delbiaggio, 2017 – 45 с.
2. Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision – Amr Suleiman, Yu-Hsin Chen, Joel Emer, Vivienne Sze, 2017 – 4 с.
3. Performance Comparison of Face Recognition Algorithm Based on Accuracy Rate – Rashmi Ravat , Namrata Dhand, 2015 – 326 с.
4. Computer Vision in C++ with the OpenCV Library – Adrian Kaehler & Gary Bradski, 2017 – 967 с.

Доцільність використання акторної моделі при побудові високонавантажених сервісів обробки замовлень

Пишний М.В.

Науковий керівник – к.т.н.,доц. Яшина О.М.

Хмельницький національний університет

Розвиток інформаційних технологій та ринкових відносин сприяв появі нових видів бізнесу, одним і з таких є Інтернет-торгівля. За останнє десятиріччя значно зросла кількість користувачів Інтернету, що в свою чергу спричинило зростання розвитку інфраструктури Інтернет-торгівлі [1].

Розвиток Інтернет-торгівлі призвів до збільшення кількості сервісів обробки замовлень. Основне завдання таких сервісів полягає у швидкій та коректній обробці замовлень. Таким чином досить актуальними є питання розробки та розширення можливостей таких сервісів. Адже стрімке збільшення користувачів призводить до збоїв у роботі сервісу.

На сьогоднішній для вирішення проблеми високих навантажень застосовують мікросервісну архітектуру [2], але питання застосування даної

архітектури залежить від складності та швидкодії розробки програмного продукту [3]. Тому більшість сучасних сервісів застосовують багатопоточну архітектуру, це зумовлено швидкістю розробки.

Однією з найбільш використовуваних різновидів даної архітектури є класична тривінева система, яка передбачає поділ додатка на три рівні:

1. Представлення (Presentation Logic);
2. Бізнес-логіка (Business Logic);
3. Доступ до даних (Database Management).

Дана архітектура має ряд переваг при побудові веб-сервісів, адже дозволяє розділити всю інфраструктуру сервісу на логічні слабо зв'язні шари.

Серед недоліків даної архітектури варто відзначити проблему масштабування [4].

Аналіз літературних джерел дозволив встановити, що більшість проблем, пов'язаних з масштабуванням виникають при проектуванні бізнес шару. Адже зазвичай використовується об'єктно орієнтована концепція сервісів, яка інкапсулює бізнес логіку та не дозволяє безпосередньо змінювати внутрішній стан сервісу.

Проблема об'єктно орієнтованої концепції полягає в її несумісності роботи з асинхронними викликами (рис. 1).

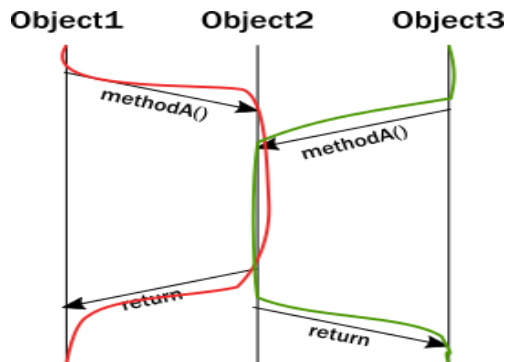


Рисунок 1 – Виклик асинхронного методу

Виклики такого виду призводять до пошкодження стану сервісу. Загальний підхід до вирішення цієї проблеми полягає в додаванні блокувань всередині методів. Хоча блокування гарантують, що не більше одного потоку буде викликати метод в будь-який момент часу, та це призводить до ряду проблем:

1. Блокування в значній мірі обмежують паралелізм, та є дуже затратними для сучасних архітектур процесорів;
2. Збільшення кількості блокувань призводить до появи взаємоблокувань (deadlock);
3. Зменшення кількості потоків, які можуть виконувати корисну роботу.

В сумі дані проблеми призводять до безвихідної ситуації.

Без достатньої кількості блокувань стан сервісу стає пошкодженим. З великою кількістю блокувань, падає рівень продуктивності та збільшується ризик появи взаємоблокувань.

Для вирішення вищенаведених проблем можна застосувати модель актора, яка була винайдена в 1973 році Карлом Х'юїттом [5] - актори представляють собою «фундаментальні одиниці обчислень, що реалізують обробку, зберігання і комунікацію».

Тобто замість виклику методів, учасники відправляють повідомлення один одному. Відправлення повідомлення не переносить потік виконання від відправника до місця призначення. Актор може відправити повідомлення і продовжити роботу без блокування.

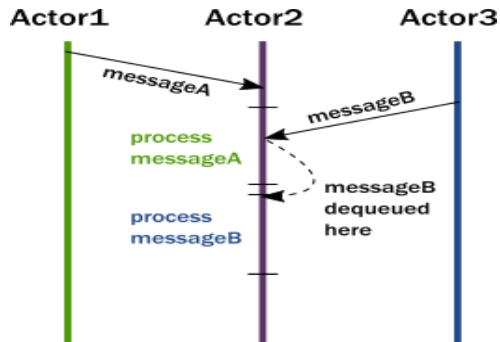


Рисунок 2 – Відправка повідомлення актору

Друге ключова відмінність, інкапсуляція. Актори реагують на повідомлення так само, як об'єкти «реагують» на використовувані ним методи. Різниця полягає в тому, що замість того, щоб кілька потоків «викликали» актора і руйнували його внутрішній стан, актор обробляє повідомлення, відправлені йому послідовно, інші актори працюють одночасно один з одним, тому система акторів може обробляти стільки повідомлень одночасно, скільки процесорних ядер є на машині.

Актор обробляє повідомлення наступним чином:

1. Актор додає повідомлення в кінець черги.
2. Якщо актор не був запланований для виконання, він позначається як готовий до виконання.
3. Прихований об'єкт планувальника приймає актора і починає його виконувати.
4. Актор вибирає повідомлення з передньої частини черги.
5. Актор змінює внутрішній стан, відправляє повідомлення іншим учасникам.
6. Актор помічається, як незапланований для виконання.

Пиклад обробки повідомлень актором (рис. 3).

Модель актора складається з:

1. Поштовий ящик (черга, де закінчуються повідомлення).
2. Поведінка (стан актора, внутрішні змінні тощо).
3. Повідомлення (фрагменти даних, що представляють сигнал, подібні до викликів методу та їх параметри).
4. Навколишнє середовище виконання (фізична машина, яка приймає дійових осіб, які мають повідомлення, щоб реагувати на коди обробки повідомлень та викликати їх).
5. Адреса актора.

Отже, проаналізувавши ряд наукових робіт, можна зробити висновок, що використання акторної моделі при побудові високонавантажених сервісів є доцільним адже, дана модель вирішує проблеми, перераховані раніше:

1. Інкапсуляція зберігається шляхом відокремлення виконання від сигналізації (спосіб передачі викликів здійснює передачу, передача повідомлень не відбувається).
2. Немає необхідності в блокуванні. Модифікація внутрішнього стану актора можлива лише за допомогою повідомлень, які обробляються по одному, усунувши гонки при спробі зберегти інваріанти.
3. Немає ніяких замків, а відправники не заблоковані. Мільйони акторів можуть бути ефективно заплановані на десяток потоків, що досягають повного потенціалу сучасних процесорів. Завдання делегування є природним способом роботи для акторів.

Разом з тим, можна зробити висновок про те, що акторна модель підходить для масштабування адже стан акторів є локальним, а не загальним, зміни та дані поширюються через повідомлення. Тобто актори можуть працювати в межах однієї машини або в скупченні сотень машин.

Література

1. Ладика Л.В. Особливості розвитку інтернет-торгівлі в Україні / Л.В. Ладика // Міжнародний науковий журнал “Інтернаука” – 2016 -№ 6. DOI 10.21267/IN.2016.6.2294.
2. Pattern: Microservice Architecture [Електронний ресурс] - Режим доступу: <https://microservices.io/patterns/microservices.html>. - Дата доступу: 02.09.2018.
3. MicroservicePremium [Електронний ресурс] - Режим доступу: <https://martinfowler.com/bliki/MicroservicePremium.html>. Дата доступу: 02.09.2018.
4. N-tier Architecture Improves Scalability, Availability and Ease of Integration [Електронний ресурс] - Режим доступу: <http://cadeiras.iscte.pt/CDSI/fich/N-tier%20Architectures-Intel.pdf>. Дата доступу: 09.09.2018.
5. Carl Hewitt. Actor Model of Computation: Scalable Robust Information Systems [Електронний ресурс] - Режим доступу: <https://pdfs.semanticscholar.org/7626/93415b205b075639fad6670b16e9f72d14cb.pdf>. - Дата доступу: 29.08.2018.

Інтелектуальне мультиагентне моделювання епідемічного процесу іксодового кліщового бореліозу

Пілецький П. Є.

Науковий керівник – к.т.н. Чумаченко Д. І.

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Іксодові кліщові бореліози (ІКБ) – група інфекційних трансмісивних природних осередків захворювань, що викликаються бореліями групи *B. burgdorferi* та передаються іксодовими кліщами. Клінічно захворювання протікає з переважним ураженням шкіри, нервової системи, опорно-рухового апарату, м'яз серця і характеризується схильністю до хронічного, а також летального перебігу.

В Україні, за даними Львівського НДІ епідеміології та гігієни, різноманітність нозологічних форм хвороби Лайма пов'язана з циркуляцією всіх трьох патогенних для людини видів борелій *B. burgdorferi*: *S. stricto*, *B. garinii*, *B. afzeli*. У найбільш активних природних вогнищах хвороби Лайма в Україні інфікованість бореліями кліщів становить 10-70%. Епідеміологічні та серологічні дослідження, проведені даними інститутом, показали, що від 10% до 42,2% населення України мають контакт із збудником бореліозу. Це свідчить про активність епідемічного процесу хвороби Лайма.

В рамках дослідження проведено аналіз захворюваності на ІКБ в Харківській області. За аналізований період (з 2000 по 2015 рр.) інфікованість всіх видів кліщів бореліями зросла з 4,4% до 23,9%. У 2014 році зменшилась інфікованість до 21,4% і чисельність кліщів (1,3 на 1 км маршруту проти 1,9 в 2013 році і 1,93 у 2015 році).

За всі роки проведеного епідеміологічного аналізу захворюваність мала виражену сезонність з максимумом в травні-липні (46,9-58,5%). Присмокування кліща найчастіше відбувалися в травні-липні (65,8%) і вересні-жовтні (20,1%), що відповідає періоду найбільшої активності кліщів.

У 2014 - 2015 роках зросла кількість осіб, що заразилися в антропоургічних осередках і склала 85,1% і 73,1%, в той час як в попередні роки їх число було трохи менше 50%. У два рази збільшилася питома вага осіб, інфікованих на дачах і присадибних ділянках – 36,0% проти 14,9%. Таким чином, слід зазначити, що ІКБ є актуальною інфекцією для Харківської області зі значним медичним, соціальним і економічним збитком.

В охороні здоров'я і епідеміології моделювання застосовується для кількісної оцінки гідності різних методів боротьби і профілактики, таких як ізоляційно-обмежувальні заходи, проведення вакцинації і вибір контингентів для імунізації, виявлення груп ризику захворюваності та ін., це необхідно для того, щоб органи охорони здоров'я могли прийняти найбільш раціональні та дієві заходи в боротьбі з інфекціями. Тільки правильно сформульовані математичні моделі дозволяють підійти до суворого вивчення всіх аспектів

цієї проблеми, незалежно від того, чи йде мова про епідеміологічні діагностики, оцінки ефективності існуючих профілактичних і протиепідемічних заходів або ж про заходи, що плануються органами охорони здоров'я та санітарно-епідеміологічної служби.

Найбільший внесок у розвиток моделювання епідемічного процесу в останні роки дали популяційні моделі. Популяційні моделі – це дискретно-подійові моделі, в яких все модельовані індивіди явно розділені на соціальні групи, які формуються з урахуванням віку індивідів, в детальних моделях може враховуватися рід занять індивіда. Поширення інфекції між індивідами може статися тільки в рамках однієї «контактної» групи. Кожен день в моделі індивіди в залежності від своєї соціальної групи формують певні контактні групи, в яких від хворого індивіда захворювання може бути передано здоровому. Контактні групи визначаються характерною структурою суспільства, яка буде залежати від модельованої території.

Мультиагентні технології пов'язані з імітацією взаємодії інтелектуальних агентів - активних елементів динамічних систем будь-якої фізичної, біологічної та соціальної природи. Поведінка і зміна стану таких систем є результатом покрокової взаємодії безлічі її активних елементів, характером відносин і зв'язків між ними, умовами досягнення локальних та глобальних цілей і т. п. У цьому випадку формалізація і моделювання процесів поведінки і взаємодії агентів дозволяють імітувати і прогнозувати виникнення якісно нових станів системи і оцінити можливості досягнення мети при різних варіантах, а також обґрунтувати прийняття рішень в складних ситуаціях ризику, невизначеності та конфлікту.

Було розроблено інтелектуальну мультиагентну модель захворюваності на ІКБ. Агент в моделі представлений як набір властивостей:

$$a = \langle s, s_t, c, t_a, l \rangle, a \in A, s \in S, c \in C, t_a \in T_a,$$

де s_t – час перебування в стані, A – множина всіх агентів, S – множина різних станів агентів, C – множина комірок робочої області, T_a – множина типів, до яких можуть належати агенти, l – тривалість життя.

Множина станів агента визначена попередньо і є постійною. Залежно від досліджуваного процесу множина може доповнюватися різними станами, початкова множина являє собою:

$$S = \{Susceptible, Exposed, Infected, Convalescent, Recovered, Dead\}.$$

Використання такого набору станів ґрунтується на ідеї розподілу всієї популяції на підмножини, виходячи з їхніх станів за епідемічними ознаками.

– Susceptible – агент здоровий (може бути схильним до зараження). В даному випадку під здоровим мається на увазі агент, який є сприйнятливим

до захворювання на інфекцію, епідемічний процес якої моделюється.

– Exposed – агент зазнав захворювання. Даний стан являє собою інкубаційний період. Протягом цього часу агент ще не є хворим, але вже має шанси передавати інфекцію.

– Infected – агент хворий. Агенти в цьому стані є найбільш ймовірними джерелами інфекції для інших агентів.

– Convalescent – агент одужує. Даний стан характеризує період, коли зникли клінічні симптоми захворювання, але агент, як і раніше, може бути носієм захворювання і джерелом інфекції. Наявність такого стану характерна для певних інфекційних захворювань.

– Recovered – агент видужав (придбав імунітет). Агенти в цьому стані більш не схильні до захворювання.

– Dead – агент мертвий від старості або внаслідок захворювання.

Порівняльний аналіз точності прогнозування методом ковзних середніх до 3 років і агентного моделювання показав, що останній описує існуючі дані краще, отже, прогноз буде виконано точніше.

Для автоматизації прогнозування захворюваності ІКБ на мові C# розроблений програмний комплекс, що дозволяє обчислити прогнозу захворюваність на підставі існуючих статистичних даних в режимі реального часу. У моїй моделі налаштування програмного комплексу включені дані за період 2006 – 2017 роки. Дані по роках включають інтенсивні показники захворюваності на 100 тис. населення, зверненнями населення з приводу укусів кліщів, результати досліджень кліщів, знятих з людей, на наявність борелій і питома вага інфікованих бореліями кліщів, зібраних на прапор, показники чисельності кліщів.

Проведена порівняльна оцінка точності прогнозування динаміки епідемічного процесу з використанням методу ковзних середніх і мультиагентного підходу. Більш висока точність відзначається при використанні мультиагентного імітаційного моделювання, на підставі якого розроблено програмний комплекс, що дозволяє обчислити очікуваний рівень захворюваності ІКБ. Адекватність перевіреного прогнозу перевірена на реальних статистичних даних щодо захворюваності на ІКБ в Харківській області з 2006 по 2017 рік.

Отриманий прогноз виявив очікуване збереження нестійкої епідемічної ситуації щодо ІКБ, що диктує необхідність розробки комплексу профілактичних заходів, спрямованих на зниження захворюваності людей ІКБ. Віртуальна перевірка ефективності таких заходів буде наступним етапом нашого дослідження.

Програмне забезпечення оптимізації роботи стоматологічного кабінету шляхом розв'язку задач лінійного програмування неграфічним методом

Подкопний Д.В.

Науковий керівник – к.т.н., доц. Гурман І.В.

Хмельницький національний університет

Програмні засоби автоматизації різних видів діяльності вже давно тісно увійшли у більшість сфер життя. Сучасний етап функціонування людства характеризується постійним ростом кількісних та якісних характеристик інформаційних потоків. Дана проблема не є новою і для більшості фахівців – це прогнозований поштовх для вдосконалення наявних, і розробки нових методів та засобів обробки інформації. В умовах розвитку та реформування різних галузей нашої держави, крім забезпечення можливої автоматизації різного функціонування також постає проблема її якості та доцільності. Зокрема, розглядаючи сучасний стан програмного забезпечення для медичної сфери, можна простежити велику кількість спеціалізованих програмних засобів для оптимізації роботи ресстратури, проведення діагностики захворювань та низки програм спрямованих на підвищення кваліфікації медичного персоналу. Проте, також можна простежити низьку якість консолідації інформаційних даних між наявними програмними продуктами, а також, в деяких випадках, відсутність доцільності їхнього застосування в окремих медичних закладах, таких як невеликі стоматологічні кабінети та клініки тощо.

Таким чином постає проблема наявності розроблених методів та готових рішень для підвищення ефективності функціонування невеликих медичних закладів вузького спрямування, а також їх відповідність потребам даних установ.

Незважаючи на те, що сучасний стан розвитку приватної медицини та реформування державних медичних закладів перебуває на одному із початкових етапів у порівнянні з іншими розвинутими країнами, вже сьогодні, можна простежити хороший рівень серед таких напрямків медицини, як діагностика, косметологія та стоматологія. Це викликано тим, що дані напрями рідко потребують стаціонарного лікування та великих приміщень для розміщення обладнання і персоналу, а також невеликої кількості документації для забезпечення правомірності надання медичних послуг для населення.

Оскільки, діагностування захворювань є лише початковим етапом лікування пацієнтів, використання програмних засобів такими закладами відбувається тільки при застосуванні спеціалізованої апаратури для проведення аналізів і консолідування даних за допомогою нескладних засобів обліку – таких як електронні таблиці. При розгляді закладів та окремих підрозділів косметичної медицини простежується більший рівень застосування програмних засобів, оскільки крім обліку пацієнтів та медичних

матеріалів тут застосовуються засоби комп'ютерного моделювання, проте, це відноситься до дуже вузького кола таких закладів, які поряд із косметичними послугами, надають послуги пластичної хірургії, що в свою чергу – виводить дані заклади за межі досліджуваного сектору. Таким чином, найбільш актуальними для проведення даного дослідження постає сфера надання стоматологічних послуг. Зокрема слід виділити її деяку, умовну незалежність від загальної медицини і водночас прямий зв'язок з іншими медичними напрямками вузького спрямування на основі:

- анатомічної близькості та прямої залежності в людському організмі окремих органів і систем;
- можливості шаблонного копіювання методів та засобів здійснення організації надання медичних послуг (облік пацієнтів, працівників, матеріальних засобів, медичних маніпуляцій та процедур, планування роботи, підвищення кваліфікації персоналу).

Дослідженнями в області застосування можливостей використання інформаційних технологій для автоматизації обліку на підприємствах різних сфер діяльності займалися ряд зарубіжних та вітчизняних науковців. Серед них –К. Леонард [1], М. Гасемі [2], П. С. Клімушин[3], О. М. Томашевський [4] та інші. Дослідження в області застосування автоматизації обліку в медичних закладах, які надають стоматологічні послуги також описані в ряді наукових публікацій [5,6,7].

На основі аналізу ряду джерел отримано висновок, що розробка програмного забезпечення для медичних закладів вузького спрямування залишається актуальною, оскільки переважна більшість із реалізованих або розроблених програмних систем має ряд недоліків, які можуть частково або суттєво нівелювати їх застосування на практиці. Дана актуальність викликана наступними узагальненими факторами:

- надлишковий функціонал;
- невідповідність наявних форм для введення та відображення даних сучасним вимогам обліку у вітчизняних медичних закладах;
- відсутність інструментів, які б оптимізували роботу медичного персоналу не тільки на рівні обліку та звітності.

Таким чином, об'єктом даного дослідження є організація діяльності невеликого медичного закладу по наданню стоматологічних послуг.

Предметом даного дослідження є застосування математичних методів та програмних рішень для оптимізації роботи невеликого медичного закладу вузького спрямування.

Мета дослідження – розробка індивідуального програмного застосування для стоматологічної клініки із врахуванням специфіки автоматизації типових облікових функцій закладу сфери надання стоматологічних послуг, а також можливості проведення автоматичних розрахунків на основі існуючих даних, для оптимізації професійної діяльності персоналу.

Досягнення поставленої мети містить два напрямки, які полягають у виборі структури організації даних з їх подальшою програмною реалізацією, а також автоматизації деяких функцій медичного персоналу шляхом застосування існуючих математичних методів оптимізації.

Реалізація даних напрямів полягає у вирішенні наступних завдань:

- дослідження вже існуючих програмних систем, присутніх в об'єкті дослідження, або схожих чи споріднених галузях;
- дослідження облікових документів, які фігурують у роботі медичного закладу, який надає стоматологічні послуги;
- дослідження задач практичної діяльності медичного та допоміжного персоналу, вирішення яких можна оптимізувати за рахунок використання програмних засобів;
- дослідження можливостей зв'язку та консолідації даних з іншими подібними системами;
- розробка алгоритму застосування математичних методів при побудові програмної системи;
- розробка архітектури та детального проекту програмного засобу;
- вибір шаблонів та каркасів для успішної та ефективної реалізації детального проекту;
- тестування та апробація створеного програмного продукту;
- надання інструкцій та рекомендацій по використанню та можливому подальшому вдосконаленню програмної системи.

Для реалізації поставлених завдань було проведено аналіз ряду сучасних інформаційних систем, які використовують у медицині та системі охорони здоров'я. В окрему групу було виділено спеціалізовані програмні продукти для автоматизації діяльності стоматологічних клінік оскільки, вони максимально наближені до інтересів дослідження. В процесі аналізу було визначено критерії оцінки даних систем, зокрема:

- наявність реалізації медико-технологічної складової;
- наявність реалізації організаційної складової;
- наявність реалізації адміністративної складової;
- відповідність організації даних сучасному стану документообігу в медичній сфері;
- кросплатформенність або можливість використання на веб-платформі;
- можливість нарощення функціоналу;
- вартість.

На основі обраних критеріїв було складено порівняльну таблицю, з визначенням слабких та сильних сторін кожної з розглянутих інформаційних систем. Зокрема до сильних сторін було віднесено:

- повна, або майже повна автоматизація обліку діяльності;
- наявність ряду вбудованих довідників та інструкцій для професійної діяльності медичного персоналу;
- наявність інструментів для нарощення функціоналу та масштабованості програмного продукту.

До слабких сторін можна віднести:

- прив'язка до однієї платформи;
- неповна відповідність документації сучасним критеріям обліку в медичних закладах, що надають стоматологічні послуги;
- автоматизація обліку діяльності медичного закладу здійснена з повним або частковим ігноруванням оптимізації його роботи в сфері розрахунків витрат робочого часу персоналу а також матеріальних засобів задіяних в процесі діяльності закладу.

Отриманні висновки стали основою для проектування та програмної реалізації комбінованої системи медичного документообігу з елементами системи прийняття рішення на веб-платформі із застосуванням вільноросповсюджуваних програмних засобів та технологій.

Вибір платформи зумовлений її універсальністю для маніпуляцій інформаційними потоками через будь-який підключений до мережі пристрій, здатний відображати веб-сторінки. Вибір вільноросповсюджуваних засобів розробки зумовлений мінімізацією витрат на створення та подальший супровід програмного продукту.

Побудова архітектури дозволить забезпечення функціонування програмної системи в режимі відсутності підключення до мережі, з тимчасовим обмеженням використання деяких інструментів.

Повна або часткова автоматизація ряду процесів інформаційної системи, зокрема вибір пломбувальних матеріалів та їх підбір, з подальшим формуванням плану закупок здійснено за рахунок використання методів оптимізації. Зокрема, ряд рішень по визначенню оптимальної кількості витрачених матеріалів на стоматологічні процедури виходячи з питань забезпечення належної якості надання медичних послуг при якомога менших витратах матеріальних, часових та людських ресурсів.

Вирішення задачі розпочато із знаходження екстремуму. Для отримання математичної моделі, виконано чотири етапи моделювання. Визначено межі системи оптимізації. Відкинуто ті зв'язки об'єкта оптимізації із наявними в системі даними, які не можуть сильно вплинути на результат оптимізації, а, точніше, ті, без яких розв'язок спрощується. Обрано змінні проектування, отримані шляхом побудови запитів до бази даних. Визначено ролі змінних, а саме поділ на керовані та некеровані змінні. Визначено значення з області допустимих рішень (керовані змінні). Визначено обмеження на керовані змінні за допомогою рівностей та нерівностей отриманих із схеми застосування матеріалів під час надання різних стоматологічних послуг. Обрано числовий критерій оптимізації. Створено цільову функцію. Побудовано ряд методів розв'язку отриманої моделі та створено форми для виведення результатів у зрозумілому для сприйняття вигляді.

Для розмежування доступу користувачів до системи організовано видимість по ролях, для кожної з яких визначено наступні параметри:

- доступність адміністративних функцій різного рівня;

- доступність окремих підсистем прийняття рішень;
- режим доступу до окремих довідників;
- режим доступу до окремих документів.

Таким чином, запропоноване застосування може бути використано в типовому медичному закладі стоматологічного профілю в якості:

- програмного забезпечення автоматизованого робочого місця працівника будь-якого структурного підрозділу;
- комутативної інформаційної платформи для персоналу та клієнтів.

Цілями подальшого вдосконалення розробленого продукту можуть стати розширення адміністративної частини для нарощення функціоналу рядовими працівниками клініки без залучення допомоги ІТ-спеціалістів, а також впровадження нових математичних і програмних рішень.

Література

1. Kimberlee Leonard. "How Is Information Technology Used in Accounting?" Small Business – Chron.com [Електронний ресурс]./ L. Kimberlee. - Chron.com, 28 June 2018 – Режим доступу: <http://smallbusiness.chron.com/information-technology-used-accounting-2101.html> (дата звернення 22.10.2018).

2. Maziyar Ghasemi. The impact of Information Technology (IT) on modern accounting systems./ M. Ghasemi, V. Shafeiepour, M. Aslani, E. Barvayeh – Procedia – Social and Behavioral Sciences 28 (2011). 112 p.

3. Клімушин П. С. Інформаційні системи та технології в економіці : навч. посібн. / П. С. Клімушин, О. В. Орлов, А. О. Серенок. — Х. : ХарPI НАДУ «Магістр», 2011. — 448 с.

4. Томашевський О. М. Інформаційні технології та моделювання бізнес-процесів : навч. посібн. / О. М. Томашевський, Г. Г. Цегелик, М. Б. Вітер, В. І. Дудук. – К. : Центр учбової літератури, 2012. — 296 с.

5. О. Я. Ковальчук. Розробка спеціальних прикладних рішень для автоматизації діяльності стоматологічної клініки засобами сучасних інформаційних технологій [Електронний ресурс]./ О. Я. Ковальчук, В. М. Періг, О. В. Єрьомін. – Ефективна економіка № 3, 2014 – Режим доступу: <http://www.economy.nayka.com.ua/?op=1&z=2858> (дата звернення 21.09.2018).

6. Burke Lillian. Information Technology for the Health Professions (5th Edition) / Lillian Burke, Barbara Weill. – Pearson, 2018. - 452 p.

7. Mattheos N. Potential of information technology in dental education [Електронний ресурс] / N. Mattheos, N. Stefanovic, P. Apse, R. Attstrom, J. Buchanan, P. Brown, A. Camilleri, R. Care, E. Fabrikant, S. Gundersen, S. Honkala, L. Johnson and al – Volume12, Issues1. Special Issue: Global Congress on Dental Education III. - February, 2008. Pages 85-92 – Режим доступу: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0579.2007.00483>. (дата звернення 21.10.2018).

8. Ю. Д. Попов. Методи оптимізації. Навчальний електронний посібник для студентів спеціальностей “Прикладна математика”, “Інформатика”, “Соціальна інформатика”. / Попов Ю.Д., Тюття В.І., Шевченко В.І. – Київ: Електронне видання. Ел. бібліотека факультету кібернетики Київського НУ ім. Тараса Шевченка, 2003.–215 с.

Чисельний підхід до знаходження наближеного розв'язку рівняння збурень Кортевега-де Вріса-Бургерса для задачі динаміки рідинного автобалансира

Сапужак О.М.

Науковий керівник – к.т.н., доц. Драч І.В.
Хмельницький національний університет

Рідинне автобалансування використовується в роторних машинах для усунення небажаних ефектів від маси дисбалансу, що постійно змінюється в часі і просторі через технічні умови роботи роторної системи. Рідинне автобалансування стало стандартною опцією для більшої частини сучасних побутових приладів, але також використовується і у важких промислових роторних машинах. Наприклад в пральних машинах моделей Samsung, LG, Bosch, Ardo [5]. Слід окремо відмітити модель пральної машини Samsung із серії EcoSilver – H1245A [6], яка під час прання автоматично вибирає нахил барабана, що в результаті запобігає дрижанням і вібрації самої машини.

Припустимо ситуацію, коли маса дисбалансу m присутня, для прикладу коли через неоднорідний розподіл білизни в баку пральної машини. Ротор має критичну кутову швидкість Ω_{cr} , яка провокується відцентровими силами незбалансованої маси і прогином ротора (переміщенням ротора), що визначається жорсткістю пружинних кріплень. Нижче цієї швидкості ($\Omega < \Omega_{cr}$) де центр мас рідини буде знаходитись «на тій самій стороні» що і маса дисбалансу, як показано на рис. 1а (тут M вказує масу порожнього ротора і M – масу рідини, що міститься усередині камери). На певній кутовій швидкості після критичної ($\Omega > \Omega_{cr}$) (припустимо, під час процесу сушіння) центр мас рідини переміститься на «протилежний бік» від маси дисбалансу, як показано на рис. 1б, що в результаті дає «балансування системи» і таким чином зменшення відцентрових сил і зменшення амплітуд коливання ротора.

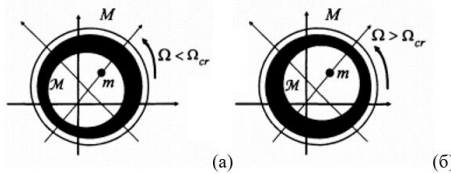


Рисунок 1 – Робочий принцип рідинного автобалансируючого пристрою (АБП)

Такий принцип роботи рідинного АБП, що появився ще в 1912, і запатентований в 1916 [2]. Початковий макет складався з одного або кількох дуже вузьких концентричних каналів частково наповнені «рідиною або дуже малими сталевими кульками, або металевим наповнювачем».

Динаміка АБП була досліджена на базі моделі ротора з двома ступенями свободи, що містить невелику кількість рідини. При обертанні

утворюється тонкий внутрішній шар рідини, який описується через теорію мілкої водяної хвилі. Товщина збуреного шару рідини описується через рівняння збурень Кортевега-де Вріса-Бургерса.

Відхилення компонентів посудини x_* , y_* і сил компонентів рідини F_x , F_y показані на рис. 2а, 2б. Тут Ω_* – безмірна швидкість обертання посудини, визначена як $\Omega_* = \Omega/\omega_s$, де ω_s – критична обертальна швидкість для пустого ротора.

На рис. 2а, 2б показано кут фазового відхилення посудини φ_d і результуючої сили рідини φ_f відповідно. Фазове кутове відхилення за невеликими обертальними швидкостями починається коло $\varphi_d \approx 0$, тобто відхилення в напрямку маси дисбалансу. Після проходження резонансу фазовий кут зсувається приблизно на 180° . Фазовий кут результуючої сили рідини (показаною цілою лінією) має подібний напрямок і закінчення, після проходження через резонанс, при значенні $\varphi_d = -190^\circ$.

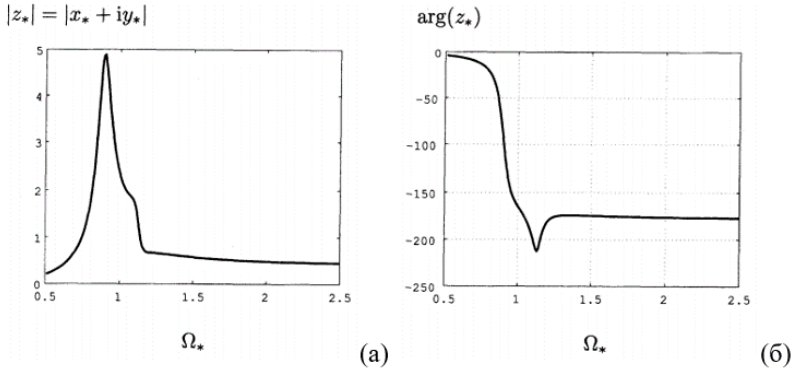


Рисунок 2 – Відхилення посудини (а) і фазовий кут (б) в градусах як функції кутової швидкості Ω_*

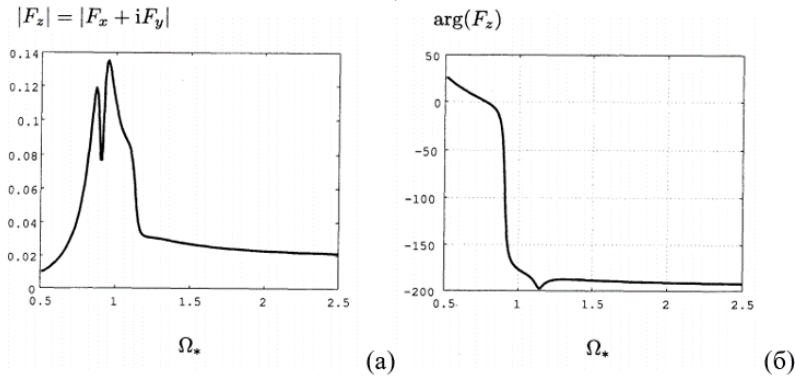


Рисунок 3 – Амплітуди сили рідини (а) і фазовий кут в градусах (б) як функції кутової швидкості Ω_*

Тому, після проходження резонансу, результуюча сила рідини починає діяти проти маси дисбалансу і призводить до генерації відхилення, яке виступає проти відхилення, утвореного масою дисбалансу. Це пояснює базову динаміку АБП.

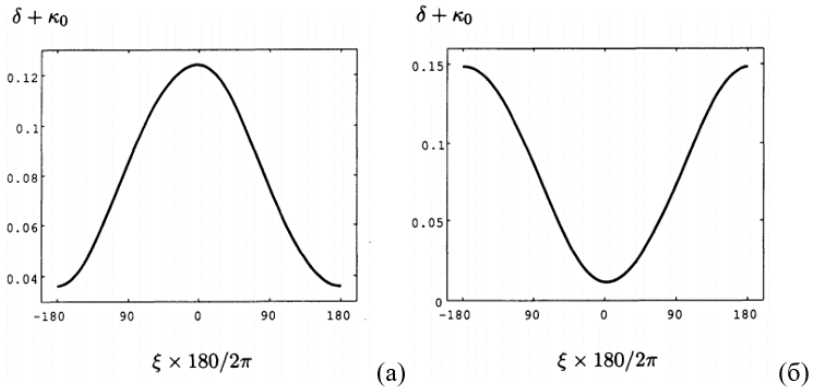


Рисунок 4: Шар рідини в посудині, описаної через $\delta + \kappa_0$, в прямокутних координатах: а – шар рідини перед резонансом ($\Omega_*=0.6$), б – шар рідини після резонансу ($\Omega_*=1.6$)

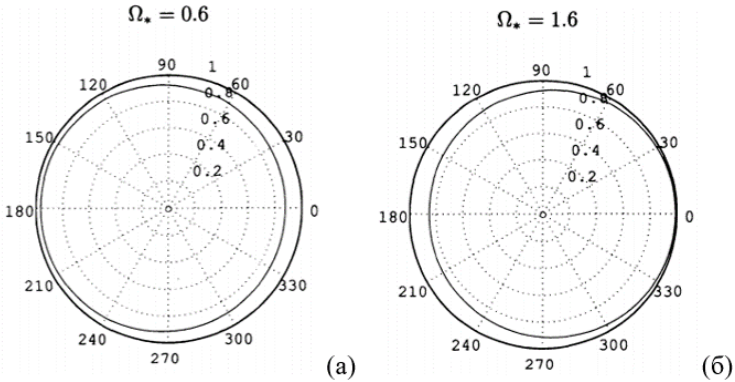


Рисунок 5 – Шар рідини, описаної в полярних координатах: а – шар рідини перед резонансом ($\Omega_*=0.6$), б – шар рідини після резонансу ($\Omega_*=1.6$)

На рис. 4 показано поверхню рідини, описаної безмірними параметрами $\delta + \kappa_0$ в квадратних координатах перед критичним значенням Ω_* ($\Omega_*=0.6$) і після критичного значення ($\Omega_*=1.6$). На рис. 5 показано поверхню рідини перед критичним значенням Ω_* ($\Omega_*=0.6$) та після критичного значення ($\Omega_*=1.6$). Рис. 4 дає розгорнуте представлення в прямокутних координатах, а рис. 5 – більш фізичне представлення в полярних

координатах.

Слід відзначити, що нефізичне рішення може утворитись при $\Omega_* \approx 1$, тому що $\delta + \kappa_0(\zeta)$ (яке має бути більше 0 для $\forall \zeta$) може стати менше 0 при певних значеннях ζ . Проблема була повідомлена і обговорена [1] і [4]. З метою уникнення цього, обмеження на вираз $\delta + \kappa_0(\zeta) > 0$ мають бути накладені на відносно велику кількість значень ζ навколо області. Це буде означати що вони будуть мати (багато) більше рівнянь ніж невідомі і замість цього вимагатиме використання хоча б методу квадратів. Ця проблема не спричиняє ніякої сингулярної поведінки в системі рівнянь і здається не викликає якісних змін частоти відклику графіків теж. Тому, на даному етапі надається перевага уникненню даної проблеми.

На рис. 4-5 спочатку ($t_*=0$) маса дисбалансу знаходиться в $\zeta=0$, потім рухається з вихором (тобто, з кутовою швидкістю $\omega-\Omega$, або $\tilde{\omega} - 1$ з точки зору безмірних параметрів). Таким чином, вершина хвилі знаходиться біля маси дисбалансу перед критичною швидкістю обертання, і навпроти маси дисбалансу після проходження критичної швидкості обертання, як зображено на рис. 1. Однак, присутнє повільне відхилення з кутовою швидкістю $1 - \tilde{\omega}$. Цей небажаний феномен був підтверджений в експериментах, і різноманітні засоби захисту були розглянуті щоб запобігти цьому, для прикладу – канал шестикутної форми і роздільні пластини [3].

Література

1. Jung C.-H. A dynamic model and numerical study on the liquid balancer used in an automatic washing machine / C.-H. Jung, C.-S. Kim, Y.-H. Choi // J. Mech. Sci. Tech. – 2008. – No 22. – P. 1843–1852.
2. Leblanc M. Automatic balancer for rotating bodies // U. S. Patent No. 1,209,730.
3. Nakamura, T. (2009). Study on the improvement of the fluid balancer of washing machines // In Proceedings of the 13th Asia-Pacific Vibrations Conference – 22-25 November 2009 – P. 1-8. University of Canterbury, New Zealand.
4. Urbiola-Soto L. Dynamic performance of the Leblanc balancer for automatic washing machines / L. Urbiola-Soto, M. Lopez-Parra // J. Vibr. Acoust. – 2011. – Vol. 133 – 041014-1-041014-8.
5. Балансировка барабана стиральной машины: инструкция. URL: <https://cosmo-frost.ru/stiralnye-mashiny/remont-i-zamena/balansirovka-barabana-stiralnoj-mashiny-instrukciya/> (дата звернення: 20.10.2018).
6. Серебро в стиральном барабане. URL: <http://irvispress.ru/catalog/krupnaja-bytovaja-tehnika/stiralnye-mashiny/serebro-v-stiralnom-barabane/> (дата звернення: 27.10.2018).

Моделювання тенденції динаміки валютного курсу гривні

Троянов Б.В., Григоруک С.С.

Науковий керівник – к.п.н., доц. Григорук С.С

Хмельницький національний університет

Вибір системи валютного курсу будь-якою країною, виступаючи найважливішою складовою макроекономічної стабільності і економічного зростання, визначається рівнем розвитку і розмірами економіки, ступенем її відкритості, станом фінансових ринків, станом платіжного балансу, рівнем конкурентоспроможності, величиною резервів іноземної валюти, ступенем залежності економіки від зовнішньої торгівлі, суспільно-політичним кліматом в суспільстві, станом національної грошової системи, природою і характером економічних потрясінь, з якими стикається та чи інша країна.

У контексті простежених трендів розвитку української економічної системи, на наш погляд, найбільш актуальною і пріоритетною проблемою для вітчизняних фірм можна відзначити саме проблеми організації зовнішньоекономічної діяльності. Дані проблеми є нерозривно пов'язаними із завданнями формування валютної політики фірм. Основним ядром цих завдань є управління валютними ризиками, які викликані коливаннями валютних курсів.

Ціль роботи: проаналізувати та спрогнозувати динаміку валютного курсу гривні (долар, євро).

Дані про курс валют було взято з сайту національного банку України [1]. Аналіз здійснювався за період від 1 січня 2013 р. до 1 вересня 2018 р. методами аналізу часових рядів за допомогою MS Office Excel. За даними національного банку було побудовано полігони у гривні та приведені гривні, у якій враховано інфляцію (дані про індекс інфляції України було взято з сайту Мінфін[2]). Динаміку зміни курсу гривні показано на рисунках 1-3.

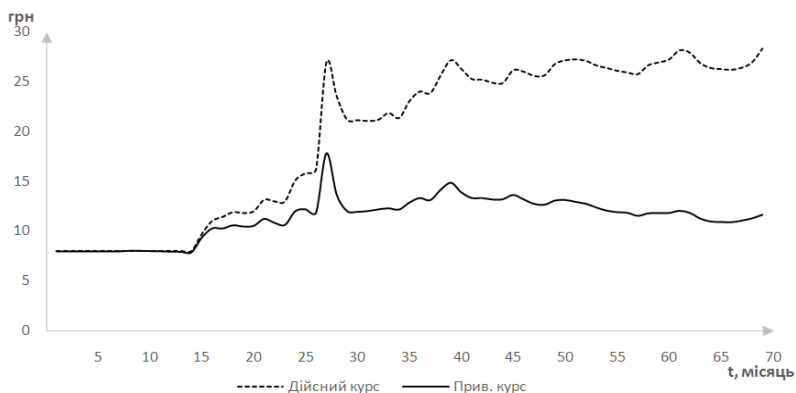


Рисунок 1 – Динаміка зміни курсу гривні відносно долара США

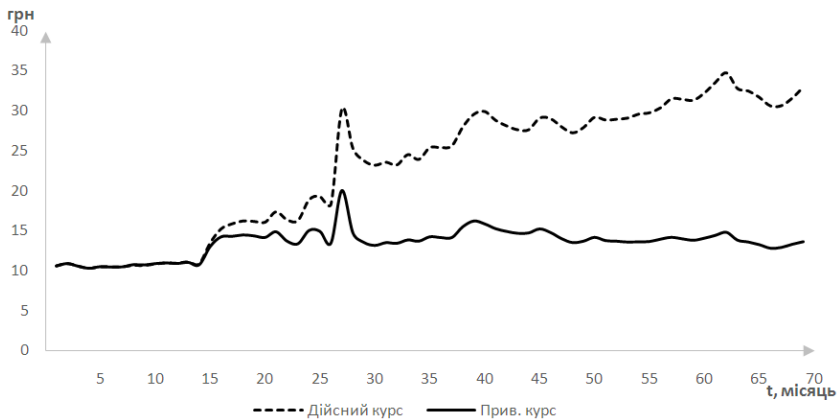


Рисунок 2 – Динаміка зміни курсу гривні відносно Євро

Візуальний аналіз побудованих полігонів свідчить, що:

- до 2014 року (0–13 місяці) курс гривні стабільний;
- у 2014–2016 роках (14–35 місяці) відбувається досить хаотична зміна курсу, що можна пояснити початком та активною стадією військового конфлікту;

- з 2016 року спостерігається тенденція до зростання валютного курсу гривні, хоча курс приведеної гривні є відносно стабільним;

- також, було припущено, що протягом року можливі сезонні коливання, а саме – ближче до зимових і після зимових свят курс гривні починає зростати, інший період часу курс гривні спадає і нормалізується.

Так як є ознаки коливного процесу, для оцінки динаміки валютного курсу гривні було обчислено середні показники динаміки [3].(Таблиця 1)

Таблиця 1 – Середні показники динаміки валютного курсу за період з 1 березня 2015 р. до 1 вересня 2018 р., прив. грн

	Долар США	Євро
Середній рівень ряду	12,413	14,014
Середній абсолютний приріст	-0,0091	0,0022
Середній темп росту	99,92	100,02
Середній темп приросту	-0,077%	0,016%

З середніх показників динаміки можна зробити висновок, що у період з березня 2015 року по вересень 2018 р.:

- середня ціна гривні за 1 долар США – 12,413 прив. грн. з урахуванням інфляційних процесів та наявна слабка тенденція до зменшення ціни за долар США на 0,0077%;

- середня ціна гривні за 1 євро – 14,014 прив. грн. з урахуванням

інфляційних процесів та спостерігається тенденція до зростання курсу 0,016%.

Статистична перевірка підтвердила наявність тренду в динамічних рядах.

У результаті були побудовані трендові моделі:

- для долара: $y = -0,003812t^2 + 0,33t + 6,04$ (рисунок 3);
- для євро: $y = -0,00296t^2 + 0,248t + 9,55$.

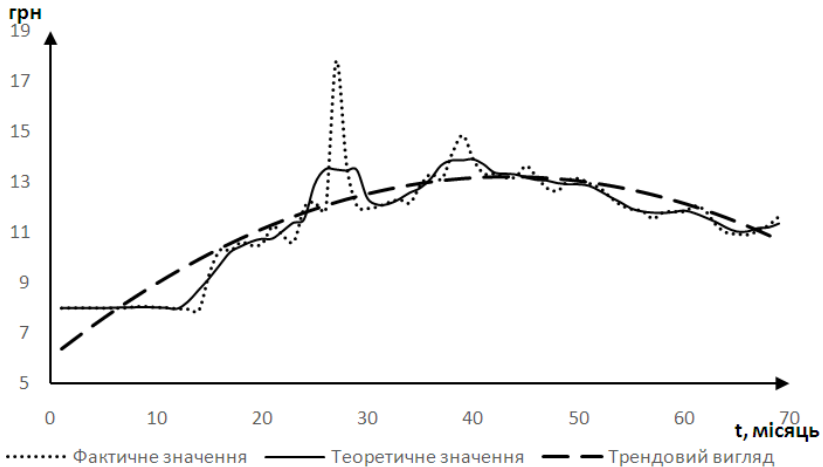


Рисунок 3 – Модель курсу гривні відносно долара США

Висновки

Таким чином, наше дослідження тенденції динаміки валютного курсу гривні дозволило зробити припущення, що спостерігається тенденція до відносної стабілізації гривні. Але для перевірки цього припущення у подальшому нашому дослідженні необхідно більш детально проаналізувати динаміку валютного курсу, зокрема проаналізувати коливні процеси.

Література

1. Офіційний курс гривні щодо іноземних валют [Електронний ресурс] / Веб-сайт НБУ – Режим доступу: <https://bank.gov.ua/control/uk/curmetal/detail/currency?period=daily> – Назва з екрана.
2. Індекс інфляції [Електронний ресурс] / Веб-сайт Мінфін – Режим доступу: <https://index.minfin.com.ua/ua/economy/index/inflation/> – Назва з екрана.
3. Герасименко С. С. Статистика : підручник / С. С. Герасименко. – К.: КНЕУ, 1998. – 468 с.

Програма-оптимізатор виробництва і реалізації продукції одягу ТОВ «Бембі»

Франчук М.Ю.

Науковий керівник- к.п.н. доц. Праворська Н.І.

Хмельницький національний університет

На сьогоднішній день економіка України виходить з застою. У порівнянні зі зруйнованою інфраструктурою 90-х років минулого століття та кризою 2008-2009 рр., сьогоднішній економічний стан можна назвати таким, що розвивається.

Актуальність даної роботи полягає економічній доцільності оптимізації виробництва текстильної направленості на прикладі ТОВ «Бембі». Підвищення ефективності призвело до зменшення собівартості одиниці продукту та кращого розуміння ситуації на ринку. Результати даного дослідження – програмний продукт та математична модель, на якій він заснований – можуть бути використані також для інших підприємств. Як було сказано вище, легка промисловість невибаглива до територіального розміщення, отже дослідження в цій області можливо використовувати в будь-якому регіоні України. Зміни в українському законодавстві спричинили зростання іноземного капіталу та внутрішніх інвестицій, що робить цю сферу діяльності перспективнішою, ніж раніше.

Мета дослідження – підвищити ефективність виробництва і реалізації продукції одягу від ТОВ «Бембі» у м. Хмельницький.

Завдання дослідження полягають у досягненні основних та допоміжних цілей, таких як:

- дослідження предметної області, вивчення термінології, що використовується;
- аналіз існуючих методів та використовуваних практик при вирішенні аналогічних задач;
- порівняльний аналіз вибраних методів для підвищення ефективності підприємства;
- побудова математичної моделі з використанням вибраної математичної моделі;
- розробка програмного продукту на основі розробленої математичної моделі.
- написання документації користувача, створення ілюстрацій з прикладами роботи програми;
- аналіз можливих шляхів розвитку програмного продукту.

Об'єкт дослідження – ефективність роботи підприємства легкої промисловості.

Предмет дослідження – підвищення ефективності роботи підприємства шляхом створення математичної моделі підприємства; її подальша оптимізація за допомогою використання створеного програмного продукту.

ТОВ «Бембі» має досвід роботи на ринку з 1998 року. Одяг під

брендом «Бембі» має ринки збуту продукції не тільки в Україні, але і в ближньому зарубіжжі – Росії, Білорусі, Молдові. Компанія з двадцятирічним досвідом стала ведучим українським виробник дитячого одягу з найширшим асортиментом – понад 400 моделей в 9 групах. Також розвивається ринок експорту на нові ринки далекого зарубіжжя – одяг постачається також в Канаду, Нову Зеландію та Німеччину.

Необхідно провести опис інструментарію, що використовувався при побудові математичної моделі та програмного продукту. Основою математичної моделі є принцип ефективності за Парето, програмний продукт написаний з використанням таких інструментів як:

- мова програмування C#;
- середовище розробки Visual Studio з використанням інтерфейсу програмування Winows Forms;

З використанням даного інструментарію було створено додаток для підвищення ефективності підприємства ТОВ «Бембі» в м. Хмельницький.

Програмний продукт призначений для проведення оптимізації, використовуючи принцип Ефективності за Парето. Програма застосовується після створення математичної моделі для аналізу впливу змін параметрів системи.

Вимоги до системи.

Для роботи програмного продукту необхідно мати такі параметри ПК:

- операційна система сімейства Windows; версія не старіша, ніж Windows Vista.
- встановлений .NET Framework 4.0 і новіше.

Апаратні характеристики системи не обмежуються програмним продуктом і можуть бути рівними системним вимогам операційної системи.

Для користувача програми – математика-аналітика, робота з програмним продуктом матиме вигляд, як показано на рисунку нижче:

Особливістю роботи з програмою є те, що її використання може бути як одноразовим для досягнення конкретних параметрів виробництва, або ж регулярним, для ітеративного покращення отриманих результатів.

Будова програми описана на рис. 2, який схематично зображає використання архітектурного шаблону MVP (Model View Presenter).

У класі View знаходиться код, що відповідає за відображення даних для користувача. Ніяких керуючих ролей у нього немає. Клас Optimizer являє собою модель, тобто всю бізнес-логіку програми, яка проводить обрахунки. Клас Presenter є поєднуючим класом і веде обробку даних, отриманих від користувача та передачу їх моделі; віддає на вивід дані на відображення, отримані від моделі.

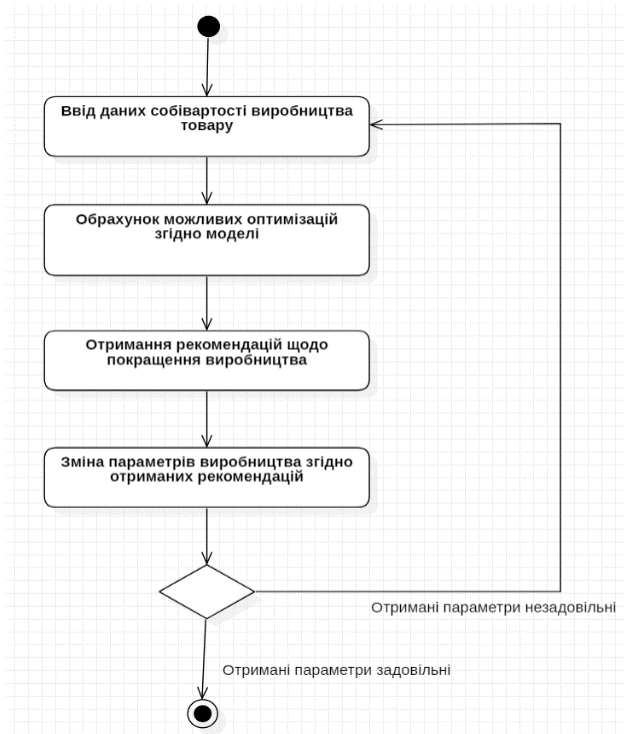


Рисунок 1 – Хід виконання програми-оптимізатора

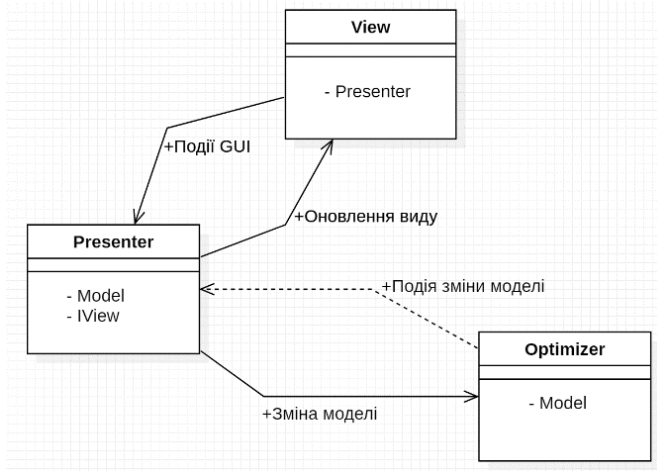


Рисунок. 2 – Будова програми згідно принципу побудови MVP

На ринку існують програмні продукти, які виконують схожу роль, так, наприклад, продукція Yokogawa Electric Corporation надає послуги по інтеграції програмних продуктів у сферах системи управління, програмного забезпечення по збору даних та інших. Однак, їх продукція розрахована на великих корпоративних замовників і цінова політика на послуги не вписується в бюджет малого бізнесу.

Отже, було розглянуто можливість підвищення ефективності підприємства легкої промисловості на прикладі ТОВ «Бембі» за допомогою створеної математичної моделі.

Варто зазначити, що хоча й використання програмного продукту, що є результатом даної роботи, і було продемонстровано на конкретному підприємстві з м. Хмельницький, але ним не обмежується. При необхідності, можливо пристосувати програмний продукт до інших підприємств по Україні зі схожою сферою праці.

Література

1. Червак, О. Ю. Теорія оптимального вибору. Підкритерії паретівської згортки критеріїв / О. Ю. Червак // Науковий вісник Ужгородського університету : Серія: Економіка / редкол.: В.П. Мікловда (голов. ред.), М.І. Пітюлич, Н.М. Надь та ін. – Ужгород: Видавництво УжНУ «Говерла», 2010. – Вип. 30. – С. 28–30.

2. Шилдт Г. Полный справочник по С# 4.0 / Шилдт Г. – Київ, 2015 – 375 с.

Проблема забезпечення швидкого шифрування інформації для передачі даних між клієнтом та сервером

Хітрін О. В.

Науковий керівник - д.т.н. Бедратюк Л. П.
Хмельницький національний університет

Вступ

Всім відомий вислів «Хто володіє інформацією, той володіє світом». А хто володіє інформацією конкурентів, має змогу отримати безпрецедентні переваги в боротьбі з ними. Прогрес зробив компанії залежними від інформаційних систем, а разом з цим - уразливими до атак хакерів, комп'ютерних вірусів та людського фактору в такій мірі, що багато власників бізнесу вже не можуть відчувати себе в безпеці. Але в той же час питання захисту цифрової інформації надає безліч можливостей задля убезпечення від витоків інформації.

Сьогодні все частіше світ інформаційних технологій зіштовхується з проблемою забезпечення належного захисту цифрової інформації. В епоху бурхливого розвитку технологій, проблеми інформаційного захисту

виявляються одними з найбільш важливих.

Основні проблеми захисту інформації в комп'ютерних системах виникають через те, що інформація не є жорстко пов'язаною з носієм. Її можна легко і швидко скопіювати і передати по каналах зв'язку. Інформаційна система схильна до як зовнішніх, так і внутрішніх загроз з боку порушників. Тому постає питання шифрування інформації.

Постановка проблеми

Проаналізувавши проблему захисту даних, було вироблено дане рішення, яке є одним з найбільш оптимальних враховуючи час на виконання операцій коування/декодування, надійності захисту та часу передачі даних. Даний клас задач вимагає пошуку вирішення наступних проблем:

- як закодувати інформацію;
- як як передати дані;
- як розкодувати дані на приймаючій стороні.

Мета роботи: сформувати рішення поставленої проблеми на основі описаного рішення що дасть змогу знайти оптимальне рішення для захисту даних при передачі по мережі.

Аналіз існуючих рішень

Здійснивши аналіз можливих методів захисту інформації, було виділено три основних схеми методів реалізації захисту інформації:

- хешування;
- симетричні методи;
- асиметричні методи

Хешування створює унікальний, фіксований підпис для повідомлення або набору даних. Кожен хеш унікальний для певного повідомлення, тому незначні зміни в цьому повідомленні будуть легко відстежуватися. Коли дані шифруються за допомогою хешування, його неможливо змінити або розшифрувати. Хешування досі є методом захисту, який все ще є корисним при передачі даних, задля усунення можливості підробки та модифікації даних.

Симетричне шифрування також називається криптографією приватного ключа, і називається так, тому що ключ, який використовується для кодування та розкдування повідомлення, повинен залишатися безпечним, оскільки кожен, хто має доступ до нього, може розшифрувати дані. Використовуючи цей метод, відправник шифрує дані за допомогою певного набору правил, надсилає дані (зашифрований текст), а потім приймач використовує ці самі правила для розшифрування даних.

Асиметричне шифрування або криптографія відкритого ключа відрізняється від попереднього методу, оскільки він використовує різні ключі для шифрування або дешифрування. Цей метод потенційно є більш безпечним за симетричні методи шифрування. За допомогою цього методу відкритий ключ вільно доступний всім і використовується для шифрування повідомлень, а одержувач використовує інший приватний ключ для

розшифрування повідомлень.

Виклад основного матеріалу дослідження

Для проведення порівняльних дослідів було наступні алгоритми шифрування інформації:

– DES — це стандарт шифрування даних, був першим стандартом шифрування, рекомендованим NIST (National Institute of Standards and Technology). Він заснований на алгоритмі IBM під назвою Lucifer. DES став стандартом в 1974 році. З того часу було зроблено багато атак та винайдено методів записані, які експлуатують слабкі сторони DES, що зробило його небезпечним алгоритмом шифрування.

– 3DES (Tripple DES) – вдосконалення алгоритму DES. У ньому використаний стандартний метод шифрування аналогічний оригіналу DES, але він застосовується 3 рази, щоб збільшити рівень шифрування. Але широко відомий факт, що 3DES працює повільніше, ніж інші методи блочного шифрування.

– AES (Advanced Encryption Standard) - це новий стандарт шифрування, рекомендований NIST для заміни DES. Алгоритм також носить назву Rijndael (вимовляється Rain Doll), він був обраний у 1997 році під час конкурсу обрання найкращого стандарту шифрування. Атака Brutforce є єдиною ефективною атакою, відома проти цього алгоритму. За цієї атаки зловмисник намагається перевірити всі комбінації символів. І AES і DES - блочні шифри.

– Blowfish - це один з найпоширеніших алгоритмів симетричного шифрування, створений Брюсом Шнайером - одним з провідних криптологів у світі та президентом консалтингової компанії Counterpane Systems, яка спеціалізується на криптографії та комп'ютерній безпеці.

Дані алгоритми були обрані через те, що вони є найбільш використовуваними при шифруванні інформації на даний момент.

Основною задачею даного дослідів є отримання відповідей на питання, який же із алгоритмів є найшвидшим та найкраще підходить для шифрування інформації при передачі даних через мережу.

Для реалізації усіх вище описаних алгоритмів було обрано мову програмування C#. Вибір обгрунтовано тим, що вона володіє всіма засобами для реалізації даних алгоритмів та оптимізована для швидкої роботи. Середовищем розробки було обрано Visual Studio 2017.

Клас Stopwatch було використано у якості засобу вимірювання швидкості виконання. Цей клас дозволяє точно виміряти період часу, який було витрачено на виконання певних інструкцій коду.

Усі алгоритми було протестовано на обчислювальній машині з такими характеристиками:

- Processor AMD A10;
- 8 GB RAM;

- Windows 10;
- .NET 4.7

Вхідними даними є текстовий файл визначеної довжини, який дозволить визначити різницю у швидкості роботи алгоритмів.

У процесі виконання алгоритмів та вимірювання результатів до уваги брався лише час, необхідний для шифрування інформації.

Результати

Після опрацювання вхідних даних в об'ємі від 64 до 256 мегабайт було отримано такий час виконання кожного алгоритму як зображено в таблиці 1.

Таблиця 1 – Час виконання алгоритмів шифрування

Назва алгоритму	МВ оброблено	Час витрачено	МВ/Сек
Blowfish	256	3.976	64.386
AES(128-bit) key	256	4.196	61.010
DES	128	5.998	21.340
DES3	64	6.499	20.783

Із даної таблиці стає зрозумілим що рішення шифрування на основі алгоритмів Blowfish та AES.

Проте необхідно врахувати, що алгоритм Blowfish є симетричним алгоритмом шифрування, тому він не здатен забезпечити належний рівень захисту інформації.

Висновки

У ході даної роботи було виділено основні вимоги задля забезпечення захищеної передачі даних, які вимагаються від систем, що здійснюють передачу даних у мережі. Також було вивчено основні підходи до вирішення даної проблеми, виокремлено їхні особливості та варіанти використання.

Було здійснено аналіз симетричних та несиметричних методів шифрування інформації. Також було проаналізовано історію використання тих чи інших методів та їхню надійність у ході практичного застосування.

Виявилося, що симетричні методи шифрування інформації є недостатньо надійними та в історії суспільства існує велика кількість випадків, коли такі методи були скомпрометовані та зловмисники отримували доступ до захищених даних.

Використання симетричних методів шифрування даних, а саме алгоритму AES, дозволяє за прийнятний час забезпечити належний рівень захисту даних та унеможливило спроби несанкціонованого доступу до закодованої інформації.

Література

1. Wireless Security Handbook: Auerbach Publications, 2015. – С. 88.
2. Applied Cryptography: BRUCE SCHNEIER, "Applied Cryptography", John Wiley & Sons, Inc – С. 308.
3. Security In Wireless LANS And MANS: Artech House Publishers, 2005 – С. 405

4. Performance Comparison: Security Design Choices – Режим доступу: <https://msdn.microsoft.com/en-us/library/ms978415.aspx>. – Дата доступу: 29.08.2018.

5. Block Cipher. – Режим доступу: https://en.wikipedia.org/wiki/Block_cipher. – Дата доступу: 21.08.2018.

Розробка навчального модуля «особові рахунки» для інформаційного забезпечення автоматизованого робочого місця касира банку

Шостак А.С.

Науковий керівник – к.ф.-м.н., доц. Ратушняк Т.В.

Університет державної фіскальної служби України

Розрахункові та касові операції належать до основних, найбільш трудомістких та відповідальних у банківській діяльності. Тому для виконання цих операцій застосовується обчислювальна техніка. Сьогодні в інтегрованих банківських системах комплекс розрахункових і касових операцій становить підсистему, головне призначення якої — автоматизувати облік, контроль та складання звітності про рух коштів на рахунках клієнтів банку. Згідно з призначенням такої підсистеми виконуються її функції: відкриття особових рахунків, їх перегляд, доповнення й коригування, формування довідкової фінансової та технологічної інформації про стан особових рахунків через такі процедури, як блокування особового рахунку на установлений термін, вилучення закритих рахунків.

Для автоматизації розрахунково-касових операцій і контролю за їх здійсненням в інтегрованих банківських системах створюються такі автоматизовані робочі місця (АРМ): операціоніста, контролера, технолога, бухгалтера, спеціаліста валютного відділу, касира, кредитного інспектора, керівника (відділу, банку), адміністратора бази даних. Кожне з перелічених АРМ має різний набір функцій, різний рівень доступу до бази даних.

Інформаційну базу (ІБ) розрахунково-касових операцій утворює сукупність певним чином структурованої (як документи чи файли) інформації, використовуваної під час виконання завдань підсистеми.

Файли оперативної інформації протягом робочого дня формуються або на підставі вхідних повідомлень, які надходять операціоністу у вигляді паперових чи електронних (по мережі) документів, або під час обробки оперативної інформації.

Архівні файли – це файли результатної інформації, сформовані після складання балансу та записані в архів. За допомогою файлів довідкової, оперативної та архівної інформації виконуються завдання з обліку та контролю розрахункових і касових операцій.

Автоматизація розрахункових операцій виконується за допомогою програмного комплексу "Операційний день банку" (ОДБ), який дає змогу в

будь-який момент його функціонування дістати інформацію про рух коштів на розрахункових, поточних, кредитних, депозитних та інших рахунках. Залежно від конструктивних характеристик програмного комплексу ОДБ облік і контроль за розрахунковими операціями здійснюються в національній валюті та валюті інших країн в одному ОДБ. У більшості програмних пакетів ОДБ передбачається, що головними виконавцями автоматизації розрахункових операцій є операціоністи, контролери, технологи. Для них у банківській системі створюються відповідні АРМ. Конкретний перелік операцій для кожного АРМ залежить від запровадженого технологічного процесу, наявності технічних засобів, обсягу документообігу тощо, але деякі роботи жорстко прив'язані до виконавців.

Технологічний процес автоматизації розрахункових операцій здійснюється за допомогою програм, які об'єднані у модуль під назвою «Особові рахунки».

Базовою функцією під час автоматизації розрахункових операцій є відкриття нових особових рахунків. При цьому зазначають такі реквізити: тип особового рахунку, рівень секретності рахунку, вид валюти рахунку, реєстраційний номер клієнта в банку, номер балансового рахунку, номер особового рахунку клієнта, назва особового рахунку, дата відкриття; підстава для відкриття рахунку; тип особового рахунку для податкової адміністрації тощо. Супроводження особових рахунків виконується на робочих місцях відповідальних виконавців-операціоністів і технолога. Крім відповідальних виконавців, доступ до особових рахунків з різними повноваженнями мають працівники валютного відділу, каси, головний бухгалтер та керівники служб банку. Зазначеним працівникам доступна функція перегляду особових рахунків. Функція перегляду особових рахунків дає змогу:

- переглянути реквізити рахунку, зокрема номер балансового рахунку, номер особового рахунку, тип рахунку, його призначення та інші довідкові ознаки;

- визначити оперативний стан особового рахунку, а саме: вхідний залишок, поточні обороти коштів, вихідне сальдо рахунку.

Банки України використовують такі програмні комплекси ОДБ: ИСАОД, 4К-Bank, Profix, SmartBank, Мебиус, Скрудж, SyBridge, В2 та інші.

В силу різних причин (зокрема, висока вартість програмних комплексів, відсутність технічних можливостей) та з міркувань безпеки, студенти вищих навчальних, зазвичай, не мають можливості під час навчання отримати навички роботи в жодному з комплексів ОДБ; винятками є лише окремі домовленості ВУЗів про проходження короткотермінових ознайомчих курсів або екскурсій у фінансові установи або банки.

Тому актуальною задачею для якісної підготовки студентів спеціальності «Фінанси» є розробка навчальних середовищ, подібних за своїм функціоналом до автоматизованих робочих місць працівника банку.

У представлений науковій роботі запропоновано принципову схему

руху коштів між особовими рахунками клієнтів банку та реалізовано засобами системи управління базами даних Access. На рисунку 1 представлено схему даних модуля «Особові рахунки». Функціонал побудованої бази даних відповідає функціоналу модуля «Особові рахунки» програмного комплексу «Операційний день банку». Функціонал дає змогу:

- переглянути реквізити рахунку, зокрема номер балансового рахунку, номер особового рахунку, тип рахунку, його призначення та інші довідкові ознаки;
- визначити оперативний стан особового рахунку, а саме залишок на рахунку;
- перевести кошти з одного рахунку на інший;
- переглянути архів операцій руху коштів між особовими рахунками клієнтів банку.

Під час розробки бази даних було вирішено наступні задачі: розробка інфологічної моделі; побудова таблиць з даними; проектування запитів, форм, звітів, макросів.

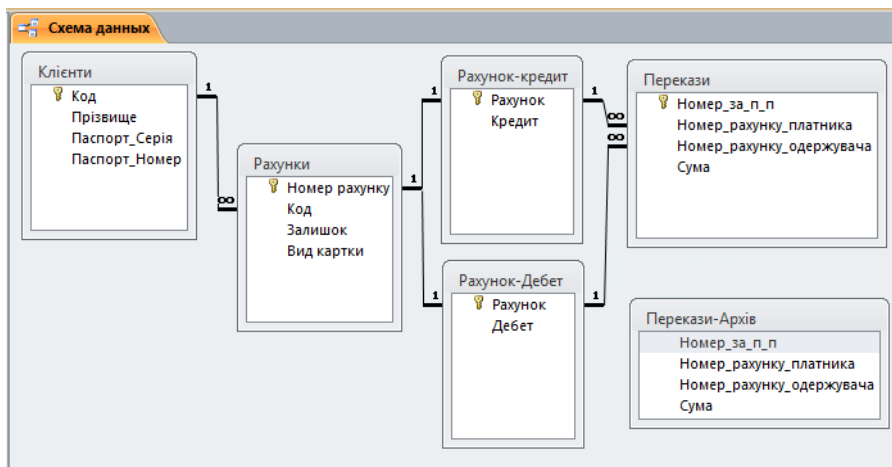


Рисунок 1 - Схема даних модуля «Особові рахунки»

Побудована база даних може бути корисною для якісної підготовки студентів спеціальності «Фінанси» в якості навчального середовища для засвоєння теоретичних знань та отримання практичних навичок роботи у комп'ютерних програмах працівників банку.

Література

1. Информатика та обчислювальна техніка: практикум / [В.А. Одинець., О.В. Гладченко, Т.В. Ратушняк та ін.]. – Ірпінь: Видавництво Національного університету ДПС України, 2015. – 394 с.

2. Практикум з інформатики з використанням MS Office 2010: Навч. посібник / [Одинець В.А., Мостіпан О.І., Ратушняк Т.В., Поденежко О.В., Гладченко О.В., Сьомка Н.К.] – Ірпінь: Національний університет ДПС України, 2014. – 538 с.

3. Ратушняк Т.В., Вишемірська Я.С. Використання інформаційних технологій в соціально-економічних дослідженнях // Теорії мікро-макроекономіки. — 2009. – Вип. 33. – С. 97 – 102.

4. Ратушняк Т.В., Вишемірська Я.С. Сучасні інформаційні технології в соціально-економічних дослідженнях // Науковий вісник Академії муніципального управління. Серія «Техніка». - 2010. – Вип. 1. – С. 134 – 1

Аналіз методу оптимізації згорткових нейронних мереж при розпізнаванні образів на мобільних пристроях

Ящук Р.В.

Науковий керівник – к.т.н, доцент Форкун Ю. В.

Хмельницький національний університет

Сьогодні нейронів навчання застосовують при розробці ПЗ для розпізнавання навколишнього середовища, фото- і відеоінформації. Умовно цю область називають комп'ютерним зором. Однак машинне навчання не обмежується тільки картинками, його також можна використовувати для роботи з аудіо- та дактилоскопічними даними (розпізнавання відбитка пальця, голосу і т. п.).

На мобільних пристроях зазвичай використовується кілька систем глибокого навчання: розпізнавання осіб, китайських ієрогліфів, кодових фраз «Ок, Google». Варто враховувати, що ці програми в більшості випадків нічому не вчать безпосередньо у користувача. Оскільки сучасні гаджети мають швидкий вихід в мережу інтернет, процес навчання можна перенести на віддалений сервер.

Найчастіше весь процес навчання відбувається на віддаленому сервері з великою кількістю GPU. Саме на нього надходять дані про запити користувачів, які згодом оброблюються системою.

Все частіше нейронні мережі навчаються безпосередньо на мобільних пристроях, що гарантує повну конфіденційність та безперервну роботу мобільного додатку. При цьому постає питання оптимізації, тому що для користувача стабільність роботи системи є одним із пріоритетів.

Один з найпростіших способів додавання можливостей Machine Learning полягає у використанні нового ML Kit з Firebase, нещодавно оголошеного в Google I/O 2018.

ML Kit - це набір API-інтерфейсів, що надаються Firebase, які забезпечують розпізнавання обличчя, сканування штрих-кодів, розпізнавання тексту, виявлення орієнтирів та маркування зображення. Деякі з цих API

забезпечують офлайн-режим, який дає змогу користуватися цими функціями, не турбуючись про те, чи користувач має підключення до Інтернету.

ML Kit відмінно підходить для випадків звичайного користування, описаних вище, але що робити, якщо у вас є якийсь особливий випадок використання? Наприклад, ви хочете мати можливість класифікації різних предметів в реальному часі за допомогою камери телефону. З вирішенням такого завдання допоможе бібліотека TensorFlow Lite.

TensorFlow Lite - це рішення TensorFlow, що застосовує не надто комплексні моделі машинного навчання і підходить для мобільних пристроїв. Вона також використовує апаратне прискорення на платформі Android за допомогою Machine Learning API.

Але при цьому постає питання тренування нейронної мережі обумовлено тим фактом, що навчання згорткових нейронних мереж є надзвичайно складною проблемою оптимізації. Вони, як правило, використовують стохастичний градієнтний спуск, щоб знайти мінімуми для функції втрат, і ця методика використовує великі набори початкових даних з мітками для ефективної мінімізації. Оскільки SGD є жадібним методом, неможливо знайти глобальні мінімуми. Проте було показано, що методика transfer learning, відома як точне налаштування параметрів, може підвищити продуктивність згорткової нейронної мережі у порівнянні з випадковою ініціалізацією [2].

Процес створення власної моделі нейронної мережі описано нижче на рисунку 1.

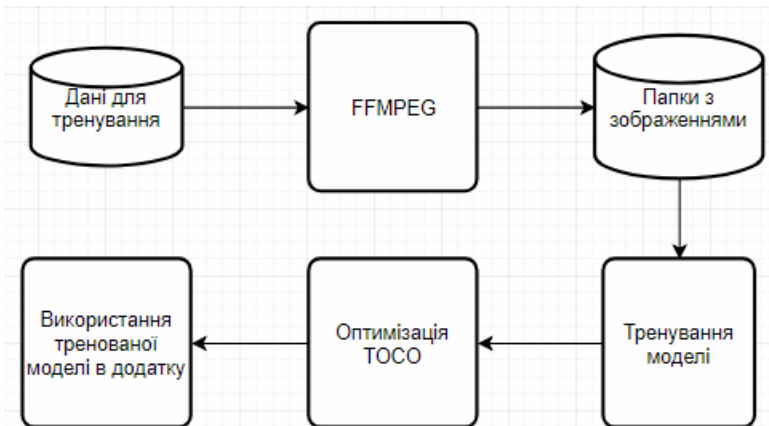


Рисунок 1 – діаграма процесу створення кастомної моделі TensorFlow.

Підготовка моделі TensorFlow може тривати довгий час і вимагати великого об'єму даних для тренування. На щастя, існує спосіб зробити цей

процес коротшим, без гігабайтів даних чи при невеликій потужності графічного процесора.

Transfer Learning (тонке налаштування нейронної мережі) - це процес використання вже підготовленої моделі та подальшої перепідготовки її для створення нової моделі.

Залежно від завдання необхідно знайти компроміс між складністю та розміром моделі. Якщо ваше завдання вимагає високої точності, то вам може знадобитися велика і складна модель. Деякі завдання можуть працювати з менш точною моделлю, для цих цілей краще використовувати не надто об'ємну модель, але менш точну. Менші моделі не тільки використовують менше місця на диску та пам'ять, але, як правило, більш швидкі та енергоефективніші. Наприклад, наведені нижче графіки демонструють компроміс між обсягом пам'яті, необхідної для моделі, та точністю при використанні моделі для класифікації зображень.

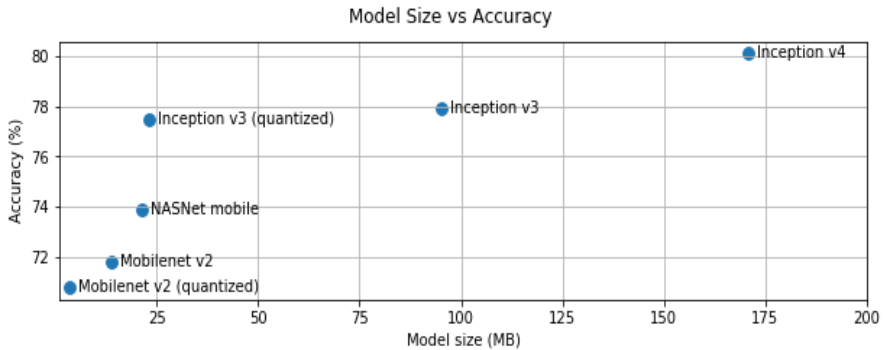


Рисунок 2 – Графік залежності розміру моделі та її точності

Було обрано модель MobileNet_V2, яку ми зможемо тренувати на власних зображеннях.

Коли ми завершимо перепідготовку нашої моделі, нам потрібно оптимізувати файл для роботи на мобільних пристроях. TOCO або "TensorFlow Lite Optimizing Converter" – це інструмент, наданий бібліотекою TensorFlow, що оптимізує графік для роботи на мобільних пристроях.

Отриманий файл `retrained_graph.pb`, що було створено перед цим, за допомогою TOCO перетворюється в оптимізований граф та текстовий файл з мітками, що будуть показуватися в реальному часі при виявленні камерою пристрою об'єктів, якими було треновано нейронну мережу.

Оскільки було обрано платформу Android, необхідно обрати мінімальну версію API. Для цього було досліджено офіційні дані Google (рисунок 3).

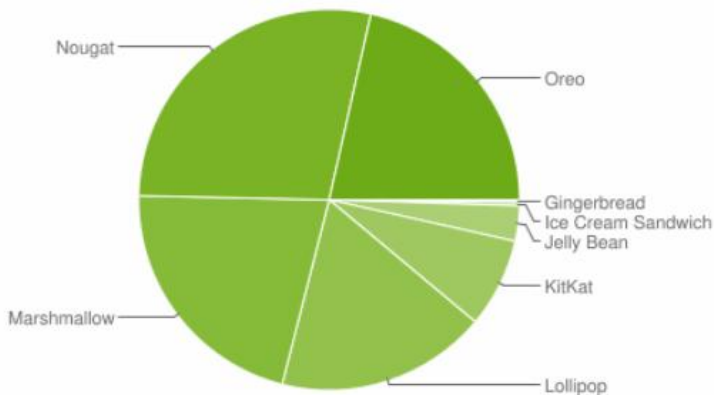


Рисунок 3 – Діаграма розповсюдження різних версій Android

В результаті було вирішено встановити мінімальною версією 5.0 «Lollipop». Цей вибір є оптимальним, ця версія операційної системи охоплює більшу частину ринку (91%).

Отже, в даній роботі проаналізовано використання згорткової нейронної мережі при обробці зображень. Обґрунтовано необхідність розв’язання задачі оптимізації для визначення оптимальних параметрів налаштування нейронної мережі. Наведено приклад, який пояснює отримання оптимальної структури згорткової мережі за допомогою TensorFlow Lite.

Література

1. Інформатика: Комп’ютерна техніка. Комп’ютерні технології. Посіб. / За ред. О.І. Пушкаря – К: Видавничий центр “Академія”, 2001. – 696с.
2. Jesse, Russell Штучна нейронна мережа / Jesse Russell. - М.: VSD, 2012. – 130 с. Jesse, Russell Нейрон / Jesse Russell. – М.: VSD, 2012. - 130 с.
3. Барський, А. Б. Логічні нейронні мережі / А.Б. Барський. – М.: Інтернет-університет інформаційних технологій, Біном. Лабораторія знань, 2007. – 352 с.
4. Mark L. Murphy Busy coder’s guide to Android development.
5. Круглов, В.В. Штучні нейронні мережі. Теорія та практика: моногр./ В.В.Круглов, В.В.Борисов. – М.: Гаряча лінія-Телеком, 2002. – 382 с.

Наукове видання

«Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ-фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.2: Математичне моделювання та інженерія програмного забезпечення – 96 с.

**Відповідальність за зміст текстів і якість редагування матеріалів
покладена на авторів і наукових керівників.**

Комп'ютерна верстка: Чешун В.М.
Дизайн: Муляр І.В.

Здано до складання 5.11.18. Підписано до друку 10.11.18. Формат 60x84/16. Папір друкарський. Тираж 50 прим. Умовних друкованих аркушів – 6. Обліково-видавничих аркушів –1,35.

Редакційний відділ ПВНЗ УЕП 29016, м. Хмельницький, вул. Львівське шосе, 51/2.

ББК 74.480.278
С.88