ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, КОМП'ЮТЕРНА ІНЖЕНЕРІЯ, КІБЕРБЕЗПЕКА

УДК 004.4:378

Баган С.В., студентка, IV курс, гр. СМ-15-1, ФАМІТ
Науковий керівник – к.т.н., доц. Мельников О.Ю.
Донбаська державна машинобудівна академія, м. Краматорськ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ВИЗУАЛІЗАТОРОВ ПРЕДСТАВЛЕННЯ БАГАТОМІРНИХ ДАНИХ

Дані, які представлені в чотирьох і більше вимірах, необхідно або перетворювати до тривимірного простору, або використати спеціальні методи: наприклад, «обличчя Чернова», що базуються на концепції кодуванні значень різних змінних у характеристиках людського обличчя; пелюсткові діаграми у вигляді кола; діаграми з паралельними координатами, де кожна з осей відображає значення по обраному показнику.

Кожний з методів має свій ареал застосування, розроблені додатки для порівняння методів і вибору кращого при візуалізації конкретних даних. У той же час жоден з існуючих програмних засобів не дозволяє користувачеві самому створити візуалізатор відповідно до власних переваг.

Було поставлене завдання проектування системи – застосування, що дозволяє користувачеві за допомогою графічних примітивів створити певний рисунок (схему) і визначити його параметри (для вимірів). Рисунок-схема з усіма описами зберігається в спеціальному файлі, а потім використовується для візуалізації даних. Створено інформаційну модель системи мовою візуального моделювання UML.

У цей час здійснюється робота з реалізації моделі в середовищі візуального програмування. Створене застосування дозволяє:

– працювати з даними (рис. 1): завантажити (імпортувати з редактора електронних таблиць, при цьому автоматично розраховується число вимірів), нормалізувати, зберегти;

– працювати з візуалізаторами;

– будувати багатовимірні діаграми.

Робота з візуалізаторами, у свою чергу, припускає або використання стандартних методів, або створення власних візуалізаторів.

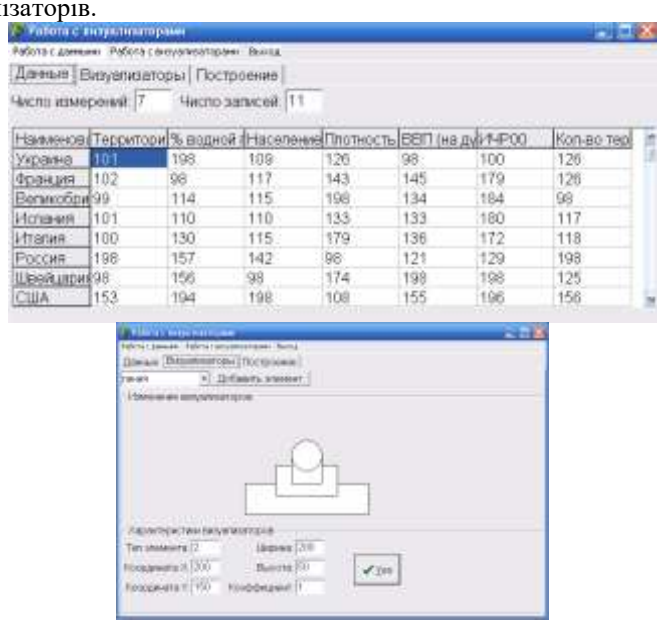


Рисунок 1 – Робота застосування: вкладки «Дані» й «Візуалізатори»

При роботі із власними візуалізаторами користувач має можливість завантажити раніше збережену схему, створити (додати) нові елементи й зберегти все створене або модифіковане на диску.

Приклад візуалізатора, показаний на рис. 1, містить 4 елементи: 3 елементи типу «прямокутник» й 1 елемент типу «коло». Користувач має можливість змінювати положення кожного елемента, а також параметр «коефіцієнт», що показує, у скільки разів максимальне нормалізоване значення буде більше мінімального (за замовчуванням дорівнює одиниці). При використанні такого візуалізатора для раніше представлених даних будуть застосовані 7 вимірів: для кожного прямокутника – ширина й висота, для кола – радіус.

Наступним етапом удосконалювання системи стане розширення переліку графічних примітивів, а також збільшення числа параметрів.

УДК 004.056.53

Байлюк Є.М., асист. каф. КІ та КБ
Покотило О.А., асист. каф. КІ та КБ
Житомирський державний технологічний університет

АНАЛІЗ СПОСОБІВ ЗАХИСТУ ПАРОЛІВ ВІД АТАК МЕТОДОМ «ГРУБОЇ СИЛИ»

Здавна паролі використовувалися як засіб аутентифікації, як спосіб кудись потрапити. Це було ще задовго до того, як з'явилися комп'ютери. Але з появою ІТ-систем люди не відмовилися від звички використовувати паролі. Це стало дуже великою проблемою для розробників тому, що вони зіткнулися з проблемою як зробити системи одночасно і зручними, і швидкими, і захищеними. Є багато методів зламу паролів. Один з них – метод «грубої сили».

Атака методом «грубої сили» (англ. Brute-force attack), або атака методом повного перебору – злам пароля шляхом перебору всіх можливих варіантів ключа. Будь-який пароль може бути підібраний шляхом повного перебору. При цьому, навіть якщо обчислення цільової функції від кожного конкретного можливого рішення задачі може бути здійснено за поліноміальний час, то в залежності від кількості всіх можливих рішень, час на виконання атаки змінюватиметься за експоненціальним законом.

Складність повного перебору залежить від кількості всіх можливих рішень задачі. Якщо простір рішень дуже великий, то такий вид атаки може не дати результатів протягом декількох років або навіть століть.

Криптографічні атаки, засновані на методі «грубої сили», є найбільш універсальними, але в той же час найбільш повільними. Вони використовуються в основному хакерами-початківцями. Атаки ефективні для нескладних алгоритмів шифрування і ключів довжиною до 64 біт. Для сучасних ключів довжиною від 128 біт є неефективними.

У криптографії на обчислювальній складності повного перебору ґрунтується оцінка криптостійкості шифрів. Зокрема, шифр вважається криптостійким, якщо не існує методу «злому» більш швидкого ніж повний перебір всіх ключів. Криптографічні атаки, засновані на методі повного перебору, є найбільш універсальними, але водночас і найбільш повільними.

Як почали працювати над безпекою паролів? Перше, що почали використовувати - криптографічні хеш-кодування. Основна особливість таких кодувань в тому, що їх не можна повернути назад. Тобто, якщо в цей хеш передали якусь інформацію та отримали на виході значення, то з нього назад інформацію отримати неможливо. Але, на жаль, вони дуже швидко обчислюються. Наприклад, сучасний кластер з 4-х відеокарт NVidia може перебирати кілька мільярдів паролів в секунду. Тобто, якщо ентропія пароля менше 40 біт, то кластер з 4-х відеокарт підбере його за хвилину, а то і швидше.

На кожен хеш є своя райдужна таблиця. Для цього беруть найпопулярніші паролі та комбінації символів, обчислюють для них хеші та зберігають їх на жорсткому диску об'ємом кілька терабайт. Коли зустрічається якийсь хеш, його можна не обчислювати, а дуже швидко знайти за цими таблицями і зіставити з паролем, який заздалегідь обчислений. Їх перевагою є те, що вони дуже швидко працюють, але потребують дуже багато місця для зберігання. В інтернеті є готові таблиці для найпопулярніших хешів, які можна скачати, або навіть купити.

Але на кожну райдужну таблицю знайдеться своя сіль. Сіль – це випадковий набір байтів, який дописується до паролю. Зберігається він у таблиці десь поруч з хешем і захищає від райдужних таблиць. Тобто людям, які отримали в руки базу з «посоленими» хешами, все одно доведеться ці хеші обчислювати. Але проблема в тому, що обчислюються ці хеші дуже швидко і сіль тут особливо не допомагає. Постає проблема уповільнення перебору хешів. Криптографи винайшли кілька функцій, які спеціально створені для уповільнення перебору паролів – вони використовують велику кількість пам'яті і всі можливі сучасні інструкції процесора. Якщо пароль, захищений цією функцією, потрапить в руки зловмисників, то їм доведеться використовувати комп'ютер з дуже потужною апаратною частиною.

Крім того, постала проблема перенавантаження власних серверів. Першими, хто вирішив дану проблему, була компанія Facebook. Вони винесли обчислення функції hmac (це хеш з ключем) на віддалений сервіс. Схема працює наступним чином: є власний сервер (бекенд), є віддалений сервіс. На цьому сервісі є якийсь секретний ключ. На бекенд заходить людина та вводить свій пароль. Він змішується з сіллю, хешується і відправляється на сервіс. Сервіс бере свій секретний ключ, обчислює функцію hmac і відправляє це все назад. Якщо у Facebook викрадуть базу користувачів, то перебирати паролі сенсу ніякого немає, тому що у них немає віддаленого секретного ключа. Але проблема підходу Facebook в тому, що якщо щось трапиться з їх віддаленим секретним ключем, то система перестане працювати.

Криптографи знайшли вирішення цієї проблеми. Було запропоновано представити паролі у вигляді числа. Це дає можливість складати, множити, перетворювати паролі або хеші від них в якісь інші числа. Також є можливість виконувати над ними і складні математичні операції. Однією з перших систем, яка почала використовувати цю технологію, стала Sphinx. Це детермінований менеджер паролів, який

використовує метод еліптичних кривих. Дана технологія має наступні переваги: вона дає можливість генерувати великі паролі, тобто захищає від перебору; якщо хакер отримає доступ до кількох паролів, він не зможе дізнатися інші, оскільки вони генеруються незалежно один від одного; якщо користувач якимось чином втратить свій майстер пароль, то це теж нічого не дасть, тому що у зловмисників не буде секретного ключа; вказана система працює дуже швидко. Але вона має недоліки. Сервер, до якого звертається користувач, нічого про нього не знає. Він просто отримує на вхід якісь випадкові числа, на щось їх примножує і відправляє назад. Клієнт також нічого не знає про сервер. Якщо сервіс перестане працювати, то користувач не матиме доступу до ресурсів. Крім того, немає можливості змінити секретний ключ.

Для покращення системи Sphinx було створено протокол Pythia. Завдяки йому сервіс «знає», хто вводить пароль, тобто на сервер окрім паролю передається ідентифікатор користувача. Це може бути якийсь випадковий id, або просто user name. В описаної системи є багато переваг: не навантажує власний сервер; якщо базу з такими числами вкрадуть, то перебирати паролі сенсу ніякого немає без секретного ключа; сервіс Pythia може блокувати спроби перебору; завдяки маскуванню сервіс нічого не знає про пароль; завдяки ZKP (Zero-knowledge proof) власний сервер завжди знає, що йому відповів саме той самий сервіс, до якого він колись спочатку звернувся; якщо зламають сам сервіс Pythia, то можна згенерувати новий приватний ключ. Недоліком цієї системи є те, що існує дуже мало бібліотек, які використовують еліптичні криві. Хоча, зважаючи на стрімкий розвиток технологій, всі перераховані мінуси можна назвати тимчасовими.

Але на цьому вчені вирішили не зупинятися. Вони задумалися над тим, чи можна домогтися тих властивостей, які надає Pythia, за допомогою математики, яка використовується багато років. В липні 2018 року було випущено новий протокол, який називається Simple Password-Hardened Encryption Services або скорочено PHE. Він має багато переваг. По-перше, використовує стандартну криву P-256, яка у всіх браузерах, продуктах застосовується по-замовчуванню. По-друге, вона працює приблизно раз в 10 швидше ніж Pythia і використовує стандартні примітиви. Її складно, але можна імплементувати власними руками, не використовуючи якісь сторонні бібліотеки. Можна використовувати OpenSSL, Bouncy Castle чи ін.

Працює PHE трохи по-іншому, ніж Pythia. Знову ж таки, є бекенд, є сервіс PHE. На бекенді є публічний ключ, на сервісі – приватний ключ. На відміну від Pythia, процес реєстрації і процес перевірки пароля проходить по-різному. Коли на сервіс приходить новий користувач і хоче зареєструватися, бекенд спочатку запитує у PHE-сервісу дані, які він може використовувати для реєстрації, якийсь Enrollment-запис. Після погодження цих даних сервісом, бекенд генерує випадкову 32-х байтну сіль (sNonce). На основі цієї солі та свого приватного ключа він генерує два числа, а також докази (Proof) того, що ці числа (або 2 точки) були згенеровані саме з використанням його приватного ключа, використовуючи протокол Шнорра. Далі бекенд перевіряє ці докази. Але пароля тут поки ще немає. У бекенда, зі свого боку, теж є особистий клієнтський приватний ключ і він, отримавши пароль від користувача, робить приблизно те ж саме, що робив сервіс, тільки додає туди ще пароль. Бекенд бере випадкову cNonce (випадкову клієнтську сіль), пароль і генерує знову 2 числа. Перше число використовується для аутентифікації, а у друге підмішується ще якесь випадкове число, домножене на приватний ключ бекенда. Це число теж розміром 32 байта і надалі може використовуватися для шифрування даних користувача і буде доступне в якості ключа тільки після того, як сам користувач ввів правильний пароль. На етапі реєстрації генерується не тільки запис для авторизації, а й ключ шифрування, унікальний для кожного користувача, яким можна зашифрувати його профіль, дані та ін. Потім бекенд просто складає те, що надіслав сервіс і те, що зробив він, та кладе в основу 2 солі (sNonce, cNonce), за допомогою яких були отримані ці числа і 2 числа, які вийшли в результаті суми. Відповідно процес аутентифікації користувача відбувається в зворотному порядку.

PHE володіє всіма основними функціями Pythia: наявна можливість випуску update token, якщо щось трапилося з приватним ключем; можна оновити базу і повернути її до початкового стану; наявний захист від перебору; сервіс нічого не знає про пароль; наявна можливість оновлювати базу і ключі.

У сервісу PHE (як і у інших сервісів) є недолік: у порівнянні з класичним підходом хеш + сіль, додаються точки відмови (мережа до сервісу, сервер з сервісом, сам сервіс) і швидкість виконання операції збільшується (як мінімум за рахунок мережевої взаємодії).

Таким чином, існує велика кількість способів захисту паролів від атак методом «грубої сили». Відповідно до проведеного аналізу, кожен з них має свої переваги та недоліки. Вчені проводили багато досліджень та покращували вже існуючі системи захисту. На даний час найкращим способом захисту паролів є протокол Simple Password-Hardened Encryption Services(PHE). Найкращим виявився протокол Simple Password-Hardened Encryption Services(PHE). Використовуючи його, перебирати паролі, як запевняють вчені, буде неможливо (або дуже довго) без приватних ключів. Крім того, в разі витоку бази або ключа, досить оновити ключ і дані в базі.

УДК 004.4'242

Васильчук А.С., студент IV курсу, гр. КІ-1
Науковий керівник – Оринчак І.А., ст. викл. кафедри КІ та КБ
Житомирський державний технологічний університет

ЗБИРАЄМО СУЧАСНИЙ FRONTEND НА ДОПОМОГОЮ WEBPACK

Програми, написані на JavaScript, постійно ускладнюються, тому виходом з цієї ситуації є використання автоматичного збирача (або Бандлера). Webpack – це статичний модульний збирач для додатків на JavaScript. Подібні інструменти дозволяють розробникам упаковувати, компілювати і, в цілому, організовувати всі ресурси, необхідні для проекту. Можна використовувати не тільки сторонні бібліотеки, а й власні файли. Подібна модульна система дозволяє домогтися кращої організації проекту, тому що він розбитий на невеликі модулі.

Webpack на даний момент є одним з найпотужніших подібних інструментів. Він має відкритий вихідний код і дозволяє вирішувати безліч різних задач.

Офіційний сайт проекту: https://webpack.js.org

Як і інші інструменти розробника, Webpack має свої плюси і мінуси.

Плюси: він відмінно підходить для роботи з односторінковими додатками (SPA – Single Page Application). Також Webpack може здійснювати поділ коду. Через ці та інші плюси зараз він є одним з найбільш популярних інструментів JS-розробки.

Мінуси: спочатку трохи складно розібратися в його роботі, частина документації застаріла через велику кількість змін у вихідному коді.

Webpack можна легко встановити за допомогою менеджера пакетів:

npm install --save-dev webpack

Це дуже гнучкий в налаштуванні інструмент. Для того, щоб почати працювати з ним, необхідно познайомитися з чотирма базовими поняттями.

- Entry – вхід;
- Output – вихід;
- Loaders – завантажувачі;
- Plugins – плагіни.

Під Entry (вхід) мається на увазі точка входу (entry point), яку Webpack буде використовувати для побудови внутрішнього графа залежностей. Після введення точки входу Webpack зможе зрозуміти, які модулі і бібліотеки безпосередньо і не безпосередньо зв'язуються. В результаті кожна залежність перетворюється в файли, які називаються бандл (bundles – з англійської можна перевести як „пакети" або „вузли").

Output (вихід) вказує, де Webpack повинен розміщувати створені бандли, і як він буде називати ці файли (за замовчуванням це ./dist). Налаштувати цю частину процесу можна в полі output в конфігурації.

Завантажники (Loaders) дозволяють Webpack'у обробляти не тільки файли JavaScript, тому що сам по собі Webpack розуміє тільки JS. Завантажники трансформують всі типи файлів в модулі, які потім можна додати в граф залежностей вашого додатку (а значить, і в бандл).

Використання завантажувачів має дві мети:

- властивість test визначає, який файл / файли повинні бути трансформовані;
- властивість use вказує, який завантажувач повинен використовуватися для виконання трансформації.

Правила (rules) визначаються для певного модуля і використовують властивості test і use. Компілятор Webpack'а зрозуміє, що коли він дійде до файлу, який вказаний у блоці „rules", після властивості „test:" перед додаванням в бандл йому потрібно буде використовувати те, що буде записано у коді після властивості „use:".

Плагіни (Plugins) – дозволяють виконувати досить широкий спектр функцій. Якщо завантажувачі використовуються для трансформації певних типів модулів, то плагіни можуть бути використані для виконання набагато більш широкого списку задач.

Для того, щоб використовувати плагін, необхідно використовувати require() і додати його в масив плагінів plugins:[]. Більшість плагінів можна налаштовувати. Так як один плагін може використовуватися кілька разів для різних цілей, необхідно створити кілька окремих екземплярів, використавши оператор new.

Багато плагінів поставляються відразу з Webpack'ом з коробки.

Враховуючи все вищевказане, можна з впевненістю говорити, що Webpack – корисний перспективний програмний продукт, гнучкий до різноманітних налаштувань, який суттєво спрощує розробку сучасного frontend'у.

УДК 004.588

Ващенко В.Д., магістр, гр. ПІ-49м
Науковий керівник – Шатківський В.М., ст. викл. каф. ІПЗ
Житомирський державний технологічний університет

РЕАЛІЗАЦІЯ ТА ПЕРЕВАГИ СИСТЕМИ ОЦІНЮВАННЯ ДІЙ ВОДІЯ У СИМУЛЯТОРІ КЕРУВАННЯ ТРОЛЕЙБУСОМ

Симуляція – це імітація певної речі, ситуації чи процесу, отже є досить велике різноманіття симуляторів і використовуються вони з різною метою:

- тренування;
- навчання персоналу;
- тестування певної технології;
- для розважальних цілей.

Зазвичай, симулятори застосовують тоді, коли взаємодія з реальним об'єктом зв'язана з недосяжністю, небезпекою чи високою вартістю такої взаємодії. Симулятор тролейбуса, про який далі піде мова, застосовується саме з метою роботи з транспортом віртуальним чином, адже небезпечно саджати недосвідченого водія за кермо автомобіля такого класу і довіряти йому життя та здоров'я не тільки пасажирів, а й усіх учасників дорожнього руху.

Система симуляції реалізує функціонал управління тролейбусом та моделювання реальних подій, що зв'язані з керуванням даним транспортним засобом. Симуляція забезпечується наборами скриптів.

Для того, аби визначити компетентність водія у симулятор було імплементовано підсистему аналізу дій користувача. Базуючись на існуючих даних, дії користувача було класифіковано і виділено чотири типи: нейтральна дія, позитивна дія, негативна дія, попередження.

Нейтральна дія і попередження ніяк не впливають на рейтинг водія і не сприймаються аналізатором. До нейтральних дій відносяться увімкнення тумблерів, відкриття дверей і тому подібне. Серед попереджень можна виділити незначне перевищення швидкості в рамках правил дорожнього руху.

Позитивна і негативна дія впливає на рейтинг водія і в подальшому аналізується. До позитивних дій відноситься, наприклад, правильне паркування на зупинці. Негативні – перевищення швидкості, рух з відкритими дверима та інші дії водія, які можуть призвести до небажаних наслідків.

Для отримання кінцевої оцінки водія було введено шкалу оцінювання. Кінцевий бал є числом від 0 до 100. Початкове значення – 80 балів. По ходу оцінювання, основуючись на подіях, що трапляються, до початкового значення додаються або віднімаються бали. За кожну позитивну дію до рейтингу додається два бали. За кожну негативну віднімається п'ять балів. Кінцева шкала оцінювання наведена у табл. 1. Відмінним результатом вважається значення більше за 90 балів, добрим – від 70 до 89. Якщо значення кінцевої оцінки менше за 70 – результат є незадовільним.

Шкала оцінювання водія Таблиця 1

Оцінка	0-69	70-79	80-89	90-100
Буквена оцінка	F (Незадовільно)	C (добре)	B (добре)	A (відмінно)

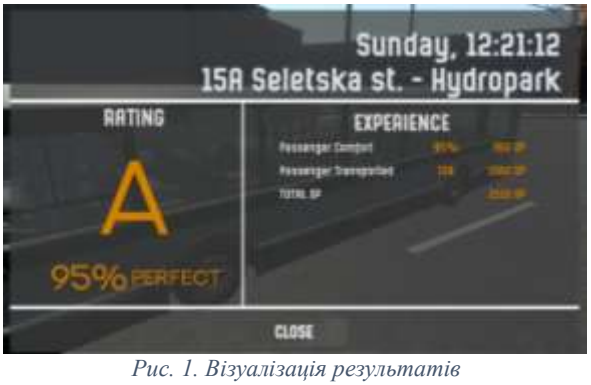


Рис. 1. Візуалізація результатів

Для відображення подій було додано спливаючі повідомлення, що контролюються відповідними скриптами, які відповідають за загальний стан спливаючих повідомлень та контроль конкретного повідомлення у черзі.

Для більш детальної текстової звітності було створено скрипт, що відстежує показ спливаючих повідомлень і записує їх у файл логу. Кожен файл іменується відповідно до дати та часу початку оцінювання та містить у вигляді списку всі повідомлення, які отримав водій під час руху. У шапці визначено версію симулятора та дату й час початку логування. Після чого йдуть записи з часом, типом, оцінкою та текстом повідомлень.

Таким чином, використання модулю оцінювання дій водія у симуляторі керування тролейбусом надає широкі можливості екзаменувати гравців та отримувати детальну інформацію про характер їх водіння та прогнозувати перспективи розвитку їх професійних навичок.

УДК 004.42

Візгалов О.Ю., студент, гр. ПІ-51
Науковий керівник – Марчук Г.В., ст. викл. каф. ІПЗ
Житомирський державний технологічний університет

АНАЛІЗ SPA У РЕАЛІЯХ СУЧАСНОЇ ВЕБ-РОЗРОБКИ

У наш час веб-технології набули стрімкого розвитку, оскільки на даний момент вони дають змогу задовольнити багато потреб та послуг у різних сферах нашого життя. Наразі важко навіть уявити звичайний день без використання браузеру: погода, новини, довідники, покупки або розважальні сервіси – все це інформація, яку сьогодні можна отримати у швидкий та зручний спосіб. Тому питання дістати потрібні нам дані у максимально комфортному та доступному вигляді стоїть досить гостро.

З моменту розробки першого браузеру – «WorldWideWeb», який був здатний відображати лише прості HTML-сторінки, пройшло чимало часу, за який винайдено чимало технологій та підходів. Проте досить довгий період розвитку веб-технологій домінували так звані MPA (Multip Page Application), що в перекладі – багатосторінкові додатки, які використовували для відображення статичні HTML-сторінки, які згодом насичились каскадними таблицями стилів (CSS) та деякою динамікою, переважно в межах сторінки, засобами мови JavaScript. Проте основною особливістю таких додатків було те, що клієнт, при навігації по розділам або виконанні певної функціональної взаємодії з додатком, виконував запит до серверу на повне перезавантаження сторінки та отримання нової HTML-структури, що можна порівняти з перегортанням сторінок звичайної паперової книги. Варто зауважити, що такий принцип роботи і у наш час досі залишається у рамках звичного нам поняття «веб-сайт». Недоліком даного підходу є те, що при необхідності незначної зміни частини сторінки, весь контент має бути перезавантажений знову (рис.1), навіть якщо він не змінився. Це створює додаткове навантаження на сервер та збільшує потік даних між клієнтом та сервером, тим самим погіршуючи досвід користування веб-додатком, оскільки сторінка «зникає» на час перезавантаження, а також страждає швидкодія.

Рис.1. Принцип роботи багатосторінкового додатку

Першим кроком для зміни ситуації стало введення тегу <iframe> у 1996 році, який давав змогу асинхронно завантажувати контент іншого документу. Пізніше, у 1999 розробники Microsoft розробили концепт для продукту Microsoft Exchange Server 2000, що представляв собою інтерфейс, названий *IXMLHTTPRequest*, який ліг в основу *XMLHttpRequest* (XHR) об'єктів. Згодом Mozilla розробила власний інтерфейс *nsIXMLHttpRequest* для браузерного рушія Gecko, який був максимально подібний до інтерфейсу розробленого Microsoft. Нарешті у 2002 році Mozilla випустила стабільну версію Gecko, який включав реалізовану обгортку над інтерфейсом *nsIXMLHttpRequest,* що працювала у середовищі мови JavaScript. Дану реалізацію інтерфейсу вони і назвали *XMLHttpRequest*. Хоча сам стандарт був затверджений лише у квітні 2006 року, даний підхід дозволив реалізувати асинхронне підвантаження та модифікацію контенту для вже завантаженої сторінки, що у свою чергу лягло в концепцію Ajax (asynchronous JavaScript and XML) дизайну. Це дало значний поштовх для розвитку веб-технологій у напрямку SPA (Single page application) – односторінкових додатків. Почали з'являтися бібліотеки, що дозволяли працювати з XHR функціоналом без прямої взаємодії з його API. Найвідомішою з яких стала jQuery. Також не можна не згадати такі бібліотеки як Knockout.js та Backbone.js, проте першим дійсно повноцінним і комплексним рішенням для SPA додатків став фреймворк AngularJS, побудований на основі клієнтської Model-View-Controller (MVC) архітектурі з підтримкою шаблонів, двосторонньою трансформацією даних та можливістю впровадження залежностей.

Як виявилося, SPA додатки отримали значну кількість переваг порівняно з звичайними багатосторінковими веб-сайтами. По-перше, односторінкові веб-сайти в процесі користування дають враження нативних додатків незалежно від платформи, на якій вони використовуються. По-друге, перевагою є зменшення трафіку даних між сервером та клієнтом при користуванні SPA у порівнянні з MPA, оскільки в процесі взаємодії користувач отримує лише необхідні дані, які оновлюють тільки окремі частини сторінки, замість повного її перезавантаження (рис.2.). З даного моменту випливає наступна,

безперечна перевага – швидкодія. Це мабуть саме те, що зробило SPA настільки популярним та актуальним на сьогодні. SPA взаємодіє з користувачем як звичайний додаток для персонального комп'ютеру, даючи можливість переглядати старий контент до моменту оновлення та динамічно реагувати на взаємодію викликану клієнтом. Це досягається і завдяки такому підходу як «reactive programming», в основі якого принцип того, що програма повинна визначати дії, які необхідно виконати у відповідь на певну подію, замість того, щоб явно казати системі, що робити у даний момент часу.

Рис.2. Принцип роботи односторінкового додатку

Аналізуючи сам процес розробки односторінкових додатків, неможливо не наголосити увагу на тому, що SPA містить дуже чітке розділення між кодом серверної та клієнтської частини. Зазвичай, клієнт спілкується з сервером через розроблений API, в якій покладено певні контракти. Це дозволяє відділяти back-end від front-end'у та замінювати кожну частину незалежно. Також принципи SPA дозволяють чітко розділити частину відображення (view layer) від даних (model layer), що у багатьох випадках дозволяє обмінюватися з сервером лише даними через модель, не зачіпаючи при цьому рівень відображення, що в свою чергу зменшує навантаження від передачі HTML-розмітки, стилів тощо. В той час як на MPA серверний та клієнтський код тісно зв'язані, що викликає складнощі при окремій модернізації кожної з компонент (наприклад, у випадку зі створенням окремого додатку для мобільних пристроїв SPA дозволяє вільно використати ту саму серверну частину, в той час як MPA у більшості випадках буде потребувати окремої серверної реалізації).

Незважаючи на велику кількість переваг односторінкових додатків над багатосторінковими, залишається все ще немала кількість недоліків та завдань, які необхідно вирішити при розробці SPA. Особливо гостро стоїть питання з підтримкою старих або деяких мобільних браузерів, оскільки SPA переважно використовують новітні стандарти JavaScript, що може викликати нестабільність або повну відмову роботи браузера з додатком такого типу. Оскільки MPA мають набагато слабшу залежність від JavaScript або можуть обмежено функціонувати навіть у браузері з відключеним JavaScript, це дає змогу підтримувати ширший набір клієнтів. Проте існують такі рішення як Babel.js, які дають можливість зворотної підтримки старих стандартів та браузерів для односторінкових додатків. Також до недоліків SPA можна віднести набагато складніший та довший процес розробки, оскільки дуже важливо розробити розширювану архітектуру з коректним розділенням відповідальності між усіма її компонентами. Варто зауважити і про складність застосування підходів SEO (Search engine optimization) з SPA додатками. Звичайно, на даний момент існує не один інструмент для спрощення даної задачі, проте важко сперечатися з тим, що налаштування взаємодії пошукових систем з багатосторінковими додатками простіша та більш стандартизована.

На сьогодні існує значна кількість фреймворків для реалізації односторінкових додатків – SPA, а саме: AngularJS, Vue.js, React, які можна поєднувати з різними серверними технологіями, реалізованими за допомогою різних мов програмування, таких як PHP, C#, Python тощо. Даний вибір дозволяє обирати розробникам підхід, який найбільш зручний для реалізації поставленої перед ними задачі. Наразі SPA перебуває у стадії постійного розвитку та вдосконалення, що говорить про безперечну важливість та актуальність даної теми у реаліях сучасної веб-розробки.

Для виконання випускної роботи буде використана бібліотека ReactJS, що дозволить досить зручно будувати клієнтську частину SPA-додатку, не прив'язуючись до стеку технологій розробника. Також дана бібліотека має документацію високого рівня та сильну спільноту, яка вже має багато готових рішень для реалізації різного роду функціоналу та вирішення загальних проблем при розробці SPA-додатку. Чітко відділяючи дані у вигляді стану від відображення (DOM структури), React має змогу продуктивно оновлювати саме ті компоненти, для яких змінилися дані, для чого також модна використати бібліотеку Redux. В свою чергу, для реалізації серверної частини буде використаний фреймворк ASP.NET Core.

Отже, наразі SPA перебуває у стадії постійного розвитку та вдосконалення, що говорить про безперечну важливість та актуальність даної теми у реаліях сучасної веб-розробки.

УДК 004.4:378

Дідевич К.С., студентка, ІІІ курс, гр. СМ-16-1, ФАМІТ
Науковий керівник – к.т.н., доц. Мельников О.Ю.
Донбаська державна машинобудівна академія, м. Краматорськ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТИ З ОСВІТНІМИ ПРОГРАМАМИ Й СТАНДАРТАМИ ВИЩОЇ ОСВІТИ

Стандарт вищої освіти – це сукупність норм, які встановлюють основну мету й завдання освіти, вимоги до змісту освіти, обсягу й рівню підготовки фахівців, що визначають спосіб діагностики якості вищої освіти.

Згідно прийнятим Міністерством освіти й науки України правилам, формат «Стандарту вищого утворення» містить два види компетенцій (загальні і спеціальні), нормативний зміст у вигляді переліку знань й умінь, а також два додатки: матрицю відповідності дескрипторів НРК (знання – уміння – комунікація – відповідальність) кожної компетенції й матрицю відповідності програмних результатів навчання компетентностям.

Аналіз доступних джерел інформації показав, що на цей час немає застосувань, що дозволяють комплексно вирішувати завдання, пов'язані з обробкою освітніх стандартів. Було сформовано завдання створення програмної системи, що дозволяла б працювати зі списком формованих компетенцій і по предметах, і програмним результатам навчання. Система повинна надавати можливість імпортувати всі наявні дані, вносити зміни в будь-який розділ і працювати з даними XLS-формату.

Реалізоване програмне забезпечення надає можливість працювати з таблицями компетентностей і результатів навчання. Таблиця компетентностей розділена на два типи: спеціальні (професійні) і загальні, також є можливість приховати одну або обидві компетенції для загальної зручності при використанні таблиць. Також програма передбачає можливість окремого перегляду таблиць. Приклади роботи наведені на рис. 1-3. Застосування полегшить роботу з навчальними документами й призначене для використання на кафедрах при підготовці матеріалів до ліцензування й акредитації.

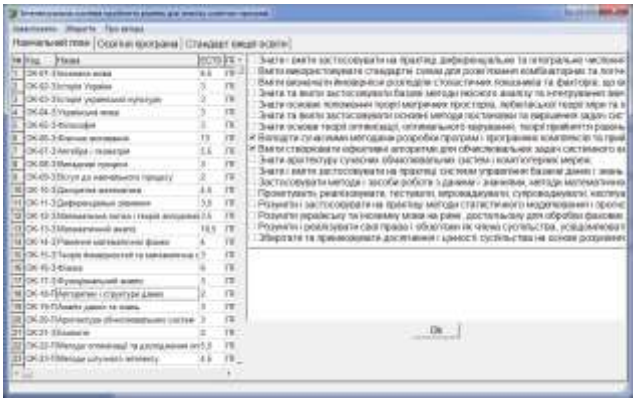


Рисунок 1 – Робота з навчальним планом

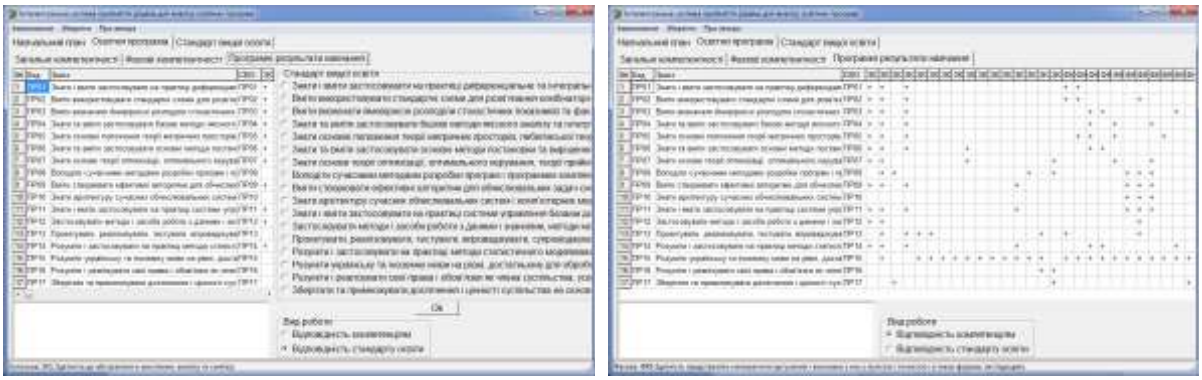


Рисунок 2 – Робота з освітньою програмою (відповідність стандарту й компетенціям)

УДК 004.424

Докійчук М.О., магістрант, гр. ПІ-50м
Науковий керівник – Левківський В.Л., аспір. ФІКТ
Гришкун Є.О., ст. викл. каф. ІПЗ
Житомирський державний технологічний університет

ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ СЛУЖБИ ДОСТАВКИ ТОВАРІВ

На сьогодні автоматизація роботи служби доставки товарів є досить актуальною. Це пов'язано з активним розвитком комерційних підприємств. Найважливішим призначенням подібних служб вважається швидка доставка документів або посилок. Також відправник та отримувач очікують якісну передачу та збереження переданого.

Термінова доставка грошових коштів, цінних паперів або посилок - саме ці послуги користуються попитом найбільше. Адже саме подібні пересилання здійснюються за лічені години і спрощують життя навіть багатьом бізнес-компаніям.

Для вирішення даної проблеми розробляється автоматизована система служби доставки, для отримання максимальної ефективності та швидкості в послугах даної сфери діяльності. Дана система надає користувачам зручно та легко відслідковувати відправлення не залежно від їх статусу, отримувача чи відправника, а саме:

- відслідковування місця знаходження вантажу;
- оптимальність вирішення проблеми доставки вантажу;
- оптимізація місця знаходження об'єктів (відділень);
- спілкування з службою підтримки в разі виникнення питань;
- ознайомлення з детальною інформацією на сайті, такою як: новини, контакти, місцезнаходження, час роботи служб доставки та загальна інформація про сервіс.

Дана система вносить зручність в користуванні системами доставки та контролювання власних різних вантажів, таких як: цінні папери, грошові перекази, малогабаритні вантажі та інші різноманітні посилки.

Вся система представляє собою веб-сайт розроблений за допомогою фреймворку laravel на базі мови програмування PHP.

На даний період часу це одна з найперших розробок яка базується на цьому фреймворку. Складність поєднання програмних систем заключалась в інтеграції API функціоналу з різними платформами, а саме - налаштування та створення серверної частини та запитів для пошуку. Для реалізації алгоритму пошуку були створені ajax-запити з інших серверів.

Також для інформації знаходження відділень на карті використовувались алгоритми гугл карт, інтегроване API та налаштування під сервер для використання інформації на інших платформах (інтеграції API).

В програмі реалізовано розмежування ролей для різних користувачів, які мають відмінний функціонал. Особливість різних ролей залежать від потреб працівників, таким чином оператор має доступ до одного з найголовнішого функціоналу (створення тікету відправлення вантажу).

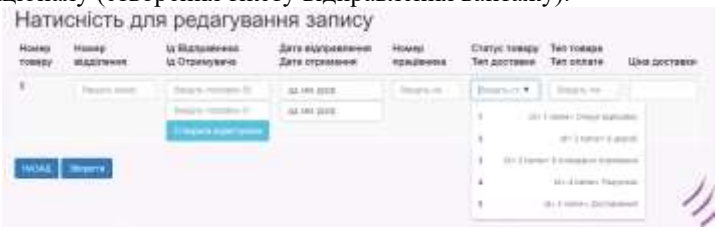


Рис. 1. Створення тікету відправки вантажу

Допоміжні технології для вирішення проблематики розробки сайту: HTML5, CSS3, PHP, MySQL, JavaScript, JQuery, Twitter Bootstrap, owl carousel, fancybox, та інші. Система базується на основі шаблону проектування MVC (Model View Controller). Програма підтримує розробку власного API для підключення функціоналу на сторонні сервіси, для прикладу підключення функції пошуку товару (рис. 2.)

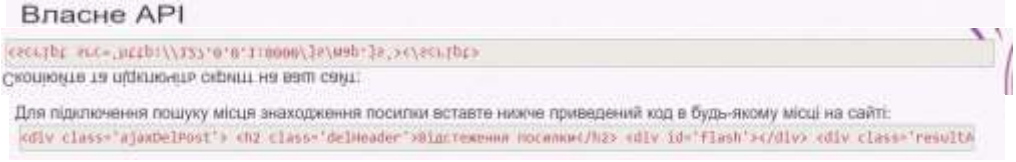


Рис. 2. Приклад підключення API до стороннього сайту

Одним з найважливіших алгоритмом програми є розподіл користувачів на ролі, адже в базі служби доставки потірбно автомно розприділяти обов'язки та права користувачів. На разі існують такі панелі доступів:

- Панель адміністратора;
- Панель Оператора;
- Панель Кур'єра.

Для прикладу (на рис. 3.) відобрежена адмін панель з переходом на різні функції керування.

Рис. 3. Функції адмін панелі

Таким чином дана роль користувача дозволяє керувати всіма записами сайту наведеними в списку.

Алгоритм пошуку оптимального вирішення доставки вантажу.

По суті алгоритм пошуку шляху досліджує граф, починаючи з однієї вершини та переходить до сусідніх вузлів і так доки не досягне цільового вузла, як правило з наміром знайти найдешевший (найкоротший) маршрут.

Основні алгоритми, такі як пошук у ширину і пошук у глибину, спрямовані на першу проблему, вичерпуючи всі можливості за допомогою грубого перебору; починаючи з даного вузла, вони ітераційно досліджують усі потенційні шляхи, доки не досягають цільового вузла. Більш складною проблемою є пошук оптимального шляху.

Загальним прикладом алгоритму пошуку шляху базованого на графі є алгоритм Дейкстри.

Псевдокод:

```
прочитати g
   d = g d[0] = 0
   mark[0] = True
   for i = 1 ... n - 1
      mark[i] = False
   for i = 1 ... n - 1
      v = -1
   for i = 0 ... n - 1
      if (not mark[i]) and ((v == -1) or (d[v] > d[i]))
         v = i
         mark[v] = True
   for i = 0 ... n - 1
   if d[i] > d[v] + g[v][i]
      d[i] = d[v] + g[v][i] вивести d
```

Алгоритм працює за даною схемою:

У гіпотетичній ситуації коли вузли A, B і C утворюють зв'язаний неорієнтований граф з ребрами AB = 3, AC = 4, та BC = -2, оптимальний шлях від A до C коштує 1, а оптимальний шлях від A до B витратить 2. Алгоритм Дейкстри починає з A, а потім спочатку розглядає вузол B, оскільки він розташований найближче. Цей шлях буде коштувати 3 і буде позначений як «закритий», а значить, його вартість ніколи не буде переоцінена.

Оптимізація місцязнаходження об'єктів.

Для оптимізації місцязнаходження об'єктів буде використовуватись мова sql. Алгоритм оптимізації буде полягати в таких варіантах:

1. Якщо кількість посилок на пошті перевищує заплановану кількість за місяць, тоді будуть надані відповідні рекомендації.
2. Якщо кількість посилок на пошті має оптимальну кількість запланованих, тоді будуть надані відповідні рекомендації.
3. Якщо кількість посилок на пошті менше ніж запланована кількість за місяць, тоді будуть надані відповідні рекомендації.

З цих фактів можна зробити висновок, що програма повинна містити різні типи модулів для легкої та автоматизованої взаємодії користувача з веб-сайтом та особливу увагу потрібно приділити оформленню веб-сайту для зручної взаємодії з ним, також впровадження модулів для відслідковування вантажів та знаходження найближчих відділень на карті, власне API для можливості підключення віджетів на інші ресурси.

УДК 004.42

Іголкін С.С., магістр, гр. ПІ-50м
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

РОЗГОРТАЄМИЙ КОМПЛЕКС ДОПОМІЖНИХ МОДУЛІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ПОБУДОВИ СКЛАДНИХ ІГРОВИХ СИСТЕМ У 3D ПРОСТОРІ

При побудові ігрової системи у 3D просторі існує коло задач, які найчастіше доводиться вирішувати незалежно від кінцевої ідеї, наприклад переміщення ігрового персонажу у просторі, здатність пошкоджувати певні ігрові об'єкти, або ж необхідність зв'язати між собою дію та її наслідки. З цього випливає проблема: виконується робота, яка витрачає час розробників продукту, який може бути витрачений на більш розгорнуту і довершену реалізацію ігрової системи. Рішенням цієї проблеми є створення комплексу допоміжних модулів, який вирішує базові задачі і дозволяє розробнику при менших витратах часу модифікувати, додавати або видаляти принципи, на яких базується його ігрова система.

Розробка ігрових систем – це процесс створення ігрових механік, явних або неявних, кінцева ціль яких створити єдиний ігровий процесс, завдяки якому гравець зможе отримати певний ігровий досвід. Розгортаємий комплекс допоміжних модулів – це комплекс, що дозволяє вирішити задачі з проєктування найпростіших базових механік і в подальшому значно полегшити процесс розробки ігрової системи за рахунок наявності у комплексі базових систем явних та неявних механік. Переваги розгортаємо комплексу у порівнянні зі створенням базових механік з нуля: наявність незалежних систем, які можуть бути модифіковані під цільову задачу, наявність зв'язку з різними частинами комплексу, реалізація складних систем, таких як сценарій рівня, система накладаємих єффектів. Переваги розгортаємого комплексу модулів у порівнянні з готовими рішеннямя (фреймворками): лояльність до модифікації, мінімальна реалізація для розширення можливостей ігрових систем, узгодженність з зовнішніми системами, відсутність глибоких закритих мезанізмів роботи, цілеспрямованість на певний вид діяльності замість універсальності.

Основними задачами розгортає мого комплексу допоміжних модулів є:

– зменшити час та витрати на розробку ігрових систем;
– створити кістяк для реалізації певних ігрових систем;
– налагодити стабільний процес розширення і доповнення ігрової системи без необхідності глобальних змін у системі;
– вирішувати задачі з узгодження систем між собою для створення цілісної системи;
– за необхідності – змінюватися у незначних деталях, зберігаючи решту кістяку та зберігаючи цілісну архітектуру.

Передбачувані результати роботи розгортаємого комплексу модулів: швидке прийняття рішень, зменшення часу необхідного для вирішення базових питань що до проєктування, архітектури, розширення ігрової системи, можливість створення більш продуманої та більш детальної ігрової системи, і, як наслідок, збільшення прибутку за рахунок більш позитивного враження про кінцевий продукт.

Для подібних систем є актуальними дві наступні проблеми: наявність узгоджуючої системи сценарію, що керує процесами ігрової системи та наявність спільної «нервової системи», що дозволяє навіть окремо від складної сценарної системи узгоджувати між собою різні за своєю структурою та призначенням модулі. На рис. 1 можна бачити основну структуру «нервової системи» розгортаємого комплексу модулів:

Рис. 1. Нервова система комплексу

«Нервова система» комплексу модулів базується на двох ключових частинах.

Перша - це так звані «збуджувачі» нервової системи, які призначенні для початкового старту сигналу, що передається до головного керуючого модулю системи. Збуджувач є шляхом для системи дізнатися про певні подіїї у самій системі, як то знаходження занадто близько до певної позиції, проходження скрізь певну зону. Збуджувачі базуються на допоміжному проміжному модулю, що дозволяє нам створюваті різні реалізації збуджувачів.

Проміжний модуль також містіть у собі ідентифікор збуджуванного сигналу, який дозвляє у подальшому рецепторам розпізнати цей сигнал. Механізми збудження можуть бути різними, як було сказано раніше, починаючи від найпростіших на дистанцію і закінчуючи комплексними системами, що слідкую за декількома критєріями взаємодії одразу, як то, наближення до цілі, знаходження у радіусі дії та віддалення від радіусу дії.

Друга частина представляє собою рецептори збуджень. Вони базуються на підключенні до центрального керуючого модулю і відстеженні активності нервової системи. У випадку, якщо ідентифікатор сигналу нервової системи співпадає з ідентифікатором рецептору, рецептор викликає сигнальні механізми, що дозволяють повідомити усіх, хто під'єднаний до нього, про те що цей рецептор був збудженний.

Сценарна система представляє собою комбінацію рецепторів нервової системи та дій, які мають бути виконані. А саме: певна кількість дій, одна або більше, об'єднані у єдиний блок. Блок має рецептори, один або декілька, які сприймають сигнали від нервової системи. Після того, як усі репептори перейдуть до стану «опрацьовані», блок впровадить дії, які були приєднані до нього. На рис. 2 зображена структура сценарію:

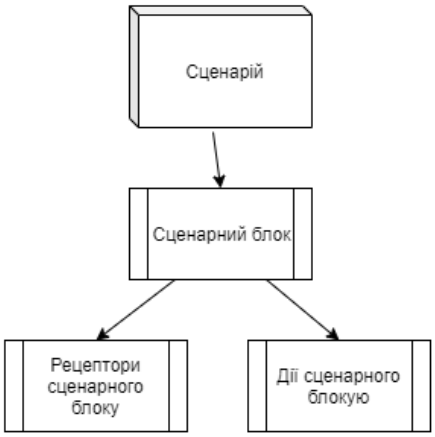


Рис. 2. Структура сценарію

Основою розгортаємого комплекса допоміжних модулів є «Ін'єкція залежностей». Завдяки ін'єкції залежностей система може отримувати доступ до необхідних модулів через використання базових типів або інтерфейсів. Окрім цього, система дозволяє перевести об'єкт до більш абстрактної форми існування. Не як фіксована структура, а як елемент глобальної бази данних, що дозволяє бачити об'єкт з різних перспектив.

Таким чином, об'єкти ігрового простору можуть бути, за необхідності, використані з різних аспектів, таких як: 3D модель об'єкта, спорядження, інвентар. Таким чином надається можливість, наприклад, зберігати окремо візуальний ефект і легко їх замінювати.

Також при побудові розгортаємого комплексу модулів слід пам'ятати про те, що у подальшому кожна з систем має бути здатна до модифікації, якщо модифікація передбачається цією системою. Так, наприклад, усі спеальні дії персонажу відносяться до модулю дій і мають базовий клас з реалізацією керування. У подальшому в основі цього классу можуть бути створені класи дій, які будуть оброблені системою.

Слід зауважити також, що розгортаємий комплекс має бути гнучким, оскільки базується на принципі «розгортання». Це означає, що під час розробки деякі властивості системи можуть бути змінені або ж видалені, оскільки вони не потрібні для розробки данної ігрової системи.

Незважаючи на те, що комплекс має значно більш високі вимоги до розробника програмного забезпечення для свого використання, як результат вдається уникнути ситуації, коли готове рішення у фреймворку іде з великою кількістю коду, який може бути скомпільованим і, окрім того, закритий для змін.

УДК 004.42

Іщенко А.В., магістр, гр. ЗПІ-18-2м
Науковий керівник – Єфіменко А.А., к.т.н.,
зав. каф. комп'ютерної інженерії та кібербезпеки
Житомирський державний технологічний університет

ВИКОРИСТАННЯ СЕРВІСУ FIREBASE ДЛЯ РОЗРОБКИ СУЧАСНИХ WEB ДОДАТКІВ

Років 10 тому створення веб-додатку, що працює в реальному часі, було непосильним завданням для багатьох програмістів.В наш час коли кожного дня з'являться по декілька нових фреймворків інколи стає дуже складно обрати технології на яких виконувати розробку web-додатків. А якщо вам все таки вдалось запустити та розробити хоча б мінімальний функціонал постає досить правильне питання, а як це все розгортати і запускати на серверах.

Тепер же з появою Firebaseвід досить знайомої нам компанії Google, будь-хто може почати розробку свого веб-додатку, не турбуючись про серверну частини проекту.Команда розробників Firebase останні роки провела низку вдосконалень сервісу та додала багато нових опцій для більш швидкої і продуктивної роботи з сервісом. Давайте подивмось на що спроможний Firebasei які інструменти він надає розробникам:

– FirebaseAuthentication -сервіс авторизації, дозволяєавторизувати користувачів без додавання серверного коду використовуючи такі популярні способи як: авторизація за номером телефону, анонімний вхід, стандартний (логін/пароль) а також за допомогою сторонніх сервісів (Facebook, Twitter, Google, GitHub). Дозволяє створювати захищені та вже звичні для користувачів способи авторизації.

– FirebaseCloudFirestore taFirebaseRealtime Database - бази даних по суті представляють собою NoSQL бази що являють собою JSON-дерева в Firebase їх масштабувати набагато простіше, ніж традиційні SQL-структури. Між собою мають невеликі відмінності: FirebaseRealtime Database більш проста по структурі та призначена для швидкого доступу та частого перезапису даних, Firebase Cloud Firestore більш просунута в плані архітектури може включати вкладені колекції в документи. Обидві бази мають простий інтерфейс доступу і при підключенні до сторінки створюють підключення через протоколWebsocket.

– FirebaseStorage - сховище файлів, дозволяє організовувати завантаження і зберігання файлів (зображення, відео, аудіо та ін.).

– FirebaseHosting-хостинг призначений для розміщення файлів сайту на серверах Firebase.Особливо зручно використовувати на невеликих проектах або в якості демо презентацій майбутніх стартапів.

– FirebaseFunctions-дозволяє виконувати серверний код в спеціальних функціях які можуть працювати з даними, а також можуть взаємодіяти з іншими підсистемами які перераховані для роботи в середовищі інфраструктури Firebase.

– Firebase ML Kit – використання машинного навчання для виконання типових задач в web-додатках. ML Kit поставляється з набором готових до використання API для звичайних випадків мобільного використання: розпізнавання тексту, виявлення обличь, сканування штрих-кодів, маркування зображень і розпізнавання орієнтирів.

– FirebaseNotifications - новий інтерфейс, побудований на базі API FirebaseCloudMessaging, дозволяє надсилати повідомлення користувачам в якості нотифікацій також можна обирати налаштування для розсилки для спеціальних груп користувачів створених на основі аналітичних даних FirebaseAnalytics.

– FirebaseDynamicLinks - покращує роботу з посиланнями.

– FIrebaseCrashReporting збирає і відсилає вам найважливішу інформацію, яка може допомогти в пошуку проблем iOS / Android-додатків після релізу.

– FirebaseTestLab - хмарне тестування додатків на реальних девайсах, які розташовані в дата-центрах, використовуються для тестування на мобільних платформах.

– FirebaseAnalytics - це інструмент для аналізу мобільних додатків. Частково він схожий на GoogleAnalytics, включає в себе цілу низку підсистем для аналізу і роботи з масивами данних що надходять від користувачів.

Отже підсумовуючи ми маємо потужний сервіс для розробки сучасних web-додатків. Використовуючи наявний інструментарій стає можливим знаючи один з фреймворків таких як (Angular, React, Vue)швидко розробляти та представляти замовнику прототипи рішень бізнес задач з гнучким налаштуванням який надає сервіс Firebase. Також одним з великих плюсів є підтримка розробки під платформи iOS та Android можливості тестування та широкий інструментарій по аналізу web-додатків з сторони маркетингу.

Все це створено для того, щоб менше турбуватися про серверної складової і більше уваги приділяти клієнтської частини проекту. Розробник пише функціонал майже не турбуючись про архітектуру додатку.

УДК 004.4:796

Кадацький М.А., студент, III курс, гр. СМ-16-1, ФАМІТ
Науковий керівник – к.т.н., доц. Мельников О.Ю.
Донбаська державна машинобудівна академія, м. Краматорськ

ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ ДЛЯ ПРИБЛИЗНОГО ЗНАХОДЖЕННЯ ПОКАЗНИКІВ СПОРТСМЕНА-МЕТАЛЬНИКА ЯДРА

Сучасний рівень розвитку спорту ставить завдання розробки нових засобів і методів спортивної підготовки, які будуть сприяти швидкому й надійному досягненню високих результатів.

У фізичній культурі й спорті нейронні мережі використовуються для аналізу й прогнозування показників фізичної підготовки спортсменів, а також результатів спортивних змагань. Ефективність використання нейронних мереж забезпечується можливістю моделювання фізіологічних процесів в організмі людини, що носять нелінійний характер, а також здатністю нейронних мереж до самонавчання.

В одному виданні наводяться дані про характеристики восьми спортсменів (вік, ріст, маса тіла, віддає перевагу методу метання), а також їхні спортивні результати (початкова швидкість польоту ядра, кут метання, висота відриву від руки й відстань польоту).

З математичної точки зору можна сформулювати два завдання прогнозування:

– за наявним даними про вік, ріст, масу тіла атлета, а також характеристиках польоту ядра визначити дальність цього польоту;

– за наявним даними про вік, ріст, масу тіла атлета, а також дальності польоту ядра визначити оптимальне сполучення характеристиках польоту – початкової швидкості, куту й висоті відриву.

Обидва завдання можна вирішити методами штучних нейронних мереж. Як модель нейронної мережі доцільно вибрати двошаровий персептрон. Оцінка кращого числа нейронів у прихованому шарі проведемо, використовуючи відому нерівність. У першому випадку величину схованого шару приймемо рівним 3 нейронам, у другому – 5. Тип активаційної функції – сигмоїда. Навчання мережі проводиться методом зворотнього поширення помилок. Розрахунок був проведений у середовищі Deductor Studio.

Name	Technique	Age [years]	Height [m]	Weight [kg]	v0 [m/s]	a0 [°]	H0 [m]	Distance [m]
Cantwell	Rotational	28	1,98	140	14	37,8	2,29	22,03
Majewski	Glide	27	2,04	132	13,8	39,3	2,43	21,91
Bartels	Glide	31	1,87	135	14	33,6	2,12	21,37
Hoffa	Rotational	31	1,82	133	14	34,4	2,06	21,28
Nelson	Rotational	34	1,83	115	14,1	32,9	2,05	21,11
Lyzhin	Rotational	28	1,89	110	13,6	39,2	2,22	20,98
Mikhnevich	Glide	33	2,02	127	13,4	37,7	2,43	20,74
Vodovnik	Rotational	31	1,96	145	13,7	33,1	2,25	20,5

Рисунок 1 - Дані про атлетів

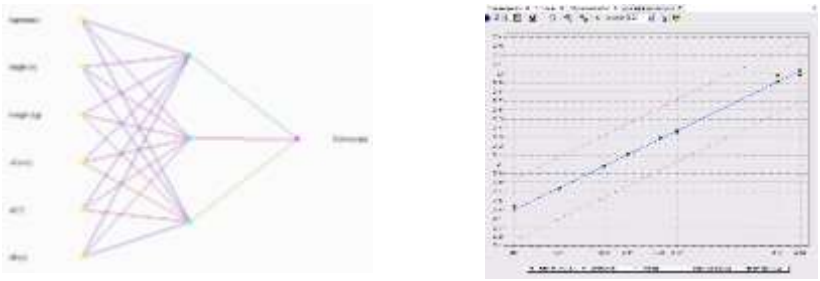


Рисунок 2 – Граф і діаграма розсіювання нейронної мережі MLP-6-3-1

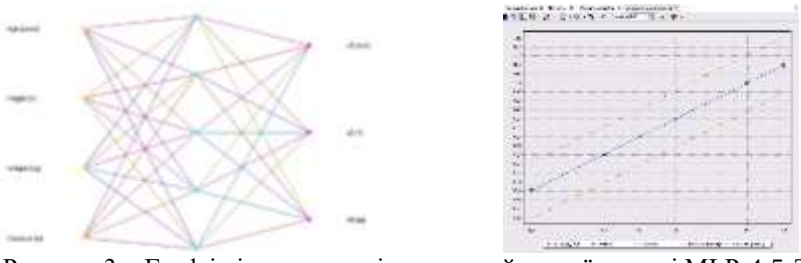


Рисунок 3 – Граф і діаграма розсіювання нейронної мережі MLP-4-5-3

УДК 004.42

Карпович О.Г., магістр, гр. ЗПІ-18-1м
Науковий керівник – Коротун О.В., к.пед.н, доц. каф. комп'ютерних наук
Житомирський державний технологічний університет

ОПИС ОСНОВНИХ ТИПІВ НЕСТРУКТУРОВАНИХ БАЗ ДАНИХ

У сучасному світі, в зв'язку зі стрімким розвитком мережі Інтернет, різко збільшується об'єм інформації яку необхідно обробляти за допомогою найбільш ефективних засобів, спеціалізованих на збереженні та обробці даних. Оскільки такі дані складно структурувати, доцільно використовувати нереляційні бази даних (БД).

Існує велика кількість нереляційних БД. Причина цього полягає в тому, що якась одна система не зможе задовольнити запити додатку набором своїх властивостей. Виокремлюють чотири основні типи неструктурованих БД: бази засновані на моделі "ключ-значення" – вони мають велику таблицю хеш-ключів і значень; документо-орієнтовані бази – вони зберігають документи, складені з елементів з тегами; бази на основі стовпців – кожен блок зберігання містить дані лише з одного стовпця; бази засновані на графовій моделі – використовують ребра і вузли для представлення та зберігання даних.

БД засновані на моделі "ключ-значення", що працюють з даними типу ключ-значення, наприклад, як словник. Тут немає місця ні структурі, ні зв'язків. Після підключення до сервера додаток може задати ключ і його значення, а як наслідок отримувати ці данні за запитом. Такі системи управління базами даних (СУБД) зазвичай використовуються для швидкого збереження базових даних, а іноді не таких вже й базових, якщо підрахувати витрати процесора і пам'яті. Вони, зазвичай, дуже швидкі, працездатні та легко масштабуються (добре використовувати такі БД для зберігання сесій, кеша, лічильників відвідувань або переглядів і т.д.). Представниками даного типу БД є Redis, Riak та MemcacheDB.

Документо-орієнтовані працюють так само як і попередні системи, але допускають набагато більшу вкладеність і складність структури даних (наприклад, документ вкладений в документ, вкладений в документ). Документи знімають обмеження вкладеності першого і другого рівнів типу ключ-значення в розподілених сховищах. В цілому, можна описати як завгодно складну структуру даних як документ і зберегти в такій БД. Незважаючи на досить великий функціонал і здатність доступу до даних по одному ключу, такі СУБД мають ряд своїх проблем. Наприклад, при доступі до одного документу користувач повністю отримуєте його у відповідь на запит, навіть якщо було необхідно тільки одне поле, що відповідно позначається на продуктивності. Представниками даного типу БД є CouchDB та MongoDB.

Бази на основі стовпців по своїй суті це наступний крок після СУБД типу ключ-значення. Такі БД відмінно працюють, створюючи колекції з одного або декількох пар ключ-значення, що в сумі відповідають запису. Відповідні СУБД не вимагають попереднього опису структури даних. Кожен запис складається з одного або декількох стовпців, містять дані, а кожен стовпець різних записів може зберігати різні типи даних. В цілому, розподілене сховище – це двовимірний масив, де кожен ключ (запис) містить одну або кілька пар ключ-значення прив'язаних до нього. Відповідна система дозволяє зберігати і використовувати великі обсяги неструктурованих даних. Представниками даного типу БД є Cassandra та HBase.

Бази засновані на графовій моделі зберігають дані в зовсім іншому вигляді. Вони використовують деревовидні структури з вузлами і зв'язками з'єднують їх. Аналогічно як в математиці, деякі операції набагато зручніше виконувати з такими даними завдяки зв'язкам між ними та їх угрупування. Такі БД часто використовуються в додатках, де потрібно мати чітко встановлені зв'язки. Наприклад, коли користувач входить у будь-яку соціальну мережу, зберігати зв'язки між ним та його друзями набагато простіше при застосуванні БД на основі графів. Представниками даного типу БД є OrientDB та Neo4J.

Отже основними перевагами неструктурованих баз даних над структурованими є:

– Швидкість – NoSQL бази даних зазвичай швидше, а іноді набагато швидше, коли справа доходить до запису. Операції читання також можуть бути досить швидкими в залежності від того яку саме БД необхідно використовувати;

– Автоматизована реплікація/масштабування – NoSQL бази даних швидко розвиваються – розробники активно вирішують основні проблеми, одна з яких – реплікація і масштабування. На відміну від реляційних СУБД, NoSQL рішення легко масштабуються і працюють з кластерами;

– Безсхемна розробка – реляційні СУБД вимагають чітко описану структуру даних до початку роботи. NoSQL рішення пропонують більш гнучкі рішення;

– Великий вибір – існує великий вибір різних моделей сховища NoSQL. Зауважимо, вплив на продуктивність розробленої системи має правильно зроблений вибір моделі сховища.

Кожний з розглянутих типів неструтурованих БД в порівнянні з іншими представниками має свої недоліки та переваги, що були описані, тому вибір певного типу БД залежить від потреб додатку, що розробляється для забезпечення його максимальної продуктивності.

УДК 004.42

Кльов В.О., магістр, гр. ПІ-49м
Науковий керівник – Єфремов Ю.М., к.т.н., доц. каф. ІПЗ
Житомирський державний технологічний університет

ПОШУКОВА СИСТЕМА ТА ДІЯЛЬНІСТЬ ПОШУКОВОГО РОБОТА

A web search engine and activity of a web crawler

Пошукова система – це спроба впорядкувати і класифікувати найрізноманітнішу інформацію на просторах Інтернету. Для користувача пошукова система виглядає, як звичайний сайт, в якому він вказує свій запит. Сайт реагує майже миттєвою відповіддю, у вигляді серпа, тобто створеною сторінкою сайту, що містить, в якості відповіді, послідовно розташовані посилання та інші елементи. Але те, що видно користувачу – "обличчя" пошукової системи – сайт, є лише інтелектуальною надбудовою величезного комплексу. Природно, вона має матеріальну базу, виражену в системі реальних машин, потужних апаратів, забезпечених спеціально написаним унікальним програмним забезпеченням. Незважаючи на те, що програми мають між собою багато спільного, їх індивідуальні особливості не підлягають розголошенню. І це обумовлює відмінність пошукових систем між собою. Крім того, пошукова система не є застиглим явищем, вона розвивається, розширюється, змінює алгоритми роботи.

Пошукова система володіє пошуковими роботами. Пошуковий робот – це найважливіший елемент пошукової системи, в завдання якого входить збір нових даних про сайти та їх оновлення. Пошуковий робот являє собою програму, яка діє приблизно так, як і браузерна програма – зчитує інформацію з веб-сторінок. Пошуковий робот, бот, краулер, пошуковий павук – це все назви одного й того ж явища, які можуть зустрічатися в інтернеті. Пошукова система може мати не один, а кілька пошукових роботів. Кожен бот являє собою автоматичний скрипт, що має свій алгоритм роботи, своє конкретне завдання для певного сайту. Він методично досліджує інтернет у пошуках нових сайтів, нових сторінок, нових файлів, зчитує, заносить їх до реєстру пошукової машини, тобто індексує. Для чого це потрібно пошуковій системі? Для того, щоб вона могла видати на запит найточнішу відповідь, що відповідає картині даних на самий останній момент. Для чого це потрібно сайту? Для того, щоб потрапити у видачу, тобто для того, щоб на пошуковий запит, пов'язаний з ним, система у своїй відповіді зазначила б саме цей сайт. Для чого це потрібно користувачеві? Для отримання правильної адекватної відповіді на своє питання.

Як вже було сказано, система володіє великою кількістю різних роботів, які виконують різні завдання: одні шукають нові сторінки, інші відповідають за знаходження "мертвих" сайтів і чистку пошукових даних, треті індексують картинки, четверті – знаходять відео. Є робот, який відповідає за перевірку коректності посилань і робот, який читає виключно коментарі.

Павукова діяльність так влаштована, що рано чи пізно сайт буде помічений і проіндексований. Однак, це може зайняти кілька місяців. Щоб пошуковий робот швидше помітив його, потрібно внести сайт в спеціальні списки-каталоги, що існують при пошукових системах. Мова в першу чергу йде про такі пошукові Гіганти, як Google і Яндекс. Одноразово проіндексувавши сайт, бот буде регулярно туди заходити. Однак, частота його відвідувань безпосередньо пов'язана з частотою оновлення сайтів. Помітивши, що сайт оновлюється приблизно раз на тиждень, бот заходить туди приблизно стільки ж, відповідно, нова веб-сторінка сайту може залишатися непоміченою кілька днів. І навпаки: існують рухливі блоги, які додають записи по кілька разів на день. Відповідно, робот контролює їх дуже часто й нові сторінки індексуються вже через кілька хвилин. Діяльність робота визначається заданим пошуковим алгоритмом, система алгоритмів гнучка і змінюється.

Для робота одне з найважливіших значень має файл robots.txt, розташований на підконтрольному сервері. Зайшовши на будь-який сайт, робот звертається в першу чергу до нього. Цей файл – інструкція для робота. По-перше, robots.txt може взагалі не допустити бота на сайт і сайт залишиться не проіндексованим. По-друге, robots.txt може закрити боту доступ до певних сторінок і файлів.

В сучасних пошукових системах нейронні мережі перетворять пошукові запити і заголовки веб-сторінок в групи чисел — семантичні вектори. Їх можна порівнювати один з одним і видавати ще більш точні результати. Існують і пошукові алгоритми, які порівнюють вектори запитів і веб-сторінки цілком, а не тільки їх заголовки. Це дозволяє системі розуміти сенс сторінок і вірно відбирати їх, коли люди описують шукане своїми словами. Для цього нейромережа перетворює тексти сторінок в семантичні вектори заздалегідь — на етапі індексування. А коли людина задає запит, алгоритм порівнює вектор запиту з уже відомими йому векторами сторінок.

Пошукових систем існує досить багато, кожна з них наділена своїми цілями і завданнями, часто – комерційними. Однак, глобальних, світового значення пошукових систем, які користуються більшою довірою клієнтів, не так вже багато. Домінуючу позицію в світі займає система Google. Цією системою обробляється 85 % запитів в Інтернеті.

УДК 004.4:378

Коноваленко Д.О., студент, II курс, гр. СА-17-1, ФАМІТ
Науковий керівник – к.т.н., доц. Мельников О.Ю.
Донбаська державна машинобудівна академія, м. Краматорськ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ДЕМОНСТРАЦІЇ РОБОТИ АЛГОРИТМУ ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ APRIORI

Асоціативні правила дозволяють знаходити закономірності між зв'язаними подіями. Прикладом такої закономірності служить правило, що вказує, що з події X потрібна подія Y з деякою ймовірністю. Знаходження таких залежностей дає можливість знаходити дуже прості й інтуїтивно зрозумілі правила. Як правило, для роботи алгоритмів пошуку використовується «Deductor», що проводить миттєві розрахунки й виводить результати у вигляді візуалізаторів «Правила», «Популярні набори», «Дерево правил», «Якщо» (рис. 1). Головним недоліком цього додатку є відсутність візуалізації процесу роботи алгоритму. Також користувачі не можуть зрівняти різні алгоритми й усвідомити переваги методу Apriori.

Рисунок 1 – Розрахунок у середовищі «Deductor» («Правила» й «Дерево правил»)

Було поставлено завдання розробки в середовищі візуального програмування застосування, що дозволяло б студентам, які вивчають алгоритми пошуку асоціативних правил, спостерігати за процесом і проводити аналіз переваг і недоліків ряду методів. Такий додаток повинен дозволяти завантажувати дані з текстового файлу, проводити пошук асоціативних правил і відображати роботу алгоритму Apriori. Розроблений додаток дозволяє генерувати файл транзакцій і проводити розрахунок (рис. 2-3).

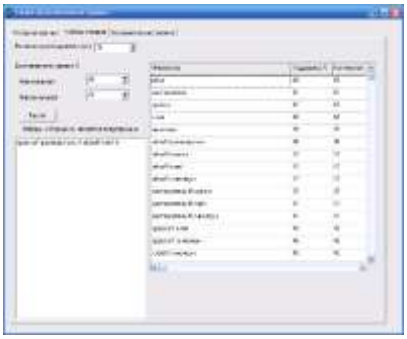


Рисунок 2 – Результат роботи додатку, вкладка «Набори товарів»

Рисунок 3 – Результат роботи додатку, вкладка «Асоціативні правила»

УДК 004.624

Криворучик Д.П., студент IV курсу, гр. КІ-1
Науковий керівник – Оринчак І.А., ст. викл. каф. КІ та КБ
Житомирський державний технологічний університет

ТЕХНОЛОГІЯ WEB-SCRAPING ДЛЯ ПАРСИНГУ ТА ОБРОБКИ ДАНИХ З САЙТІВ

Величезна кількість інформації в мережі Інтернет суттєво ускладнює можливості пошуку інформації Спростити пошук може масштабований спосіб збору інформації, її організації та аналізу. Для цього потрібно використовувати web-scraping (дослівно – „*вишкрібання"*). Створивши додаток web-scraping-у для автоматичного отримання даних можна зекономити час та опрацювати набагато більше сайтів.

Web-scraping, у порівнянні з людиною, працює набагато швидше та ефективніше. Комп'ютерна програма-парсер:

- швидко обійде тисячі веб-сторінок;
- акуратно відокремить технічну інформацію від «людської»;
- безпомилково відбере потрібне і відкине зайве;
- ефективно упакує кінцеві дані в необхідному вигляді.

Отже, web-scraping - це метод, який дозволяє автоматично отримувати велику кількість інформації з веб-сайту, що може заощадити час та зусилля. Використання подібних додатків є дуже актуальною темою сучасних розробок програмних засобів.

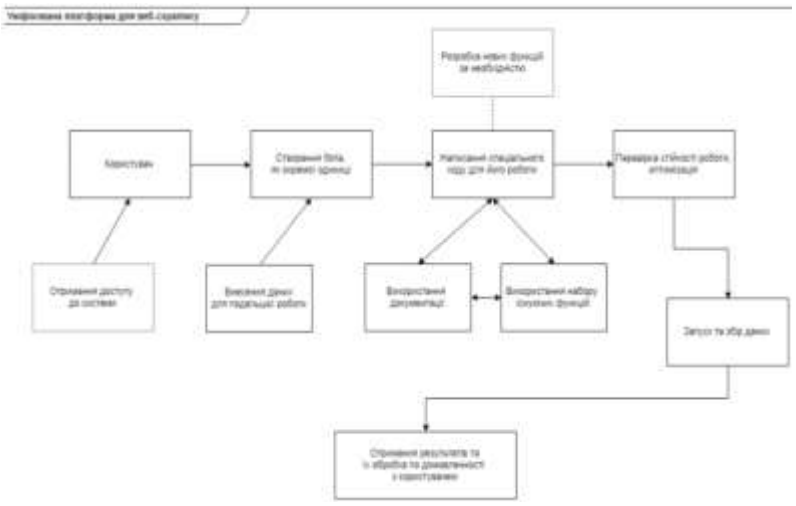
Здається, що все діє досить просто, але є одна проблема. Вона полягає у тому, що близько 95% всіх веб-сторінок або веб-сайтів – це унікальні ресурси. Тому виникає задача по створенню унікальних додатків під кожний веб-ресурс. Але, з точки зору програміста, в унікальних додатках можливе часткове використання того-самого способу отримання даних або частково алгоритмів.

Розглянемо, що собою являє програма або додаток для web-scraping-у.

Як правило, це комп'ютерна програма, що імітує поведінку людини в Інтернеті, або з'єднуючись з веб-сервером напряму по протоколу HTTP, або за допомогою керування повноцінним веб-браузером. Web-scraping включає в себе завантаження та вилучення. Спочатку завантажується сторінка (що робить браузер, коли ви переглядаєте сторінку), після цього можна добувати потрібну інформацію. Зміст сторінки може бути проаналізовано, переформатовано, його дані скопійовані в електронну таблицю тощо. Веб-скрапери, як правило, беруть щось зі сторінки, щоб використати це для інших цілей та потреб.

Існують методи, за допомогою яких певні веб-сайти намагаються запобігати web-scraping-у. Наприклад, виявлення та заборона ботів для сканування (перегляду) своїх сторінок. У відповідь на це були розроблені системи, які спираються на використання методів аналізу об'єктної моделі документа, комп'ютерного бачення та обробки тексту на природній мові, щоб імітувати пошук людини, щоб дозволити збирати вміст веб-сторінок для автономного синтаксичного аналізу.

Як уже згадувалося раніше, якщо всі веб-сайти різні і унікальні, то з'явилася ідея розробити невелику уніфіковану платформу, яка дозволить за допомогою команд отримувати дані з будь-якого веб-ресурсу. Схема роботи даної розробки представлена на рисунку.



В ході розробки буде створена система, що використовуватиме набір функцій, які задовольнять більшу частину вимог від користувача для збору інформації, а також, за необхідністю, є можливість для розширення даної системи. Крім того, розроблятиметься система відслідковування дій – деякі системні повідомлення про події з конкретним процесом.

Дана система розробляється як окремий веб-додаток.

Найголовнішим є те, що система зможе використовуватись майже для будь-якого веб-ресурсу і зможе надати досить широкий спектр можливостей для її використання.

УДК 004.42

Лазорко Н.В., магістр, гр. ПІ-49м
Науковий керівник – Яремчук С.І., к.ф-м.н., проф.
Житомирський державний технологічний університет

ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ В АЛГОРИТМІ ГОМОРІ РОЗВ'ЯЗАННЯ ЗАДАЧІ РОЗМІЩЕННЯ ДЖЕРЕЛ ФІЗИЧНОГО ПОЛЯ

Задачу розміщення джерел фізичного поля на фіксовані місця можна описати наступним чином. Є область $\Omega \subset R^n$; N джерел фізичного поля $D_i, i \in [1:N]$; N посадкових місць $n^j \in \Omega, j \in [1:N]$ та K контрольних точок. Необхідно розмістити джерела фізичного поля на посадкові місця таким чином, щоб максимальне із значень поля в контрольних точках було найменшим. Кожне джерело повинно займати одне посадкове місце та на одне посадкове місце повинно призначатися лише одне джерело. При проектуванні нових технічних систем важливо отримати результат, який задовольняє наперед заданим обмеженням, накладеним на параметри системи, та при якому значення обраного критерію якості набуває найкращого значення.

Відповідно до змістовної постановки задачі була побудована математична модель (1-3).

Керовані змінні.

$$x_{ij} = \begin{cases} 0, \text{якщо i} - \text{те джерело не призначається на j} - \text{те місце} \\ 1, \text{якщо i} - \text{те джерело призначається на j} - \text{те місце} \end{cases}$$

Обмеження.

Так як кожне джерело може займати лише одне посадкове місце та на одне посадкове місце повинно призначатися лише одне джерело, то керовані змінні повинні задовольняти наступним умовам:

$$\begin{cases} \sum_{i=1}^{N} x_{ij} = 1, j \in [1:N], \\ \sum_{j=1}^{N} x_{ij} = 1, i \in [1:N] \end{cases} \quad (1)$$

$$x_{ij} \in \{0,1\}, i \in [1:N], j \in [1:N] \quad (2)$$

Функція цілі.

$$f(x) = \max_{k \in [1:K]} f_k(x) \to min, \quad (3)$$

де $f_k(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}^k x_{ij}$, c_{ij}^k – вклад джерела, що знаходиться на посадковому місці, в значення поля в контрольній точці.

На практиці часто розв'язуються задачі великої розмірності. Тобто, якщо є n – обмежень та n-посадкових точок, то розмірність такої задачі вже буде досягати n^2.

Для підвищення ефективності обрахунків задач, використовуються різні підходи та способи розпаралелення, тобто відбувається розбиття алгоритму на блоки або окремі гілки, що передаються процесорам та можуть обраховуватися незалежно один від одного.

Щоб вдосконалити алгоритм Гоморі для розв'язання задач розміщення джерел фізичного поля на фіксовані місця, було проаналізовано та досліджено способи розбиття процесорної здатності комп'ютера та обрано той, що може бути застосований до даного алгоритму.

При побудові паралельних способів виконання матричного множення поряд з розглядом матриць у вигляді наборів рядків і стовпців широко використовується блокове уявлення матриць.

При блоковому розбитті даних для визначення базових підзадач природним представляється взяти за основу обчислення, що виконуються над матричними блоками.

Для виконання всіх необхідних обчислень базовими підзадачами повинні бути доступні відповідні набори рядків і стовпців матриць. Розміщення всіх необхідних даних в кожній підзадачі неминуче призведе до дублювання та значного зростання обсягу використовуваної пам'яті. Отже, обчислення повинні бути організовані таким чином, щоб в кожний поточний момент часу підзадачі містили лише ту частину необхідних для проведення розрахунків даних, а доступ до решти даних забезпечувався б за допомогою передачі даних між процесорами.

Підхід, що був обраний, має назву алгоритм Фокса, що включає в себе конвеєр та паралельні операції. Його основні принципи – це:

- розпаралелення суттєво послідовних операцій. В даному випадку, це перехід від однієї симплекс таблиці до іншої, тобто пошук кутової точки на виході. За таким же принципом відбувається і побудова відсічення на кожному кроці.

- З'єднання процесорів таким чином, щоб результат роботи одного процесора потрапляв на вхід іншого (лінійна топологія).
- Розбиття складної операції на декілька послідовних стадій, кожна з якої виконується своїм процесором.

Постановка задачі.

1. Реалізувати послідовний алгоритм множення матриць.
2. Реалізувати програму блочного множення матриць (Алгоритм Фокса.
3. Розрахувати теоретичне прискорення і ефективність.
4. Провести набір тестів. Порівняти прискорення послідовного і паралельних алгоритмів.

Приклад реалізації алгоритму має містити наступні функції:

- визначення допоміжних комунікаторів з декартовою топологією;
- завдання вихідних матриць в головному процесі і їх висновок на екран;
- пересилання блоків матриць в процеси декартової решітки (з використанням допоміжного типу MPI_BLOCK);
- власне алгоритм множення матриць, в якому, в свою чергу, викликаються функції пересилання блоку однієї матриці в усі процеси рядки декартової решітки, множення блоків матриць та циклічне пересилання блоків іншої матриці в усі процеси стовпчика декартової решітки;
- пересилання отриманих блоків в головний процес;
- тестування результату в головному процесі шляхом його порівняння з результатом непаралельності множення матриць;
- звільнення виділеної пам'яті.

Маємо основний алгоритм:

- процесор 1 передає дані процесору 2;
- процесор 2 додає свої дані і передає процесору 3 і так далі до отримання розв'язку ЗЛП. Результат передається на початок процесору 1;
- процесор 1 починає перевірку умови цілочисельності. Якщо розв'язок не задовольняє умовам, то застосовується паралельна реалізація, де кожен процесор обраховує свою дію, а саме шукає дробову частину вільного члена того рядка, що не відповідає умовам цілочисельності. І по закінченню кожен процесор віддає свої результати обрахунків, з яких обирається максимальна дробова частина;
- всі результати розрахунків кожного з процесорів повертаються назад до процесора 1, а саме в результаті отримуються розв'язуючий рядок та розв'язуючий елемент;
- будується правильне відсічення та процес знову повторюється до отримання оптимального розв'язку задачі: цілочисельного або частково цілочисельного.

Аналіз ефективності алгоритму. Оцінка часу.

Час паралельних розрахунків складається з часу роботи процесора та часу виконання передачі (4):

$$k_e = \frac{T_1}{T_p} = \frac{1}{\frac{q}{n} + \frac{2q(p-1)\tau_0 + \tau_c \log_2 p}{2n^2 - n}} \qquad (4)$$

Час вирішення m задач на одному процесорі для конвеєра буде мати вигляд: $T_1 = mpt$
А його прискорення (5):

$$k_e = \frac{\text{mpt}}{(p-1)t_0 + pt + (m-1)(t+t_0)} \qquad (5)$$

Дослідивши алгоритми розпаралелення, можна зробити висновки, що алгоритм Фокса, порівняно з іншими:

- достатньо ефективний при великих розмірностях матриці;
- можливість розпаралелення принципово послідовних операцій;
- можливість одночасного виконання передачі даних та їх обробки (асинхронні операції);
- в момент використання паралелізації потребує синхронізації, що знижує ефективність. Але в даному випадку обрахунки не містять великого об'єму даних.

УДК 004.42

Остроухов М.С., аспір., каф. ІПЗ
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДПРИЄМСТВА З ОБЛІКУ, КОНТРОЛЮ ТА КЕРУВАННЯ ВИТРАТАМИ НА ЗВ'ЯЗОК ТА ІТ-ПОСЛУГИ

Створення умов для налагодження взаємовигідних, збалансованих стосунків між державою, владою, її громадянами та суб'єктами підприємницької діяльності, великими організаціями є одним із важливих аспектів співвідношення потреб та інтересів суспільства і держави у фінансовій сфері України. Провідне місце у цих відносинах посідає проблематика обліку податків та їх контролю, з одного боку, та податкового планування і прийняття управлінських рішень щодо оптимізації витрат господарюючими суб'єктами в електронному бізнесі.

В діяльності різних типів підприємств у міру зростання якісних і кількісних параметрів бізнесу, доводиться обробляти великі потоки інформації. Цей процес потребує досконалих форм управління фінансовими і податковими відносинами для обробки потоку інформації. Саме це стало причиною використання автоматизованих інформаційних систем (АІС) на найпоширеніших платформах у повсякденній діяльності підприємства. Можливе застосування АІС як потужних, та надійних платформ для розробників унікальних інформаційних рішень, які можуть задовольняти будь яким вимогам конкретного підприємства і в той же час відповідати вимогам Українського законодавства.

Для досягнення всього перерахованого необхідний постійний моніторинг та контроль ключових показників діяльності, який неможливо здійснювати без впровадження і використання сучасних інформаційних технологій (ІТ). Здатність компанії своєчасно обробляти і аналізувати великі об'єми інформації безпосередньо залежить від рівня автоматизації її діяльності

У сучасному світі складно уявити собі нормальне функціювання будь-якої організації без засобів зв'язку. Зі збільшенням числа співробітників і філій кожна організація рано або пізно змушена буде налагоджувати зв'язок (телефонія, інтернет, мобільна, супутникова) і, відповідно, обслуговувати її.

Системи, що обчислюють вартість послуг зв'язку для кожного клієнта, ще й зберігають інформацію про всі тарифи й інші вартісні характеристики, які використовуються телекомунікаційними операторами для виставляння рахунків абонентам і взаєморозрахунків з іншими постачальниками послуг, називаються білінговими; цикл виконуваних ними операцій називається білінгом. Білінгова система (БС) — це бухгалтерська система, програмне забезпечення, розроблене спеціально для телекомунікаційних операторів. Тобто мова не ведеться лише про операторів стільникового зв'язку. БС використовуються також операторами звичайного (стаціонарного) зв'язку. Ір-Телефонія — інша область застосування БС. Інтернет-Провайдери теж використовують БС, наприклад, для формування рахунків, обліку трафіку. Будь-яка БС створюється на основі певної системи керування базами даних (СКБД). Більшість БС у світі створювалося на основі СКБД Oracle. Серед інших СКБД можна виділити Sybase і Informix як розраховані на більші обсяги інформації. Для прикладу, наведемо назви деяких білінгових систем: BIS, Flagship, CBOSS, Arbor, Bill-2000-prepaid. Варто згадати, що під БС може матися на увазі й апаратне забезпечення, що бере участь в організації білінгу. Тому що БС призначена для автоматизації розрахунків із клієнтом, то вона й повинна забезпечувати цю автоматизацію починаючи з укладання договору до виписки рахунків за послуги стільникового зв'язку. За допомогою підсистем автоматичних послуг і автоматичного збору даних БС повинна надавати абонентам можливість самообслуговування.

Для комплексного рішення автоматизації процесів обліку білінгу підприємства, що об'єднав в одній системі підприємства найбільш популярні методики, є програмне забезпечення, що розробляється індивідуально для потреб підприємства, призначене відділу фінансів та службі моніторингу витрат, а також служб управління персоналом.

Аналізуючи існуючі інформаційні білінгові системи, можна зробити висновок, що жодна з них не відповідає в повному обсязі всім вимогам, які висуваються підприємствами на сучасному етапі розвитку інформаційно-інноваційного процесу, тобто не враховує всіх факторів та можливих обмежень, що впливають на життєздатність інформаційної системи, яка повинна включати в себе надійну базу даних, що зберігається на сервері підприємства, а також клієнтську частину, реалізовану на будь-якій з сучасних мов програмування, яка буде мати актуальність і простоту в підтримці штатними програмістами. Можна перерахувати коло задач, які необхідно вирішити на теоретичному й практичному рівнях при створенні інформаційної білінгової системи на рівні підприємства:

- Ведення кількісного та цінового обліку обсягів витрат і послуг.
- Основною вимогою є ведення бази даних, облік витрат мобільного зв'язку і мобільного інтернету з розшифровкою.

- База даних повинна бути розрахована на велику корпорацію, тобто великі об'єми інформації.
- Облік повинен вестися в розрізі підприємств, країн, відділів та ділянок.
- Введення даних в базу може відбуватися як в ручному режимі, так і в автоматичному через заповнення необхідної інформації з структурованих файлів (.xml, .dbf).
- Облік користувачів програмного продукту, налаштування прав доступу до інформації.
- Отримання звітів за результатами оцінки, можливість формувати власні звіти безпосередньо користувачем або керівником.
- Забезпечення конфіденційності. З цієї причини не можна використовувати стандартне корпоративне програмне забезпечення від провайдерів, наприклад, стільникового зв'язку або інтернет-провайдерів. Служби безпеки великих компаній не можуть допустити витік інформації з прив'язкою персональних даних своїх працівників до номерів телефонів. Тому потрібна своя база даних де, номер телефону буде прив'язаний до персональних даних співробітника і його місці в компанії. Підприємство отримує за контрактом від мобільного оператора тільки номери корпоративних телефонів і вже всередині організації призначає ці номери співробітникам з занесенням цієї інформації в базу даних, для подальшого обліку та розшифровки.
- Щомісячне отримання підприємством від провайдера послуг в структурованих файлах (.xml, .dbf) розшифровки з вартістю послуг з прив'язкою тільки до номерів телефонів. Ця інформація втягується до бази даних і зіставляється зі співробітниками, виходячи з наданого йому корпоративного номера телефону.

За рахунок впровадження інформаційної системи підприємства з обліку, контролю та керування витратами на зв'язок та IT-послуги досягається наступний ряд переваг на різних ієрархічних рівнях організації.

По перше: було виконано детальний аналіз ринку, на предмет надійного засобу зберігання даних, надійних методів проектування і реалізації поставленого завдання. Була знайдена оптимальна система, яка дозволяє автоматизувати введення інформації, отримання звітів діяльності компанії, централізовано вести облік і обробку даних, здійснювати електронну обробку даних і обмін інформацією, оперативно аналізувати діяльність з подальшим визначенням фінансових результатів.

По друге: перед початком розробки рішення, було зроблено ефективний вибір на користь розробки унікальної інформаційної бази і отримано дуже широкі можливості для проектування і реалізації системи. Ця система, з одного боку буде відповідати поставленим перед нею вимогам, і реалізовувати завдання підприємства в повному обсязі, а з іншого боку мати, широкий функціонал для подальшого розвитку:

1. Можливість інтеграції з Обліковими системами бухгалтерського та управлінського обліків, що дасть можливість за коштами формування бухгалтерських проводок безпосередньо впливати на облік всього підприємства;
2. Мати можливість імпорту та експорту даних в будь-якому обсязі та гнучкому вигляді електронних таблиць або структурованих файлів.
3. Доповнювати функціонал, використовуючи можливості регістрів. так, наприклад, використовуючи регістр розрахунку ми можемо раз розробити аналітичний модуль для прогнозу витрат;
4. Використання гнучких інструментів для побудови звітності, що забезпечує керівників підприємства актуальними даними.
5. Використання облікової системи для контролю витрат в різних країнах світу з підтримкою декількох валют.

Мати не високу ціну підтримки програмної та апаратної частин - це є досить суттєвим фактором при обранні ІС

Результатом проведеної роботи стала інформаційна система, що працює на реальному підприємстві України, яка істотно вирішила такі завдання як:

- оптимізація робочого часу співробітників;
- отримання інструменту аналізу витрат, для керівників;
- істотна простота в адмініструванні;
- гнучка та зрозуміла система звітності;
- збереження конфіденційності особистих даних працівників;
- мати безкінечності гнучкість для доробок і модернізації.

Таким чином, була виконана задача вивчення предметної області та завдань підприємства для пошуку оптимального, гнучкого і багатофункціонального інструменту. Розроблена необхідна інформаційна система, з урахуванням всіх поставлених вимог і завдань.

З усього перерахованого вище можна зробити висновок, що добре організована і автоматизована робота співробітників зі спеціально розробленим програмним комплексом, безпосередньо пов'язана з успішною і прибутковою роботою підприємства.

УДК 004.42

Петрів Ю.М., магістр, гр. ПІ-50М
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

РЕКОМЕНДАЦІЙНА СИСТЕМА ДЛЯ ПОШУКУ ПОТЕНЦІЙНИХ КАНДИДАТІВ ТА СПІВАВТОРІВ

З появою глобальної мережі кількість інформації в усьому світі стала стрімко зростати, що зумовило необхідність створення сервісів, основною метою яких є спрощення життя Інтернет-користувачів. Оскільки орієнтуватися у настільки великому об'ємі інформації надзвичайно важко, на сучасному ринку IT виникли рекомендаційні системи як механізм для заміни статичному списку рекомендацій при пошуку на веб-сайтах. Основна мета таких систем – сформувати рейтинговий перелік об'єктів на основі різних критеріїв.

Сьогодні рекомендаційні системи почали широко використовувати у своїй діяльності різноманітні компанії, які займаються рекрутинговою діяльністю. Час величезних папок з профілями співробітників, довгих співбесід і резюме-анкет вже пройшов – технології змінюють всі сторони офісного життя, а роботу HR-відділів особливо. Дедалі все більше HR-спеціалістів користуються різноманітними сервісами, які значно полегшують їм робочий процес. Рекрутинг за допомогою ПЗ значно спрощує процес пошуку та найму кандидатів, що допомагає організаціям швидше набрати необхідних співробітників.

В наш час ринок стає переповненим постачальниками, які пропонують ідентичну функціональність. Все більш важливими диференціаторами стають зручність використання, налаштування, аналітика та управління даними. Сучасні сервіси для пошуку потенційних кандидатів надають зручний інтерфейс та легкість в керуванні, проте не вирішують всіх проблем користувачів. Існує велика кількість різних схем роботи для таких сервісів. Одні функціонують в ролі соціальних мереж для пошуку та встановлення ділових контактів, інші виконують функції соціальних платформ для знаходження клієнтів та виконання завдань. Досить великої популярності сьогодні набули автоматизовані пошукові системи та чат-боти, які пропонують користувачам найрізноманітнішу функціональність. Однак навіть попри велику кількість сервісів для пошуку кандидатів, чимало проблем досі залишаються не вирішеними. Оскільки найважливішими аспектами залишаються ефективність, доцільність та якість. Та найголовніше – на пошуки вдалого претендента, який би відповідав всім вимогам, доводиться витрачати досить багато часу. Оптимальним рішенням за таких обставин є підбір кандидата відповідно до його вмінь і знань, з якими можна наочно ознайомитись. Тобто, рекрутер проглядає весь «багаж» потенційного робітника і відповідно до цього приймає рішення. Для реалізації такої системи найму постає необхідність безпосередньої співпраці з веб-сервісами для хостингу IT-проектів для отримання необхідної інформації.

Основною метою проекту є розробка онлайн сервісу у вигляді веб-сайту, який орієнтований на допомогу в рекомендації та підборі потенційних робітників та співавторів.

Для вирішення поставленого завдання необхідно проаналізувати ряд сервісів з подібною функціональністю. В якості аналогів серед них обираються найпопулярніші. Такими виявились LinkedIn, Upwork, HeadHunter, SuperJob, Workle, PersiaHR, Indeed, HackerRank, Pymetrics, Amazing Hiringтаінші. Порівнявши дані рішення та визначивши їхні плюси та мінуси можливо розробити план по розробці функціоналу системи.

Для реалізації відмовостійкої, високодоступної, надійної та оперативної системи необхідно проаналізувати ряд технологій та вибрати ті, що максимально підходять до специфіки роботи системи. Для написання клієнтського та серверного коду була обрана мова програмування Java Script.

В ході дослідження у сфері механізму підбору персоналу пропонується новий підхід до оптимального вибору потенційного кандидата або співавтора – на основі рекомендацій відповідно до відкритих даних отриманих з веб-сервісу для хостингу IT-проектів – GitHub. Таким чином, HR зможе отримати максимально вичерпну інформацію щодо можливого працівника: ознайомитись з його проектами, дізнатись якими технологіями володіє, в якій країні знаходиться, скільки підписників має тощо. Та на основі цих даних обрати потрібного працівника або співавтора. Такий алгоритм підбору персоналу виключає необхідність складання резюме та проходження додаткових тестових завдань для оцінки здібностей кандидата. А це, в свою чергу, значно пришвидшує процес найму.

УДК 519.85:004.41

Постова С.А., магістрант, гр. ЗПІ-18-2м, ФІКТ
Науковий керівник – Колос К.Р., д.пед.н., проф. каф. комп'ютерних наук
Державний університет «Житомирська політехніка»

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ

Системи масового обслуговування (а інколи і їх мережі): інтернет-банкінг, інтернет-магазини, диспетчерські системи посадки літаків тощо, – наразі широко використовуються у повсякденному житті, в тому числі й в інформаційних системах. Метод імітаційного моделювання (ІМ) широко застосовується в програмуванні складних систем і полягає в створенні логічно-аналітичної (математичної моделі системи та зовнішніх впливів), імітації функціонування системи. Тобто ІМ дає можливість визначити часові зміни стану системи під впливом зовнішніх впливів, а також отримати вибірки значень вихідних параметрів, за якими визначаються основні характеристики систем, в тому числі і систем масового обслуговування (СМО).

Процес функціювання системи масового обслуговування полягає в наданні тієї чи іншої послуги, що визначається функційністю призначення системи. Об'єкт обслуговування в СМО називається вимогою або заявкою. Кожна СМО призначена для обслуговування деякого потоку заявок (вимог), які потрапляють в деякі випадкові моменти часу.

Процес функціювання будь-якої СМО включає в загальному випадку наступні етапи: надходження заявок; очікування (при необхідності) в черзі; обслуговування в пристрої; вихід вимоги з системи. Кожна СМО складається з деякої кількості обслуговуючих одиниць, які називаються каналами обслуговування (це станки, роботи, канали зв'язку, касири, продавці тощо). Обслуговування заявки триває деякий час, після чого канал звільняється й готовий до прийому наступної заявки. Надходження заявок до системи може носити як детермінований характер, так і випадковий (ймовірнісний). В обох випадках доволі часто виникають ситуації, під час яких в певні періоди часу на вході СМО накопичується велика кількість заявок (вони або стають в чергу, або залишають СМО не обслуженими). Також СМО можуть працювати з недозавантаженням або взагалі простоювати.

Для формалізації СМО необхідно описати: процес надходження заявок в систему; процес обслуговування заявок в системі; дисципліну обслуговування. Дисципліною обслуговування (ДО) є правило, за яким заявки вибираються на обслуговування з черги. Розрізняють наступні ДО: 1) обслуговування в порядку надходження або дисципліна FIFO (First Input, First Output — першим прийшов, першим пішов); 2) обслуговування в зворотному порядку або дисципліна LIFO (Last Input, First Output — останнім прийшов, першим пішов); 3) обслуговування у випадковому порядку, коли заявка на обслуговування вибирається випадково серед заявок, що очікують.

Під час обслуговування заявок, крім дисципліни її обслуговування, важливо враховувати до якого класу пріоритетів вона належить. Розрізняють наступні класи пріоритетів на обслуговування: а) ДО без пріоритетна, коли між заявками різних класів немає пріоритетів (пріоритет – це переважне право на обслуговування); б) ДО з відносними пріоритетами, коли пріоритети заявок враховуються тільки в моменти вибору їх з черги на обслуговування; в) ДО з абсолютними пріоритетами, коли пріоритети враховуються також і під час обслуговування – високопріоритетні заявки переривають обслуговування низькопріоритетних; г) ДО зі змішаними пріоритетами, коли заявки даного класу мають до заявок одних класів відносний пріоритет, до заявок другого – абсолютний, а до заявок третього – не мають пріоритету.

Розглянемо покроковий алгоритм моделювання роботи багатоканальної СМО з ДО FIFO для безпріоритетних заявок складається з наступних кроків:

1. Заявка, надходить до N-канальної СМО, перевіряє чи вільний перший пристрій.
2. Якщо вільний, то одразу обслуговується. Час виходу такої заявки дорівнює: час входу + час обслуговування.
3. Якщо пристрій зайнятий, то заявка перевіряє чи вільний наступний пристрій.
4. Якщо вільний, то одразу обслуговується. Час виходу такої заявки дорівнює: час входу + час обслуговування.
5. Якщо пристрій зайнятий, то заявка перевіряє чи вільний наступний пристрій і т.д. до тих пір, поки не буде перевірено на зайнятість останній пристрій.
6. Якщо усі пристрої зайняті, вона перевіряє чи може стати в чергу. Якщо так, то стоїть в черзі до тих пір, поки не звільниться будь-який пристрій. Коли пристрій звільняється, то заявка переходить до нього й обслуговується. Час виходу такої заявки дорівнює: час входу + час очікування + час обслуговування.
7. Якщо ні, то заявка залишає СМО не обслуженою. Час виходу такої заявки дорівнює часу входу.

Існує декілька типів алгоритм ІМ СМО. Так, алгоритм імітаційного моделювання за принципом особливих станів зводиться до наступних дій: 1) визначення події з мінімальним часом надходження – найбільш ранньої події; 2) модельному часу надається значення часу настання найбільш ранньої події;

3) визначається тип події; 4) залежно від типу події починаються дії, спрямовані на завантаження пристроїв і просування заявок у відповідності з алгоритмом їх обробки, й обчислюються моменти настання майбутніх подій (реакція моделі на події); 5) перераховані дії повторяються до закінчення часу моделювання. Під час моделювання здійснюється вимірювання та статистичне опрацювання значень вихідних характеристик. Узагальнена схема алгоритму моделювання за принципом особливих станів наведена на рис. 1.

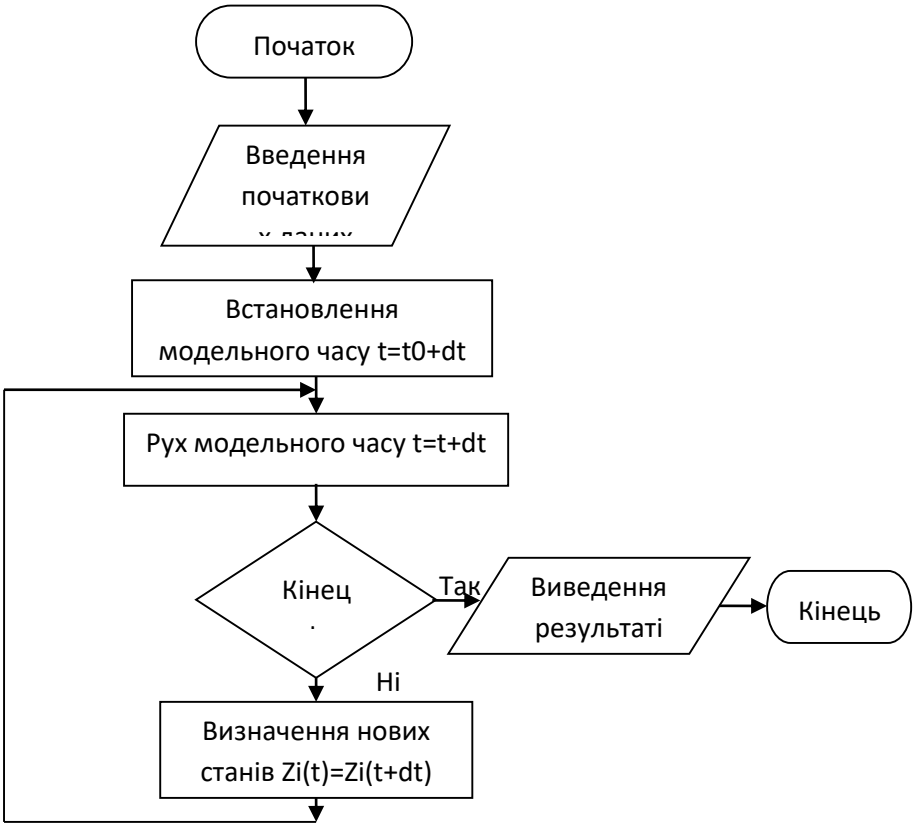


Рис. 1. Узагальнений алгоритм моделювання систем за принципом особливих станів

Альтернативним алгоритмом імітування роботи СМО є алгоритм на базі принципу Δt. Сутність роботи даного алгоритму полягає в наступному. На початку ініціалізується програма, а саме вводяться значення $Z_i(t_0)$, i=1,2,…k, які характеризують стан системи в k-мірному фазовому просторі станів в початковий момент часу t_0. Модельний час встановлюється $t = t_0 = 0$. Основні операції з імітації системи здійснюються в циклі. Функціювання системи відслідковується за послідовною схемою станів $Z_i(t)$. Для цього модельному часу дається деякий приріст dt. Потім за вектором поточних станів визначаються нові стани $Z_i(t + dt)$, які стають поточними. Для визначення нових станів за поточними в формалізованому описі системи повинні існувати необхідні математичні залежності. Під час імітації вимірюються, обчислюються, фіксуються необхідні вихідні характеристики. При моделюванні стохастичних систем замість нових станів обчислюються розподіли ймовірностей для можливих станів. Конкретні значення вектора поточних станів визначаються за результатами випадкових досліджень. В результаті проведення імітаційного експерименту отримуємо одну з можливих реалізацій випадкового багатовимірного процесу в заданому інтервалі часу (t_0, T_k). Моделюючий алгоритм, що базується на застосуванні dt, є застосовним для ширшого діапазону систем, ніж алгоритм, побудований за принципом особливих станів. Але під час його реалізації виникають проблеми визначення величини dt. Для стохастичних систем існують також специфічні алгоритми, призначені саме для обробки ймовірнісних подій.

Велике значення під час реалізації будь-якої імітаційної моделі за допомогою виважений вибір мови програмування. Мова програмування повинна відображати внутрішню структуру понять при описі широкого кола понять. Високий рівень мови моделювання значно спрощує програмування моделей. Основними моментами під час її вибору є: проблемна орієнтація; можливості збирання, опрацювання, виведення результатів; швидкодія; простота налагодження; доступність сприйняття.

Цими властивостями володіють зазвичай процедурні мови високого рівня. Для моделювання також можуть бути використані мови імітаційного моделювання та загального призначення. Найвідомішими мовами імітаційного моделювання, які використовуються для СМО, є SIMULA, SIMSCRIPT, GPSS, SOL, CSL.

УДК 519.85:004.41

Прилуцький О.В., магістрант, гр. ЗПІ-18-2м, ФІКТ
Науковий керівник – Колос К.Р., д.пед.н., проф. кафедри комп'ютерних наук
Державний університет «Житомирська політехніка»

РОЗРОБКА ТА ОПТИМІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ ЄВРОПЕЙСЬКОЇ ПРОГРАМИ ІНДУСТРІАЛІЗАЦІЇ «INDUSTRIE 4.0» ДЛЯ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗАЦІЇ ВИРОБНИЧОГО ПРОЦЕСУ

Телевізор, комп'ютер, смарт-годинник, мобільний телефон, автомобіль чи будь-який інший сучасний пристрій, – усі вони оснащені, а саме вироблені з використанням електронних друкованих плат. Електронні друковані плати закладені в основі будь-якої електроніки.

Жодна сучасна техніка, жодна сучасна виробнича машина на базі числового програмного управління (ЧПУ), жодний побутовий пристрій, не можуть бути побудовані без використання друкованих плат.

Аналізуючи світові тенденції розвитку людства показує, що потреби у електронних пристроях, як на побутовому рівні, так і у виробництві та оборонній промисловості, будуть з часом лише зростати. Беручи до уваги останні десять років, чітко прослідковується динаміка розвитку у цьому напрямку.

Це обумовлює подальше збільшення навантаження на виробництво, а отже, необхідність у зменшенні виробничого часу при збережені якості друкованих плат, що надасть змогу задовольнити зростаючі потреби та збільшити прибуток.

Вирішення цієї проблеми є комплексним питанням: щонайменше втручання й поліпшення потребує механічна частина виробничих машин та їх програмне забезпечення, – що в свою чергу, може бути поділене на декілька основних напрямків.

Друкована плата (з англ. Printing circuit board, PCB) – це радіоелектронний компонент, пластина, на якій чи в середині якої сформований хоча б один шар з провідними доріжками. На друковану плату монтуються електронні компоненти різного розміру та призначення, які з'єднані своїми виводами з елементами провідного рисунка паянням, або значно рідше – зварюванням, у результаті чого складається електронний модуль.

Для автоматизації та зменшення втручання оператора у процес паяння друкованих плат, а саме змонтованих на ній компонентів, та скорочення виробничого часу, використовується модуль розпізнавання продукту. Цей модуль являє собою поєднання механічної частини, сканера та програмного забезпечення. Таким чином, на кожній друкованій платі нанесене спеціальне маркування, що зчитується сканером ще до початку процесу паяння компонентів.

Далі, за допомогою розробленого програмного забезпечення, проводиться співставлення отриманого коду з бібліотекою даних. На підставі отриманої інформації про друковану плату та розташованих на ній компонентів, в автоматичному режимі, машина обирає відповідну програму для паяння. В залежності від комплектації машини та потреб, сканер для зчитування коду може бути, як інтегрованим так і мобільним. Програмне забезпечення, в свою чергу, надає можливість корегування оператором отриманих даних зі сканера.

А отже, такий модуль усуває ймовірність запуску неправильної програми паяння, що унеможливлює зіпсування друкованої плати або цілої партії.

Значною перевагою є ще й те, що стає можливим більш глибока автоматизація виробничого процесу, у рамках європейської програми індустріалізації "Industrie 4.0", а отже, скорочення часу на виробництво та збільшення об'ємів готової продукції.

Сканер коду – це пристрій, для зчитування коду (barcode, data matrix code, QR-code). нанесеного на певну ділянку, й передачі інформації на комп'ютер або іншій термінал, для її подальшого декодування.

Кодування та декодування використовується для швидкої ідентифікації продукту.

Industrie 4.0 – Четверта промислова революція (*з англ. The Fourth Industrial Revolution*) – прогнозована подія, масове впровадження кіберфізичних систем у виробництво й обслуговування людських потреб, включаючи побут, працю, дозвілля. Зміни охоплюють усі різноманітні сторони життя: ринок праці, життєве середовище, політичні системи, технологічний устрій, людську індивідуальність та інше.

Для написання програмної частини модуля розпізнавання продукту, було обрано мову програмування C#. Ця мова є надзвичайно поширеною, а отже, існує велика інтернет-спільнота спеціалістів, де постійно відбувається обмін досвідом, що дає можливість, у разі необхідності звернутися до них за допомогою.

Враховуючи високу популярність цієї мови програмування, створено багато різноманітних бібліотек та готових елементів коду, що полегшує написання кінцевого коду продукту. Ще однією за вагомих причин є універсальність цієї мови, тобто незалежність від платформи, будь-то Windows чи Linux або Mac. Безпосередньо ця мова програмування продовжує свій стрімкий розвиток.

УДК 004.42

Романченко Д.М., магістр, гр. ЗПІ-18 2м
Науковий керівник – Єфіменко А.А., к.т.н., зав. каф. комп'ютерної інженерії та кібербезпеки
Житомирський державний технологічний університет

ВИКОРИСТАННЯ ПАТЕРНІВ ПРОЕКТУВАННЯ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ТЕСТУВАННЯ

На сьогодні у зв'язку з стрімкою інформатизацію сучасного суспільсва, людсво все більше цифроєзується. Отже, наявність якісного програмного продукту стає фактичною необхідністю в умовах сучасності. Розробка максимально якісного програмного продукту стає є життєво необхідною складовою в умовах сильної конкуренції серед ІТ компаній в усьому світі та в Україні як складової глобального ринку. Контроль якості програмного продукту на всіх етапах розробки є обов'язковим для всіх, незалежно від розміру компаній, від потужних світових гігантів до інді розробників та навіть одинаків. Автоматизація процесів тестування допомагає зменшити використання людського ресурсу на ці цілі та збільшити покриття коду.

Тестування - це складний процес націлений на попередження та виявлення дефектів програмного продукту під час розробки та експлуатації. В сучасній комп'ютерній науці використовують термін Quality assurance (QA) що довільно можна перекласти як контроль якості.

Автоматизація процесу тестування – це процес впровадження програмно-апаратних комплексів автоматизації для виявлення дефектів програми/коду. Оскільки цей процес складний, ресурсо та часо затратний, тому для автоматизація це найкращий вихід для покриття вже відтестовного коду (регресійне тестування) та покриття нового коду (White box testing). Патерни (або шаблони) проектування описують типові способи вирішення поширених проблем при проектуванні програм.

Основними задачами використання патернів проектування для автоматизації процесу тестування є наступні:

- Використання перевірених рішень ІТ спеціалістів з усього світу .
- Витрати менше часу завдяки використанню готових рішень
- Стандартизація та оптимізація коду.
- Використання уніфікованих рішень, оскільки всі приховані в них проблеми вже давно знайдено.
- Використання загального словника програмістів, що підвищує ефективність підтримки коду іншими спеціалістами у подальшому.
- Пошук, генерація та використання валідних тестових даних для автотестів.

У роботі будуть використовуватись *Породжувальні, Поведінкові* та *Структурні* патерни для розв'язання завдання вдосконалення та реалізації із використанням патернів проектування моделей та методів створення (генерації) тестових даних, взаємодії систем та передача даних під час автоматизованого процесу тестування.

Породжувальні патерни проектування відповідають за зручне та безпечне створення нових об'єктів або навіть цілих сімейств об'єктів, що в свою чергу значно полегшує генерацію тестових даних особливо в умовах обов'язкової унікальності та/або в зв'язку з необхідністю покрити різноманітні варіції/комбінації. З породжувальних патернів буде використано наступні методи: Фабричні методи, Абстрактна фабрика, Будівельник, Прототип та Одинак.

Структурні патерни проектування відповідають за побудову зручних в підтримці ієрархій класів. Використання даних патернів дає змогу об'єктам з несумісними інтерфейсами взаємодіяти, розділяти класи на окремі ієрархії — абстракцію та реалізацію, дозволяючи змінювати код в одній гілці класів, незалежно від іншої та багато іншого. До структурних патернів відносять наступні методи: Адаптер, Міст, Компонувальник, Декоратор, Фасад, Легковаговик та Замісник.

Поведінкові патерни вирішують завдання ефективної та безпечної взаємодії між об'єктами програми. До них входять наступні методи: Ланцюжок обов'язків, Команда, Ітератор, Посередник, Знімок, Спостерігач, Стан, Стратегія, Шаблонний метод та відвідувач.

Звичайно цілком успішно можна реалізовувати контроль якості за допомогою автоматизації і не використовуючи патерни напряму. Більше того, часто спеціалісти могли вже не раз реалізувати який-небудь з патернів, навіть не підозрюючи про це. Але якраз свідоме володіння інструментом відрізняє професіонала від аматора. Ви можете забити цвях молотком, а можете й дрилем, якщо дуже сильно постараетесь. Але професіонал знає, що головна "фішка" дриля зовсім не в цьому. Підводячі підсумки з усього вище описаного ми приходим висновку, що знання та використання патернів для автоматизації процесів тестування суттєво покращує їх ефективність і як результат значно підвищує перевагу програмного продукту на ринку і як результат конкурентоспроможністю компанії в цілому.

УДК 004.42

Романченко М.М., магістр, ЗПІ-18-1м
Науковий керівник – Коротун О.В., к.пед.н., доц. каф. комп'ютерних наук
Житомирський державний технологічний університет

ОПИС ВЕБ-ОРІЄНТОВАНОЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ПОШУКУ ФІЛЬМІВ, СЕРІАЛІВ І КОМП'ЮТЕРНИХ ІГОР

На сьогодні створено сотні тисяч фільмів, десятки тисяч серіалів і комп'ютерних ігор. Їх кількість постійно зростає. Великий обсяг інформації про такі твори мистецтва зумовив необхідність у каталогах фільмів, серіалів і комп'ютерних ігор. З появою мережі Інтернет, люди почали створювати цифрові (електронні) каталоги у вигляді сайтів, зокрема каталоги з переліком фільмів, серіалів і комп'ютерних ігор, їх описом, фільмографіями режисерів, сценаристів, акторів тощо.

Розробка веб-орієнтованої рекомендаційної системи пошуку фільмів, серіалів і комп'ютерних ігор дозволить значно скоротити час на пошук необхідної інформації про них, забезпечить доступ до цієї інформації широкому колу людей і зробить процес пошуку більш зручним.

Для створення веб-орієнтованої рекомендаційної системи пошуку фільмів, серіалів і комп'ютерних ігор доцільно застосувати: мови PHP 7, HTML5, CSS3, JavaScript; технологію AJAX; веб-сервер Apache; середовище розробки PhpStorm 10; СУБД MariaDB. Розглянемо інструментальні засоби, які варто використовувати при розробці відповідної системи:

– мова програмування PHP (PHP: Hypertext Preprocessor – PHP: гіпертекстовий препроцесор) була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів;

– мова розмітки гіпертекстових документів HTML5 (HyperText Markup Language, version 5) використовується для структурування і представлення змісту сторінок сайтів;

– мова каскадних таблиць стилів третього покоління CSS3 (Cascading Style Sheets 3) переважно використовується, як засіб опису і оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, також застосовується до будь-яких XML-документів;

– скриптова мова програмування JavaScript, що підтримує об'єктно-орієнтований, імперативний і функціональні стилі. Реалізація мови ECMAScript. JavaScript найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінок.

При розробці JavaScript була мета зробити цю мову схожу на мову Java, але при цьому, щоб JavaScript була легка для використання не програмістами;

– AJAX (Asynchronous Javascript And Xml) – технологія асинхронного перезавантаження (динамічного оновлення) сторінок сайта за допомогою JavaScript, HTML, PHP і XMLHttpRequest. Використання AJAX дозволяє обмінюватися даними між сторінками сайту і веб-сервером з частковим перезавантаженням сторінок сайту, і в результаті цього сервер може надсилати у відповідь дані тільки для тієї частини сторінки сайту, яка потребує оновлення; зменшується час на очікування відповіді від сервера; користувач може продовжувати переглядати зміст сайту, поки сервер все ще обробляє запит;

– Apache – відкритий кросплатформовий веб-сервер, який підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Apache є найбільш популярним веб-сервером у світі. Apache передусім використовується для передачі через HTTP статичних і динамічних веб-сторінок у всесвітній павутині;

– PhpStorm 10 – кросплатформове інтегроване середовище розробки для PHP з інтелектуальним редактором, який підтримує PHP версій 7.2-5.3. PhpStorm розроблене компанією JetBrains на основі платформи IntelliJ IDEA;

– система керування базами даних MariaDB – відгалуження від реляційної системи керування базами даних MySQL. MariaDB поширюється під вільною та відкритою ліцензією GNU GPL. MariaDB має необмежений розмір бази даних, є кросплатформовим програмним забезпеченням.

У веб-орієнтованої рекомендаційної системи пошуку фільмів, серіалів і комп'ютерних ігор буде серверна структуру збереження даних, багатокористувацький сайт-додаток для реалізації функціональних можливостей і засобів керування доступом.

Впровадження такої системи пошуку фільмів, серіалів і комп'ютерних ігор допоможе компаніям, що створюють або продають фільми, серіали і комп'ютерні ігри рекламувати відповідну продукцію. Глядачам фільмів і серіалів, а також користувачам комп'ютерних ігор дана система допоможе швидше знайти необхідну інформацію і з меншими зусиллями отримати загальні і індивідуальні рекомендації щодо фільмів, серіалів і комп'ютерних ігор, а крім того, отримати прогнозні оцінки успіху фільмів у глядачів.

УДК 004.9

Сергутін С.О., студент V курс, гр. ПІ-50м
Науковий керівник – Ковальчук А. М., к.т.н., доц. каф. комп'ютерних наук
Житомирський державний технологічний університет

ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ МАРШРУТІВ РУХУ (ПЕРЕМІЩЕННЯ) НЕПОВНОЛІТНЬОЇ ОСОБИ ЗА ДАНИМИ GPS ТРЕКІНГУ

Актуальність теми дослідження. Полягає в необхідності забезпечити сучасним батькам можливості контролю пересувань та рухової активності дітей, через отримання геоданих з мобільного пристрою через мобільну мережу Інтернет. Така технологія може застосовуватись для визначення місцезнаходження неповнолітньої особи і локалізації на карті, навіть під час пошуку загублених.

Мета і завдання дослідження. Розробити моделі та алгоритми інформаційної системи геолокації осіб за допомогою мобільних пристрої. Найбільш складними завданнями в такому випадку є: реалізації фільтру корекції вхідних даних від помилок та алгоритм визначення місця розташування неповнолітньої особи за даними від інерційних датчиків і GPS.

Технологія геопозиціювання, що використовує апарати мобільного зв'язку, вимагає встановлення на телефоні клієнтського програмного забезпечення для визначення місця його розташування. Ця методика визначає розташування пристрою шляхом обчислення її розташування за допомогою ідентифікації стільника, потужності сигналу домашньої і сусідньої комірки, яка безперервно посилається на носій. Крім того, якщо телефон також обладнаний GPS, то зі смартфона надсилається значно більш точна інформація про місцезнаходження.

Програмний комплекс який реалізовано в процесі дослідження грунтується на багатоланковій архітектурі та складається з трьох основних компонентів: серверна частина (сервер, що надає API), головний мобільний додаток (клієнт API) і підпорядкований додаток збирає дані про переміщення (клієнт API). Перелік основних функціональних можливостей системи наступний:

1. На web (мобільному) додатку:
– бачити поточне місце розташування в реальному часі неповнолітньої особи;
– дивитися історію переміщень за останні декілька днів;
– самостійне налаштування оновлень місцезнаходження;
– сповіщення про відключення GPS навігатора;
– низька витрата заряду батареї під час використання додатку.
2. На клієнтському додатку:
– передача даних на основний додаток про місця розташування дитини або декількох дітей одночасно;
– відправка сповіщення про відключення GPS навігатора;
– низька витрата заряду батареї під час використання додатку.

Мобільний додаток з якого береться інформація про переміщення є джерелом даних, і відправляє їх на сервер для зберігання, а основний додаток є споживачем даних і відображає їх на призначеному для користувача інтерфейсі. Для отримання даних головний додаток посилає запити на сервер до REST API(рис. 1).

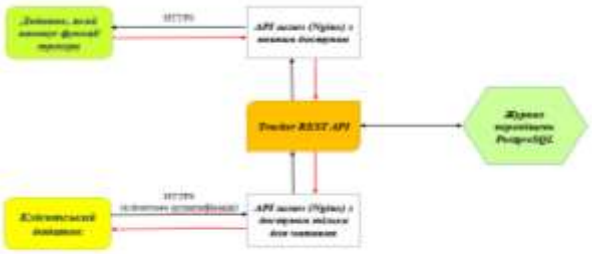


Рис.1. Архітектура програмного комплексу

На рівні додатку серверна частина надає API за технологією REST. Дані передаються без застосування додаткових шарів, тому REST вважається менш ресурсомістким, оскільки не треба робити семантичний аналіз запиту щоб зрозуміти, що він повинен зробити і не треба переводити дані з одного формату в інший. Дані про переміщення неповнолітньої особи будуть зберігатися в журналі геоданих PostgreSQL. Взаємодія між компонентами цих додатків буде відбуватися за допомогою протоколу HTTPS. HTTPS - це звичайний HTTP протокол, який опрацьовує шифровані транспортні механізми TLS. Він повинен забезпечувати безпечну передачу даних про переміщення.

УДК 004.42

Скоковський І.Й., магістр, гр. ПІ-50м
Науковий керівник – Єфремов М.Ф., доц., к.т.н., доц. каф. ІПЗ
Житомирський державний технологічний університет

ВИКОРИСТАННЯ OPENCL ДЛЯ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

В даний час з'являється все більше завдань, пов'язаних з обробкою великих обсягів інформації. Вони відносяться до різних областей діяльності: економічні розрахунки, фізичне моделювання, архітектура, мультимедіа, медицина, обробка графіки і так далі. Одним з основних прийомів, використовуваних при вирішенні подібних завдань є метод паралелізму, коли одна велика задача розбивається на кілька підзадач, що виконуються паралельно і незалежно. Для систем, які вирішують подібні завдання, потрібна велика кількість обчислювальних ресурсів. Раніше такі ресурси могли надати тільки великі кластери в обчислювальних центрах, однак, з розвитком технологій, обчислювальні ресурси ставали доступнішими, з'являлися нові способи створення високопродуктивних систем.

OpenCL (Open Computing Language) – це новий промисловий стандарт і однойменний фреймворк, що реалізує техніку GPGPU, для паралельних задач і паралельних даних гетерогенних розрахунків на різних сучасних процесорах, графічних процесорах, DSP і інших конструкціях мікропроцесорів (рис. 1), які можна знайти на персональних комп'ютерах, серверах, мобільних пристроях і вбудованих платформах.

Рис. 1 Область застосування OpenCL

В останній час стандарт OpenCL досяг набагато ширшої аудиторії завдяки зростаючій кількості пристроїв, що підтримують його. У той же час, спостерігається збільшення відмінностей між пристроями, що підтримують цей стандарт. Ця ситуація пропонує розробникам, які хочуть отримати високу продуктивність, широкий спектр платформ. Враховуючи додаткові параметри платформи OpenCL уздовж конкретних параметрів прикладних програм, проектний простір для дослідження є серйозно великим. Крім того, наявність більш ніж одного виду пристрою дозволяє розподіляти обчислення на гетерогенних платформах.

Програми на OpenCL призначені для виконання розрахунків на відеокартах з підтримкою стандарту OpenCL 1.1 або вище. Сучасні відеокартии містять сотні невеликих спеціалізованих процесорів, які одночасно виконують прості математичні операції над вхідними потоками даних. OpenCL охоплює SMP і SIMD рівні областей паралелізму. Мова OpenCL бере на себе організацію таких паралельних розрахунків і дозволяє досягти великого прискорення для великого класу задач.

Основні особливості стандарту:

1. Вихідний код додатків легко портується на інші платформи.
2. Підтримка широкого класу пристроїв досягається за рахунок введення узагальнених моделей даних системи: модель платформи (platform model), модель пам'яті (memory model), модель виконання (execution model), модель програмування (programming model).
3. Всі моделі є абстрактними (не прив'язаними до конкретних пристроїв), реалізація надається виробником.

Основною перевагою OpenCL є переносимість між різними обчислювальними платформами. На даний момент OpenCL є унікальним засобом такого роду.

УДК 004.42

Соколовський С. В., магістр, гр. ЗПІ-18 2м
Науковий керівник – Єфіменко А.А., к.т.н., зав. каф. комп'ютерної інженерії та кібербезпеки
Житомирський державний технологічний університет

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯУПРАВЛІННЯЗАКУПІВЕЛЬНОЮ ДІЯЛЬНІСТЮ УРОЗДРІБНІЙТОРГІВЕЛЬНІЙ МЕРЕЖІ З ДЕЦЕНТРАЛІЗОВАНОЮ ЗАКУПІВЛЕЮ ТОВАРІВ

В умовах сучасного динамічного розвитку роздрібної торгівлі продовольчими товарами та товарами першого попиту, керування залишками товарів та ланцюгами постачання є найбільш вразливим з бізнес-процесів. Зважаючи на величезний обсяг інформації, який потрібно опрацювувати, зберігати та формувати, виникає потреба у розробці програмного забезпечення для керування залишками товарів та їх постачанням.

Тільки якісна система управління закупівлями може забезпечити оптимальну заповнюваність магазину товарами через контроль ланцюжків поставок товару та спостереженням за ходом продажів та швидке реагування на зміни обсягів реалізації. Тому програмне забезпечення для управління закупівлями повинна бути оснащена наступними функціями:

- фіксація і накопичення статистики продажів через певні проміжки часу;
- надання інформації про фактичні залишки товару;
- формування даних необхідних для якісного аналізу при створенні замовлення на постачання товарів;
- контролювати стан виконання замовлення, попереджувати про існування ризику невиконання умов постачання;
- забезпечувати оперативне реагування на значні зміни в товарообігу та/або на зміни умов постачання товарів;
- забезпечувати побудову та збереження товарної (асортиментної) матриці;
- мати зрозумілий інтерфейс;
- містити інструменти для запобігання виникнення дефіциту товару або створення надлишкового товарного запасу.

Основною концепцією для розробки програмного забезпечення повинен стати повний перехід до електронного документообігу для процесу закупівель. Будь який етап процесу повинен бути реалізований як документ. Такий документоорієнтований підхід дозволить: повністю формалізувати процес закупівлі, зменшити витрати часу на отримання та оброблення попередньо підготовлених даних та прийняття відповідних рішень, а також встановити повний контроль за замовленням від передачі його представнику постачальника до моменту надходження товару на торгівельну точку.

Основою для визначення обсягу та складу замовлення на постачання товару, завжди є аналіз продажів за певний період, який дозволяє визначити які товари мають найбільший та найстабільніший обсяг реалізації в поєднанні з найбільшим прибутком. Крім того повинна бути вирішена проблема оптимізації рівня запасів товарів, які мають невеликий споживчий попит, але обов'язково повинні бути присутні на полицях роздрібних точок. В якості інструменту, який допоможе виконати таку обробку даних обрано поєднання подвійного ABC аналізу та XYZ аналізу. Подвійний ABC аналіз дозволить визначити товари, які мають найбільший попит та приносять найбільший прибуток. XYZ аналіз надасть можливість визначити стабільність рівня попиту. Поєднання цих результатів надасть можливість ранжувати товари по групам за їх відповідною«привабливістю» для закупівлі. Для оптимізації використання, ці дані доцільно формувати один раз на тиждень, у перший робочий день наступного тижня, з обов'язковим збереженням результатів дослідження у зовнішній базі даних.Така реалізація дозволить користувачам програмного продукту, миттєво отримувати доступ до інформації.

Разом з даними аналізу доцільно зберігати інформацію щодо середньоденного попиту на товар та кількість днів в яких був відсутній товар на полицях магазину.

Основним документом який буде регулювати обсяг та асортимент буде програмна реалізація товарної (асортиментної) матриці, в якій буде зазначено: який товар і у якого постачальника потрібно закупати, значення мінімального залишку, максимальний залишок, якому потрібно дотримуватися на роздрібній точці, строк доставки товару після замовлення, середньоденний товарообіг.

Результатом впровадження розробленого програмного продукту очікувано буде: оптимізація товарних залишків у роздрібній мережі, вивільнення обігових коштів за рахунок істотного зменшення запасів товарів які не продаються, запобігання виникнення дефіциту товарів маючих високий попит у покупців, гнучке управління асортиментного матрицею, суттєве збільшення ефективності працівників зайнятих закупівлею товарів.

УДК 004.942

Сотніченко А.В., магістр, гр. ПІ-49м
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

ВИКОРИСТАННЯ ТА МОЖЛИВОСТІ РОЗШИРЕННЯ NAVMESH COMPONENTS ДЛЯ РЕДАКТОРА UNITY

В 2017 році для редактора Unity вийшло розширення, розроблене командою Unity Technologies. Воно доповнює можливості стандартного компонента NavMesh і знаходиться у відкритому доступі.

NavMesh створює поверхню, яка видима тільки в редакторі. Ця поверхня відповідає за знаходження найкоротшого шляху до вказаної точки. Використовується для обрахування переміщення NavMesh Agent. NavMesh дуже спрощує роботу з переміщенням і пошуком шляхів, але в нього є один великий недолік - він абсолютно статичний.

Що б працювати з динамічними об'єктами доводилось шукати обхідні шляхи в розробці. Так, наприклад, при генерації випадкової локації потрібно було завчасно створювати невеличкі шматки території і на них прораховувати розташування NavMesh, а вже потім генерувати локацію з цих завчасно підготовлених фрагментів.

Завдяки новому розширенню з'явилась можливість динамічно формувати NavMesh прямо з коду. Розширення не йде одразу з редактором. Його потрібно скачувати та встановлювати окремо.

Основними компонентами розширення є:

1. NavMeshSurface
2. NavMeshModifier
3. NavMeshModifierVolume
4. NavMeshLink

NavMeshSurface відповідає за створення NavMesh на певній області для переміщення агентів певного типу. Кожну область можна окремо збудувати і переміщення персонажа на них буде відрізнятись. Цей компонент прикріпляється до об'єкта на сцені. До нього можна звертатися з коду і вже в коді викликати створення NavMesh. Його можна створювати відносно графіки або відносно фізичних коллайдерів. Коллайдери - це компоненти, які описують форму об'єкта для фізичних зіткнень. Вони не дають об'єктам проходити одне через одного. Коллайдери краще підійдуть коли необхідно моделювати моменти пов'язані з фізикою. Якщо використовувати графіку то NavMesh побудується по геометрії моделей на які буде накладатись. Також в налаштування можна виставити відносно якого шару це буде працювати. Можна зробити так що б об'єкти з певним шаром ігнорувались.

NavMeshModifier дозволяє точно налаштовувати поведінку конкретного об'єкта під час генерації NavMesh. Цей компонент призначається до ігрових об'єктів. Він є заміною стандартного, який доступний в редакторі. Але на відміну від стандартного він є динамічним. Налаштування стандартного можна змінити тільки в редакторі і за його допомогою згенерувати статичний NavMesh. А що б змінити його доведеться перезапускати гру і заново створювати новий NavMesh.

NavMeshModifierVolume дозволяє відмічати певну область як тип в той час як NavMeshModifier призначає тип області певним ігровим об'єктам. Його можна використовувати для позначення якихось рухомих областей. Такою областю може бути область дверей і за певних умов персонаж не зможе пройти цю область. В нього можна налаштувати розмір області яка буде модифікована. Також можна вказати центр модифікатора відносно об'єкта до якого він буде прикріплений.

NavMeshLink дозволяє створювати зв'язок між двома віддаленими точками NavMesh. Завдяки цьому можна реалізувати стрибки, переміщення по мостах. Коли об'єкт входить в одну з точок то одразу ж переміщується в іншу, ігноруючи будь-які перешкоди на шляху. В параметрах можна вказати тип агента який може використовувати цей елемент. Завдяки цьому можна легко зробити так що б тільки певні персонажі змогли переходити через якусь область. Початкову та кінцеву точки. Між ними і буде відбуватись переміщення об'єкта. Можна вказати довжину цього переходу. Також можна змінити в яку сторону буде працювати цей компонент. Він може працювати як в обидві сторони. А може працювати тільки в одну. І вже коли об'єкт переміститься то не зможе потрапити назад тим самим шляхом.

Кожен з модифікаторів впливає на всю ієрархію об'єктів. Тобто якщо об'єкт має дочірні об'єкти то обраний модифікатор застосується до кожного дочірнього. Вони отримають такі ж властивості як і їх батьківський об'єкт.

Підсумовуючи все вищесказане можна зробити висновок що нові NavMesh Components є дуже зручними і корисними в застосуванні. Вони легко вирішують складні проблеми, в них не важко розібратись і тому, завдяки ним, розробнику програмного забезпечення не доводиться шукати обхідні шляхи для реалізації певних задумок.

УДК 004

Сугоняк І.І., к.т.н, доц.
Лугових О.О., студент, гр. ЗПІ-18-2м
Житомирський державний технологічний університет

ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ РУХУ ТЕХНОЛОГІЧНОГО ОБЛАДНАННЯ

Вимірювання різноманітних механічних величин широко застосовуються на підприємствах по видобутку та обробці природного каменю. Перш за все, це параметри руху виробничих об'єктів. В тому числі – це параметри руху технологічного обладнання та виробів з природного каменю при їх виготовленні. Результати вказаних вимірювань використовуються для керування виробничими процесами і дотримання технологічних норм при виготовлені виробів з природного каменю, контролю їх якості та підвищення конкурентоспроможності.

В сучасних умовах постійно підвищуються вимоги до якості та конкурентоспроможності промислової продукції, що виготовляється з природного каменю. Для цього необхідно підвищувати науково-технічній рівень розробок засобів вимірювань механічних величин, які використовуються для вимірювань і контролю у цій галузі. Від вирішення цієї проблеми залежить точність та надійність функціонування складних виробничих систем, якість промислової продукції, що виготовляється з природного каменю. Все це обумовлює необхідність підвищення точності та швидкодії засобів вимірювання параметрів руху виробів з природного каменю та технологічного обладнання.

Результати вимірювань необхідно зберігати, упорядковувати, перетворювати, обраховувати, порівнювати, оцінювати, представляти графічно, оновлювати. Тому постає задача в створенні інформаційної системи для визначення та контролю параметрів руху технологічного обладнання.

Інформаційна система (у загальному розумінні) — це системи, яка здійснює або в якій відбуваються інформаційні процеси: пошук, збирання, зберігання, передавання й опрацювання інформації.

В якості інформаційної системи можна використати будь-який файл з інформацією, а саме, зберігання інформації нормативно-довідкового характеру.

Опираючись на тему дипломної атестаційної роботи магістра, яка називається Інформаційна система для визначення та контролю параметрів руху об'єктів з програмно-алгоритмічною обробкою відеозображень, можна запропонувати інформаційну систему на базі двох підходів:

- Файлова (текстові файли);
- Організація у вигляді баз даних (MS Excel).

Дана система буде оперувати отриманими даними механічних величин (координата, швидкість, прискорення), з оброблених відеозображень технологічного обладнання та вирішувати наступні задачі: аналіз даних, діагностика, моніторинг, прогнозування, планування, підтримка прийняття рішень.

Отримані дані структуровані в текстовому файлі, потім інтегрувались в MS Excel. В MS Excel були обраховані необхідні значення та побудовані графіки залежностей.

Створення інформаційної системи значень механічних величин за допомогою інших прикладних програм можна відкинути, так як обрахунок даних там присутній, але графічне представлення наявне не у всіх.

Для обробки та контролю параметрів руху технологічного обладнання є безліч програм, але вони призначені для вузького кола задач. Тому для безпосереднього вирішення виробничих задач потрібно розробити спеціалізовані програми. Це можливо реалізувати в програмному пакеті Matlab, тому що окрім використання вже вбудованих функцій, в ньому існує підтримка деяких мов програмування, що дає змогу зробити вставку написану на іншій мові програмування. Або використати власний синтаксис програмного пакету Matlab. Також створити інформаційну систему можливо використавши ряд інших мов програмування: C++, C#, Java.

C#, C-sharp, сі-Шарп - мова програмування, що поєднує об'єктно-орієнтовані і аспектно-орієнтовані концепції. Розроблено в 1998-2001 роках групою інженерів під керівництвом Андерса Хейлсберга в компанії Microsoft як основна мова розробки додатків для платформи Microsoft .NET. Компілятор з C # входить в стандартну установку самої .NET, тому програми на ньому можна створювати і компілювати навіть без інструментальних засобів на кшталт Visual Studio.

В якості мови програмування MATLAB має ряд суттєвих недоліків: обов'язкове встановлення пакету Matlab та відповідних бібліотек для створення інформаційної системи, неможливість запуску та роботи готової програми без Matlab, збитковість програми. Тому, враховуючи дані недоліки, запропоновано створити інформаційну систему моніторингу параметрів руху технологічного обладнання на мові програмування C#.

УДК 004.42

Талько Д.В., магістр, гр. ПІ-50м
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

ANDROID-ДОДАТОК ДЛЯ ПІДБОРУ ОДЯГУ ЗА КОЛЬОРОВОЮ ГАМОЮ

У сучасному світі у зв'язку зі стрімким розвитком легкої промисловості та моди людям стає дедалі складніше правильно обирати одяг за поєднанням, кольором. Одним із варіантів вирішення даної проблеми є швидких та зручний помічник android-додаток на телефон. Android-додаток можна буде завантажити на будь-який смартфон.

Переваги використання додатку заключаються в наступному:

- швидка можливість дізнатись до чого підходить певний елемент одягу;
- вибрати річ необхідного кольору;
- додаток самостійно визначає стиль одягу за допомогою камери смартфону.

Основними задачами android-додатку є:

– підбір одягу за вибраним елементом одягу;
– легкість у користуванні;
– можливість дізнатися сумісність обраних кольорів;
– поради користувачу щодо сумісності елементів одягу.

Передбачувані результати роботи додатку: легко, швидко та головне зі стилем одягнутись для прогулянки, зустрічі чи святкування.

Також актуальним для користувача є оптимальне та водночас просте управління додатком. Необхідно проаналізувати потреби користувачів для формування меню керування.

Основою додатку є доступність та легкість сприйняття інформації. Отримані за його допомогою результати, в тому числі і візуальні, будуть миттєво втілені на практиці. Додаток буде розроблено з використанням штучного інтелекту.

Штучний інтелект — дуже перспективна галузь досліджень, започаткована 1956 року. Визначення штучного інтелекту зводиться до вивчення методів розв'язання завдань, які потребують людського розуміння. Отже, мова іде про те, щоб навчити мобільний додаток розв'язувати тести інтелекту. Це передбачає розвиток способів розв'язання задач за:

- аналогією,
- методів дедукції та індукції,
- накопичення базових знань,
- вміння їх використовувати.

Штучний інтелект вивчає методи розв'язання задач, для яких не існує способів розв'язання або вони не коректні. Завдяки такому визначенню інтелектуальні алгоритми часто використовуються для розв'язання NP-повних задач, наприклад, задачі комівояжера. Штучний інтелект займається моделюванням людської вищої нервової діяльності. Штучний інтелект — це системи, які можуть оперувати зі знаннями, а найголовніше — навчатися. В першу чергу мова ведеться про те, щоб визнати клас експертних систем інтелектуальними системами. Останній підхід, що почав розвиватися з 1990-х років, називається агентно-орієнтованим підходом. Цей підхід зосереджує увагу на тих методах і алгоритмах, які допоможуть інтелектуальному агенту виживати в довкіллі під час виконання свого завдання. Тому тут значно краще визначаються алгоритми пошуку і прийняття рішення.

Хоча проблема «штучного інтелекту» тісно пов'язана з потребами практики, однак тут немає єдиної загальної практичної задачі, яка б однозначно визначала розвиток теорії, проте є багато задач, які є частковими, вузькими. Тому проблема «штучного інтелекту» — це фактично цілий комплекс проблем, які характеризуються різним ступенем загальності, абстрактності, складності й розробленості і кожній з яких властиві свої принципові й практичні труднощі. Це такі проблеми, як розпізнавання образів, навчання й самонавчання, евристичне програмування, створення загальної теорії самоорганізовуваних систем, побудова фізичної моделі нейрона та ін., багато з яких мають велике самостійне значення.

Результатом створення android-додатку є:

1. скорочення часу на підбір одягу,
2. легкість в отриманні інформації,
3. можливість дізнатися сумісність обраних кольорів.

Використання результатів дослідів з кольором одного з найкращих дизайнерів XX сторіччя Йоганеса Іттена допоможе знайти гармонійні поєднання певної речі за кольором. За допомогою автоматизації цих результатів та поставлених задач у пересічного користувача з'явиться можливість одягнутися швидко та зі смаком - це стане на багато простіше. Android-додаток для спрощення підбору одягу значно полегшує користувачу підбір одягу та дає можливість обрати те, що йому більше подобається з обраних стилів.

УДК 004.42

Федорченко М. Ю., магістр, гр. ПІ-49м
Науковий керівник – Петросян Р.В., ст. викл., каф. КН
Житомирський державний технологічний університет

ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ВІДДАЛЕНОГО ДОСТУПУ

Організація віддаленого доступу до апаратних технічних засобів через інтернет в даний час дуже актуальна задача. Сучасний рівень розвитку інформаційно-комунікаційних технологій дозволяє організувати дистанційно збір, обробку та управління будь-яким обладнанням в реальному режимі часу.

Метою є встановлення зв'язку та передача даних між клієнтською і серверною частинами додатків. Встановивши зв'язок, клієнт відправляє повідомлення на сервер. Сервер має можливість переглянути повідомлення, використовуючи з'єднання сокету. Один сокет, слухає конкретний порт під конкретним IP, щоб сформувати з'єднання. Сервер формує сокет-слухач, коли клієнт звертається до сервера. Перш ніж поглибитися в код сервера та клієнта, настійно рекомендується використовувати модель TCP / IP.

Модель TCP / IP – це коротка версія моделі OSI що містить чотири рівні:

1. Рівень процесу та застосування. Виконує функції трьох верхніх шарів моделі OSI: Application, Presentation і Session Layer. Відповідає за зв'язок вузлів і контролює специфікації інтерфейсу користувача. Приклади протоколів: HTTP, HTTPS, FTP, TFTP, Telnet, SSH, SMTP, SNMP, NTP, DNS, DHCP, NFS, X Window, LPD. HTTP і HTTPS - протокол передачі гіпертексту. Використовується в Інтернеті для управління комунікаціями між веб-браузерами та серверами. HTTPS - Ефективний у випадках, коли браузер повинен заповнити форми, увійти, аутентифікувати та здійснити банківські операції. SSH - програмне забезпечення емуляції терміналу, подібне до Telnet. Причина, за якою SSH є кращим, полягає в його здатності підтримувати зашифроване з'єднання. Встановлює безпечний сеанс через з'єднання TCP / IP. NTP – це протокол мережевого часу. Використовується для синхронізації годин на нашому комп'ютері з одним стандартним джерелом часу.
2. Транспортний рівень, хост. Аналогічний транспортному шару моделі OSI. Відповідає за зв'язок та надійну доставку даних. Захищає програми верхнього рівня від складності даних. Два основних протоколи, присутні в цьому шарі: Протокол управління передачею TCP - забезпечує надійний зв'язок між кінцевими системами. Виконує секвенування і сегментацію даних. Також має функцію підтвердження і керує потоком даних через механізм керування потоком.
3. Інтернет рівень. Визначає протоколи, які відповідають за логічну передачу даних в мережі. Основними протоколами, що знаходяться на цьому рівні, є: IP протокол - відповідає за доставку пакетів від вихідного хоста до хоста призначення, дивлячись на IP-адреси в заголовках пакетів. IP має 2 версії: IPv4 і IPv6. IPv4 - використовують більшість веб-сайтів. Але IPv6 зростає, оскільки кількість адрес IPv4 обмежена за кількістю користувачів. ICMP – це протокол Інтернет-повідомлень. Він інкапсулюється в IP-дейтаграми і відповідає за надання хостам інформації про проблеми мережі. ARP - протокол дозволу адреси. Завдання знайти апаратну адресу хоста з відомої IP-адреси. Має декілька типів.
4. Рівень мережевого доступу та зв'язку. Шукає апаратну адресацію і протоколи, присутні в цьому рівні, дозволяє здійснювати фізичну передачу даних.

Програмування сокета – це спосіб підключення двох вузлів до мережі для взаємодії один з одним.

Перш ніж створювати сокетк-лієнта, користувач повинен вирішити, до якої IP-адреси буде підключатися. Через метод «connect», з'єднаємо сокет з сервером. Перш ніж надсилати будь-яке повідомлення, потрібно перетворити інформацію в байти та записати в масив, після чого повідомлення може бути відправлене на сервер через метод «send». Завдяки методу «receive» отримаємо масив байтів як відповідь сервером.

Так само нам потрібна IP-адреса, яка ідентифікує сервер, щоб дозволити клієнтам підключатися. Після створення сокета метод «bind» зв'язує IP-адресу з сокетом. Метод «listen» відповідає за створення черги очікування, яка буде пов'язана з кожним відкритим сокетом, приймає в якості максимального числа клієнтів, які можуть залишатися в черзі очікування.

В роботі спроектували та реалізували інформаційно-комунікаційну систему управління віддаленим доступом, яка складається з клієнтської та серверної частин. Рооблена система дозволяє виконувати наступні функції:

спілкуватися за допомогою текстового чату;

спілкуватися за допомогою голосового та відеочату;

управління віддаленим комп'ютером (наприклад для налаштування); обмін файлами.

УДК 621.398

Фоменко С.О., студент, гр. АТ 22-2м
Кушнір Н.О., м.н.с.
Науковий керівник – Локтікова Т.М., ст. викл.
Житомирський державний технологічний університет

ЗАСТОСУВАННЯ СИГНАЛЬНИХ МІКРОКОНТРОЛЕРІВ ПРИ РОЗРОБЦІ ПРИСТРОЇВ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ ТЕЛЕМЕХАНІЧНИХ КОМПЛЕКСІВ

Основу будь-якого функціонального модуля інформаційно-управляючого телемеханічного комплексу (ІУТК) становить мікроЕОМ, яка здійснює управління модулем. Наприклад, в ІУТК «Гранит-микро» це мікроЕОМ AT89S52 або її аналоги фірми Atmel.

Прилад AT89S52 являє собою економічний високопродуктивний мікроконтролер (МК) фірми Atmel, орієнтований на вирішення задач управління.

До складу мікроконтролера AT89S52 входять: центральний процесорний пристрій (ЦПП) із розрядністю, рівною 8, і максимальною тактовою частотою, рівною 24 МГц; Flash постійний запам'ятовуючий пристрій (ПЗП) ємністю 8 кбайтів; оперативний запам'ятовуючий пристрій (ОЗП) ємністю 256 байтів; 32 програмованих лінії введення-виведення; три 16-розрядних таймера-лічильника подій; вісім джерел сигналів переривань; дуплексний канал послідовного зв'язку UART; сторожовий таймер; вбудований генератор і формувач тактових імпульсів.

Основні схемотехнічні параметри МК AT89S52 такі: тактова частота ЦПП – 0…24МГц; розрядність – 8 бітів; ємність ОЗП – 256 байтів; кількість ліній введення-виведення – 32; порт – дуплексний послідовий UART; діапазон робочих температур – мінус 40 … +85ºС; напруга живлення – мінус 5 … +5В.

Із появою нових мікроконтролерів і розширенням функціональних можливостей модулів ІУТК до їх складу вводяться більш потужні мікроЕОМ.

У цьому сенсі значний інтерес представляють мікроконтролери фірми STM. Провідною лінійкою МК зазначеної фірми є STM32. Засновані на архітектурі ARM Cortex-M4 мікроконтролери серії STM32F4 є продовженням лінійки STM32, маючи ще більш високу продуктивність. Ці мікроконтролери виготовляються за 90нм-технологією і використовують запатентований ST Microelectronics ART Accelerator для досягнення найкращих результатів тестів серед заснованих на ядрі Cortex-M МК, досягаючи показників у 220 DMIPS / 606 CoreMark і працюючи з Flash-пам'яттю на частоті 180 МГц. Інструкції DSP (Digital Signal Processing) і модуль операцій з плаваючою комою надають можливість широко застосовувати дані контролери. Динамічне споживання живлення дозволяє значно зменшити споживання струму при виконанні програми, розміщеної у Flash-пам'яті. Мікроконтролери серії STM32F4 є результатом ідеального поєднання можливості управління МК у реальному часі і продуктивністю обробки сигналів, властивої сигнальним процесорам. Серія складається з п'яти класів приладів.

Розглянемо більш детально мікроконтролер STM32F407VET6.

Даний МК являє собою високопродуктивний, економічний мікроконтролер, заснований на архітектурі ARM Cortex-M4. Також потрібно відзначити, що мікроконтролер STM32F407VET6 належить до нової перспективної групи мікроконтролерів, а саме – сигнальних мікроконтролерів, які підтримують DSP-інструкції.

Окрім загальновідомих частин (ЦПП, Flash-пам'ять, ОЗП тощо), до складу мікроконтролера STM32F407VET6 входять: внутрішні RC-генератори на 16МГц і 32кГц (для RTC); зовнішнє джерело тактування 4 ... 26МГц і для RTC – 32,768кГц; модулі налагодження SWD / JTAG, модуль ETM; три 12-розрядних АЦП на 24 вхідних канали; два 12-розрядних ЦАП; DMA-контролер на 16 потоків даних із підтримкою пакетної передачі; 17 таймерів (16- і 32-розрядні); два сторожових таймери (WDG і IWDG); контролер SDIO (карти SD, SDIO, MMC, CE-ATA); інтерфейс цифрової камери (8/10/12/14-розрядні режими); FSMC-контролер (Compact Flash, SRAM, PSRAM, NOR, NAND і LCD 8080/6800); апаратний генератор випадкових чисел; апаратне обчислення CRC, 96-розрядний унікальний ID; модуль шифрування AES 128, 192, 256, Triple DES, HASH (MD5, SHA-1), HMAC.

Основні схемотехнічні параметри МК STM32F407VET6: тактова частота ЦПП – 168 МГц; розрядність – 32 біти; ємність FLASH – 1 Мбайт; ємність ОЗП – 192 кбайти; кількість ліній введення-виведення – 82; комунікаційні інтерфейси – I2C, USART, SPI, CAN, USB OTG, Ethernet; діапазон робочих температур – мінус 40…+105 ^{0}С; напруга живлення – 1,8…3,6 В.

На базі мікроконтролера STM32F407VET6 був розроблений багатофункціональний пристрій для сучасного ІУТК. Структурна схема розробленого пристрою зображена на рисунку 1.

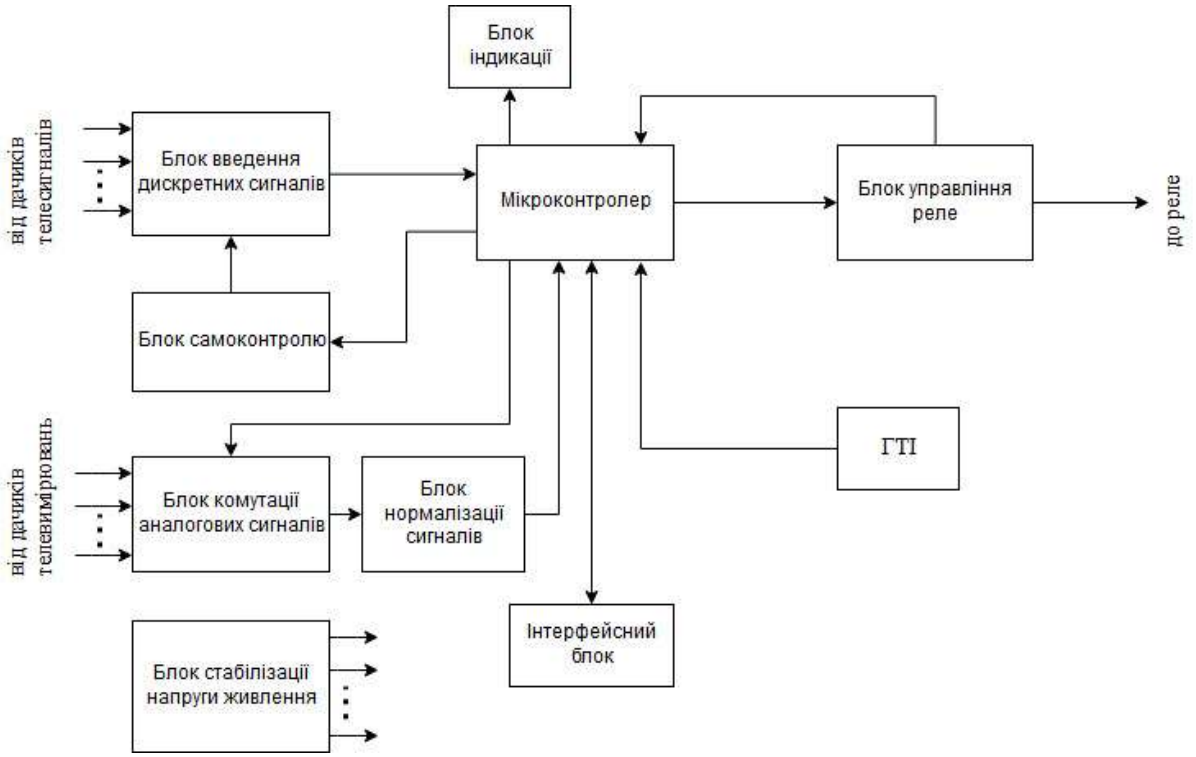


Рис. 1 – Структурна схема багатофункціонального пристрою для ІУТК

Запропонований пристрій поєднує в собі три найбільш важливі функціональні модулі ІУТК – модуль приймання команд телеуправління, модуль введення телевимірювань поточних значень параметрів, модуль введення дискретних сигналів, які забезпечують виконання відповідних основних телемеханічних функцій – телеуправління, телевимірювання, телесигналізації.

Багатофункціональний пристрій призначений для приймання команд телеуправління, формування сигналів управління виконавчими механізмами (реле), здійснення спорадичної передачі телевимірювань поточних значень параметрів при зміні сигналу від датчика на величину, що перевищує деякий встановлений поріг, приймання команд виклику даних від пристрою пункту управління – команд опитування датчиків дискретних сигналів.

Багатофункціональний пристрій забезпечує управління двома об'єктами; введення, перетворення й обробку аналогових сигналів від чотирьох датчиків телевимірювань 0…5мА, мінус 5…0…+5мА, 0…20мА, 4…20мА; введення дискретних сигналів від восьми датчиків телесигналів.

Також були розроблені принципіальна електрична схема багатофункціонального пристрою і програма для мікроконтролера. Програма написана на мові С. Були проведені експериментальні дослідження розробленого пристрою.

УДК 004.588

Шатківський В.М., ст. викл. каф. ІПЗ
Житомирський державний технологічний університет
аспірант
Інститут інформаційних технологій і засобів навчання НАПН України

ОГЛЯД СТАНДАРТІВ ВЕБ-ОРІЄНТОВАНИХ СЕРЕДОВИЩ НАВЧАННЯ

Веб-орієнтоване навчання – це одна з тенденцій і потреб сучасності, що значно покращують процес та результат освітньої діяльності. Згідно зі статистичними дослідженням міжнародного агенства "We are social", що оприлюднені у звіті "Digital in 2018" Інтернетом в Україні користуються близько 25,56 млн. чоловік (58% населення країни), з них 72 % - щодня. Цілком природньо, що веб-орієнтовані середовища навчання стають пріоритетними в контексті організації масової дистанційної освіти. Науковці визначають першочергові завдання щодо розроблення і розповсюдження програмних продуктів та сервісів для вирішення освітніх завдань, виробництва і розвитку засобів та інструментів, що створюють передумови для ефективної співпраці в умовах дистанційного навчання. Одним зі шляхів виконання поставлених завдань можуть бути науково обґрунтоване використання веб-орієнтованих середовищ навчання (ВОСН). Їх використання в педагогічних системах сучасного комп'ютерно орієнтованого освітнього середовища надає можливість залучити усі доречні форми, методи та засоби навчання та контролю, а також усі процеси взаємодії між учасниками освітнього процесу

Що стосується веб-орієнтованих середовищ навчання, то вони дозволяють крім вище описаного створити умови для організації навчального процесу, розподіленого в просторі і в часі, при обов'язковій мережевій взаємодії всіх учасників освітнього процесу. Вітчизняними і зарубіжними дослідниками розглядаються різні аспекти використання веб-орієнтованих середовищ навчання. Проте актуальним залишається питання дослідження, аналізу, вивчення та дотримання специфікацій та стандартів систем організації електронного навчання під час проектування, створення та використання ВОСН.

Суттєвим недоліком веб-орієнтованих середовищ організації навчання являється відмінність у різних виробників навчального контенту функцій управління, відстеження, використання, підготовка звітів про результати. Це являється причиною підвищення вартості навчальних матеріалів. На це є декілька причин: постачальники навчального контенту формують різні формати навчальних матеріалів для окремих середовищ організації навчання; розробники середовищ організації навчання повинні самостійно фінансувати розробку навчального контенту. Для зменшення впливу даного недоліку було започатковано низку міжнародних ініціатив для визначення специфікацій та стандартів, які можуть сприяти повторному використанню технологій навчання. Розробляються стандарти метаданих, що встановлюють необхідний набір атрибутів навчальних матеріалів й оцінки навчальних об'єктів. Вагомими атрибутами яких є тип, ім'я автора, ім'я власника, терміни розповсюдження і формат. У разі потреби до списку атрибутів включаються опис атрибутів педагогічного характеру, а саме стиль викладання або взаємодії викладача зі здобувачем освіти, рівень попередньої підготовки та рівень отриманих знань та ін.

Основними результатами таких ініціатив в даній області є створення наступних стандартів:

- AICC стандарт;
- SCORM (scorm.com) ;
- Tin Can API (xAPI, xapi.com);
- IEEE LTSC (ltsc.ieee.org);
- IMS стандарт (imsglobal.org).

AICC стандарт - документи комітету AICC (Aviation Industry Computer Based Training Committee - Комітет з комп'ютерної підготовки авіаційної промисловості), що стосуються електронної освіти містять два додотки:

Appendix A - стандартом формалізовано спосіб обміну даними між навчальними матеріалами та системою управління навчанням через прямі HTTP-запити (HACP - HTTP AICC/CMI Protocol). Реалізовані модифіковані (з урахуванням Веб) правила створення мета-даних і упаковки створюваних навчальних матеріалів.

Appendix B - (API-Based CMI Communication) визначив спосіб обміну даними між навчальними матеріалами та системами дистанційної освіти (СДО) за допомогою виклику навчальним модулем функцій JavaScript спеціального інтерфейсу (API), що надається системою навчання. Використання API дозволяє розробнику навчальних матеріалів не використовувати прямі HTTP-запити (що вимагає додаткових знань), обмежившись простими викликами JavaScript-функцій. Стандарт AICC, залишається досить актуальним, так як підтримується численними СДО.

SCORM - набір стандартів та специфікацій, який був розроблений для СДО. Даний стандарт включає вимоги до організації навчального матеріалу, а також усієї СДО. SCORM забезпечує сумісність компонентів та можливість їх багатократного застосовування: навчальний матеріал поданий невеликими

блоками, які можуть бути включені у різноманітні навчальні курси та використовуватись СДО незалежно від автора, часу та засобів створення курсу. SCORM заснований на стандарті XML.

Tin Can API (Experience API – xAPI) - це API з відкритим вихідним кодом, що створена на архітектурі REST і використовує формат обміну даними JSON. Tin Can поступово замінює застарілий стандарт SCORM і має ряд переваг у порівнянні зі своїм попередником:

- можливість роботи з матеріалом офлайн. при цьому весь прогрес навчання зберігається і при появі Інтернет-з'єднання дані відправляються в СДО;
- підвищений рівень безпеки - підтримує відкритий протокол авторизації OAuth;
- значно розширено перелік показників, що збирається для статистики;
- не «прив'язаний" до СДО - використовуючи LRS, матеріал можна завантажити на сайт, в блог або соціальні мережі.
- дозволяє враховувати види навчальної активності: навчання за допомогою мобільних пристроїв, ігри, симуляції, очне та змішане навчання. xAPI дозволяє навчальним системам спілкуватися між собою шляхом відстеження і запису навчальних занять всіх видів. Інформація про навчальну діяльність зберігається в спеціальну базу - сховище навчальних записів (LRS - Learning Record Store). LRS надають можливість створювати дуже глибоку аналітику електронного навчання через великі обсяги навчальних даних, які вони записують і зберігають. Традиційні специфікації електронного навчання, такі як SCORM, обмежуються збереженням простих точок даних, наприклад, остаточної оцінки, або того, що курс був запущений або завершений. У структурі операторів, які записують LRS, існує багато точок даних, які можна використати для аналізу освітнього процесу. Такі бази навчальних записів можуть бути як частиною СДО, так і функціонувати самостійно.

IEEE LTSC (Institute of Electrical and Electronics Engineers Learning Technologies Standards Committee) - міжнародна організація, що активно розвивається, і розробляє базові стандарти, що описують вимоги до елементів навчального процесу у середовищі новітніх освітніх технологій.

Ключовим завданням визначено розробку структури, яка підтримує педагогічне різноманіття та інновації, одночасно сприяючи обміну та сумісності матеріалів електронного навчання.

У завдання комітету включено стандартизацію:

- форматів зберігання і пошуку навчальних джерел;
- елементів освітнього контенту;
- форматів обміну даними;
- інформації про учасників навчального процесу;
- принципів побудови систем управління навчанням;
- форматів і принципів розробки навчальних матеріалів.

IMS стандарт (Instructional Management System) - міжнародна некомерційна організація, яка забезпечує впровадження сучасних навчальних технологій у сфері освіти. Дана організація визначає технічні специфікації для організації взаємодії програм і сервісів, що використовуються у СДО, а також підтримує впровадження специфікацій IMS в процес виробництва програмних продуктів і сервісів у глобальному масштабі. Основними стандартами для дистанційного навчання дана організація визначає:

CC (Common Cartridge Specification) - вимоги до структури та організації зберігання навчальної інформації.

CP (Content Packaging Specification) - вимоги до структури даних, яка може бути використана для обміну даними між системами, імпорту та експорту матеріалу, агрегатування та розділення на модулі навчального контенту. Дана специфікація також використовується у стандарті ADL SCORM.

LRM (Learning Resource Meta-data Specification) - формалізація метаданих ресурсів, що використовуються у процесі навчання. Дана специфікація відповідає специфікації Learning Object Meta-data (LOM) Scheme, створеній організацією IEEE LTSC, а також використовується у стандарті ADL SCORM.

LIS (Learning Information Services) - стандарт обміну між системами управління інформацією, що описує користувачів, групи, курси, та результати в контексті навчання.

QTI (Question & test Interoperability Specification) - вимоги до XML-даних, що використовуються для організації обміну навчальними матеріалами, що призначені для тестування знань і оцінювання результатів тестування.

LTI (Learning Tools Interoperability) - стандартизація взаємодії навчального інструментарію з навчальним контентом.

Перспективним вбачається створення засобів і технологій єдиного інформаційно-освітнього середовища закладів освіти та наповнення їх комп'ютерно-орієнтованого навчального середовища якісними інформаційними ресурсами навчального і наукового призначення, що відповідають світовим стандартам та навчального матеріалу із забезпеченням доступу до цих ресурсів.

УДК 519.71

Штоюнда Н.В., магістрант, гр. ПІ-50м
Науковий керівник – Левківський В.Л., аспір. ФІКТ
Левченко А.Ю., к.т.н., ст. викл. каф. ІПЗ
Житомирський державний технологічний університет

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ СИСТЕМ РЕГУЛЯЦІЇ ГЛІКЕМІЇ ПАЦІЄНТІВ З ЦУКРОВИМ ДІАБЕТОМ

Принцип застосування математичного моделювання для обрахунку оптимальної інсулінотерапії був висловлений давно, проте його реалізація довгий час була неможливою через фізіологічну неадекватність та неефективні математичні моделі вуглеводного обміну. Незважаючи на їх велику численність, всі моделі базуються на основі однієї з двох оригінальних базових моделей: модель перорального тесту толерантності до глюкози (ПТТГ) що розроблена Больє у 1961 році та модель внутрішньовенного тесту толерантності до глюкози (ВТТГ) Бергмана-Кобеллі (1979).

Модель ПТТГ:

$$\begin{cases} \mathrm{Vg'} = \mathrm{f1(g,i)} + \mathrm{G'} \\ \mathrm{Vi'} = \mathrm{f2(g,i)} + \mathrm{I'} \end{cases} \quad (1)$$

V — об'єм компартмента, g — глікемія, i — інсулінемія, штрих означає похідну від часу

G' I' — інтенсивність надходження в компартмент інсуліну

Перше рівняння системи (1) спирається на гіпотетичний, фізіологічний зв'язок динаміки глікемії та інсуліномії. Згідно з цим швидкість утилізації глюкози визначається рівнем інсуліну в крові. Ця теорія суперечить клінічно підтвердженому фармакологічному правилу: щоб знизити рівень глюкози на певне значення Δg, потрібно також ввести відповідну кількість інсуліну Δi.

Тобто ведена кількість Δg *та* Δi *пропорційна*

$$\Delta g = \gamma \Delta i \quad (2)$$

З коефіцієнтом пропорційності γ, який лежить в межах

$$2 * 10^{-3} \frac{\text{мг}}{\text{мкОд}} \leq \gamma \leq 5 * 10^{-3} \frac{\text{мг}}{\text{мкОд}}$$

де глюкоза вимірюється в мг, відповідно інсулін в мкОд

Вище описана модель Больє вузька у використанні, зокрема, вона взагалі непридатна для опису експоненціального спадання глікемічної кривої ВТТГ.

Модель ВТТГ

$$\begin{cases} \frac{dg(t)}{dt} = -\alpha[g(t) - g_b] - X(t)g(t), \quad g(0) = g_0, \\ \frac{dX(t)}{dt} = -\beta X(t) + \gamma[i(t) - i_b], \quad X(0) = 0 \end{cases} \quad (3)$$

$g(t)$ та i(t) — глікемія та інсуліномія, g_b i_b — їх базові значення, X(t) — невимірювана величина, яка не має визначеного фізіологічного сенсу, α β γ — коефіцієнти.

Основним недоліком цієї моделі — на відміну від моделі Больє є те, що інсулін і (t) є вхідною змінною, значення якої визначають клінічно

Класифікація моделей

Розрізняють два класи математичних моделей:

Емпіричні моделі

Перевага моделей полягає в простоті і швидкості реалізації. Проте дані моделі не забезпечують розуміння процесу зміни концентрації ГК і інсуліну в різних тканинах і органах, так як не враховують його фізіологію. До того ж, ці моделі потребують введення нових даних в потоковому режимі від об'єкта, щоб передбачити зміни концентрації ГК в найближчому майбутньому.

Теоретичні моделі в свою чергу можна додатково розділити на змішані і повні теоретичні.

Змішані моделі обмежуються числом динамічних рівнянь, в яких функції різних органів або тканин тіла змішуються в один або більше компартментів. Параметри і коефіцієнти даних моделей визначаються на основі результатів різного роду клінічних тестів та досліджень, що обумовлює їх напівемпіричний характер. *Теоретичні моделі.*

Повні теоретичні моделі розглядають фізіологію детально. В системах окремо змодельовані динамічні зміни концентрації глюкози і інсуліну в різних органах і тканинах також враховується взаємодія між цими органами. Повні теоретичні моделі складні в реалізації тому потребують великої кількості часу.В силу своєї опосередкованості, змішані теоретичні моделі, мають найбільш практичний розвиток і являються найбільш перспективними для моделювання системи регуляції глікемії. Математичне моделювання допоможе оцінити ефективність лікарських засобів та процедур для різного типу діабету.

УДК 004.42

Шулятицький В.С., магістр, гр. ПІ-49м
Науковий керівник – Грабар О.І., к.т.н., доц., каф. ІПЗ
Житомирський державний технологічний університет

ВИКОРИСТАННЯ ФРЕЙМВОРКІВ ПРИ РОЗРОБЦІ ПРОЕКТУ НА PHP

Розробка повноцінного проекту складається з великої кількості етапів залежних між собою. Всі етапи поєднанні в загальну структуру проекту. Сучасні принципи розробки спрямовані на виконання трьох основних переваг: високу швидкість розробки, низьку вартість та високу якість проекту. Концепція організації технологічного процесу розробки програмних продуктів, орієнтована на максимально швидке отримання якісного результату в умовах сильного обмеження по термінах виконання та, як правило, нечітко визначених вимог до продукту. Для забезпечення швидкої та повноцінної розробки доцільно використовувати фреймворки.

Фреймвок – це заготовка або шаблон для програмної платформи, яка забезпечує її структуру, програмне забезпечення, полегшує розробку і поєднує в собі різні модулі програмного проекту. Популярні PHP-фреймворки такі, як Laravel, Yii чи Symfony використовують всі ці підходи приділяючи особливу увагу в розробці архітектури та інструментарію, які потім будуть використовувати інші розробники у проектах.

Використовуючи фреймворки не потрібно писати з нуля десятки, а то і тисячі рядків коду, тим самим досягається велика продуктивність. Стає простіше супроводжувати проект, вносити зміни і виправляти помилки, «склеювати» компоненти в більш складні системи. Фреймворк допомагає організувати структуровану архітектуру проекту. Тому використання готового шаблону - це один із способів значно підвищити швидкість розробки проекту. Щоб не витрачати час на планування архітектури додатку і оцінку різних бібліотек, розробник починає роботу з повністю функціонального шаблону, маючи змогу на створення відразу функціоналу, характерного саме для конкретного проекту.

Фреймворк робить додаток більш надійним: рішення, включені в сучасні фреймворки постійно тестуються великою кількістю розробників на різних проектах. Оскільки шаблони мають відкритий код, проблеми пов'язані з надійністю та безпекою, які можна пропустити під час розробки проекту на чистому PHP коді, швидко помічаються та виправляються. Більш проста підтримка проекту забезпечує надійну структуру для додатку вимагаючи використання передових методів та шаблонів у розробці програмного забезпечення. Важливо пам'ятати, що використання готової інфраструктури не означає, що можна забути про забезпечення надійної безпеки і розробки хороших методів. Тому важливо використовувати фреймворк саме за призначенням, постійно оновлювати базу основних елементів структури і реалізовувати власні методи в перевірки параметрів надійності та безпеки.

Фреймворки направляють розробників писати тісно пов'язаний код з мінімальним повторенням. Практично всі популярні на сьогоднішній день PHP-фреймворки базуються на об'єктно-орієнтованому програмуванні і розробці з урахуванням автоматичного тестування. Іншими словами, проект легко читати, тестувати, підтримувати та розвивати у майбутньому. Класи, методи, функції та їх використання: без знання проектних рішень і повної документації складно увійти у розробку або підтримку проекту розробленого на чистому PHP коді. Саме тому використання фреймворків допоможе новим розробникам швидше почати роботу над проектом. І навіть якщо структура незнайома, то документація і знайомі шаблони допоможуть швидше освоїтися в матеріалі з мінімальними зусиллями. Значну перевагу у використанні фреймворків надає активна спільнота розробників. Всі сучасні фреймворки привертають увагу розробників відкритим кодом, постійно розробляються віджети, бібліотеки та додаткові компоненти, тому розробники можуть використовувати готові рішення у проектах. Також створюються навчальні матеріали, відео, щоб пояснити тонкості того, як краще використовувати той чи інший фреймворк у розробці.

Сучасні фреймворки є потужним інструментом у розробці програмного забезпечення, вони можуть заощадити розробнику час і зусилля. Але не потрібно забувати, що кожен проект в своїй мірі унікальний і тому навіть каркас, який ідеально підходить для одного проекту, може зовсім не підійти для розробки іншого. Існує велика кількість фреймворків і кожен базується на різній структурі та функціональних можливостях. Тому головна ідея полягає в правильності вибору інструменту для майбутньої розробки.

УДК 004.91

Якобчук А.С., магістрант, гр. ПІ-49м
Науковий керівник – Плечистий Д.Д., к.т.н., доц.
Житомирський державний технологічний університет

РОЗРОБКА СЕРВІСУ ДЛЯ АВТОМАТИЗАЦІЇ ВЕДЕННЯ ОБЛІКУ ДАНИХ УСПІШНОСТІ СТУДЕНТІВ

Складність ведення обліку даних успішності студентів у вищих навчальних закладах загострює потребу в сучасних комп'ютеризованих системах для автоматизації ведення звітності про успішність студентів. Актуальність теми обумовлена необхідністю розробки сервісу для автоматизації ведення обліку даних успішності студентів вищих навчальних закладів з урахуванням потреб різних структурних підрозділів навчального закладу.

На сьогодні є багато видів обліку і контролю за даними про успішність студентів, які ведуться старостами груп, кураторами, викладачами, проректорами з навчальної роботи та деканами факультетів. Зокрема це такі види контролю як: поточна успішність студента, інформація про успішність студента за кожен місяць, відомості про академічні заборгованості й абсолютної успішності студентів за станом на останній день сесії, результати іспитів і заліків, накази про зарахування студентів на стипендію на наступний після екзаменаційної сесії семестр, облік відвідуваності студентами лекцій, семінарів, лабораторних робіт та інші. Ці дані зберігаються в журналах груп, екзаменаційних та залікових відомостях, довідках, наказах, списках.

Дані про студентів одночасно можуть знадобитися старості, викладачу, деканові. Складнощі обліку успішності обумовлюють:

- значну кількість документації;
- розподіленість споживачів та інформації.

Доцільно розробити централізовану систему створення, заповнення та контролю над відомостями про успішність студентів, що в свою чергу призведе до наступних результатів:

- зменшить витрачання часу на заповнення відомостей;
- забезпечить оперативне отримання відомостей про поточну успішність студентів;
- надасть можливість перегляду відомостей дистанційно;
- виключить помилки викладача при заповненні відомостей;
- автоматизує проведення розрахункових операцій згідно з вимогами навчального закладу;
- надасть можливість ведення електронного архіву журналів.

Для реалізації проекту за основу було обрано клієнт-серверну архітектуру.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

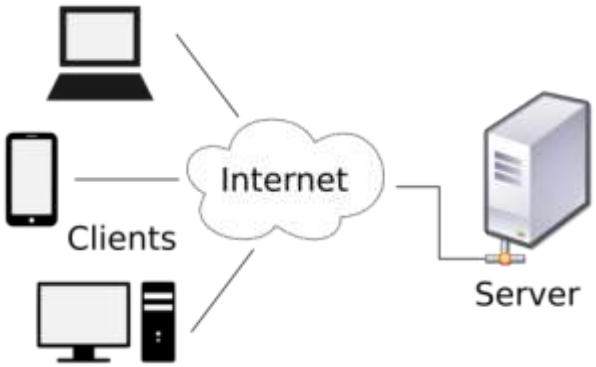


Рис. 1 – Клієнт-серверна архітектура.

Отже основною метою створення системи для автоматизації ведення обліку даних успішності студентів є забезпечення відкритості та прозорості навчального процесу, полегшення ведення відомостей про успішність та зменшення часу на їх заповнення, оперативне отримання необхідних даних про успішність студентів.