

Krivuliak V. V., instructor

PARALLEL PROGRAMMING APPROACHES: TASK VS THREAD*Zaporizhzhya State Engineering Academy, department SoAS*

A fundamental difference between thread-based and task-based programming is the higher level of abstraction that task-based embodies. It frees you from the details of thread management, an observation that reminds the need to summarize the three meanings of “thread” in concurrent software:

- **Hardware threads** are the threads that actually perform computation. It is very generic term that refers to multithreading achieved mostly by duplicating thread state and sharing most everything else in a processing core. Multithreading achieved by duplicating most everything, the whole “core”, is what multicore and many-core designs are all about. Processors and coprocessor can have both “hardware threads” and lots of cores [1].
- **Software threads** (also known as OS threads or system threads) are the threads that the operating system¹ manages across all processes and schedules for execution on hardware threads. It’s typically possible to create more software threads than hardware threads, because when a software thread is blocked (e.g., on I/O or condition variable), throughput can be improved by executing other, unblocked, threads.
- **Threads** (C++11 `std::thread`, `Thread` in java or C#) are objects in process that act as handles to underlying software threads. Some of these objects represent “null” handles, i.e., correspond to no software thread, because they’re in a default-constructed state (hence have no function to execute), have been joined (the function they were to run has finished), or have been detached (the connection between them and their underlying software thread has been severed).

Software threads are a limited resource. If you try to create more than the system can provide, a corresponding exception (`std::system_error` in C++) is thrown. Well-written software must somehow deal with this possibility, but how? One approach is to run `doAsyncWork` on the current thread, but that could lead to unbalanced loads and, if the current thread is a GUI thread, responsiveness issues. Another option is to wait for some existing software threads to complete and then try to create a new thread again, but it’s possible that the existing threads are waiting for an action that `doAsyncWork` is supposed to perform (e.g., produce a result or notify a condition variable).

Even if you don’t run out of threads, you can have trouble with oversubscription. That’s when there are more ready-to-run (i.e., unblocked) software threads than hardware threads. When that happens, the thread scheduler (typically part of the OS) time-slices the software threads on the hardware. When one thread’s time-slice is finished and another’s begins, a context switch is performed. Such context switches increase the overall thread management overhead of the system, and they can be particularly costly when the hardware thread on which a software thread is scheduled is on a different core than was the case for the software thread during its last time-slice. In that case, (1) the CPU caches are typically cold for that software thread (i.e., they contain little data and few instructions useful to it) and (2) the running of the “new” software thread on that core “pollutes” the CPU caches for “old” threads that had been running on that core and are likely to be scheduled to run there again.

Avoiding oversubscription is difficult, because the optimal ratio of software to hardware threads depends on how often the software threads are runnable, and that can change dynamically, e.g., when a program goes from an I/O-heavy region to a computation-heavy region. The best ratio of software to hardware threads is also dependent on the cost of context switches and how effectively the software threads use the CPU caches. Furthermore, the number of hardware threads and the details of the CPU caches (e.g., how large they are and their relative speeds) de-

pend on the machine architecture, so even if you tune your application to avoid oversubscription (while still keeping the hardware busy) on one platform, there's no guarantee that your solution will work well on other kinds of machines [2].

Life will be easier if you dump these problems on somebody else and use asynchronous approach for example C++'s `std::async` or Java's `Executor::execute`. This shifts the thread management responsibility to the implementer of the library. The likelihood of receiving an out-of-threads exception is significantly reduced, because it doesn't guarantee that it will create a new software thread. It is more likely to use an existing worker thread to run the function, or to place it in a queue to wait for a worker thread to become available [3]. When it comes to load balancing, however, the runtime scheduler is likely to have a more comprehensive picture of what's happening on the machine than you do, because it manages the threads from all processes, not just the one your code is running in. State-of-the-art thread schedulers employ system-wide thread pools to avoid oversubscription, and they improve load balancing across hardware cores through work-stealing algorithms. The C++ Standard does not require the use of thread pools or work-stealing, and, to be honest, there are some technical aspects of the C++11 concurrency specification that make it more difficult to employ them. The `boost::thread_group` can address this problem, but it creates dependency on Boost library. C# .NET has `ThreadPool` – the wrapper around a pool of threads maintained by the CLR. `ThreadPool` gives you no control at all; you can submit work to execute at some point, and you can control the size of the pool, but you can't set anything else. You cannot even tell when the pool will start running the work you submit to it. Using `ThreadPool` avoids the overhead of creating too many threads. However, if you submit too many long-running tasks to the thread-pool, it can get full, and later work that you submit can end up waiting for the earlier long-running items to finish. In addition, the `ThreadPool` offers no way to find out when a work item has been completed (unlike `Thread.Join()`), nor a way to get the result [4].

Compared to thread-based programming, a task-based design spares you the travails of manual thread management, and it provides a natural way to examine the results of asynchronously executed functions (i.e., return values or exceptions). .NET `Task` class from the TPL offers the best of both worlds. Like the `ThreadPool`, a task does not create its own OS thread. Instead, tasks are executed by a `TaskScheduler`; by default it simply runs on the `ThreadPool`. Unlike the `ThreadPool`, `Task` also allows you to find out when it finishes, and (via the generic `Task<T>`) to return a result. You can call `ContinueWith()` on an existing `Task` to make it run more code once the task finishes (if it's already finished, it will run the call-back immediately).

Conclusions:

- The thread API offers no direct way to get return values from asynchronously run functions, and if those functions throw, the program is terminated.
- Thread-based programming calls for manual management of thread exhaustion, oversubscription, load balancing, and adaptation to new platforms.
- Thread-pool is best used for short operations where the caller does not need the result.
- Task-based programming via asynchronous functions handles most of these issues for you.

Sources:

1. Forums > Intel MIC Architecture [Web – resource], what is the relation between “hardware thread” and “hyperthread”? – access <https://software.intel.com/en-us/forums/topic/515522> open access.
2. Scott Meyers, Effective Modern C++, O'REILLY Media, Canada, 2015.
3. Concurrency: Executor interfaces [Web – resource], Oracle Documentation – The Java™ Tutorials – access <https://docs.oracle.com/javase/tutorial/essential/concurrency/exinter.html> open access.
4. Threads vs. Tasks [Web - resource] Slacks.Blog 11 Oct. 2013 – access <http://blog.slaks.net/2013-10-11/threads-vs-tasks/> – open access.

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДІВ ВІЗУАЛІЗАЦІЇ КОЛІРНИХ ХАРАКТЕРИСТИК ВІДЕОСИГНАЛУ

Запорізька державна інженерна академія, кафедра ПЗАС

Відео контент використовується практично у всіх областях на сьогоднішній день. Раніше відеозапису, як і взагалі будь-які мультимедійні матеріали використовувалися лише на телебаченні, проте швидкий розвиток інформаційних та електронних технологій зробило відео контент дуже доступними в наш час. У справжні дні мультимедійні матеріали оточують нас практично скрізь: телебачення, фільми, інтернет-сервіси, що дозволяють переглядати і розмішувати відео (youtube) та інше.

Раніше створенням відео займалися лише професіонали, а прямі трансляції існували переважно на телебаченні. Однак у наш час записувати відео може практично будь-яка людина, що має мобільний телефон, оскільки технології, що дозволяють робити захоплення відео, стали дуже доступними на сьогоднішній день і присутні практично скрізь, починаючи від мобільних телефонів і відеореєстраторів, закінчуючи професійними відеокамерами, що використовуються на телебаченні і в різних кінокомпаніях. Як наслідок усього вищеприведеного, з'явилася можливість потокового мовлення відео в режимі реального часу. Подібні технології прийнято називати broadcasting, video-broadcasting, streaming (від stream - потік). Іншими словами те, що раніше робили лише на телебаченні, тепер доступно кожному. Для того, щоб просто віщати в мережу (віщати можна куди завгодно, не обов'язково в інтернет), людині достатньо мати якесь захоплююче відео пристрій (смартфон з вбудованою камерою, відеокамера, реєстратор, будь-яке інше пристрій, що має необхідний інтерфейс), а також спеціальне програмне забезпечення, яке зможе цей сигнал приймати і відправляти в інтернет. Інші люди (користувачі) автоматично отримують можливість переглядати віщати відео контент.

У середовищі професіональних відео-продюсерів та операторів виникає питання про якість зображення під час мовлення. Проблема полягає в тому, що всі відеокамери досить різні і дають різного роду зображення. На професійних майданчиках, під час записів і мовлення великих заходів, кількість камер може виявитися досить значним, з кожною з камер може надходити зображення різної якості і містити різну кількість інформації. Таким чином, перемикання між різними камерами може створювати досить помітні ефекти, що впливають на сприйняття зображення, у зв'язку з чим камери зазвичай калібрують, оскільки камери, розташовані в різних кінцях сцени можуть давати різне зображення. На зображення, одержуване з камери, можуть впливати різні чинники:

- Власні характеристики відео-камери.
- Рівень освітлення на різних ділянках і напрямок світла.
- Різні зовнішні фактори навколишнього середовища (environment).

У зв'язку з цим усім виникає проблема калібрування відеокамер між собою, з тією метою, щоб зображення, одержуване з різних камер, було найбільш наближене до природної обстановці (як якщо б людина сам був присутній на місці камери). Алгоритми калібрування існують різні, часто вони засновані на калібруванні з якоїсь "абсолютної" (приблизно добре налаштованої) камері, проблема калібрування камер один з одним, в цілому, не є повністю вирішеною. Однак незалежно від того, ручна чи використовується калібрування або автоматична, необхідно мати деяку інформацію про зображення, а також його характеристики.

Якщо узагальнити вищесказане, користувачеві (якщо калібрування проводиться вручну) або розробнику (прикладному програмісту, що розробляє алгоритм калібрування) необ-

хідно мати певну інформацію про поточний стан відеосигналу, щоб бачити як ті чи інші настройки можуть впливати на одержувану картинку. В цілому можна виділити два основних категорії характеристик відео-сигналу:

- Яскравість (luminance).
- Колір (chrominance).

У даній роботі ставиться мета розробка таких компонентів, які дозволили б отримувати інформацію про відеосигналі \ зображенні і відображати її різними способами: як у текстовому (якісь статистичні дані зображення, значення яких-небудь параметрів), так і в графічному (візуалізація цих кольорних і яскравості характеристик). Додатковою умовою є дослідження можливостей оптимізації даних методів з метою пристосування їх до реальних умов, придатним для практичного використання (описана вище візуалізація не повинна гранично завантажувати CPU і повинна працювати оптимально).

Під час досліджень можливих таких методів, мною було з'ясовано, що основними методами оцінки якості зображення, що дозволяють візуалізувати результати роботи, є:

- Waveform
- Vectoroscope

Перший є способом візуалізації інформації про яскравість зображення. Другий же є емуляцією реального осцилографа для отримання характеристик поступаючого сигналу.

В результаті можна виділити наступні основні модулі системи:

- Модуль захоплення входять кадрів, який може мати кілька режимів роботи: захоплення одного зображення (картинки), витяг кадрів з відео-контейнера (avi, mp4 і т.д.), захоплення відео з відеокамери.
- Модуль аналізу одержуваних кадрів, завдання якого полягає у витяганні необхідних характеристик.
- Модуль компонента Waveform.
- Модуль компонента Vectorscope.

Розглянута в роботі програмна система дозволить отримувати характеристики входять кадрів, аналізувати їх кольорні і яскравість характеристики, а потім візуалізувати їх і виводити їх характеристики на екран.

Висновки:

- проведено дослідження предметної області;
- вивчені різні популярні кольорні простори;
- вивчені основні характеристики відеосигналу і способи їх аналізу;
- вивчені методи роботи декодерів та їх використання;
- досліджено можливості візуалізації отриманих характеристик.

Література

1. Tektronix - Television Measurements: NTSC Systems
2. <http://en.wikipedia.org/wiki/Vectorscope>
3. Danny Greer - Video Essentials: How to Read a Vectoroscope and Waveform Monitor
4. NewTek Discussion Forums - Vectoroscope Related Topics
5. MontiVision Imaging Technologies - MV Smart ActiveX Control: DirectShow Applications

Бобирь Б.М., магістрант, Скрипник І.А., к.т.н., доцент, науковий керівник

АВТОМАТИЗОВАНА СИСТЕМА РОСПІЗНАВАННЯ МОВИ ЖЕСТІВ НА ОСНОВІ ТЕХНОЛОГІЇ OPENCV

Запорізька державна інженерна академія, кафедра ПЗАС

Технічний прогрес не стоїть на місці. Кожного дня з'являються нові технології, які ще вчора могли здаватися тільки фантастикою. І, як правило, займають свої ніші в різноманітних галузях. Одна з самих важливих галузей – медицина, зараз завдяки ІТ набуває нових рис. Інноваційні технології допомагають лікарям проводити об'єктивну діагностику захворювань, накопичувати й ефективно використовувати отриману інформацію на всіх стадіях лікувального процесу і, що найважливіше для медичної науки, є неоціненними у науковому пізнанні.

Однією з таких інноваційних технологій, яка, хоч і вважається порівняно молодого, але дуже перспективною, є комп'ютерний зір. Комп'ютерний зір активно використовується в аналізі відеоданих, які отримують за допомогою мікроскопії, рентгенографії, ультразвукових досліджень та томографії. Прикладом інформації, яка може бути отримана з таких відеоданих є виявлення пухлин, атеросклерозу чи інших злоякісних змін.

В наш час, не зважаючи на видимий науковий прогрес та величезні досягнення в медицині, залишається відкритою проблема фізично обмежених людей, а саме глухонімих людей та людей з поганим слухом. Лікування таких недугів в наш час майже неможливо, і тому людям з цим залишається тільки жити. І хоч в XVII столітті француз Шарль де Ліпі створив язык знаків, який кардинально змінив життя глухонімих людей в кращу сторону, все-одно проблема комунікації з іншими людьми, які не потерпають від таких недугів, має бути вирішена. Це дасть змогу зменшити прірву нерозуміння між такими групами людей. Прикладом може стати випадок, коли глухоніма людина, будучи відвідувачем лікарні, не зможе пояснити аптекарю, які ліки і в якій кількості їй потрібні.

Рішенням такої проблеми може стати система для розпізнавання мови знаків глухонімих людей та конвертації цих знаків в слова та словосполучення, які зможе зрозуміти звичайний чоловік. Так, за допомогою тільки відео записуючого пристрою (веб-камера, смартфон і т.д.), направленої на людину, яка розмовляє мовою жестів, та підключеної до комп'ютеру з системою розпізнавання жестів, існує можливість, що такі дві групи людей порозуміються [1-3]. Оскільки в наш час веб-камери вже не таке дороге задоволення, а комп'ютери настільки розповсюджені, що їх відсутність сприймається як дикунство та виликає здивування, то проблем з технікою майже не виникне.

Отже, в роботі описується створення системи розпізнавання мови жестів, базуючись на вхідному відео потоці з даними із зовнішнього джерела. Система, аналізуючи кожен кадр потоку, за допомогою бібліотеки OpenCV (*Open Source Computer Vision Library* – бібліотека комп'ютерного зору з відкритим вихідним кодом) розпізнає жест як комбінацію пікселів, відсіюючи не потрібний шум та задній фон. Далі, за допомогою нейронної мережі, розпізнане зображення спрощується, даючи змогу швидше ідентифікувати жест та знайти відповідний йому символ, слово або словосполучення. Під спрощенням мається на увазі зведення отриманого відфільтрованого кадру до набору базових точок.

Підводними каменями при створенні даної системи є те, що в "амслензі" (інша назва мови жестів) є слова (або словосполучення), які можна показати тільки комбінацією жестів. Тому система повинна мати змогу ідентифікувати як і окремі жести, що можуть означати букви, так і цілі комбінації жестів(масиви з кадрів), що можуть означати цілі словосполучення. Одним з рішень такої проблеми є особлива структура бази жестів, по якій проводиться по-

шук та ідентифікація жесту, де індексом є перший «піджест» комбінації. Другою проблемою є те, що на фоні у зображення може бути шум, який буде тільки заважати розпізнаванню. Рішення – використання відео пристроїв з більш високою якістю зйомки.

В результаті, можна виділити такі основні модулі системи:

- Захват відео з пристрою – модуль системи, який безпосередньо записує відео з зовнішнього пристрою та розділяє на кадри.
- Попереднє відсіювання кадрів – модуль системи, який відсіює зайві кадри, якщо наступний кадр майже не відрізняється від попереднього, тим самим збільшуючи швидкість ідентифікації жесту.
- Аналіз кадрів – аналіз отриманих кадрів на наявність деякого жесту.
- Спрощення жесту – фільтрування зображення від зайвого шуму і отримання базового набору точок для пошуку по базі жестів
- База жестів – особлива структура даних, яка зберігає набір жестів
- Ідентифікація жесту – модуль для пошуку набору базових точок в базі жестів та отримання вихідної букви, слова або словосполучення.

Отже, розглянута в роботі програмна система, дає змогу зрозуміти мову жестів звичайним людям, які до цього ніколи не мали змоги її вивчити [4]. Це зменшує прірву непорозуміння між глухонімими людьми та людьми без вад. В майбутньому, коли такі прототипи, як наприклад, окуляри Google Glass, які вміщують в собі як комп'ютер, так і камеру, стануть повсякденною річчю, системи такого рівня будуть допомагати в спілкуванні і знищать будь-які мовні бар'єри.

Висновки:

- проведено ряд досліджень у галузі застосування комп'ютерного зору в промисловості та медицині;
- вивчено історію створення мови жестів;
- вивчено базові жести мови глухонімих;
- проведено дослідження в розпізнаванні окремих образів в відеопотоці;
- досліджено переваги та недоліки, а також можливості, які надає бібліотека OpenCV;
- розроблено автоматизовану систему для розпізнавання мови жестів за допомогою бібліотеки OpenCV.

Література

1. Абакумов В.Г. Интерпретация движений рук расширяет возможности интерактивного управления в интеллектуальных системах/ В.Г. Абакумов, Е.Ю. Ломакина// Природные и интеллектуальные ресурсы Сибири. – 2009. – С.199-202
2. Lomakina O.Y. Gestures Recognition as a New Information Input Device for Automatic System Control / O.Y. Lomakina // «Modern Problems of Radio Engineering, Telecommunications, and Computer Science»: Proceedings of X-th International Conference TCSET'2010. – Lviv-Slavske, 2010. – P. 100-103.
3. Hand tracking and gesture recognition for human-computer interaction / [C. Manresa et al.] // Electronic Letters on Computer Vision and Image Analysis. –2005. – №5(3). – P. 96-104
4. Абакумов В.Г. Застосування жестів рук при людино-машинному інтерфейсі/ В.Г. Абакумов, О.Ю. Ломакіна, О.Б. Яровенко// Електроніка и связь. – Тематический выпуск: Электроника и нанотехнологии. – 2011

Борисов С. В., магістрант, Шамровський О.Д., д.т.н, професор, науковий керівник

ДОСЛІДЖЕННЯ СИСТЕМ МОДЕЛЮВАННЯ ПРОЦЕСУ НАВЧАННЯ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНИХ АГЕНТНИХ СЕРЕДОВИЩ

Запорізька державна інженерна академія, кафедра ПЗАС

Стрімкий розвиток технологій спонукає до змін практично в усіх галузях суспільства. Серед них і освітня галузь, яка останнім часом проходить непростий період трансформацій. На наших очах відбувається бурхливий розвиток і перші кроки практичного застосування дистанційної освіти в Україні.

На сьогодні існує декілька класифікацій систем навчання. Засоби навчання з елементами штучного інтелекту складають значний клас програмних засобів [2]. Головною відмінністю даної групи засобів є використання концепцій моделювання інтелектуальної діяльності учня як такої. Крім того, засоби штучного інтелекту дозволяють створювати системи навчання, які не потребують фізичної присутності вчителя під час навчання – його дії імітує система. Поєднання інтелектуальних систем з новими технологіями дає змогу вивести навчальний процес на інший рівень, основні переваги якого відображаються в наступних формулюваннях: «освіта без меж», «освіта крізь усе життя», «освіта за меншу вартість» [1].

Проте у цілому всі ці системи навчання, окрім своїх переваг, мають ряд недоліків. Часто такі системи створені для певної цільової аудиторії. Наприклад для конкретного вищого навчального закладу, для школи тощо. Це накладає обмеження на використання таких систем. До того ж, хоч в даній області і ведуться постійні дослідження, такі системи навчання досі не досягли того рівня викладання, який забезпечують нам «реальні» викладачі.

Більш актуальною проблемою є неможливість швидко перевірити ефективність системи навчання на практиці. Впровадження нової системи навчання у тому ж вищому навчальному закладі – це великий ризик. Результати таких змін проявляться лише через кілька років. Дуже важко передбачити успіх чи невдачу. У випадку, якщо нова система виправдає себе, ВНЗ випустить висококваліфікованих спеціалістів. Інакше ж, рівень навчання студентів може сильно знизитись.

Виникла потреба створення такої системи, яка б моделювала процес навчання і робила висновок про придатність застосованого методу навчання на практиці. В даній системі віртуальні учні навчаються за заданою програмою у віртуальних викладачів. Після закінчення навчання система видає графік успішності та за певними критеріями оцінює застосований метод.

Для створення такої системи було використано багатоагентне інтелектуальне середовище. Багатоагентна система (рос. МАС, англ. Multi-agent system) [4] — це система, утворена декількома взаємодіючими інтелектуальними агентами.

Багатоагентні системи можуть бути використані для розв'язання таких проблем, які складно або неможливо вирішити за допомогою одного агента або монолітної системи (англ.). Прикладами таких завдань є онлайн-торгівля, ліквідація надзвичайних ситуацій, моделювання соціальних структур тощо.

Виходячи з характеристик мультиагентних середовищ (МАС), вони найбільш підходять і для моделювання процесу навчання.

У багатоагентній системі агенти мають кілька важливих характеристик:

- Автономність: агенти, хоча б частково, незалежні
- Обмеженість уявлення: у жодного з агентів немає уявлення про всю систему, або система занадто складна, щоб знання про неї мало практичне застосування для агента.

- Децентралізація: немає агентів, що керують усією системою

МАС належать до систем, що самоорганізуються, тому що в них шукається оптимальне розв'язання задачі без зовнішнього втручання. Під оптимальним розв'язанням розуміється розв'язання, на яке витрачене найменша кількість енергії в умовах обмежених ресурсів.

Головна перевага МАС — це гнучкість. Багатоагентна система може бути доповнена й модифікована без переписування значної частини програми. Також ці системи мають здатність до самовідновлення й мають стійкість до збоїв, завдяки достатньому запасу компонентів і самоорганізації.

Існує багато середовищ розробки МАС. Для поставленої цілі більш за все підходить одна з таких систем, а саме Java Agent Development Framework [3].

JADE - програмне середовище розробки мультиагентних систем та програм, що підтримує FIPA - стандарти для інтелектуальних агентів. The Foundation for Intelligent Physical Agents (FIPA) - є органом з розробки та встановлення стандартів на програмне забезпечення для гетерогенних і взаємодіючих агентів та агентних систем на їх основі.

JADE включає в себе:

- Середовище виконання агентів. Агенти реєструються і працюють під управлінням середовища;
- бібліотеку класів, які використовуються для розробки агентних систем;
- набір графічних утиліт для адміністрування та спостереження за життєдіяльністю активних агентів.

Програмне середовище Jade підключається до будь-якому проекту на мові Java. Це дуже зручно, оскільки мовою розробки системи була обрана саме Java.

Java — об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Синтаксис мови багато в чому схожий на C та C++. У офіційній реалізації, Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. Розроблену на цій мові систему моделювання процесу навчання, в перспективі можна легко перенести на будь-яку платформу.

Висновки:

- проведено ряд досліджень у галузі застосування штучного інтелекту у навчальній системі
- вивчено історію розвитку навчальних систем та основну класифікацію типів систем навчання
- досліджено застосування багатоагентних систем для моделювання процесу навчання
- вивчено програмне середовище розробки мультиагентних систем Java Agent Development Framework
- розроблено систему моделювання процесу навчання з використанням багатоагентної системи

Література

1. Гриценко В.И. Дистанционное обучение: теория и практика/ Гриценко В.И., Кудрявцева С.П., Колос В.В., Веренич Е.В. – Киев: Наукова думка, 2004. – 375 с.
2. Компьютерная технология обучения. Словарь-справочник. / Под ред. В.И. Гриценко, А.М. Довгялло. – Киев: Наукова думка, 1992. – 650 с.
3. Основи нових інформаційних технологій навчання/ За ред. Ю.І. Машбиця. – К.: ІЗМН, 1997. – 264 с.

РОЗРОБКА ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ІГРОВИХ ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ТА ГЕНЕТИЧНИХ АЛГОРИТМІВ

Запорізька державна інженерна академія, кафедра ПЗАС

Нейронні мережі - це один з напрямків досліджень в галузі штучного інтелекту, заснований на спробах відтворити нервову систему людини. А саме: здатність нервової системи навчатися і виправляти помилки, що має дозволити змодельовати, хоча і досить грубо, роботу людського мозку.

Історично, штучні нейронні мережі за свою вже більш ніж півстолітню історію відчували як періоди стрімких злетів і підвищеної уваги суспільства, так і змінювали їх періоди скепсису і байдужості.

У наші дні нейронні мережі знову в зеніті слави завдяки винаходу методу преднавчання «без вчителя» на основі Обмежених больцманівських Машин (Restricted Boltzmann Machines, RBM), що дозволяє навчати глибокі нейронні мережі (тобто з екстра-великим, порядку десятків тисяч, кількістю нейронів) і успіхам глибоких нейронних мереж в практичних завданнях розпізнавання усного мовлення і зображень.

Найцікавішим є прикладний аспект нейромереж, їх здатність навчатися на зібраних даних і вирішувати задачі розпізнавання. З багатьма практичними завданнями класифікації та прогнозування чудово справляються добре опрацьовані, відносно невеликі моделі багатошарових перцептронів (Multilayer Perceptron, MLP) та мережі радіальних базисних функцій (Radial Basis Function network, RBF).

Але що як відійти від найбільш розповсюджених шляхів використання нейронних мереж та використати їх унікальні особливості для моделювання поведінки персонажів у комп'ютерних іграх? Традиційні методи створення штучного інтелекту в іграх дозволяють досить успішно, але дещо жорстко описувати поведінку персонажів. Такий підхід не включає в себе досить важливий фактор — різноманітні стилі та манери гри різних людей. Штучний інтелект на основі нейронних мереж, або його гібридний варіант (використання нейронних мереж у тих модулях, де важлива гнучка та швидка адаптація до різних ігрових ситуацій) здатний пристосовуватись до унікальних особливостей ведення гри різними людьми та швидко адаптувати стратегію під різні ігрові ситуації. Такий штучний інтелект здатен моделювати поведінку більш схожу на поведінку людини, ніж це можна досягнути тільки традиційними методами, тому ігрові сесії з таким опонентом набагато цікавіші та динамічніші.

Переваги використання нейронних мереж:

- *Адаптація до змін навколишнього середовища.*

Нейронні мережі мають здатність адаптуватися до змін навколишнього середовища. Зокрема, нейронні мережі, навчені діяти в певному середовищі, можуть бути легко перевчені для роботи в умовах незначних коливань параметрів середовища. Більш того, для роботи в нестаціонарній середовищі (де статистика змінюється з часом) можуть бути створені нейронні мережі, перенавчатися в реальному часі. Чим вище адаптивні здібності системи, тим більш стійкою буде її робота в нестаціонарній середовищі. При цьому слід зауважити, що адаптивність не завжди веде до стійкості; іноді вона призводить до абсолютно протилежного результату. Наприклад, адаптивна система з параметрами, швидко змінюються в часі, може також швидко реагувати і на сторонні збудження, що викличе втрату продуктивності. Для того щоб

використовувати всі переваги адаптивності, основні параметри системи повинні бути досить стабільними, щоб можна було не враховувати зовнішні перешкоди, і досить гнучкими, щоб забезпечити реакцію на істотні зміни середовища.

- *Потенційно надвисока швидкодія.*

Нейронні мережі мають потенційно надвисоку швидкодю за рахунок використання масового паралелізму обробки інформації.

Однак, що стосується загального підходу використання нейронних мереж на практиці, він кардинально відрізняється від звичайного детермінованого підходу «запрограмував, працює - значить, працює завжди». Нейронні мережі за своєю природою є імовірнісними моделями, і підхід до них повинен бути зовсім іншим.

Існує велика кількість алгоритмів навчання нейронних мереж (*Левенберга-Марквардта, BFGS, Conjugate Gradients, ...*). Для оптимізації навчання нейронних мереж широко використовуються генетичні алгоритми. Генетичний алгоритм є ітераційним і дає лише наближене рішення, що компенсується областю його застосування та швидкістю обчислень при обмежених обчислювальних ресурсах. Він може застосовуватися для підстроювання ваг прихованих і вихідних шарів при фіксованому наборі зв'язків і широко використовується в задачах оптимізації та навчання нейронних мереж.

У рамках розробки штучного інтелекту для ігрового застосунку з використанням нейронних мереж можна виділити 2 основних компоненти: Систему навчання та донавчання нейронних мереж та сам ігровий застосунок. Ці 2 компоненти будуть зв'язані каналом передачі даних по якому буде передаватись нейронна мережа від ігрового застосунку до системи навчання (для донавчання) та у зворотньому напрямку(для безпосереднього використання).

Висновки:

- проведено ряд досліджень у галузі застосування нейронних мереж та генетичних алгоритмів для розробки штучного інтелекту в ігрових застосунках.
- вивчено історію розвитку нейронних мереж, основну класифікацію типів та топологій нейронних мереж.
- досліджено засоби розробки кроссплатформених ігрових застосунків.
- Досліджено методи уніфікації підходу до передачі даних між застосунками, що запущені на різних апаратно – програмних платформах.

Література

1. Hinton G., Deng L., Yu D., Dahl G., Mohamed A., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T. and Kingsbury B. Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, Vol. 29, No. 6, 2012, pp. 82 – 97.
2. Ciresan D., Meier U., Masci J and Schmidhuber J. Multi-column Deep Neural Network for Traffic Sign Classification. Neural Networks, Vol. 34, August 2012, pp. 333 – 338
3. С. Осовский. Нейронные сети для обработки информации – пер. с польского. М.: Финансы и статистика, 2002.–344с.
4. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006 – 738 p.
5. С. Хайкин. Нейронные сети: полный курс. Вильямс, 2006.

ДОСЛІДЖЕННЯ ТА СТВОРЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДІВ

Запорізька державна інженерна академія, кафедра ПЗАС

Навчання існує з самого початку історії людства. Ще у стародавньому світі люди створювали примітивні гуртки, де передавали знання з покоління в покоління. Освітня галузь розвивалася разом із людством, оскільки знання та досвід накопичувалися у геометричній прогресії, що призводило до створення більш досконалих методів навчання. Такий зріст знань та учнів призвів до створення цілих навчальних закладів, які виконували об'єднуючі та керуючі функції.

У наш час такі навчальні заклади мають складні структури з багатьма підрозділами і стає досить складно керувати навчальним процесом. Інформаційна база вищих навчальних закладів, зазвичай, є недостатньо структурованою та нецентралізованою. Також, такі бази зазвичай зберігаються у паперовому вигляді, що, нажаль, є дуже не зручним та ненадійним при такому обсязі даних. Тому стає все більш актуальнішою ідея створення централізованої автоматизованої системи управління навчальним процесом, яка б дозволила структурувати дані навчального закладу та спростити роботу з ними.

Отже, автоматизована система управління (АСУ) навчальним процесом вищих навчальних закладів — це інструмент або набір інструментів для комплексної або часткової автоматизації процесів керування ВНЗ.

На сьогодні вже існує певна кількість таких систем. Але, нажаль, досить часто потреби окремих навчальних закладів не можуть бути узагальнені, що унеможливорює створення універсальної системи, яка б задовольнила усім вимогам. Слід зазначити, що все ж таки існують загальні системи, які можуть бути адаптовані під більшість навчальних закладів.

Розробка АСУ передбачає створення централізованої бази даних та окремих логічних модулів для роботи зі створеною базою. Розглянемо структуру АСУ ВНЗ на прикладі деяких існуючих рішень.

АСУ "ВНЗ"

Ця система є найбільш відомою в Україні, яка підтримується з 2000 року. До складу цієї системи входять такі компоненти як:

- АС «Приймальна комісія» — автоматизує роботу приймальної комісії та забезпечує зберігання особистих справ абітурієнтів та інформації про результати вступу для використання в підрозділах установи;
- АС «Деканат» — основна частина системи АСУ «ВНЗ», яка виконує головні функції з автоматизації адміністративних та навчально-методичних процесів;
- АС «Студмістечко» — автоматизується діяльність дирекції студентського містечка навчального закладу, спрощується та вдосконалюється документообіг, підвищується оперативність, узгодженість і достовірність даних.

Кожен з цих компонентів в свою чергу також поділяється на більш дрібні модулі. Що забезпечує гнучкість цієї системи. Але, оскільки ця система орієнтована на багату кількість навчальних закладів, вона є досить громіздкою та зазвичай потребує додаткового супроводу.

АСУ "Університет"

Дана система є менш узагальненою та орієнтована виключно на ВНЗ III та IV рівня акредитації.

Складається з окремих модулів:

- Структура ВНЗ — є основним модулем. Моделює структуру ВНЗ;
- WEB – сайт ВНЗ — надає Web-інтерфейс;
- Навчальна частина — модуль для керування навчальним процесом;
- Кафедра — модуль керування даними про співробітників, включаючий в себе особові справи, розподіл навчального навантаження та ставок, а також аналіз та звітування пов'язаних процесів;
- Навчальний розклад — модуль, який відповідає за формування навчального розкладу та зручні засоби конвертації у поширені файлові формати;
- Абітурієнт — модуль, який автоматизує усі напрямки роботи Приймальної комісії.

Наведені приклади є не єдиними у своєму роді. Окрім цих рішень існує багато індивідуальних АСУ для різних навчальних закладів.

Незважаючи на те, що дані системи існують більш ніж 10 років, велика кількість ВНЗ досі не мають навіть часткової інтеграції таких автоматизованих систем, що значно уповільнює та ускладнює їх роботу.

Висновки:

- АСУ вищих навчальних закладів має досить складну структуру та, зазвичай, є громіздкою;
- Розробка такої АСУ потребує значних зусиль, оскільки охоплює дуже велику кількість даних, але, на щастя, добре структурованих;
- Більшість АСУ узагальнені та потребують супроводу;
- АСУ, як правило, мають модульну структуру, що дозволяє проводити поступову інтеграцію.

Література

1. Про систему | АСУ «ВНЗ». [Електронний ресурс]. — <http://vuz.osvita.net/>
2. АСУ «ВНЗ». [Електронний ресурс]. — http://uk.wikipedia.org/wiki/АСУ_«ВНЗ».
3. Автоматизована система управління вищим навчальним закладом III - IV рівня акредитації. [Електронний ресурс]. — <http://www.unitex.com.ua>

Додоєнко В. Є., магістрант, Скрипник І. А., доцент, науковий керівник

СЕМАНТИЧНЕ ПОРІВНЯННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ ЗАСОБАМИ TEXT MINING

Запорізька державна інженерна академія, кафедра ПЗАС

Вперше «ручні» техніки Text Mining з'явилися в середині 1980-х, а в наступне десятиліття розвиток технологій дозволило значно їх вдосконалити. У міждисциплінарному сенсі Text Mining лежить на стику пошуку інформації, Data Mining, машинного самонавчання, статистики та комп'ютерної лінгвістики.

В даний час багато провідних виробників програмного забезпечення пропонують свої продукти та рішення в області Text mining. Найбільш помітні серед них це: Intelligent Miner for Text (IBM), TextAnalyst, WebAnalyst, Oracle Text, Text Miner, та інші. Як правило - це масштабовані системи, в яких реалізовані різні математичні та лінгвістичні алгоритми аналізу текстових даних. Вони мають розвинені графічні інтерфейси, багаті можливості візуалізації та маніпулювання з даними, надають доступ до різних джерел даних, функціонують в архітектурі клієнт-сервер.

Технологія Text Mining являє собою одну з різновидів методів Data Mining і має на увазі процеси вилучення високоякісної інформації з текстових масивів. Як і більшість когнітивних технологій, Text Mining - це алгоритмічне виявлення раніше не відомих зв'язків і кореляцій у вже наявних текстових даних. Як правило, Text Mining має на увазі процес структурування ввідних текстових даних, витяг шаблонів з уже структурованих даних, і фінальну оцінку та інтерпретацію отриманих результатів. По суті, аналіз текстів і Text Mining - це набір лінгвістичних, статистичних технік, а також технік машинного самонавчання, які здатні моделювати і структурувати інформаційний контент. Типові стадії і завдання Text Mining включають в себе:

1. Пошук інформації і виявлення вихідних даних - це підготовчий крок, який включає в себе збір або виявлення набору текстових матеріалів для аналізу.
2. Використання статистичних методів аналізу, а також процесів обробки природної мови і засобів лінгвістичного аналізу.
3. Виявлення смислів - використання статистичних та інших технік для виявлення пойменованих ознак тексту – згадувань людей, організацій, місць, символів, аббревіатур і так далі. Контекст допомагає визначити, що саме позначає те чи інше слово в конкретному входженні.
4. Виявлення шаблонів - можна виявити, в яких шаблонах в тексті представлені ті чи інші смисли.
5. Виявлення перехресних посилань - виявлення визначень та інших ознак, які відносяться до одних і тих же об'єктів.
6. Виявлення взаємозв'язків, фактів і подій - пошук зв'язків між різними смислами, укладеними в текстової інформації.
7. Змістовний аналіз - розпізнавання суб'єктивного (а не фактичного) матеріалу і виявлення різних форм оціночної інформації - смислів, думок, настроїв, емоцій. Технології аналізу дозволяють вивчати смисли на рівні тем, концептів.
8. Кількісний аналіз тексту - використання набору технік, запозичених їх соціальних наук, які полягають в тому, що людина або комп'ютер витягує семантичні або граматичні зв'язки між окремими словами, щоб зрозуміти сенс стилістичних шаблонів, провести психологічне профілювання і так далі.

Важливе завдання технології Text Mining пов'язана з витяганням з тексту його характерних елементів або властивостей, які можуть використовуватися як метадані документа, ключових слів, анотацій. Інше важливе завдання полягає у віднесенні документа до деяких категорій із заданої схеми їх систематизації. Text Mining також забезпечує новий рівень семантичного пошуку документів. Можливості сучасних систем Text Mining можуть застосовуватися при управлінні знаннями для виявлення шаблонів в тексті, для автоматичного «виштовхування» або розміщення інформації по цікавлять користувачів профілям, створювати огляди документів. Основні елементи Text Mining Відповідно до вже сформованої методологією до основних елементів Text Mining відносяться:

- класифікація (classification) При класифікації текстів використовуються статистичні кореляції для побудови правил розміщення документів в певні категорії. Завдання класифікації - це класична задача розпізнавання, де за деякою контрольною вибірці система відносить новий об'єкт до тієї чи іншої категорії. Особливість систем Text Mining полягає в тому, що кількість об'єктів та їх атрибутів може бути дуже великою, тому повинні бути передбачені інтелектуальні механізми оптимізації процесу класифікації.

- кластеризація (clustering) Кластеризація базується на ознаках документів, які використовує лінгвістичні та математичні методи без використання певних категорій. Результат - таксономія або візуальна карта, яка забезпечує ефективний охоплення великих обсягів даних. Кластеризація в Text Mining розглядається як процес виділення компактних підгруп об'єктів з близькими властивостями. Система повинна самостійно знайти ознаки і розділити об'єкти по підгрупах. Кластеризація, як правило, передуює класифікації, оскільки дозволяє визначити групи об'єктів. Розрізняють два основних типи кластеризації - ієрархічну і бінарну. Кластеризація застосовується при реферування великих документальних масивів, визначення взаємопов'язаних груп документів, спрощення процесу перегляду при пошуку необхідної інформації, знаходження унікальних документів з колекції, виявлення дублікатів або дуже близьких за змістом документів.

- побудова семантичних мереж Побудова семантичних мереж або аналіз зв'язків, які визначають появу дескрипторів (ключових фраз) у документі для забезпечення навігації.

- витяг фактів, понять (feature extraction) Одержання деяких фактів з тексту з метою поліпшення класифікації, пошуку та кластеризації.

- суммаризація (summarization),
- відповідь на запити (question answering),
- тематичне індексування (thematic indexing),
- пошук за ключовими словами (keyword searching).
- засоби підтримки та створення таксономії (ontologies) і тезаурусів (thesauri).

Висновки:

- проведено ряд досліджень у галузі застосування штучного інтелекту у семантичному аналізі текстів
- досліджено передумови появи подібних систем і області їх застосування
- досліджено вже розроблені системи та рішення в області Text mining
- досліджено можливості та алгоритми які відносяться до Text mining
- розроблено програмну систему семантичного порівняння текстів

Література

1. Пескова О.В. Алгоритмы классификации полнотекстовых документов // Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. — М.: МИЭМ (Московский государственный институт электроники и математики), 2011. — С. 212.
2. Глубинный анализ текстов. Технология эффективного анализа текстовых данных. / Дмитрий Ланде. — СМР Ukraine 10 - 2003.
3. [Електронний ресурс] / Text Mining в системе управления знаниями. — Режим доступа: <http://www.smart-edu.com/text-mining-v-sisteme-upravleniya-znaniyami.html/>, вільний.

Заряєв Д.В., Заряєва Д.М., Лехман Р.В., Близнюк С.І., студенти,
Безверхий А. І., доцент, науковий керівник

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ПРИЙМАЛЬНОЇ КОМІСІЇ ВНЗ

Запорізька державна інженерна академія, кафедра ПЗАС

Одним з підрозділів Запорізької державної інженерної академії (надалі ЗДІА) є приймальна комісія, що відповідає за прийом абітурієнтів, молодших спеціалістів, бакалаврів на різні курси ЗДІА денної та заочної форм навчання, поновлення студентів, переведення з інших ВНЗ, тестування знань студентів та генерації документації. Система складається з 3-х компонентів – “Matriculant”, “Test system” та “Card builder”.

Архітектура системи має вигляд:

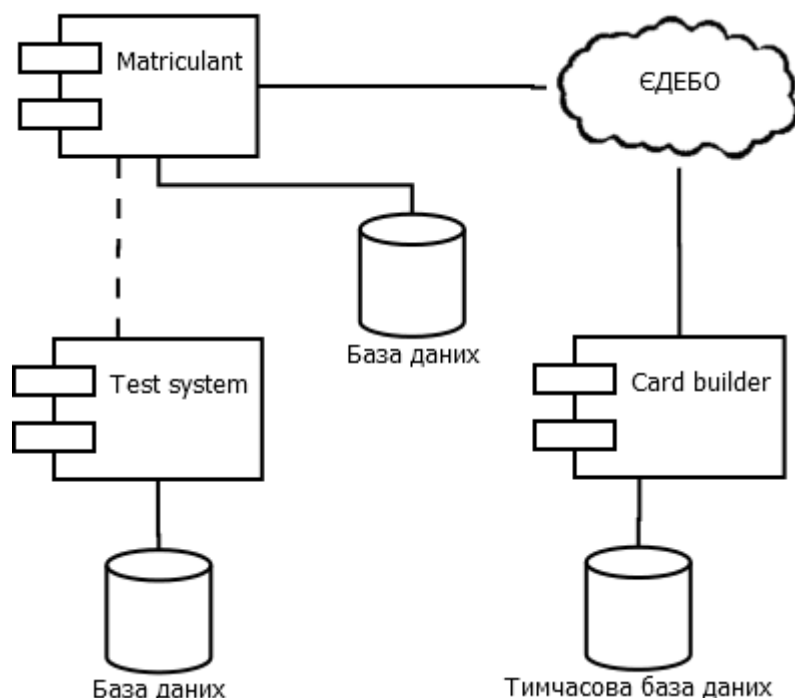


Рис. 1. Архітектура системи.

Локальна частина системи розташована на двох серверах:

1. “Matriculant”, “Card builder” та їхні бази даних знаходяться на одному сервері.
2. “Test system” та відповідна база даних знаходяться на іншому сервері.

Призначення системи:

1. “Matriculant”
 - облік вступників;
 - генерація документації;
 - синхронізація з ЄДЕБО;
 - редагування даних.
2. “Test system”

- тестування вступників;
 - додавання та редагування даних;
 - генерація документації.
3. “Card builder”
- синхронізація з ЄДЕБО;
 - генерація документації.

Були сплановані наступні вимоги до майбутнього застосунку:

- висока надійність;
- задовільна швидкість роботи;
- зручність;
- оптимальність інтерфейсу;
- коректна обробка помилок без втрати введеної інформації;
- комбінації клавіш;
- естетичний вигляд застосунку;
- використання вільного програмного забезпечення.

Згідно з вимогами до застосунку були обрані наступні засоби реалізації:

- реалізація клієнтського застосунку на серверній платформі Node.js, що використовує мову програмування JavaScript;
- використання JavaScript-фреймворка AngularJS;
- використання вільної об'єктно-реляційної системи управління базами даних PostgreSQL.

Висновки:

- Спроектвана та розроблена архітектура застосунку, що забезпечує інформаційну підтримку приймальної компанії ВНЗ.
- Створене програмне забезпечення, що реалізує базові елементи застосунку.
- Налагоджені зв'язки між окремими підсистемами застосунку та синхронізація з ЄДЕБО.
- Систему було протестовано протягом вступної компанії 2014 року. Знайдено та виправлено незначні дефекти архітектури та застосунків.

Література

1. Електронний ресурс: <http://angularjs.org>
2. Електронний ресурс: <http://nodejs.org>
3. Електронний ресурс: <http://getbootstrap.com>

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ПОШУКУ ОПТИМАЛЬНОГО МІСЦЯ ПАРКУВАННЯ

Запорізька державна інженерна академія, кафедра ПЗАС

Система підтримки прийняття рішень (СППР; англ. Decision Support System, DSS) — комп'ютеризована система, яка шляхом збору та аналізу великої кількості інформації може впливати на процес прийняття управлінських рішень.

СППР — це інтерактивна комп'ютерна система, яка призначена для підтримки різних видів діяльності при прийнятті рішень із слабоструктурованих або неструктурованих проблем. Інтерес до СППР, як перспективної галузі використання обчислювальної техніки та інструментарію підвищення ефективності праці в сфері управління економікою, постійно зростає. У багатьох країнах розробка та реалізація СППР перетворилася на дільницю бізнесу, що швидко розвивається.

Штучний інтелект — це штучні системи, створені людиною на базі ЕОМ, що імітують розв'язування людиною складаних творчих завдань. Створенню інтелектуальних інформаційних систем сприяла розробка в теорії штучного інтелекту логіко-лінгвістичних моделей. Ці моделі дають змогу формалізувати конкретні змістовні знання про об'єкти управління та процеси, що відбуваються в них, тобто ввести в ЕОМ логіко-лінгвістичні моделі поряд з математичними. Логіко-лінгвістичні моделі — це семантичні мережі, фрейми, продукувальні системи — іноді об'єднуються терміном «програмно-апаратні засоби в системах штучного інтелекту».

Сучасні системи підтримки прийняття рішень виникли у результаті злиття управлінських інформаційних систем і систем управління базами даних, як системи, що максимально пристосовані до розв'язування задач щоденної управлінської діяльності, і є інструментом, щоб надати допомогу тим, хто вирішує (робить вибір). За допомогою СППР може проводитись вибір рішень у певних неструктурованих і слабо структурованих задачах, у тому числі й тих, що мають багато критеріїв.

СППР, як правило, узагальнюють результати досліджень з кількох дисциплін, та включають у себе елементи теорії баз даних, штучного інтелекту, інтерактивних комп'ютерних систем, методів імітаційного моделювання.

Як зазначено у [1], "... з моменту появи перших розробок у галузі СППР, не було дано чіткого визначення СППР...".

Ранні визначення СППР (на початку 70-х років минулого століття) показували такі моменти:

1. Можливість оперування з неструктурованими чи слабкоструктурованими задачами, на відміну від задач, із якими має справу дослідження операцій.
2. Інтерактивні автоматизовані (тобто реалізовані на базі комп'ютера) системи.
3. Розподіл даних та моделей.

Аналіз еволюції систем СППР дає можливість виділити 2 покоління СППР:

- перше покоління розроблялося в період із 1970 до 1980 р.;
- друге - з початку 1980 р. і дотепер.

Перше покоління СППР майже цілком повторювало функції звичайних управлінських систем у відношенні допомоги (комп'ютеризованої) у прийнятті рішень. Основні компоненти СППР мали такі ознаки:

-керування даними - велика кількість інформації, внутрішні і зовнішні банки даних, обробка та оцінювання даних;
-керування обчисленням (моделюванням) - моделі, розроблені спеціалістами в галузі інформатики для спеціальних проблем;
-користувацькі інтерфейси (мова спілкування) - мови програмування, розроблені для великих ЕОМ, що використовуються винятково програмістами.

СППР другого покоління вже мають принципово нові ознаки:

-керування даними - необхідна і достатня кількість інформації про факти згідно з прийняттям рішень, що охоплюють сховані припущення, інтереси і якісні оцінки;
-керування обчисленням і моделюванням - гнучкі моделі, що відображають засіб мислення особи, приймаючої рішення, у процесі прийняття рішень;
-інтерфейс користувача - програмні засоби дружні користувачу; звична мова, безпосередня робота кінцевого користувача.

Ціль і призначення СППР другого покоління можна визначити так:

-допомога у розумінні розв'язуваної проблеми. Сюди належить структуризація проблеми, генерування постановок задач, визначення переваг, формування критеріїв;
-допомога у рішенні задач: генерування і вибір моделей і методів, збір і підготування даних, виконання обчислень, оформлення і видача результатів;
-допомога у проведенні аналізу типу «Що? . Коли?» і т.п., пояснення ходу рішення; пошук і видача аналогічних рішень у минулому і їхні результати.

Дружні людині СППР дають можливість вести рівноправний діалог із ПЕВМ, використовуючи звичайні мови спілкування. Системи можна підбудовувати під стиль мислення користувача, його знань і фахової підготовки, а також під засоби роботи.

Висновки:

- проведено ряд досліджень у галузі застосування СППР;
- вивчено історію розвитку СППР та їх основну класифікацію;
- досліджено область застосування СППР (пошук оптимального місця паркування);

Література

1. Keen P.G.W. Decision Support Systems: The next decades // Decision Support Systems, 1987. – v. 3. - pp. 253-265.
2. Ларичев О.И., Петровский А.В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер.Техническая кибернетика. - Т.21. М.: ВИНТИ, 1987,
3. Сараев А.Д., Щербина О.А. Системный анализ и современные информационные технологии //Труды Крымской Академии наук. - Симферополь: СОНАТ, 2006. - С. 47-59

Критина І. А., спеціаліст, Скрипник І.А., к. ф.-м. н., доцент, науковий керівник

СТВОРЕННЯ WEB-БРАУЗЕРНОЇ БАГАТОКОРИСТУВАЦЬКОЇ ОНЛАЙНОВОЇ РОЛЬОВОЇ ГРИ

Запорізька державна інженерна академія, кафедра ПЗАС

Багатокористувацька онлайнова рольова гра або ММОРПГ (англ. Massively multiplayer online role-playing game, MMORPG) - комп'ютерна гра, в якій жанр рольових ігор поєднується з жанром масових онлайн-ігор. MMORPG можуть бути представлені в тому числі і в браузерному вигляді, проте основною рисою жанру є взаємодія великого числа гравців у рамках віртуального світу.

Як і в будь-якій іншій рольовій грі, гравець приймає на себе роль персонажа (часто належить до фентезійного або науково-фантастичного світу) і починає керувати різноманітними його діями. MMORPG відрізняються від однокористувацьких або невеликих багатокористувацьких рольових онлайн-ігор не тільки кількістю гравців, але й постійно існуючим ігровим світом (який зазвичай підтримується силами видавця гри), існуючим незалежно від виходу з нього окремого гравця.

У MMORPG грають по всьому світу. Доходи світової індустрії MMORPG в 2005 році перевищили 500 млн доларів США, тоді як доходи західних видавців у 2006 році перевищили US \$ 1 млрд. До 2008 року плата за підписку західних гравців зросла до US \$ 1,4 млрд. У World of Warcraft, популярної MMORPG, в 2013 році було більше 7 млн гравців. Після запуску в 2011 році Star Wars: The Old Republic була оголошена «самої стрімко розвивається MMORPG в історії» через те, що протягом 3 днів до неї було привернуто увагу понад 1 млн гравців.

Сучасні MMORPG вельми істотно відрізняються від ранніх ігор у жанрі, проте їх усіх об'єднує ряд основних особливостей. Такими є ряд спільних рис: постійно існуюче ігрове оточення, різні форми розвитку, соціальні взаємодії всередині ігор, культура ігрового світу, особливості архітектури, членство в групі, персоналізація ігрових персонажів.

Більшість сучасних MMORPG використовують мережеву архітектуру клієнт-сервер. На сервері підтримується постійно існуючий віртуальний світ, а гравці можуть до нього підключатися через клієнтські програми. Через клієнтську програму гравець може отримати доступ або до всього ігровому світу без обмежень, або тільки до базової частини гри, а до деяких областей з «розширень» гри може вимагатися оплата цього контенту. Приклади ігор, що використовують другу модель - EverQuest і Guild Wars. Зазвичай гравці повинні одноразово придбати клієнтську програму, проте наростаючою тенденцією для MMORPG є використання такого заздалегідь доступного «тонкого клієнта» як браузер.

Залежно від числа гравців і особливостей архітектури, MMORPG можуть працювати на численних серверах, кожен з яких представляє окремий незалежний ігровий світ, при цьому гравці, що знаходяться на різних серверах, не можуть взаємодіяти один з одним. Яскравим прикладом тут є World of Warcraft, в якому кожен сервер може вміщати декілька тисяч ігрових персонажів. Як правило, в MMORPG кількість персонажів, одночасно присутніх в ігровому світі, обмежено декількома тисячами. Хорошим прикладом зворотного концепції є EVE Online, де сервер здатний вміщати декілька десятків тисяч гравців час від часу (понад 60 тисяч в червні 2010). У деяких іграх одного разу створений персонаж може вільно переміщатися між світами, але в кожен момент часу він повинен бути присутнім тільки одному сервері (наприклад, Seal Online: Evolution), в інших іграх персонаж повинен знаходитися лише в тому світі, де був створений.

Реалізація браузерних MMORPG суттєво відрізняються від реалізації звичних клієнтських ігор цього жанру. В браузерному варіанті достатньо значна частина бізнес-логіки «перекладається» на систему керування базами даних. По суті браузерну гру можна назвати таким собі «інтерфейсом» до бази даних.

Розглянемо переваги і недоліки браузерних MMORPG.

Переваги:

- Відносна простота в розробці (в порівнянні з клієнтськими іграми).
- Для підтримки гри потрібно значно менше ресурсів.
- Гравці мають змогу відразу ж розпочати грати (немає необхідності встановлювати клієнт гри).
- Ігрові сервери браузерних MMORPG можуть «витримувати» колосальну кількість гравців (в порівнянні з серверами клієнтських ігор).
- Система значно легше масштабується.

Недоліки:

- Низька інтерактивність ігрового процесу.

Обрані засоби реалізації:

- Мова: PHP
- Розроблений php-фреймворк: MvcBox
- СКБД: PostgreSQL
- ОС: Debian
- Web-server: NGINX
- Кеш: APC, Redis / Memcache
- HTTP кеш: Varnish

Висновки:

- Проведено ознайомлення з предметною областю.
- Отримані і закріплені знання з деяких технологій (JS, CSS, HTML, PHP).
- Реалізований php-фреймворк.
- Спроектвана архітектура системи.
- Спроектвана архітектура бази даних.
- Реалізована гра.

Література

1. Офіційний сайт PostgreSQL <http://www.postgresql.org/>
2. Офіційний сайт PHP <http://php.net/>
3. Матеріали з інтернет-ресурсу <http://habrahabr.ru/>
4. Матеріали з інтернет-ресурсу <https://www.wikipedia.org/>
5. Матеріали з інтернет-ресурсу <http://www.php-fig.org/>

БАГАТОПЛАТФОРМНИЙ ЗАСТОСУНОК ІЗ ВИКОРИСТАННЯМ ФРЕЙМВОРКІВ ANGULARJS TA IONIC BACKING TRACKER

Запорізька державна інженерна академія, кафедра ПЗАС

Backing Tracker – застосунок, який встане у нагоді будь-якій творчій людині, що не може існувати без музики. Ідея створити цей застосунок з’явилась тому, що досі не існує жодного, який допоміг би розвиватись музично.

Які ж переваги цього застосунку?

- універсальність (кожен може користуватись, як співак, так і барабанщик);
- дуже зручний інтерфейс (все, що потрібно зробити, - змістити вибраний повзунок);
- багатоплатформність (підтримка систем Tizen, webOS, Android, Apple iOS, Blackberry, Samsung Bada, Windows Phone).

Apache Cordova — фреймворк для створення мобільних застосунків, що продовжує розвиток платформи PhoneGap, після передачі проекту компанією Adobe в руки фонду Apache. Одночасно компанія Adobe представляє заснований на єдиній кодової базі з Apache Cordova продукт PhoneGap, функціонально ідентичний Apache Cordova.

Для виконання дипломної роботи я обрав Apache Cordova, тому що фреймворк дозволяє створювати універсальні мобільні застосунки, що працюють на різних мобільних платформах, з використанням стандартних веб-технологій (HTML5, CSS3 і JavaScript). Використання Apache Cordova дозволяє створювати застосунки, що функціонують на широкому спектрі мобільних платформ, включаючи Tizen, webOS, Android, Apple OS, Blackberry, Samsung Bada і Windows Phone.

AngularJS – це JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових застосунків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Ionic - це html5 фреймворк для створення гібридних мобільних застосунків. Перший реліз фреймворку пройшов у листопаді 2013. Ionic побудований на AngularJS і використовує його функціонал для оперування DOM, в той час як Ionic надає користувацький інтерфейс.

В результаті роботи було:

- вивчено існуючі засоби для рішення задачі;
- розроблено багатомодульну систему для навчання/розвитку музично;

Модулі реалізованої програмної системи:

- графічний інтерфейс за допомогою Ionic Framework;
- директива програвання аудіо доріжки з візуалізацією та налаштуваннями;
- база даних;
- головний контролер для керування застосунком.

Під час виконання даної роботи було проведено ряд досліджень бібліотек для створення застосунків для мобільні пристрої. Було детально вивчено історію створення бібліотек, можливості, переваги та недоліки різних бібліотек. Вивчено бібліотеки AngularJS та Ionic, можливості цих бібліотек та порівняно їх ефективність роботи.

Висновки:

- Розглянуті основні бібліотеки для вирішення поставленої задачі.
- Досліджено можливість застосування фреймворків Apache Cordova, AngularJS, Ionic.
- Розроблено програмну систему.

Література

1. [електронний ресурс] <https://docs.angularjs.org/guide> - документація фреймворку
2. [електронний ресурс] <http://ionicframework.com/docs/> - документація фреймворку
3. [електронний ресурс] <http://habrahabr.ru/> - найкращий IT - ресурс

Матвій М.В., студент, Яковлева С.О., ст. викладач, науковий керівник

РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ ОБРОБКИ ТА РОЗМІЩЕННЯ ФОТОГРАФІЙ І ЗОБРАЖЕНЬ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Запорізька державна інженерна академія, кафедра ПЗАС

Початком популярності соціальних мереж можна вважати 2003 – 2004 роки, коли були засновані такі мережі як Facebook, LinkedIn та MySpace. В країнах СНД соціальні мережі стали популярними в 2006 році, з появою Однокласників та Вконтакті. Станом на липень 2014 року кількість користувачів тільки Facebook досягла відмітки 1.32 млрд користувачів, які відвідували соціальну мережу хоча б раз на місяць, а добова активна кількість становить 720 тисяч користувачів. Добова аудиторія найпопулярнішої в країнах СНД мережі Вконтакті становить 70 тисяч відвідувань (станом на січень 2015 року). Таким чином дослідження проблем і взаємодії з соціальними мережами є надзвичайно актуальними, оскільки майже кожна п'ята людина на планеті користується однією з мереж або навіть декількома з них.

Останнім часом все більш популярним стає завантаження фотографій з кожного відвіданого місця та заходу до соціальних мереж. Для такої мети були створені навіть окремі соціальні мережі, наприклад Instagram та Picasa. Тому є доцільним створення застосунку для спрощення процесу розміщення фотографій у реальному часі. Аналіз ринку на подібні продукти показав, що існують подібні застосунки, наприклад Everypost, який дозволяє завантажувати знімки та відео зі свого смартфона до декількох соціальних мереж одночасно. Такий підхід зручний, якщо користувач фотографує власноруч і має прямий доступ до своїх знімків, але не дозволяє переглядати і розміщувати фотографії, зроблені іншою людиною на цьому ж заході.

За наявності на будь-якого роду заході фотографа, який синхронізує зроблені знімки з потоком на комп'ютері і використанні цього застосунку користувач отримує ряд переваг, таких як: можливість у реальному часі переглядати щойно зроблені фотографії з місця події, застосувати до необхідних фотографій фільтри, наприклад сепія або чорно-білий, після чого завантажити обрані знімки до бажаної соціальної мережі, додавши підпис. Такий підхід дозволяє користувачу не очікувати надання фотографій фотографом заходу, а також економити час на фільтрації і обробці знімків після завершення заходу.

Перед застосунком постають такі задачі як відстеження появи нових фотографій, можливість зробити фотографію використовуючи під'єднану до комп'ютера веб-камеру, вибір фото, які потрібно завантажити та можливість її розміщення до обраної соціальної мережі. Одним з найголовніших завдань була розробка алгоритму віртуалізації відображення списку наявних фотографій, а оскільки їх може бути дуже багато (10000 і більше), завантажувати всі одночасно у оперативну пам'ять не можна. Також необхідно було розробити екранну клавіатуру для можливості використання застосунку на touch-моніторі або телевізорі.

Тож, для успішного виконання поставленої задачі необхідно реалізувати:

- Елемент керування для перегляду списку фотографій з вбудованим механізмом віртуалізації даних для підвищення швидкості роботи.
- Механізм синхронізації фотографій у темі з фотографіями, наявними для перегляду в програмі.
- Тривимірний відображувач колекції зображень – Cover Flow.
- Додавання візуальних ефектів до фотографій.

Для реалізації програми було обрано мову програмування C# та технологію створення графічного інтерфейсу WPF. Windows Presentation Foundation (WPF, кодова назва —

Avalon) — графічна (презентаційна) підсистема в складі .NET Framework 3.0, що має пряме відношення до XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista, а також доступна для установки в Windows XP Service Pack 2 і Windows Server 2003. Це перше реальне оновлення технологічного середовища призначеного для користувача інтерфейсу з часу випуску Windows 95. Воно включає нове ядро, яке повинне замінити GDI і GDI+, використовувани на нинішній Windows-платформі. WPF є високорівневим об'єктно-орієнтованим функціональним шаром (англ. framework), що дозволяє створювати двовимірні та тривимірні інтерфейси.

Розроблений застосунок складається з чотирьох модулів:

- **Модуль відстеження фотографій**

Для реалізації відстеження додавання або видалення фотографій в теці в реальному часі було створено механізм, який сканує теку кожні 5 секунд і порівнює список файлів з попереднім і додає нові фото до застосунку. Інтервал сканування необхідний для зменшення навантаження на процесор комп'ютера, але при цьому не повинен бути занадто великим щоб не змушувати користувача очікувати. Інтервал у 5 секунд задовольняє цим критеріям, тому його і було обрано.

- **Модуль відображення фотографій**

Основною проблемою цього модулю була економія оперативної пам'яті при наявності великої кількості фотографій, які повинні відображатися в застосунку для вибору користувачем. Тому була розроблена система віртуалізації, яка завантажує в пам'ять лише ті фотографії, які видно в даний момент часу. Під час навігації по списку, фотографії, які виходять за межі екрану, видаляються з оперативної пам'яті, а на їх місце завантажуються нові. Такий підхід дозволяє оброблювати необмежену кількість фотографій.

- **Модуль завантаження фотографій**

Для взаємодії з соціальними мережами були використані їх API. У кожній мережі API різний, тому уніфікованого рішення для завантаження фотографії у будь-яку мережу немає. На даний момент реалізовано завантаження фотографій у Facebook, VK, Twitter, Tumblr, Foursquare, а також відправка на електронну пошту.

- **Модуль відображення графічного інтерфейсу користувача**

- Так як застосунок буде використовуватись переважно на різного роду заходах, то роздільна здатність екрану і його розміри можуть бути різними. Тому важливо, щоб інтерфейс був гнучким і не його вигляд не залежав від пристрою, на якому виконується застосунок. Для цього була використана технологія WPF, яка дозволяє досягнути необхідного результату дуже просто.

Висновки:

- Була досліджена методологія розробки програмного забезпечення з використанням технології WPF та мови програмування C#.
- Досліджена проблема віртуалізації даних в програмних застосунках.
- Досліджена проблема взаємодії користувачів з різними соціальними мережами.

Література

1. Натан А. WPF 4.5 Unleashed – 2013. 864 с.
2. Віртуалізація даних у WPF [Електронний ресурс] – Режим доступу: <http://habrahabr.ru/post/208792/>, вільний.

Міхайлуца О.М., слухач гр. ЦБО СП-13, Полякова Н.П., доцент, науковий керівник

АВТОМАТИЗОВАНА КОМП'ЮТЕРНА СИСТЕМА ОБЛІКУ КВАЛІФІКАЦІЙ ТАНЦЮРИСТІВ З УРАХУВАННЯМ ІСТОРІЇ ТАНЦЮВАЛЬНИХ ЗМАГАНЬ

Запорізька державна інженерна академія, кафедра ПЗАС

Асоціацію сучасного естрадного танцю України (АСЕТУ) розглядають як організацію створену молодими танцюристами і викладачами сучасного танцю з метою задоволення та захисту національно-культурних, спортивних та інших спільних інтересів своїх членів. Разом з тим, вона є структурою, що сприяє розвитку сучасних та естрадних танців на території України, підвищенню ролі даних видів діяльності у загальному і гармонійному розвитку особистості.

На даний момент АСЕТУ найпотужніша і прогресивна танцювальна організація в Україні, діяльність якої активно проходить в 17 областях України, членами якої є понад 300 танцювальних об'єднань і більше 20.000 танцюристів, які розвивають більше 25 напрямків танцю. Щорічно відбувається більше 100 заходів, на будь-який смак, підсумки яких представлені в календарі заходів АСЕТУ. Незважаючи на те, що членами АСЕТУ є колективи та танцювальні центри в різних містах України, організація є установою з чітко сформованою структурою. Спланована функціональність та зв'язки між гілками асоціації сприяють організованості, відсутності помилок та безперервному процесу роботи. Досягнення основних завдань АСЕТУ залежить від сформованості її змісту, тобто від того, як поставлено процес реалізації програм розвитку сучасного та естрадного танцю на території кожного регіону країни. Кожен центр має керівника танцювального колективу та декілька тренерів. Танцюристи, що займаються у тренерів танцювальних центрів, проходять он-лайн реєстрацію для участі у змаганнях, що проводяться АСЕТУ. Танцюристи, що є дійсними або асоціативними членами організації можуть отримувати знижку на участь у турнірах та бонуси на проходження майстер-класів.

Для того, щоб змагання з сучасного та естрадного танцю виглядали цікаво та був присутній спортивний дух змагань, у спортивних танцях введена система класів (ліг), що відображає рівень підготовки танцюристів (аналогічно спортивним розрядам в різних видах спорту, «поясам» в єдиноборствах) і система вікових категорій, що розподіляє танцюристів за віковими групами. Для виходу на перше змагання танцюристам присвоюється один з самих нижчих класів (Початківці), який вони згодом можуть змінити на більш високий, зайнявши на змаганнях певні місця і заробивши певні очки. Подібна система дозволяє створити досить рівноцінну конкурентну боротьбу на танцювальному майданчику і полегшити роботу суддів на змаганнях. Правила віднесення танцюристів до тієї чи іншої вікової категорії та класу регламентуються правилами АСЕТУ. Клас майстерності присвоюється за результатами змагань. Танцюристи, які показали хороші результати на змаганнях - наприклад, які пройшли до фіналу і / або посіли призові місця - отримують певну кількість очок. Участь у змаганнях і бали записуються до кваліфікаційної книжки спортсмена. Коли танцюристи набирають певну кількість очок, їм присвоюють наступний клас, про що також робиться запис у кваліфікаційній книжці. У високих класах для переходу з класу в клас необхідно виконати додаткові умови, наприклад, успішно виступити в серйозних рейтингових турнірах.

При реалізації описаних вище вагомих функцій асоціації сучасного естрадного танцю з'являється проблема збереження та відстеження зібраних даних. Для вирішення цього питання можна використовувати паперовий документообіг або спеціально розроблене програмне забезпечення. Важливими критеріями безпомилкової роботи організації є зменшення впливу людського фактору (помилки при виконанні основних операцій), полегшення пошуку та систематизації даних, які зберігаються у паперовому вигляді. Тому для відповідності цим ознакам було вирішено автоматизувати частину роботи АСЕТУ, яка стосується реєстрації

танцювальних колективів для участі у заходах, що проводяться організацією, нарахування балів у відповідності до рейтингу турніру та зайнятого на змаганнях місця, переведення танцюриста до іншої вікової категорії та ліги згідно діючим правилам асоціації.

Насамперед створена повнофункціональна система допоможе відслідковувати керівникам танцювальних колективів та тренерам класифікаційний рівень своїх учнів, зробити прозорою систему рейтингування, надасть можливість дійсним членам асоціації (які мають зареєстровану класифікаційну картку танцюриста) аналізувати свої виступи у всіх проведених турнірах, як на території країни так і за її межами, оптимізувати роботу президії АСЕТУ, а також готувати звітні данні з реалізації програм розвитку сучасного та естрадного танцю, фізичної культури в місцях та регіонах країни до обласних управлінь по справах молоді та спорту.

Система являє собою клієнт-серверний веб-застосунок, клієнтська частина якої реалізована мовою програмування PHP та з використанням системи управління базами даних MySQL.

Основними користувачами розроблюваного програмного продукту є керівники танцювальних колективів та тренера, а також безпосередньо дійсні або асоціативні члени організації, яким система надає наступні можливості:

- додавання, видалення, редагування даних про танцюристів у базі даних;
- здійснення реєстрації танцюристів на відкриті змагання;
- перегляд інформації щодо кількості зареєстрованих танцюристів з розбивкою за віковим категоріям та кваліфікаційним рівням;
- пошук інформації про кваліфікаційний рівень танцюриста за заданими параметрами;
- формування відповідних звітів.

Для виконання запитів створена та розгорнута база даних, яка складається з логічно поєднаних таблиць. Сутності Категорії, Ліги, Керівники та Конкурси допомагають віднайти інформацію про кваліфікаційний рівень дійсного члена асоціації. Наприклад, на заходах АСЕТУ відкриті категорії, до яких закріплені визначені ліги, згідно яких керівники танцювальних колективів можуть здійснювати реєстрацію учасників на відкриті танцювальні змагання. Відображення даних цього запиту неможливе без повної інформації про танцюристів, що є дійсними або асоціативними членами АСЕТУ. Тому в базі існують таблиці: Танцюристи, Книжки, Рейтинг, Оплати, Конкурсанти, Класифікаційні картки, які призначені для збереження загальної інформації про танцюристів: прізвище, ім'я, дата народження, місто, клуб, керівник, код танцювальної картки, ліга, бали тощо.

Як відзначалося раніше, у системі є своя категорія користувачів – це керівники танцювальних колективів АСЕТУ та судді, які отримують доступ до веб-сайту організації. Права адміністратора надаються членам президії асоціації.

Висновки:

З метою автоматизації роботи асоціації сучасного естрадного танцю України була розроблена програмна система, яка:

- надає тренерам можливість швидкого та зручного пошуку даних танцюриста свого танцювального колективу;
- дозволяє керівництву організації відслідковувати правильність переводу танцюристів до кваліфікаційних ліг;
- надає можливість дійсним членам АСЕТУ переглядати та аналізувати історію своєї участі у танцювальних змаганнях;
- реалізує можливість членів президії асоціації контролювати зв'язки танцювальних центрів асоціації та обласних відділень АСЕТУ.

Міхайлуца О.А., магістрант, Полякова Н.П., доцент, науковий керівник

ДОСЛІДЖЕННЯ ТА СТВОРЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ НАВЧАННЯ ЖИВОПИСУ

Запорізька державна інженерна академія, кафедра ПЗАС

З розвитком інформаційних технологій змінюються системи та методики навчання в освіті суспільства. Серед них і викладання живопису, яке охоплює довгий проміжок часу, починаючи з епохи палеоліту до наших днів. Протягом століть кожний період в історії живопису проходив перетворення, які сприяли виникненню різноманітних цільних напрямків та технік, існуючих донині. Новаторство живопису ХХІ століття полягає в оновленні форми, що пов'язано з новими ІТ-технологіями.

У час інформаційної культури, коли до всіх видів людської діяльності впроваджені комп'ютерні технології, мова мистецтва розширює свої можливості засобами комп'ютерної графіки, яка охоплює наступні області застосування: мультимедіа, наукова, ілюстративна, рекламна графіка, анімація тощо [2]. За допомогою комп'ютерних систем традиційні способи роботи з зображенням модифікувалися у більш ефективні та доступні. У художній практиці все частіше виставляються твори, виконані засобами комп'ютерної графіки.

Ринок програмного забезпечення переповнений засобами для обробки візуальної інформації – графічними редакторами, які дозволяють створювати, переглядати, обробляти та редагувати цифрові зображення. Специфіка даної групи програм полягає у великому наборі функцій, що замінюють художні матеріали у реальному житті, завдяки яким можна досягнути певних цілей. Поєднання нових технологій з увічненими знаннями про живопис дає змогу вивести процес навчання на сучасний рівень, основні переваги якого спостерігаються у збільшенні можливостей для безперервної освіти та зменшенні витрат на придбання інструментів і посібників.

Завдяки поєднанню інформаційних технологій з навчальним процесом створюються додаткові пристрої, такі як графічні планшети або дигітайзери. Вони працюють разом з настільними версіями графічних редакторів та складаються з робочої поверхні і безпроводного електронного пера, замінюючи холсти та пензлі відповідно. Графічні планшети досить зручні у використанні, тому більшість дизайнерів не уявляють реалізацію без них реалізацію своїх проєктів. Але сьогодні існує ще один вид пристроїв, яким притаманні всі переваги традиційного настільного персонального комп'ютера – Tablet PC (електронні планшети). Головна відмінність Tablet PC від ноутбуків в тому, що працювати з ними дійсно можна стоячи. За допомогою електронних планшетів можна не витрачати кошти на придбання ПК, і дигітайзера. На додаток велика кількість сучасних моделей Tablet PC підтримує малювання пальцем з декількома точками дотику - мультисенсорне введення, що повністю оновлює звичні засоби ілюстрування. З недавнього часу великі компанії-розробники Adobe та Autodesk модернізують свої графічні редактори під планшетні пристрої з сенсорним керуванням процесу малювання. Правильний вибір цих редакторів впливає на отримання успішного результату.

Одним з рішень в плані зручності для навчання живопису є графічний редактор SketchBook Pro від компанії Autodesk Inc. Розглядаючи основні характеристики SketchBook Pro, відмітимо його широкі можливості для безмежної практики у малюванні:

- управління найдрібнішими деталями завдяки 2500-процентному збільшенню;
- вибір понад з десяти інструментів, що забезпечують природній досвід малювання;

- підтримка шарів, з якими можливі різні операції, необхідні для отримання закінченого малюнка. Шари можна міняти місцями, змінювати прозорість, вимикати видимість, відключати фон шару, підтримувати злиття;
- перевага у підтримці симетричного малювання;
- менші витрати часу на виконання функцій, ніж в інших додатках, за GOMS оцінкою.

Зручність використання, функціонал, наявність десктопної версії - все це робить SketchBook Pro одним з кращих додатків для професіональних художників та початківців.

Ще одним лідером серед графічних редакторів під планшетні пристрої для навчання живопису є додаток Water Color Pencil для роботи з акварельними олівцями та фарбами. У живописі акварель представляє собою техніку, в якій поєднуються багатство тону, простір кольору з активною роллю паперу, відсутністю специфічної рельєфності мазка, яка є характерною для поверхні [4]. Головною перевагою Water Color Pencil є позиціонування у якості єдиного редактора, який орієнтований на навчання малюванню аквареллю, а також редактора з наступними особливостями:

- реалістичність нанесення такими інструментами, як кольоровий олівець та акварельна фарба, ефект яких заснований на рідині;
- реалістичні різноманітні текстури листа для малювання аквареллю.

Разом з тим при роботі з Water Color Pencil можна використовувати інші матеріали та інструменти, що виводять його на один рівень у функціонуванні і виділяє серед інших графічних редакторів водночас.

У роботі графічний редактор розроблений під операційну систему Android на мові програмування Java [1].

Отже, розроблена система надає змогу користувачу безперервно навчатися живопису, допомагає розкрити творчий потенціал завдяки малюванню де-завгодно і будь-коли. Програма буде користуватися успіхом у художників та дизайнерів початкового рівня підготовки завдяки заощадженню коштів на придбання матеріалів для малювання. Професіоналів зацікавить широкий вибір інструментів та додаткових технік живопису, як гризайль, пастель, туш, сухий пензель, масло [3], яких не передбачено в інших графічних редакторах.

Висновки:

- проведено ряд досліджень у галузі застосування сучасних технологій у навчанні живопису;
- вивчено історію розвитку основних напрямків та техніки живопису
- досліджено ринок графічних редакторів, їх переваги та недоліки, проаналізовано їх поширеність у роботі художників та дизайнерів;
- виявлені вимоги цільової аудиторії щодо функціоналу графічних редакторів;
- розроблено програмну систему під операційну систему Android на мові програмування Java.

Література

1. Медникс З. Программирование под Android/ Медникс З., Дорнин Л., Дж. Блейк Мейке – Санкт-Петербург: Питер, 2012. – 250 с.
2. Петров М. Компьютерная графика (3-е изд.). – Санкт-Петербург: Питер, 2011. – с. 320-325.
3. Вибер Ж. Живопись и ее средства. – М.: Издательство Академии художеств СССР, 1961. – 230 с.
4. Ключинг А. Материалы акварельной живописи / Художник. 1975. – с. 60-62.

Небеснюк В.О., студент гр. СП-13-1д, Михайлуца О.М., доц., науковий керівник

СТВОРЕННЯ WEB ІНТЕРФЕЙСУ ДЛЯ УПРАВЛІННЯ ЧЕМПІОНАТАМИ З НАСТІЛЬНОГО ФУТБОЛУ

Запорізька державна інженерна академія, кафедра ПЗАС

На даний момент в сучасному суспільстві існує гостра потреба в самовираженні. Одним з ефективних способів проявити себе є конкурси та змагання, в тому числі й спортивні. Основною проблемою проведення масштабних чемпіонатів є масовість їх проведення і доволі довгий процес відстеження результатів матчів та формування загальної бази з усіма даними, починаючи з інформації про кожного з учасників і закінчуючи результатами фінальних змагань.

Одну з найголовніших ролей у спрощенні обробки великої кількості інформації відіграють інформаційні системи. Існує декілька методів вирішення проблеми автоматизації і збереження великих об'ємів даних. Один із них – реалізація WEB інтерфейсу для управління і проведення чемпіонатів. Вибір цього методу є обґрунтованим, оскільки в наш час майже у всіх користувачів є доступ до Інтернету з зовсім різних комп'ютерних пристроїв та гаджетів, і було б дуже зручно завжди мати змогу подивитися результати тих чи інших турнірних матчів, склад команд чи іншу потрібну інформацію.

На сьогоднішній день існує дуже багато схожих інтерфейсів для генерування та проведення різного роду змагань. Усі ці веб-застосунки тим чи іншим способом спрощують обробку зібраних даних та прискорюють підрахунок проміжних результатів. Але усі вони мають доволі серйозний недолік – візуалізацію, яка підтримує відображення графічних об'єктів для комп'ютерів чи ноутбуків, забуваючи про те, що в сучасному світі є достатньо інших пристроїв з доступом до Інтернету. Тому, наприклад, на смартфонах часто перегляд таких веб-інтерфейсів стає неможливим. Ця проблема має декілька варіантів вирішення, але й потребує набагато більш поглибленого розуміння процесів відображення графічних елементів.

В розробленому WEB інтерфейсі застосовано фреймворк Bootstrap3, який дає змогу більш детально розглядати зображення одного й того ж інтерфейсу на різних пристроях і бути впевненим у можливості зручного перегляду потрібної інформації. Це надає розробленому веб-застосунку перевагу у порівнянні з сучасними аналогами.

Виділяють такі основні модулі в розробці WEB інтерфейсу:

1. Модуль реєстрації користувачів у системі за допомогою соціальних мереж чи заходами самого веб-застосунку. Запис та зберігання всіх необхідних даних щодо користувача та його подальшої активності в системі.
2. Модуль відображення графічних елементів застосунку. Інтуїтивно зрозумілий інтерфейс для більш комфортного користування.
3. Модуль обробки та зберігання великого об'єму інформації. Використання реляційної системи управління базами даних MySQL та веб-застосунок для їх адміністрування phpMyAdmin.
4. Модуль обробки виключень. Коректні відповіді на запити користувачів.

Для реалізації даного WEB інтерфейсу було порівняно між собою достатню кількість середовищ розробки веб – інтерфейсів та переглянуто велику кількість документації по мовам програмування, які використовуються для вирішення такого типу задач.

Засоби реалізації веб-застосунку

Основними засобами, на яких ґрунтується робота WEB інтерфейсу, є наступні використання :

- фреймворку Bootstrap 3 (HTML 5, CSS 3) для написання адаптивного, кросбраузерного та головного валідного коду відображення графічних елементів застосунку;
- скриптової мови програмування JavaScript для поліпшення швидкодії реагування на запити;
- середовища розробки PhpStorm, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA;
- мови програмування PHP 5 для більш зручної роботи з асоціативними масивами даних[1];
- реляційної системи управління базами даних MySQL та веб-застосунка для адміністрування баз phpMyAdmin.

Застосовування алгоритмів для вирішення проміжних задач

При розробці даного програмного продукту було застосовано декілька математичних алгоритмів для сортування даних та формування груп даних (команди, які складаються з кількох учасників). В алгоритмі сортування даних було використано Timsort, який дає помітне прискорення при збільшенні ступеня упорядкованості даних і працює на рівні Quicksort в звичайних випадках [2].

Timsort - гібридний алгоритм сортування, що поєднує сортування вставками і сортування злиттям, опублікований в 2002 році Тімом Петерсом. В даний час Timsort є стандартним алгоритмом сортування в Python, OpenJDK 7 і реалізований в Android JDK 1.5. Враховуючи популярність цих платформ, можна зробити висновок, що рахунок комп'ютерів, хмарних сервісів і мобільних пристроїв, що використовують Timsort для сортування, йде на мільярди.

Основна ідея алгоритму :

- За спеціальним алгоритмом вхідний масив поділяється на під-масиви.
- Кожен під-масив сортується за допомогою сортування вставками.
- Відсортовані під-масиви збираються в єдиний масив за допомогою модифікованого сортування злиттям.

Алгоритм побудований на тому розумінні, що в реальному світі сортований масив даних часто містить в собі впорядковані (не важливо, за зростанням або за спаданням) під-масиви. На таких даних Timsort істотно швидше багатьох алгоритмів сортування. Принципові особливості алгоритму в деталях, а саме в алгоритмі поділу та модифікації сортування злиттям.

Висновки:

- Досліджено різноманітні методи реалізації веб-застосунків.
- Проаналізовано роботу алгоритмів та зроблені оцінки їх ефективності.
- Реалізовано продукт, що дає змогу дуже швидко обробляти велику кількість інформації. Використання його у спортивній сфері (проведення турнірів чи змагань) дозволить значно скоротити час на підрахунок основних результатів.

Література

1. PHP 5/ Д. В. Котеров, А. Ф. Костарев. – 2-е изд., перераб. И доп. – СПб.: БХВ-Петербург, 2014. – 1104 с.: ил. – (В подлиннике)
2. Пышкин Е.В. Основные концепции и механизмы объектно-ориентированного программирования. - СПб.: БХВ-Петербург, 2005. - 640с.

Ничипорчук А.О., магістрант, Полякова Н. П., доцент, к.т.н. науковий керівник

ДОСЛІДЖЕННЯ СТАНДАРТІВ СУМІСНОСТІ СИСТЕМ E-LEARNING З МЕТОЮ СТВОРЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ВНЗ

Запорізька державна інженерна академія, кафедра ПЗАС

Дистанційне навчання - одна з найбільш ефективних і перспективних систем підготовки фахівців. Поява і активне поширення дистанційних форм навчання є адекватним відгуком систем освіти різних країн на процеси інтеграції в світі, рух до інформаційного суспільства [1].

В Україні та багатьох інших країнах дистанційна форма навчання до недавнього часу не застосовувалася в широкому масштабі через недостатній розвиток і відсутність технічних засобів нових інформаційних технологій [1]. На даний час створені технічні передумови для повсемісного використання дистанційного навчання в освіті. В останні роки набув значного поширення термін e-learning, що означає процес навчання в електронній формі через мережу Інтернет з використанням систем управління навчанням.

Програмне забезпечення для e-learning, представлено як простими статичними HTML сторінками, так і складними системами управління навчанням (LMS) та навчальним контентом (LCMS), що використовується в корпоративних комп'ютерних мережах.

Успішне впровадження електронного навчання ґрунтується на правильному виборі програмного забезпечення, яке має відповідати вимогам, цілям і завданням конкретної організації.

Кожна така система дистанційного навчання (СДН) має свій внутрішній формат представлення матеріалу. Відсутність загального підходу ускладнює процеси інтеграції ресурсів, створених в одній системі до іншої. І саме тому проблема забезпечення незалежності контенту від СДН є дуже актуальною.

Одним із варіантів вирішення проблеми є розробка універсального сховища навчальних ресурсів, яке дозволить використовувати дистанційні курси незалежно від програмних платформ. Це сховище дозволить сформулювати єдине інформаційне середовище, яка забезпечить інформаційну взаємодію між користувачами різних СДН [3].

Для реалізації такого сховища потрібно обрати представлення ресурсів в форматі, який не залежить від середовища виконання. Інтероперабельність учбових курсів та їх незалежність від конкретної СДН можливо забезпечити за рахунок використання загальноприйнятих стандартів представлення навчального матеріалу. Одним з таких стандартів є SCORM (Sharable Content Object Reference Model). Він містить вимоги до організації навчального матеріалу та всієї системи дистанційного навчання [3].

Перша офіційна версія стандарту, названа SCORM 1.0, була представлена в січні 2000 року. Це була тестова версія, яка поширювалася у вузьких колах для випробування та збору відгуків.

У січні 2001 року виходить перша "робоча" версія стандарту - SCORM 1.1. У цій версії були чітко прописані правила взаємодії Sharable Content Object і СДН, однак були відсутні правила "упаковки" навчальних матеріалів.

Цей недолік був швидко усунутий, і вже в жовтні 2001 року виходить версія SCORM 1.2, яка на кілька років стала основною. У цій версії вже строго визначені правила "упаковки" навчальних матеріалів - CAM (Content Aggregation Model, запозичена у консорціума IMS). Таким чином, ця версія стандарту стала першою дійсно зручною у використанні.

У SCORM 2004 були зроблені зміни в порівнянні з попередніми SCORM версіями. Ці зміни можна розбити на кілька категорій: уточнення понять, уточнення вимог, зміни в стандартизації та специфікації, додавання поліпшених методів, усунення помилок.

У форматі SCORM учбові курси зручно зберігати, перетворювати та обробляти. Але, незважаючи на це, кожна СДН має свою індивідуальну специфіку роботи з цим стандартом, в деяких системах реалізований тільки імпорт або експорт навчальних матеріалів, має місце часткова втрата функціональності [2].

Однією з задач, що виникають при розробці механізмів конвертації, є завдання аналізу і зіставлення моделей подання вмісту.

СДН зберігають навчальні курси в деяких внутрішніх форматах, прихованих від кінцевих користувачів (наприклад, на рівні структури бази даних, з якою користувач не працює напряму). Більшість СДО дозволяють користувачеві експортувати навчальний курс з системи, наприклад, для вирішення завдань резервного копіювання, переходу між версіями і т. п. [4]. Формат експорту ресурсів залежить від конкретної СДН, тобто не інваріантний.

Таким чином, створення універсального сховища, яке підтримує стандарт SCORM, дозволить вирішити проблему сумісності систем дистанційного навчання.

Висновки:

- досліджено види програмного забезпечення для e-learning;
- вивчено історію розвитку стандарту SCORM;
- досліджено можливості, які надає стандарт SCORM;
- досліджено проблему сумісності дистанційних навчальних систем;
- досліджені види зберігання даних СДН;
- спроектовано універсальне сховище навчальних ресурсів.

Література

1. Соловов А. В. Введение в проблематику дистанционного обучения.
2. Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM) 2004. / Перевод с англ. Е.В. Кузьминой. - М.: ФГУ ГНИИ ИТТ "Информика", 2005.
3. Концепция единого хранилища информац. образов. ресурсов для дист. обуч. – Насадкина Ю.О. , Марахтанов А.Г., Власова А.Г., Байтимиров Л.З., Фотина Е.В. 2009.
4. Анисимов А.М. Работа в системе дистанционного обучения Moodle. Учебное пособие. 2-е изд. испр. и дополн. – Харьков, ХНАГХ, 2009.

Павлов О.Є., ст. гр. СП-14-1с, Скрипник І.А., к.т.н., доцент, науковий керівник

КОМПЮТЕРНА СИСТЕМА ЗАХИСТУ ПРИВАТНОЇ ІНФОРМАЦІЇ ПРИ ПЕРЕДАЧІ ДАНИХ

Запорізька державна інженерна академія, кафедра ПЗАС

Сучасність складно уявити без інформаційних технологій, мобільного зв'язку, Інтернету. Деяка інформація настільки цінна, що їй намагаються заволодіти ті, кому вона не призначена. І звичайно можуть робити це методами, що знаходяться за рамками закону.

Сьогодні найактивнішим середовищем передачі інформації є Інтернет. Люди можуть миттєво отримувати або обмінюватися різною інформацією. Не дивно, що і зловмисники використовують Інтернет для крадіжки і псування чужої інформації (і не тільки).

Мало приємного в тому, що розмова з близькою людиною може бути прочитана іншими людьми. Рішенням подібної проблеми може стати система, яка би кодувала інформацію від користувача при передачі, а розкодувати інформацію могли би тільки ті, кому вона призначена. Для написання додатка використані HTML / CSS, JS на стороні клієнта і JS на стороні сервера.

JavaScript -універсальна скриптова мова з великими можливостями. Використання цієї мови широко варіюється від маніпуляції із зовнішнім виглядом веб-сторінок і до побудови високонавантажених серверних додатків.

Я вибрав означені мови програмування, щоб розроблюваний застосунок був кросс-платформним, а код програми могли розглянути люди, які не довіряють безпечності розроблюваної програми.

Альтернативою могло би стати open-source програмне забезпечення.

Веб-застосунок може мати проблеми зі швидкістю і недостатньою швидкістю Інтернету, але з розвитком Інтернет-технологій (включаючи мови для їх підтримки), ці проблеми стають не такими вагомими.

На стороні клієнта використовується фреймворк jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML.

Серверна частина працює на платформі node.js

Node.js — платформа з відкритим сирцевим кодом для написання серверної частини веб-реалізації на JavaScript, автором якої є Раян Дал (Ryan Dahl).

Node.js характеризується такими властивостями:

- асинхронна однопотокова модель виконання запитів
- неблокуючий ввід/вивід
- система модулів CommonJS
- движок JavaScript Google V8

На базі Node.js використовується Socket.IO - JavaScript бібліотека для веб-додатків і обміну даними в реальному часі. Складається з двох частин: клієнтської, яка запускається в браузері і серверної для node.js. Обидва компоненти мають схожу API. Подібно node.js, Socket.IO подієво-орієнтована.

Розроблювана система складається з таких модулів:

- Клієнт:
 - Модуль візуальної частини. Відповідає за вивід даних, обгортка даних у теги, реагування на дії користувача (нажаті кнопки) і т.д..
 - Зв'язок з сервером. Модуль базується на клієнтській частині socket.io. Передає/отримує зашифровані дані, очікує новини від сервера, передає запити користувача до сервера (наприклад, знайти користувача А)

- Криптомодуль – шифрує/розшифровує дані, які на нього передаються
- Контролер. Усі модулі взаємодіють через контролер. Також контролер балансує навантаження.
- Сервер. Передає повідомлення користувачів, обробляє запити користувачів, звертається до бази даних.
- База даних. Зберігає зашифровані повідомлення(їх неможна прочитати), щоб навіть нові користувачі, які підключаються до діалогу, могли його прочитати, зберігає дані для авторизації користувачів і т.д.

Використані алгоритми шифрування:

1. Симетричні

Шифрування з симетричними ключами — схема шифрування, у якій ключ шифрування, та ключ дешифрування збігаються, або один легко обчислюється з іншого та навпаки, на відміну від асиметричного, де ключ дешифрування важко обчислити.

Використані алгоритми:

- **AES** Advanced Encryption Standard (AES), також відомий під назвою Rijndael — симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт), фіналіст конкурсу AES і прийнятий в якості американського стандарту шифрування урядом США.
- **Blowfish** Blowfish (вимовляється [блоуфіш]) — криптографічний алгоритм, який реалізує блочне симетричне шифрування.
- **XXTEA** - розширення шифроалгоритму XTEA, опубліковане в 1998 році. Виправлена уразливість в алгоритмі, знайдена Markku-Juhani Saarela. Відмінною особливістю алгоритму є також можливість зашифрування блоків будь-якого кратного 32 бітам розміру (мінімальний розмір 64 біта), у той час як інші алгоритми даного сімейства TEA, XTEA, XTEA-3, Riden і RTEA використовують блоки фіксованого розміру, рівного двом або чотирьом 32-х бітовим числам.

2. Асиметричні

Асиметричні криптосистеми — ефективні системи криптографічного захисту даних, які також називають криптосистемами з відкритим ключем. В таких системах для зашифрування даних використовується один ключ, а для розшифрування — інший ключ (звідси і назва — асиметричні). Перший ключ є відкритим і може бути опублікованим для використання усіма користувачами системи, які шифрують дані. Розшифрування даних за допомогою відкритого ключа неможливе. Для розшифрування даних отримувач зашифрованої інформації використовує другий ключ, який є секретним. Зрозуміло, що ключ розшифрування не може бути визначеним з ключа зашифрування.

Використані алгоритми:

RSA став першим алгоритмом такого типу, придатним і для шифрування і для цифрового підпису. Алгоритм використовується у великій кількості криптографічних застосунків.

Отже, розглянута в роботі програмна система, надає можливість користувачам спілкуватись, не хвилюючись, що їх інформації буде прочитана третьою стороною.

Висновки:

- Розглянуті основні бібліотеки для вирішення поставленої задачі.
- Досліджено можливість застосування платформи Node.js та бібліотеки socket.io
- Розроблено програмну систему.

Література

1. [електронний ресурс] <http://api.jquery.com/> - документація фреймворку
2. [електронний ресурс] <http://socket.io/docs/> - документація фреймворку
3. [електронний ресурс] <http://habrahabr.ru/> - IT - ресурс

Паталах В.О., магістрант, Заяц В.І., доцент науковий керівник

ДОСЛІДЖЕННЯ МЕТОДІВ СТВОРЕННЯ ПІДСИСТЕМ АСУП ВНЗ ДЛЯ РОБОТИ ДЕКАНАТУ

Запорізька державна інженерна академія, кафедра ПЗАС

Автоматизована система управління (скорочено АСУ) - комплекс апаратних і програмних засобів, а також персоналу, призначений для управління різними процесами в рамках технологічного процесу, виробництва, підприємства. АСУ застосовуються в різних галузях промисловості, енергетиці, транспорті і т. д.. Термін «автоматизована», на відміну від терміна «автоматична», підкреслює збереження за людиною-оператором деяких функцій, або найбільш загального характеру, або функцій, що не піддаються автоматизації. АСУ з Системою підтримки прийняття рішень (СППР), є основним інструментом підвищення обґрунтованості управлінських рішень.

До складу АСУ входять наступні види забезпечень: інформаційне, програмне, технічне, організаційне, метрологічне, правове та лінгвістичне.

Кожна функція АСУ реалізується сукупністю комплексів задач, окремих завдань і операцій. Функції АСУ в загальному випадку включають в себе наступні елементи (дії):

- планування та (або) прогнозування;
- облік, контроль, аналіз;
- координацію і (або) регулювання.

Необхідний склад елементів вибирають залежно від виду конкретної АСУ. Функції АСУ можна об'єднувати в підсистеми за функціональною та іншими ознаками.

До категорії АСУП прийнято відносити реалізації методологій MRP (англ. Material Requirements Planning - планування потреби в матеріалах) і ERP (англ. Enterprise Resource Planning, планування ресурсів підприємства).

У галузі освіти під АСУВ, як правило, розуміють систему управління навчанням. Прикладом може служити система Moodle. Однією з перших вітчизняних систем управління навчальним процесом, є комплекс інформаційних систем «АСУ ВУЗ». В даний час активно розвивається АСУ ВУЗ «Universys WS», оперативно враховує мінливі реалії сучасної освіти.

Автоматизована система управління для вищого навчального закладу - ефективний інструмент для комплексної або часткової автоматизації процесів керування ВНЗ.

АС Деканат - основна частина системи АСУ ВНЗ, яка виконує головні функції з автоматизації адміністративних та навчально-методичних процесів.

У загальному випадку, систему управління можна розглядати у вигляді сукупності взаємопов'язаних управлінських процесів і об'єктів. Узагальненою метою автоматизації управління є підвищення ефективності використання потенційних можливостей об'єкта управління. Таким чином, можна виділити ряд цілей:

- надання особі, що приймає рішення (ОПР), релевантних даних для прийняття рішень
- прискорення виконання окремих операцій зі збору та обробки даних
- зниження кількості рішень, які має брати ЛПР
- підвищення рівня контролю та виконавської дисципліни
- підвищення оперативності управління
- зниження витрат ЛПР на виконання допоміжних процесів
- підвищення ступеня обґрунтованості прийнятих рішень

Навчально-методичні відділи в АС Деканат автоматизовано ведуть облік та здійснюють управління студентським контингентом і професорсько-викладацьким складом, в системі:

- формуються навчальні плани (в т.ч. індивідуальні);
- розподіляються навантаження по кафедрах;
- готуються екзаменаційні відомості;
- формуються зведені дані щодо проведення і результатів сесії;
- створюються відповідні звіти;
- дані про результати іспитів автоматично вносяться в особисті справи;
- формуються дані для друку відомостей, звітів та додатків до диплому;
- розклад занять;
- електронний журнал успішності.

Керівництво ВНЗ отримує оперативну інформацію по студентському і професорсько-викладацькому складу, звіти про надходження коштів за навчання.

У відділі кадрів ведуться особисті картки співробітників і студентів, заповнюються журнали наказів по анкетах студентів з автоматичним внесенням їх до особової справи.

Бухгалтерія отримує оперативну інформацію про академічну успішність та нарахування стипендій, керує інформацією про надходження платежів за навчання.

Висновки:

- проведено ряд досліджень у галузі застосування АСУП;
- розглянуто основну класифікацію та функції АСУП;
- досліджено область застосування АСУП (створення підсистеми управління деканатом ВНЗ)

Література

1. Меньков А.В., Острейковский В.А.. Теоретические основ автоматизированного управления. Оникс, 2005.
2. Сараев А.Д., Щербина О.А. Системный анализ и современные информационные технологии //Труды Крымской Академии наук.- Симферополь: СОНАТ, 2006. - С. 47-59

Причиненко М.А., магістрант, Шамровський О.Д., професор, науковий керівник

ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ ТА СТВОРЕННЯ СИСТЕМИ ДОКУМЕНТООБІГУ У ВНЗ

Запорізька державна інженерна академія, кафедра ПЗАС

У загальному обсязі управлінської роботи велику частку займають операції з документами. Організація документообігу залежить від масштабу діяльності підприємства, його функцій, кількості ланок управління і обсягу документопотоків. Спеціальні служби допомагають адміністрації вирішувати управлінські завдання, забезпечуючи підготовку та рух документів, доставку їх до виконавця та зацікавлених осіб. До таких служб в науковій установі належать канцелярія, відділ кадрів, бухгалтерія, юридичний відділ, архів та інші [1].

Ефективність управління значною мірою залежить від часу, що витрачається на створення та одержання документів, контроль за їх виконанням, організацію збереження, пошук та використання. Оптимізація цих процесів може значно зекономити час та ресурси організації. Одним з варіантів такої оптимізації є введення системи електронного документообігу (СЕД).

Законодавство України визначає електронний документообіг як сукупність процесів створення, оброблення, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів. Його порядок визначається державними органами, органами місцевого самоврядування, підприємствами, установами та організаціями всіх форм власності згідно з законодавством [2].

Підвищення ефективності управлінської діяльності при впровадженні СЕД відбувається за рахунок прискорення руху документів та зменшення трудомісткості їх опрацювання.

Програмні засоби автоматизації документаційного забезпечення охоплюють три основні завдання: документування (створення документів, які підтримують і реєструють управлінську діяльність, тобто їх підготовку, оформлення, узгодження та виготовлення); організація документообігу (забезпечення руху, пошуку, зберігання і використання документів); систематизація архівного зберігання документів.

Крім того, електронний документообіг має цілий ряд переваг: можливість вміщення в документ, крім тексту, мультимедійних даних; можливість використання заздалегідь заготовлених форм; висока швидкість передачі інформації за великою кількістю адрес; економія паперу; висока компактність архівів; висока швидкість пошуку і одержання інформації; можливість захисту документів від несанкціонованого доступу та розмежування прав доступу співробітників до інформації [3].

До СЕД висувається наступний ряд вимог:

- Масштабованість – система повинна підтримувати будь-яку кількість користувачів, а її здатність нарощувати потужність визначається тільки потужністю відповідного програмного забезпечення.
- Розподіленість – архітектура систем документообігу повинна підтримувати взаємодію розподілених майданчиків для роботи з документами в територіально-розподілених організаціях.
- Модульність – система повинна складатись з окремих модулів, інтегрованих між собою, на випадок, якщо користувачу системи не потрібно одразу впровадження всіх компонентів системи документообігу, або спектр завдань організації вузький, ніж весь спектр завдань документообігу.

- Відкритість – система повинна мати відкриті інтерфейси для можливого доопрацювання і інтеграції з іншими системами.

На поточний момент існує багато систем для автоматизації документообігу в організації. Тому актуальним є питання критеріїв вибору тієї чи іншої системи. Серед найважливіших характеристик СЕД, як правило, виділяють наступні:

- програмна платформа;
- типи документів, з якими працює система;
- можливості масштабування;
- максимальна кількість користувачів; загальна кількість та число рівнів структур, які відображають особливості внутрішньої організації підприємства;
- можливість роботи за «вільною» схемою, без жорсткої фіксації маршрутів;
- засоби для визначення маршрутних схем проходження документів;
- можливості контролю за проходженням документів; засоби повідомлення про порушення у регламенті проходження документів, спосіб повідомлення посадових осіб;
- підтримка роботи з кількома версіями документа, можливість інтеграції з іншими додатками; особливості налаштування програмного продукту на потреби конкретного замовника;
- засоби регламентації доступу та криптографічного захисту.

Електронний документообіг дає змогу створити в організації єдиний інформаційний простір, інтегруючи у інформаційний вузол всі документальні системи. Типова СЕД забезпечує повну підтримку процесів створення, управління, обробки і поширення електронних документів між відповідальними посадовими особами та контроль за потоками документів. Головне призначення таких систем - автоматизувати весь комплекс документаційних потоків: введення документів у систему, їх реєстрація, розподіл і розсилання, редагування, оперативне збереження, пошук і перегляд, відтворення, контроль виконання, розмежування доступу до документів, прискорення термінів опрацювання документів, вдосконалення механізмів організації та виконання документів, тощо.

Висновок

Таким чином, правильно створена система електронного документообігу значно зменшує трудомісткість опрацювання документів та прискорює їхній рух у організації, що, в свою чергу, підвищує ефективність управлінської діяльності, економить час та ресурси організації.

Література

1. Величkevич М.Б., Митрофан Н.В., Кунанець Н.Е. Електронний документообіг, тенденції та перспективи // УДК 651.2:004.91
2. Закон України Про електронні документи та електронний документообіг.
3. Корнута В.А. Основи організації електронного документообігу, ІФОЦППК, СЕРІЯ: Довідково-інформаційні матеріали, Випуск 58/12, Івано-Франківськ, 2012, 45 с.

Романенко Ю.О., магістрант, Попівський В.І., доцент, науковий керівник

ВПЛИВ МОВИ ПРОГРАМУВАННЯ SWIFT НА ІТ-ІНДУСТРІЮ

Запорізька державна інженерна академія, кафедра ПЗАС

Нова мова програмування Swift[1], розроблена Apple для iOS і OS X, здатна істотно змінити способи розробки добре знайомих користувачам цих систем додатків, а також помітно понизити планку рівня підготовки, достатнього для створення таких програм.

Мова Swift була офіційно представлена на черговій Міжнародній конференції розробників Apple WWDC 2014 і була зустрінутий надзвичайно тепло. Професіонали з самого початку оцінили великий потенціал новачка, але що ж може дати Swift іншим категоріям користувачів? Давайте розберемося.

Чому фахівці на WWDC 2014 були відразу так натхнені появою Swift? Вся справа в першій же фразі, в якій старший віце-президент Apple з розробки ПЗ, Крейг Федеріго, згадав нову мову: «Що, якщо у нас був би Objective-C без рудиментів C? І ми не просто подумали про це, ми зробили це. У нас є нова мова програмування. Вона називається Swift і вона дуже крута».

До останнього часу для розробки додатків для платформи Apple можна було використовувати мови C, C++ або Objective-C [2]. Останній був створений відомим програмістом Бредом Коксом на початку 1980-х, і з 1983 року він використовується в Apple для створення різноманітних додатків. Objective-C багато запозичив з C і C++, принциповою відмінністю було введення об'єктно-орієнтованої моделі програмування, реалізованої в стилі Smalltalk, в якій об'єктам надсилаються повідомлення. Ця модель була зручніше, ніж реалізована в могопарадигменном C++ і, тим більше, ніж у класичному процедурному C.

Проте, базування на мові C родом з початку сімдесятих робило програмування на Objective-C набагато більш трудомістким завданням, ніж на таких сучасних мовах, як Python або Ruby. Однак, при всій його громіздкості для програміста, серед ключових достоїнств Objective-C можна було назвати надзвичайно швидке виконання цілого ряду функцій.

Очевидно, що в Apple поставили завдання зберегти в Swift все хороше, що було в Objective-C, зокрема, згадану продуктивність коду, з'єднавши її з звичними сьогодні зручностями інтерпретованих мов. В Swift позбулися безлічі рудиментарного сміття, що дістався Objective-C від попередників. Крім того, Swift дозволяє програмісту вирішувати поставлені завдання в декількох різних моделях, серед яких функціональне програмування, при якому всі дії представлені у вигляді математичних обчислень; процедурне, де дії виконуються послідовно і лінійно, і об'єктно-орієнтоване, де всі дані представлені в якості об'єктів[3].

Swift дає на виході високо оптимізовані LLVM-сумісний байткод (тобто низько рівневі інструкції, що виконуються безпосередньо пристроєм) і працює з тими ж API, що і Objective-C або C. Завдяки такій сумісності можна інтегрувати код Swift у вже існуючі проекти, бібліотеки та програми. А якщо вам буде потрібно опуститися на рівень нижче і вручну задати, наприклад, тип змінних або параметри управління пам'яттю, ви завжди зможете повернутися до Objective-C.

Xcode[4] — інтегроване середовище розробки (IDE) виробництва Apple. Дозволяє створювати програмне забезпечення з використанням таких технологій як GCC, GDB, Java та ін. На сьогодні є єдиним засобом написання «універсальних»(Universal Binary) прикладних програм для Mac OS X.

Xcode включає в себе більшу частину документації розробника від Apple та Interface Builder - застосунок, який використовується для створення графічних інтерфейсів.

Пакет Xcode містить змінену версію вільного набору компіляторів GNU Compiler Collection і підтримує мови C, C++, Objective-C, Swift, Java, AppleScript, Python і Ruby з різними моделями програмування, включаючи (але не обмежуючись) Cocoa, Carbon і Java. Створені розробниками реалізована підтримка GNU Pascal, Free Pascal, Ada, C#, Perl, Haskell і D. Пакет Xcode використовує GDB в якості back-end'a для свого відналаджувача.

Swift створений для розробки застосунків для продукції Apple, в тому числі і для мобільних застосунків на платформі iOS. Нас цікавить така категорія застосунків, як ігри. Досить велика частина застосунків у AppStore це ігри і вони завжди будуть користуватися попитом, тому цей напрямок вважається дуже перспективним. Деякі ігри дуже прості, але більшість цікавих ігор мають складні алгоритми штучного інтелекту[5] для того, щоб користувачу було цікаво в них грати. Swift, як і всі високо рівневі мови програмування дозволяє використовувати будь-які алгоритми, які потрібні програмісту.

З професійними розробниками розібралися. А що може дати Swift студентам, тільки роблять перші кроки в програмуванні? Цілком може очікувати того, що класична фраза «Hello World» зовсім скоро буде виводитися на екран переважно за допомогою Swift. Чому? Тому що Swift простий в освоєнні, наочний і інтуїтивно зрозумілий, може використовувати різні моделі програмування і при цьому забезпечує високі швидкості Objective-C.

Нарешті, що поява Swift дасть звичайним користувачам, більшість з яких не є програмістами і мало що знає про якісь там мови? Для споживача важливо, щоб додаток чітко виконувало заявлені функції і було зручним і стабільним.

Очевидно, що впровадження Swift помітно позначиться на всій програмній екосистемі Apple. Оскільки розробка додатків для iOS і OS X стає все простіше і легше, багатьом професіоналам в інших мовах і платформах захочеться спробувати свої сили на новому полі діяльності. Але що це може означати?

Насамперед, це означає, що до мобільних і комп'ютерних платформ Apple буде залучено більше число розробників. Більше розробників - це більше додатків і більше вибору для споживача. А от питання про якісний рівень такого софту доведеться залишити відкритим. Поки що Swift занадто нова технологія і вона ще не освоєна навіть професійними програмістами для пристроїв Apple. До того ж вона більш доступна в розумінні для студентів та інших людей, які роблять перші кроки в програмуванні. До чого все це може призвести, поки сказати важко.

Висновки:

- вивчено історію розвитку мов програмування для пристроїв Apple
- досліджено засоби розробки застосунків, їх переваги та недоліки
- досліджено можливості, які надає середовище розробки Xcode для розробки програмних застосунків
- встановлено, що Swift добре підходить для розробки мобільних застосунків, в тому числі і ігор

Література

1. Apple Inc. The Swift Programming Language – iBooks, 2014. – 500 с.
2. Мэтт Нойбург Программирование для iOS 7. Основы Objective-C, Xcode и Cocoa – Вильямс, 2014. – 384 с.
3. Apple Inc. Using Swift with Cocoa and Objective-C – iBooks, 2014. – 100 с.
4. Електронний ресурс, <https://developer.apple.com/>
5. Алекс Дж. Искусственный интеллект в компьютерных играх – Вильямс, 2007. – 768 с.

Сучков Р.О., спеціаліст, Попівший В.І., доц., к.ф.-м.н. – науковий керівник

РОЗРОБКА СИСТЕМИ КЕРУВАННЯ САЙТОМ НА ОСНОВІ ФРЕЙМВОРКУ SYMFONY 2

Запорізька державна інженерна академія, кафедра ПЗАС

З появою веб-технологій комп'ютер починають використовувати абсолютно нові верстви населення. Інформацією, що зберігається на сайтах керує система керування вмістом (СКВ). Це найбільш сучасна технологія створення веб-сайтів. Symfony2 – є одним з найбільш розвинутих фреймворків для створення веб-застосунків на поточний час [1]. Через це, тема дипломної роботи є актуальною. Метою дипломної роботи є створення СКВ за допомогою фреймворка Symfony2, яка задовольняє усім сучасним вимогам.

Загальні вимоги до СКВ:

1. Можливість легко додавати та редагувати інформацію на сайті.
2. Можливість змінювати структуру сайту.
3. Користувачі можуть реєструватися, та писати коментарі на сторінках.
4. СКВ забезпечує надійний захист даних в БД.

Переваги Symfony2

Розширення, потужна система ORM, гнучка система тестування, наявність спільноти, потужна система захисту, яка підтримує ролі, ACL на рівні класів, інтеграція з ORM менеджерами, легка інтеграція з NoSQL, сервісна архітектура, інтеграція з Composer, командний рядок

Недоліки Symfony2

Солянка з технологій.

Dependency injection, service location, неймспейси, події, бандли, файрволи, компіляція кешу, автолоадінг класів, компіляція ассетів. Таким чином Symfony2 дозволяє використовувати дуже багато технологій, проте вони погано стандартизовані.

Необхідність вивчення Doctrine та Twig

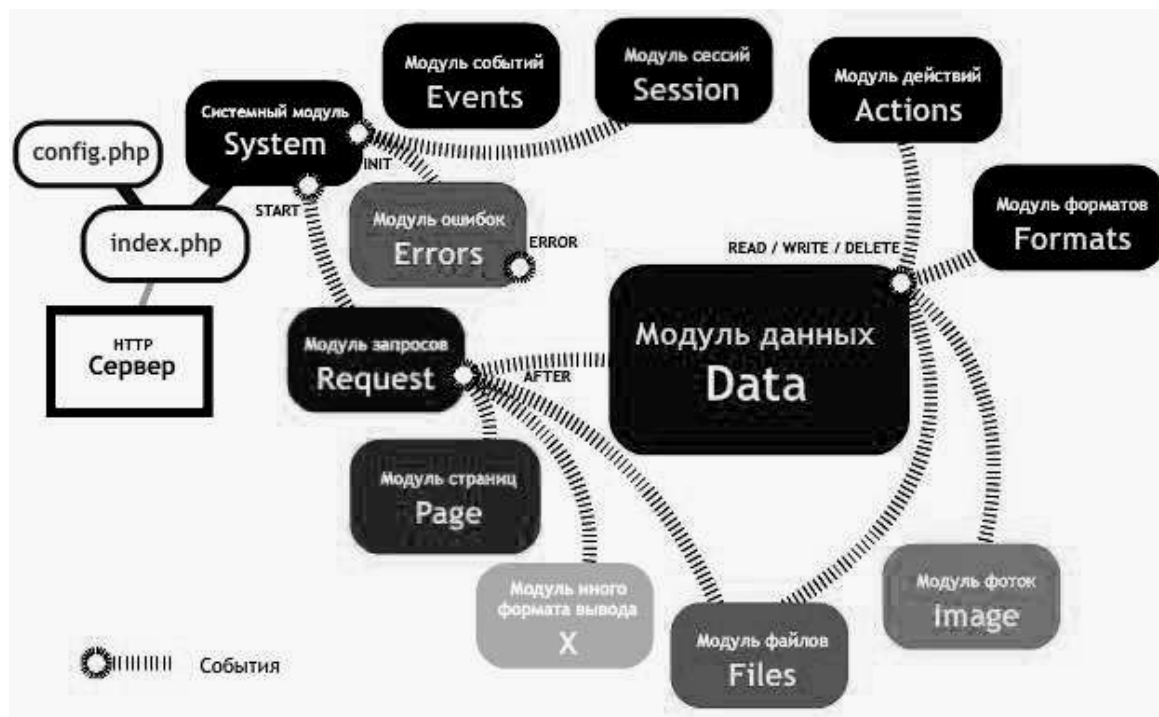
Як ми вже з'ясували Doctrine-це потужна ORM, а Twig – це шаблонізатор.

Ви звісно можете вибирати ORM та шаблонізатор власноруч, але це стандарт де-факто і більшість прикладів саме на них. В будь-якому разі вам доведеться вивчати як мінімум 2 фреймворка, при чому досить складних.

Форми.

Якщо ви хочете змінити будь-що, починаючи з шаблону виведення або валідації даних, все викликає складність і більшість часу припадає саме на роботу з формами.

Архітектура СКВ



Модуль даних- модуль, який працює з БД за допомогою ORM Doctrine.

Модуль сторінок – відповідає за систему роутінгу.

Модуль файлів – відбувається завантаження та оновлення файлів.

Модуль системи - зберігає всю інформацію про поточний стан ситеми.

Модуль помилок - в модулі знаходяться обробники помилок.

Модуль захисту – забезпечує захист від атак на веб-сайт (SQL-injection, XSS і т.і).

Висновки:

При розробці системи були проаналізовані сучасні веб-технології, що дозволяють створювати інтерактивні веб-сторінки. Це такі технології як: Symfony2, Doctrine, Twig.

Розроблена система керування вмістом, яка задовольняє всім вимогам, поставленим на етапі постановки завдання. При розробці системи були використані готові модулі аутентифікації, а також встановлені додаткові плагіни.

Література

1. Леонтьев В. П. Создай свой интернет сайт - ЗАО «ОЛМА Медиа Групп» М. 2013, 250 с.
2. Нидерст М.В. Web-мастеринг для профессионалов. – М. : Вільямс, 2014. – 768 с.
3. Локхарт Д.Сучасний php.-Орейлі,2015.-371 с.
4. Офіційна документація Symfony2 <http://symfony.com/>

Христич А.Ю., магістрант, Лимаренко Ю.О., доцент, науковий керівник

ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ІНТЕРНЕТ- МАГАЗИНІВ

Запорізька державна інженерна академія, кафедра ПЗАС

Сьогодні сфера послуг та продажу товарів онлайн займає провідне місце на ринку Веб-рішень по всьому світу. Існує безліч технологій та платформ для вирішення проблеми розробки успішного, а головне підходящого інтернет-магазину для певного замовника [1]. Для виявлення важливих особливостей був проведений порівняльний огляд найпопулярніших з технологій розробки відповідних Веб-рішень.

Перша система управління інтернет-магазинами має назву Magento., у 2011 році стала найпопулярнішою у світі й заявила про себе як достатньо потужна платформа.

Основні особливості:

- З єдиної адміністративної частини можна керувати торговою мережею інтернет-магазинів на різних доменах, мовах та з різним товарним асортиментом
- Має гнучкі налаштування зовнішнього вигляду («теми» та «шаблони») вітрини
- Відображення цін у різній валюті (у тому числі за вибором покупця)
- Багатомовність
- Різні методи оплати та доставки

За зовнішній вид в Magento відповідає тема. Тема - це набір шаблонів відображення блоків сайту на екрані; набір правил, який визначає які блоки у якому конкретному місці сторінки відображати (блоки і правила з'єднуються у файли логічної розмітки); набір ресурсних файлів: CSS, зображень та скриптів на JavaScript. Magento враховує різноманітні системні типи товарів (простий товар (SimpleProduct), збірний товар (BundleProduct), цифровий товар (VirtualProduct) і т.д.).[4]

Головним недоліком є те, що CMS побудована на ZendFramework, що зразу визначає її об'єм: вона велика та важка. Також для створення ресурсу за допомогою даної системи потребується значний бюджет, адже система є платною.

OpenCart - це система управління інтернет - магазину, навколо якої сформувалася велика спільнота (більше 46000 учасників), завдяки чому існує більше 8500 безкоштовних і комерційних доповнень, які дозволяють змінювати і доповнювати функції магазину різноманітним чином. У порівнянні з конкурентами володіє більш високою швидкістю роботи і меншими вимогами до ресурсів сервера [3].

Особливості:

- Сумісність з усіма основними браузерами.
- Вбудована багатомовність, за замовчуванням встановлений англійська.
- Необмежене число розділів і товарів.
- Статистика товарів і замовників
- Підтримка декількох валют
- Всі замовлення зберігаються в базі даних для швидкого і ефективного пошуку (історія покупок для покупців)
- Система оплати та доставки
- Підтримка численних типів платежів (чеки, платіжні доручення)
- Безліч модулів розрахунку доставки і розрахунок податків.

Недоліки системи здебільшого є платою за швидкість роботи скриптів і простоту їх доопрацювання:

- Не повністю налагоджена робота з SEO (можливість дивитися одні й ті ж сторінки за різними URL і відсутність підтримки 404-ї сторінки)
- У товару не може бути полів для окремих властивостей і, як наслідок, не реалізувати в принципі фільтри за властивостями. (Фільтр по атрибутам доступний у вигляді платного доповнення)

OpenCart має відкритий код й як вже було зазначено раніше, великі можливості до зміни та налаштування магазину безкоштовно, що робить її однією з найпривабливіших платформ.

Система osCommerce дає багатий набір інструментів «з коробки», функціональні можливості кошика, а також більш ніж 5900 доповнень, які доступні безкоштовно.

Особливості:

- Підтримує необмежену кількість продуктів і розділів категорій
- Багатомовна підтримка
- Підтримка декількох валют
- Всі замовлення зберігаються в базі даних (історія покупок для покупців)
- Швидкий і дружній інтерфейс пошуку
- Підтримка численних типів платежів (чеки, платіжні доручення)
- Безліч модулів розрахунку доставки.

На жаль, osCommerce не володіє системою шаблонів для налаштування дизайну, але є ряд модулів, які вирішують дану задачу.

osCommerce розповсюджується безкоштовно за ліцензією GNU General Public License, що суттєво зменшує вартість розробки у ряді випадків [2].

Висновки:

- існує декілька доволі гарних рішень для створення інтернет-магазинів при обмеженому бюджеті, серед них osCommerce та OpenCart
- якщо при розробці головною метою є швидкість та легкість роботи за великою кількістю товарів, то варто обрати OpenCart
- щоб створити захищений магазин, який матиме унікальний функціонал, а також невелику кількість товарів, правильним вибором стане Magento
- проведено ряд досліджень у галузі систем управління контентом
- виявлено особливості та недоліки, які допомагають обрати рішення, спираючись на потреби замовника

Література

1. Малишева Н., Віртуальні покупки. Електронна версія тижневика «Аргументи і факти» - <http://www.aif.ru>
2. Акила К., Еремеевский А. Интернет-магазин с нуля. Полное пошаговое руководство; Питер - Москва, 2013. - 176 с.
3. Рассел Джесси Интернет-магазин; Книга по Требованию - Москва, 2012. - 100 с.
4. Соловьев Д., Писарев А. Интернет-магазин без правил; Питер - Москва, 2013. - 670 с.

РОЗРОБКА МОБІЛЬНОЇ НАВЧАЛЬНОЇ ГРИ З МЕРЕЖНИХ ТЕХНОЛОГІЙ

Запорізька державна інженерна академія, кафедра ПЗАС

Якщо раніше з мобільного телефону можна було тільки зателефонувати або відправити СМС, то тепер він став ще й незамінним помічником для оперативної роботи з різними документами, перегляду фото і відео, прослуховування музики, ну і, звичайно ж, виконання мобільних застосунків. Розробка програм під Android у наші дні впевнено «набирає обертів», адже мобільні пристрої, які працюють під керуванням цієї операційної системи користуються великою популярністю.

Сьогодні ігри використовуються не тільки для розваги, один з нових напрямків e-learning - це «гейміфікація навчання», тобто процес отримання знань шляхом гри. Корпорація CISCO з 1997 року фінансує розробку електронних курсів для вивчення мережних технологій. Ці курси завжди якісно спроектовані і використовують найновіші здобутки у галузі педагогіки та електронного навчання. Тож не дивно, що під час навчання в мережній академії Cisco студентам пропонується низка ігор, які мають зробити процес навчання захоплюючим та більш ефективним: Peter packer, Binary game, Subnet game, Wireless explorer, Subnet troubleshooting game, Secure volunteer game, Edge quest, Edge quest 2, Cisco mind share. Ці ігри можна безкоштовно використовувати на:

1. https://learningnetwork.cisco.com/community/learning_center/games
2. <http://www.cisco.com/web/learning/netacad/packetville/pr/games>.

Але, не зважаючи на велику кількість ігор, що охоплює собою майже всі технології, які вивчаються студентами, лише **Binary game** може використовуватися на мобільних пристроях. На жаль навіть ця гра не є досконалою, і не на всіх мобільних пристроях працює успішно. Гра Peter Packet була розроблена компанією CISCO у 2006 році, як flash гра. Основна ціль гри – управляти пакетом так, щоб він дійшов до отримувача, по дорозі оминаючи віруси та інші загрози, вона не потребує від користувача особливих знань з мережних технологій і слугує цілям мотивації школярів до навчання за Програмою мережних академій Cisco. В наш час гра втратила свою популярність через захоплення школярів мобільними застосунками, тож виникла необхідність розробки гри Peter Packet під мобільну платформу.

Оскільки застосунок ігровий, розробка починається з вибору 2D движка. Дослідження цього питання показали, що є багато варіантів 2D движків для написання ігрових застосунків під Android, також за різними джерелами немає однієї думки щодо кращих з них. Після міркувань оснований на опиті експертів з розробки застосунків під Android було обрано движок **AndEngine**. AndEngine один з найпопулярніших та найпотужніших відкритих для використання на платформі Android 2D OpenGL ігрових движків, що був розроблений Ніколасом Гремлічем. На даний момент існує дві версії ігрового движку AndEngine : GLES1 і GLES2. Можна знайти багато прикладів використання GLES1, але оскільки GLES1 все одно використовується і в наступній версії, то можна сміливо починати з GLES2.

GLES2 має наступні вимоги [1] :

1. GLES2 працює з версіями Android 2.2 та нижче, але в нашому випадку, через піксельну колізію та деякі складі анімації потрібно використовувати версію Android 3.0 та вище.
2. Потрібно мати встановлену версію Android 4.0 або вищу в IDE, яке використовується для розробки.
3. Тестувати застосунок на даному движку можливо лише на фізичних пристроях, на емуляторі він не буде працювати.
4. Також треба оновити ваш ADT до останньої версії(рекомендовано).

Але, движок це не найбільша проблема. Для того щоб якомога краще відтворити гру під платформу Android, спочатку потрібно було отримати ігрові об'єкти: модель гравця, інших живих ігрових об'єктів, навколишнього середовища і т.п. Для досягнення цієї мети було обрано застосунок **JPEXS Free Flash Decompiler**. Він представляє собою відкритий для користування SWF декомпілятор та модифікатор. Дістає ресурси із застосунків, конвертує SWF у FLA формат, модифікує ActionScript, замінює малюнки, звуки, текст або шрифти у застосунку на завантажені користувачем. Має багатий набір вихідних типів даних. Працює з платформою Java на Windows, Linux або MacOS системах.

Після отримання ресурсів з flahs-гри, нам потрібно перевести їх у векторний формат для більш гнучкого використання, в залежності від пристроїв, на яких застосунок буде запускатися. Для цього використовується Inkscape. **Inkscape** - це висококласний професіональний інструмент для роботи з векторною графікою у Windows, Mac OS X і Linux операційних системах. Він широко використовується любителями і професіоналами по всьому світу для створення ілюстрацій, іконок, логотипів, діаграм, карт, а також веб-графіки. Inkscape використовує відкритий стандарт SVG (Scalable Vector Graphics) від W3C в якості формату по замовчуванню, а також сам є вільним і відкритим програмним забезпеченням. Inkscape містить просунуті інструменти для роботи, які витримують порівняння з можливостями Adobe Illustrator, CorelDRAW та Xara Xtreme. Дозволяє відкривати та зберігати зображення у багатьох форматах, включаючи SVG, AI, EPS, PDF, PS і PNG. Inkscape має всеосяжний набір інструментів та простий інтерфейс, підтримку багатьох мов, а також можливість розширення з використанням користувацьких доповнень.

Потім на основі отриманих ресурсів, вже у потрібному розмірі нам необхідно створити на основі них спрайти. **Спрайт** — це графічний об'єкт в комп'ютерній графіці. Частіше за все — растрове зображення, яке вільно переміщується по екрану. Спостереження за спрайтом під невідповідним кутом призводить до розкриття ілюзії. Тобто легше за все сприймати спрайт, як проекцію, що переміщуються в просторі так, що різниця не помітна. Тобто, частіше за все в комп'ютерних та мобільних застосунках спрайти використовуються як анімація. Для того щоб створити спрайти було використано **Instant Sprite**. Instant Sprite – це відкритий для використання, та відкритий для перегляду коду онлайн ресурс призначений створення спрайтів.

Для того, щоб мати змогу зберігати поточний стан гри, і потім відтворювати його для користувача, потрібно використовувати БД. Android використовує для роботи з базами даних відому бібліотеку SQLite. SQLite зарекомендувала себе у якості надійної системи баз даних, яка використовується в багатьох побутових електронних пристроях і програмах, включаючи деякі MP3-програвачі, iPhone, iPod Touch, Mozilla Firefox і ін. За допомогою SQLite ви можете створювати для свого застосунку незалежні реляційні бази даних. Android зберігає бази даних в каталозі/data/data/<ім'я вашого пакету>/databases на вашому пристрої (або емуляторі). По замовчуванню усі бази даних закриті, доступ до них можуть отримати тільки ті застосунки, які їх створили. До того ж потрібно пам'ятати, що в Android вже є готовий клас SQLiteOpenHelper, від якого потрібно успадковуватися [2]. Використовуючи ці знання інтегрувати SQLite в свій застосунок дуже просто.

Висновки:

1. Дослідження функціональні можливості прототипу гри, який було розроблено під стаціонарні пристрої.
2. Спроектовано ігровий застосунок для Android.
3. Вивчено та використано наступні інструменти як : AndEngine, SQLite, JPEXS Free Flash Decompiler, Inkscape, Instant Sprite.
4. Реалізований мобільний ігровий застосунок для популяризації навчання мережним технологіям Peter Packet.

Література

1. Джеймі Скуердер «AndEngine for Android Development cookbook». : , 2013, 380 с.
2. Майк Оуенс «The Definitive Guide to SQLite» :- 2010, 238с.

