

УДК 004.9

О.К. Пандорін

Харківський національний економічний університет ім. Семена Кузнеця, Харків

СКЛАДОВІ ІНСТРУМЕНТАЛЬНИХ КОМПЛЕКСІВ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ЕЛЕКТРОННОГО НАВЧАННЯ

В статті подано аналіз специфічних рис використання складових опрацювання мультимедійних компонент створення систем підтримки електронного навчання. Розглянуто основні проблеми, які вирішуються в процесі вибору архітектури таких систем з точки зору гейміфікаційної парадигми. Наведено набір ключових рис, які визначають архітектуру та склад системи з точки зору розробника.

Ключові слова: компонентний підхід, мультимедійні бази даних, електронне навчання, гейміфікація.

Вступ

Постановка проблеми. Ціла низка подій у сучасному світі може значно змінити технології розроблення та підходи до проектування архітектурних вирішень систем підтримки мультимедійних дидактичних комплексів. Складність проблеми модернізації сучасної освіти відповідно до вимог впровадження систем електронного навчання потребує для їх вирішення принципово підходів що відрізняються від традиційних. Насамперед, це стосується гейміфікації – концепції, заснованої на використанні ігрових елементів та ігрових методів у неігровому контексті [3; 4].

Сучасні студенти віртуозно володіють інформаційно-комунікативними технологіями й активно користуються електронними носіями. Це дозволяє винести частину навчання online, за рамки академічних годин, що дає можливість розширити аудиторне навантаження, при цьому використовуючи ті елементи освітнього процесу, які студентам цікаві та доступні.

Завдання гейміфікації – зацікавлення користувача, стимулювання виконання дій або процедур через ігрові механіки, формування позитивного досвіду взаємодії.

Основний принцип гейміфікації забезпечення постійного, вимірюваного зворотного зв'язку від користувача, що забезпечує можливість динамічного коригування поведінки користувача і, як наслідок, швидке освоєння всіх функціональних можливостей. Ще одним методом є створення легенди, історії, з використанням драматичних прийомів. Подібна легенда має супроводжувати процес використання програми. Це сприяє створенню у користувачів відчуття причетності, внеску загальну справу, інтересу до досягнення будь-яких вигаданих цілей.

Аналіз останніх досліджень і публікацій. Результати аналізу літературних джерел [1–4] свідчать

про наявність описів основних технологій електронного навчання, методик управління локальною та розподіленою обробкою інформації. Проте в спеціалізованій літературі є відсутнім цілісний аналіз специфічних рис використання міждисциплінарного підходу стосовно створення системи підтримки електронного навчання.

Мета статті. Мета даної статті – дати огляд та аналіз специфічних рис побудови архітектури інструментальної середовищ їх складових, особливо мультимедійних баз даних, при використанні підходу гейміфікації при створення системи підтримки електронного навчання.

Виклад основного матеріалу дослідження

В процесі розробки електронного мультимедійного видання дидактичного характеру з використанням підходу гейміфікації використовується інструментальний широкий спектр систем. Їх раціонально розбити на наступні класи.

Специфічні системи побудови інтерактивного дидактичного контенту (наприклад, Adobe Captivate, Camtasia Studio).

Системи побудови інтерактивних електронних ігор (наприклад, Unity 3D, Microsoft XNA, UDK – Unreal Development Kit, CryEngine) як особистий випадок конструктори ігор (StencylWorks, Game Maker, Scirra Construct, CraftStudio Game Editor, NeoAxis Game Engine, 3D Game Studio) [1]. Спрощений інтерфейс дозволяє створювати ігри в освітніх організаціях з урахуванням їх вбудовування в навчальний процес.

Найбільш універсальний з ігрових движків Unity – інструмент для розробки двох- і тривимірних додатків та ігор, що працює під операційними системами Windows, Linux і OS X. Створені за допомогою Unity програми працюють під операційни-

ми системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а також на ігрових приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One.

Системи забезпечення простору зберігання ігрової інформації як на стадії розробки, так на стадії використання: Саме огляду основних архітектурних рішень цього класу систем і посвячена основна частина статті [5].

Можливо уточнити класифікацію за декількома критеріями.

За місцем зберігання інформації – безпосередньо на клієнті, у локальній мережі, у базі даних з он-лайн доступом (як окремий випадок, в хмарному сховищі).

За архітектурою сервера бази даних – SQL СУБД, постреляційні БД, NOSQL.

За кількістю одночасно підключених користувачів – одно чи багатокористувацька.

За орієнтацією контенту – скалярний контент чи векторний (окремий випадок – мультимедійний).

По-перше, розглянемо особливості представників за архітектурою сервера.

SQL СУБД. Типові представники: пропрієтарні MS SQL Server, Oracle, IBM DB2, Informix або безкоштовні MySQL, PostgreSQL, SQLite, Firebird, Microsoft SQL Server Compact.

Головні переваги пропрієтарних СУБД:

високий рівень підтримки на всіх стадіях життя дидактичного продукту (наявність розвинутих інструментальних засобів розробки та сервісного обслуговування; бібліотек компонент доступу);

можливість ефективного опрацювання мультимедійного контенту.

Таким чином, використання пропрієтарних СУБД забезпечує дуже високий рівень комфорту на стадіях розробки, розгортання, супроводу дидактичного продукту.

Головним недоліком є висока вартість володіння.

Головною перевагою безкоштовних є відносно низька вартість володіння.

Головним недоліком є менш високий рівень підтримки на всіх стадіях життя дидактичного продукту (порівняно невелика розвинутих інструментальних засобів розробки та сервісного обслуговування; бібліотек компонент доступу).

Постреляційні БД. Об'єктна модель SQL:1999, SQL:2003 включає два основні об'єктні компоненти – структурні, визначуваних користувачами типів даних (User Defined Type – UDT) і таблиці, що типізуються (Typed Table). Перший компонент дозволяє визначати нові типи даних, які можуть бути набагато складнішими, ніж вбудовані типи даних мови SQL. При визначенні структурного UDT потрібно специфікувати не лише елементи даних, що містять-

ся в ньому, але і семантику типу даних, тобто його поведінку на основі інтерфейсу викликів методів. Другий компонент – таблиці, що типізуються, – дозволяє визначати таблиці, рядки яких є екземплярами (або значеннями) UDT, з яким явно асоціюється таблиця. У багатьох відношеннях рядок таблиці, що типізується, схожий на об'єкт класу в об'єктно-орієнтованій системі.

Швидкість виконання запитів в них зростає до двадцяти разів порівняно з реляційними СУБД. Проте перехід від реляційних баз даних, які одержали повсюдне поширення, до постреляційних пов'язаний зі значними витратами і носить поки обмежений характер.

Незважаючи на методичну доцільність та привабливість використання стандартних постреляційних можливостей мови SQL, зараз їх ще не реалізовано у таких останніх реалізаціях найпоширеніших у видавничих БД серверах, як MS SQL та MySQL. Прикладом постреляційної бази є Cache фірми Inter Systems Corporation.

Таким чином, головним недоліком постреляційних СУБД є висока вартість входження для розробників.

Термін NoSQL (*від англ. Not Only SQL* – не лише SQL) став відомий відносно недавно і був введений для опису різних технологій баз даних, виниклих для задоволення вимог, відомих як Web-scale або Internet-scale.

Є дві причини, за якими люди розглядають можливість використовувати [5; 6] бази даних NoSQL.

- Ефективність розробки додатків. Більшість зусиль, пов'язаних з розробкою додатків, витрачаються на відображення даних з структур, що зберігаються в пам'яті, в реляційні бази даних. База даних NoSQL може забезпечити модель даних, краще задовольняє потреби додатків, спростивши тим самим цю взаємодію і зменшивши кількість коду, який необхідно написати, налагодити і розвинути.

- Великомасштабні дані. Організації цінують можливість зберігати більш великі обсяги даних і швидше їх обробляти. Вони вважають занадто затратним використовувати для цього реляційні бази даних. Головна причина полягає в тому, що реляційні бази даних призначені для роботи на одному комп'ютері, в той час як великі обсяги даних і програми для їх обробки економніше зберігати на кластерах, що складаються з численних невеликих і дешевих комп'ютерів. Багато баз даних NoSQL розроблені спеціально для кластерів, тому вони краще вписуються в сценарії оброблення великих обсягів даних. Хоча різноманітність моделей зростає, усі вони мають загальні риси:

- обробляють величезні об'єми даних, розділяючи їх між серверами;

- підтримують величезну кількість користувачів;

обмежену схемою структури бази даних.

Кожне рішення в рамках технології NoSQL використовує свою власну модель. Ці моделі поділяються на чотири категорії: ключ-значення, документ, сімейство стовпців і граф. Перші три моделі мають загальну властивість, яка називається агрегатною орієнтацією (aggregate orientation). Агрегатна орієнтація враховує необхідність оперувати даними, що мають більш складну структуру, ніж набір кортежем. Бази даних типу "ключ-значення", документ і сімейство стовпців можуть містити складні записи-агрегати (Aggregate). Моделі даних типу "ключ-значення" інтерпретують агрегати як "чорний ящик", тобто шукати можна тільки цілі агрегати, – ви не можете подати запит на отримання частини агрегату. У документній моделі агрегат є прозорим для бази даних. Це дозволяє посилати запити до фрагментів агрегату і здійснювати часткове витягнення даних.

Моделі типу "сімейство стовпців" поділяють агрегат на групи стовпців, інтерпретуючи їх як одиниці даних в агрегаті-рядку. Це накладає на агрегат структурні обмеження, але дозволяє базі даних використовувати цю структуру для поліпшення доступу.

Найбільш відомою з колонкових СКБД є BigTable – високопродуктивна пропрієтарна база даних, що підтримує стискування, побудована на основі Google File System (GFS), Chubby Lock Service і деяких інших продуктах Google. Зараз BigTable використовується в різного роду додатках Google, таких як MapReduce, яке часто використовується для створення і модифікації даних тих, що зберігаються в BigTable, Google Reader, Google Maps, Google Book Search, Search_History, Google Earth, Blogger.com, Google Code hosting, Orkut і YouTube.

Графові бази даних спеціально призначені для зберігання веб-інформації з дуже маленькими вузлами і численними зв'язками між ними.

Частиною стандартів розвинутих під час реалізації HTML 5.0 є IndexedDB – стандарт зберігання великих об'ємів структурованих даних на клієнті. IndexedDB служить для зберігання великих об'ємів структурованих даних, з можливістю індексації. Потреба в такому інструментарії назріла давно, що і привело до його появи в специфікаціях HTML5.соціальних мереж, товарних переваг або правил прийому на роботу.

Головною перевагою безкоштовних баз є висока ефективність обробки при великому обсязі даних або кількості користувачів.

Головним недоліком є менш високий рівень підтримки на всіх стадіях життя дидактичного продукту (порівняно невелика розвинутих інструментальних засобів розробки та сервісного обслуговування; бібліотек компонент доступу), відсутність чітких

принципів вибору конкретного рішення для дидактичного додатку.

Засоби розробки. Високу комфортність розробника забезпечують наступні інтегровані оболонки розробки – IDE (англ. Integrated development environment) Microsoft Visual Studio IDE для Windows та Mac, Google Android Studio, NetBrains, PhpStorm, NetBeans, Delphi XE.

Ці інтегровані оболонки розробки зазвичай підтримують програмування на найбільш популярних мовах програмування-C++,Java, JavaScript, PHP,C#,C.

У складі систем побудови інтерактивних електронних ігор (Unity 3D, Microsoft XNA, UDK – Unreal Development Kit , CryEngine) існують інтегровані чи окремі системи розробки ігрових світів та написання сценаріїв, зазвичай на скрипкових мовах. Дещо осторонь перебуває Unity 3D, в якому в якості мови написання сценаріїв використовується C#.

Майже всі перераховані засоби розробки підтримують мультиплатформеність, розробку ігор для Android, Microsoft Windows, iOS.

З точки зору розвитку бібліотек доступу до СУБД найбільш розвинутими можливостями володіє Unity, який використовує такі розвинені технології, як Microsoft .NET та Active Server Pages.

Висновки

Таким чином, з моєї точки зору, як це впливає з аналізу, що проведений, інструментом першого вибору є Unity, незважаючи на досить високий поріг входження, особливо при використанні Unity, як розробників студентів спеціальностей, що пов'язано з розробкою мультимедійних інтерактивних видань, які добре засвоїли сукупність використаних в цій оболонці парадигми.

Список літератури

1. Карпенко О.М. Геймифікація в електронному навчанні / О.М. Карпенко, А.В. Лукьянова, А.В. Абрамова, В.А. Басов // Дистанційне і віртуальне навчання. – 2015. – № 4.
2. 5 лучших материалов о геймификации образования в 2013 году [Електронний ресурс]. – Режим доступу: <https://te-st.ru/2014/01/22/best-education-gamification-2013/>.
3. Технология создания электронных обучающих систем [Електронний ресурс]. – Режим доступу: <http://www.dissercat.com/content/tehnologiya-ozdaniya-elektronnykh-obuchayushchikh-sistem>.
4. Тренды будущего: распределенное производство, геймификация, спортивный интернет [Електронний ресурс].— Режим доступу: <http://habrahabr.ru/post/153797/>.
5. Геймификация в образовании [Електронний ресурс]. – Режим доступу : <http://te-st.ru/2012/12/21/gamification-education>.

6. Фаулер М. NoSQL: новая методология разработки нереляционных баз данных / М. Фаулер, П. Садаладж: – М.: ООО "И.Д. Вильямс", 2013. – 192 с.

7. Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Э. Редмонд, Дж. Уилсон. – М.: ДМК Пресс, 2013. – 384 с.

8. Ткаченко В. Облачные вычисления (Cloud computing) обзор [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.ispring.ru/elearning->

insights/gameschool/ (07.06.15): [//www.lessons-tva.info/archive/nov031.html](http://www.lessons-tva.info/archive/nov031.html). – Назв. з екрану.

Надійшла до редколегії 10.05.2017

Рецензент: д-р техн. наук проф. В.Ф. Сорокін, Національний аерокосмічний університет ім. Н.С. Жуковського «ХАІ», Харків.

СОСТАВЛЯЮЩИЕ ИНСТРУМЕНТАЛЬНЫХ КОМПЛЕКСОВ ПОСТРОЕНИЯ СИСТЕМ ПОДДЕРЖКИ ЭЛЕКТРОННОГО ОБУЧЕНИЯ

А.К. Пандорин

В статье представлен анализ специфических черт использования составляющих обработки мультимедийных компонент создания систем поддержки электронного обучения. Рассмотрены основные проблемы, которые решаются в процессе выбора архитектуры таких систем с точки зрения геймификационной парадигмы. Приведен набор ключевых черт, определяющих архитектуру и состав системы с точки зрения разработчика.

Ключевые слова: компонентный подход, мультимедийные базы данных, электронное обучение, геймификация.

COMPONENTS OF THE TOOLING TOOLS FOR BUILDING E-LEARNING SUPPORT SYSTEMS

A. Pandorin

The paper presents an analysis of specific features using multimedia components processing component creation of e-learning support. The main problem to be solved in the selection process architecture such systems in terms of gamification paradigm. An set of key features that define the architecture and composition of the system in terms of the developer.

Keywords: component approach, multimedia databases, e-learning, gamification.