

ВДОСКОНАЛЕННЯ SHAREPOINT LEARNING KIT ДЛЯ РОБОТИ З НАВЧАЛЬНИМ КОНТЕНТОМ, СТВОРЕНИМ НА ОСНОВІ FLASH

СЕРГІЄНКО О. В.

АКІМЕНКО В. І.

Суми

Упровадження інформаційних технологій у систему навчання з кожним роком відкриває перед нами все нові й нові можливості під час організації освітнього процесу. Як показує практика, електронна форма навчання – це один із найефективніших, унікальних способів передачі знань, в основі якого лежить поєднання трьох ключових компонент: електронних навчальних матеріалів, системи управління навчанням та методичного забезпечення процесу електронного навчання. У свою чергу, багаторазове підвищення якості освітнього процесу здійснюється саме за рахунок використання електронного навчального контенту, що забезпечує не лише наочність подання інформації, але й, поєднуючи в собі різноманітні способи передачі знань, надає можливість для прояву всіх видів навчальної активності.

Під час організації навчального процесу в електронній формі досить важливим є не лише питання розповсюдження електронних навчальних матеріалів серед студентів, що може бути вирішено за допомогою звичайних автоматизованих систем документообігу, але й неабиякої уваги заслуговує питання автоматизації процесу проведення контрольних заходів задля перевірки залишкових знань. На сьогоднішній день існує цілий ряд безкоштовних програмних комплексів (систем управлін-

ня навчальним контентом), які дозволяють вирішувати задачі подібного роду, до них можна віднести: MOODLE, Claroline, Dokeos, ATutor, ILIAS, SAKAI, LAMS, OLAT, OpenACS, LRN, COSE, LON-CAPA, Colloquia, OpenLMS, The Manhattan Virtual Classroom, DodeboLMS, Acollab, Microsoft SharePoint Learning Kit. Дамо коротку характеристику деяких із них [6] (табл. 1).

Аналізуючи інформацію з наведеної табл. 1, можна відзначити, що більшість із представлених систем мають майже однаковий функціонал. Проте для подальшого розгляду нами була обрана система SharePoint Learning Kit, що пояснюється широким спектром її можливостей:

- ✦ підтримкою адаптивного тестування, мультимедійного та інтерактивного контенту;
- ✦ базовою підтримкою будь-яких електронних документів;
- ✦ широкими можливостями по доопрацюванню;
- ✦ тісною інтеграцією з Microsoft SharePoint.

Саме можливість інтеграції з SharePoint є основоположним фактором, оскільки на об'єкті впровадження (кафедра економічної кібернетики ДВНЗ «Українська академія банківської справи Національного банку України») внутрішній документообіг здійснюється саме на його основі.

Як вже згадувалося раніше, SLK підтримує використання навчального контенту створеного за наступними стандартами SCORM 2004, SCORM 1.2 та формат Microsoft Class Server (LRM). Тому в процесі розробки електронних навчальних матеріалів для організації перевірки залишкових знань на основі цих стандартів до-

Таблиця 1

Порівняльна характеристика систем управління навчальним контентом

	MOODLE	LAMS	Sakai	ATutor	Claroline	Dokeos	OLAT	SLK [3]	ILIAS
SCORM	+	-	+	+	+	+	+	+	+
IMS	+	-	+	+	+	+	+	+	-
Мова додатків	PHP	Java	Java	PHP	PHP	PHP	Java	ASP.Net	PHP
СУБД	MySQL	MySQL	MySQL, Oracle, hsqldb	MySQL	MySQL	MySQL	MySQL Postgre SQL	MS SQL Server	MySQL
Ліцензії	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	Ms-CL	GNU/GPL
Російська мова	+	-	+	+	+	+	+	+	+
Інші мови	більше 54	20	28	більше 50	36	38	34	29	43
Система перевірки знань	тести, завдання, семінари, активність на форумі	тести	тести, завдання, активність на форумі	тести	тести, вправи	тести	тести, завдання	тести, завдання	тести

силь важливим є питання вибору програмного інструментарю для їх створення з точки зору можливостей:

- ✦ вибору різних типів запитань;
- ✦ включення мультимедійних матеріалів;
- ✦ формування завдань з набору тестів випадковим чином;
- ✦ обмеження часу виконання завдань;
- ✦ здійснення роботи над помилками і т. п.

На сьогоднішній день на ринку програмних засобів присутній цілий спектр додатків, проте особливої уваги, на наш погляд, заслуговують системи, які дозволяють створювати навчальні матеріали на основі Flash, що в першу чергу пов'язано зі зручністю й швидкістю їх створення та публікації.

До таких систем можна віднести: AWind Quiz-Builder, ViewletQuiz Professional, A-PDF Quizer, Quiz-ForceRU, Cabasoft QuizMaker, Wondershare QuizCreator, iSpring Presenter, Tanida Quiz Builder, Articulate Quiz-maker, Adobe Captivate, Course Lab. Всі вони дозволяють створювати матеріали у вигляді пакетів, сумісних з форматом SCORM (SCO) для LMS, маючи при цьому весь спектр описаних вище можливостей.

Під час вивчення можливостей SLK нами було виявлено деякі недоліки в роботі даної системи, що пов'язані з відстеженням процесу проходження студентом виданих йому завдань, для матеріалів створених на основі Flash. Сутність самої проблеми полягає в тому, що викладач не має можливості переглянути відповіді, що були обрані студентом, що унеможливило у майбутньому проведення роботи над помилками. Більш детальний розгляд даної задачі показав, що на сьогоднішній день не представлено жодного з варіантів її рішення, тому вона має неабияку актуальність та потребує додаткового дослідження.

Мета дослідження – визначити причини неможливості перегляду викладачем відповідей студентів на запитання для SCO створеного на основі Flash в системі SLK. Запропонувати можливі варіанти рішення даної задачі у разі можливості їх виокремлення та реалізувати їх.

Аналізуючи процеси видачі завдання викладачем студенту, виконання завдання студентом та перевірки результатів тестування викладачем в системі SLK (більш детально див. [4]) було виявлено таке:

- ✦ під час виконання завдання студент змушений перервати цей процес з метою подальшого його продовження:

1) студент повторно запускає завдання з комп'ютера, за яким розпочинав його виконання – SCO пропонує продовжити виконання, при цьому попередньо обрані відповіді зберігаються;

2) студент повторно запускає завдання з іншого комп'ютера ніж той, за яким розпочинав його виконання – SCO пропонує продовжити виконання, але при цьому попередньо обрані відповіді не зберігаються;

- ✦ у процесі перевірки результатів проходження тестів викладач намагається проглянути відповіді окремо обраного студента – запускається SCO, але при цьому обрані студентом відповіді не відображаються.

Як можна помітити, ця проблема пов'язана з процесом обміну даними між SLK та SCO. Даний висновок може бути зроблений саме на основі того факту, що обрані студентом відповіді зберігаються лише на локальному комп'ютері, на якому було запущено SCO і не можуть бути завантажені з централізованої бази даних LMS у разі продовження виконання завдання на іншому комп'ютері. Слід зазначити, що дана тенденція прослідковується лише для SCO на основі Flash, які в свою чергу створюються за стандартом SCORM, в основі якого лежить клієнт-серверна модель обміну даними (при цьому сервером у ній виступає LMS).

Для того, щоб здійснювати взаємодію між клієнтською і серверною частинами навчального середовища, необхідною є наявність єдиного програмного інтерфейсу (Application Program Interface (API) – набору функцій, до яких може звертатися SCO. У частині RTE стандарту SCORM описано, який API може використовуватися SCO-об'єктами для взаємодії з службами середовища виконання (run-time services (RTS) – програмний додаток, що контролює процеси виконання і доставки навчального матеріалу, а також може надавати додаткові сервіси: розміщення елементів навчального матеріалу, виконання завдань за розкладом, контроль вводу-виводу, управління даними). Таким чином, API робить можливим взаємодію між навчальним матеріалом і RTS, доступ до яких зазвичай надається LMS у вигляді загальних функцій API, доступних через функції мови «JavaScript». При цьому, те, яким чином здійснюється взаємодія між реалізацією API, закладеною в конкретній LMS, і серверною частиною цієї LMS, не описано в даному стандарті, і ця реалізація може бути здійснена таким чином, який є необхідним для розробників LMS. Врахування даного факту ще раз підтверджує правильність припущення щодо наявності проблеми обміну даними між SLK та SCO.

Для остаточного підтвердження нашого припущення спробуємо проаналізувати взаємодію між SCO та SLK. Спочатку розглянемо типову функціональну архітектуру LMS (рис. 1 [7]).

Аналізуючи представлену вище схему, слід звернути увагу на такий її компонент, як «Локальне сховище контенту». Дане сховище призначене для зберігання на стороні клієнта (локальний комп'ютер на якому запущено SCO на виконання) SCO пакету на період проходження завдання. Виходячи з цього маємо, що дані сховища є тимчасовими і з точки зору операційних систем пов'язані з використанням тимчасових файлів, аналіз вмісту яких дозволить остаточно підтвердити або спростувати наше припущення.

Для відстеження процесів видачі, проходження тестів та аналізу їх результатів на основі «Локальних сховищ контенту» у випадку SLK налаштуємо на сервері режим відладки для Basic Web Player [5], та спробуємо видати нове завдання і пройти його. Вікно звіту щодо взаємодії пакету SCORM і SLK фреймів представлено на рис. 2.

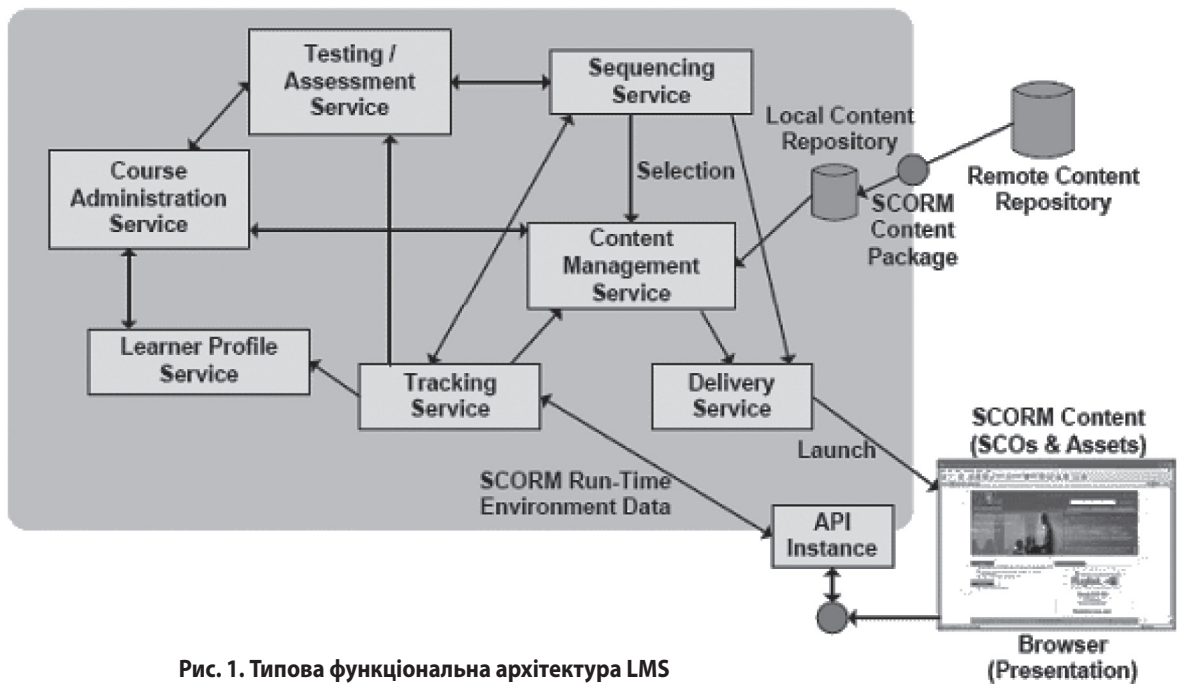


Рис. 1. Типова функціональна архітектура LMS

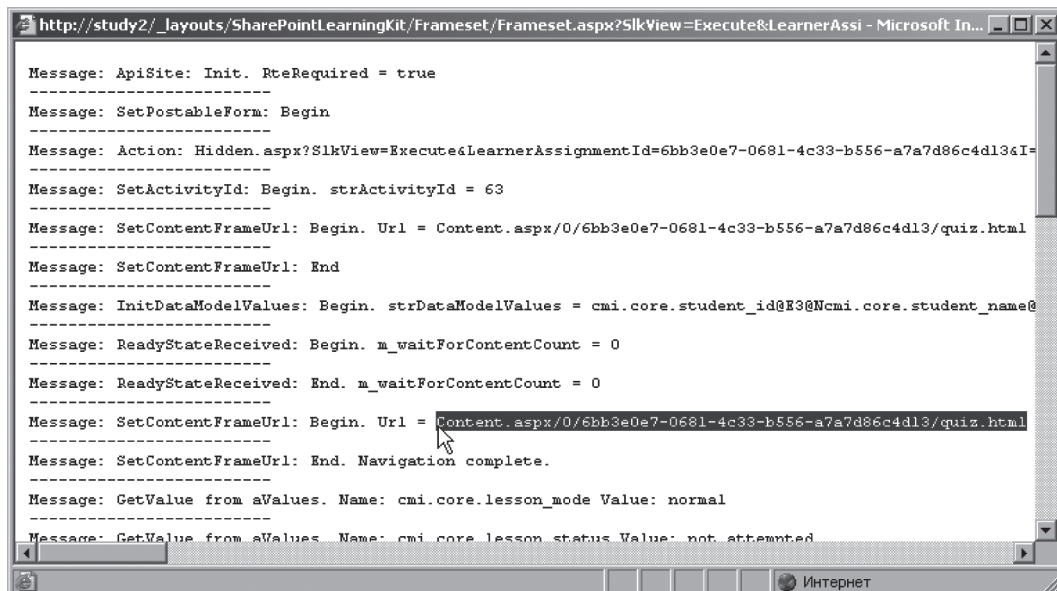


Рис. 2. Звіт щодо взаємодії пакету SCORM і SLK фреймів

Як ми можемо побачити з рис. 2, у модулі Basic Web Player на основі методу «SetContentFrameUrl()» здійснюється завантаження контенту «Url = Content.aspx/0/6bb3e0e7-0681-4c33-b556-a7a7d86c4d13/quiz.html». Здійснивши пошук по тимчасовим файлам на локальному комп'ютері, було виявлено, що під час виконання призначеного завдання відбувається завантаження навчального контенту в тимчасову папку локального профілю користувача з назвою «6bb3e0e7-0681-4c33-b556-a7a7d86c4d13» (ідентифікатор SCO в базі даних SLK). При цьому в даній папці створюється файл формату *.sol.

SOL-файли (загальні локальні об'єкти (Local Shared Objects) – так звані флеш-куки (Flash Cookies), фрагменти даних з сайтів, що використовують Adobe Flash, які зберігаються на комп'ютері користувача. Вони

можуть бути використані для збереження налаштувань деяких веб-додатків створених на основі Flash.

Що стосується SCO, то вони використовують *.sol файли для збереження всіх маніпуляцій користувача (студента) з навчальним контентом, включаючи й збереження обраних студентом варіантів відповідей.

Вивчення вмісту даних файлів дозволило встановити такі факти:

- ✦ під час отримання нового завдання студентом та запуску SCO на локальному комп'ютері, здійснюється перевірка наявності в локальному профілі папки з ідентифікатором даного SCO в базі даних SLK, так як дана папка відсутня – створюється нова папка з заданим ім'ям і порожнім файлом типу *.sol, в якому потім відображаються всі маніпуляції з SCO;

✦ при відновленні проходження завдання можливі дві ситуації:

1) студент проходить завдання на тому ж комп'ютері, що й починав – здійснюється перевірка наявності папки з ідентифікатором даного SCO в базі даних SLK, оскільки така папка вже присутня в його профілі – SCO зчитує дані з *.sol файлу, таким чином перед студентом з'являються всі попередньо відзначені відповіді;

2) студент проходить завдання на іншому ніж той, що починав комп'ютері – здійснюється перевірка наявності створеної папки з ідентифікатором даного SCO в базі даних SLK, так як дана папка відсутня – створюється нова папка з заданим ім'ям і порожнім файлом типу *.sol, оскільки щойно створений файл буде порожнім то SCO не вдасться зчитати ніякої інформації стосовно попередніх відповідей студента і всі попередньо відзначені варіанти будуть втрачені;

✦ викладач перевіряє результати проходження тесту студентом, завантажуючи їх на свій комп'ютер – здійснюється перевірка наявності створеної папки з ідентифікатором даного SCO в базі даних SLK, так як дана папка відсутня – створюється нова папка з заданим ім'ям і порожнім файлом типу *.sol, оскільки щойно створений файл буде порожнім то SCO не вдасться зчитати ніякої інформації стосовно попередніх відповідей студента і всі попередньо відзначені варіанти будуть втрачені.

Проведене дослідження дає можливість стверджувати той факт, що API SLK не дозволяє SCO, яке створене на основі Flash повною мірою здійснювати взаємозв'язок (операції запису та зчитування) з базою даних SLK.

Враховуючи можливість всебічного доступу до загальних локальних об'єктів, створених Flash-додатками в

рамках одного сайту [2], а також того факту, що SLK розповсюджується за ліцензією Microsoft Community License (Ms-CL) for SharePoint Learning Kit і має відкритий програмний код [3] – одним із варіантів вирішення даної проблеми є занесення в кінці кожної сесії взаємодії студента з SCO вмісту отримуваних файлів типу *.sol в базу даних SLK, та в майбутньому – підміна відповідних файлів при повторній ініціалізації виконання завдання студентом або при спробі перевірки результатів викладачем.

Отже, рішення, яке було отримане у результаті дослідження, додає у робочий процес LMS-тестування на базі SLK та SCORM, декілька нових ланок. Основні етапи алгоритму представлені на рис. 3.

Як бачимо, даний алгоритм ґрунтується на підміні інформації, яка міститься у *.sol-файлі вчителя, інформацією із *.sol-файлу учня. Даний процес є можливим завдяки використанню основної бази даних SharepointLearningKit у якості сховища для інформації, яка отримується із файлу учня після закінчення складання ним тесту.

Для втілення у життя даного алгоритму, потрібно було подолати декілька проблемних питань:

1. Як знайти потрібний *.sol-файл учня, адже даний файл використовується виключно Flash-плеєром, та доступу до його адреси з SLK немає?
2. Коли саме, у яке поле бази даних, і яким чином записувати отримані з файлу дані у базу?
3. Коли саме виконувати зчитування інформації з бази даних?
4. Яким чином знайти відповідний *.sol-файл для підміни інформації?

Суть першого питання полягає в тому, що шлях до *.sol-файлу неможливо отримати програмно з коду SLK, оскільки він використовується у рамках процесу Flash-

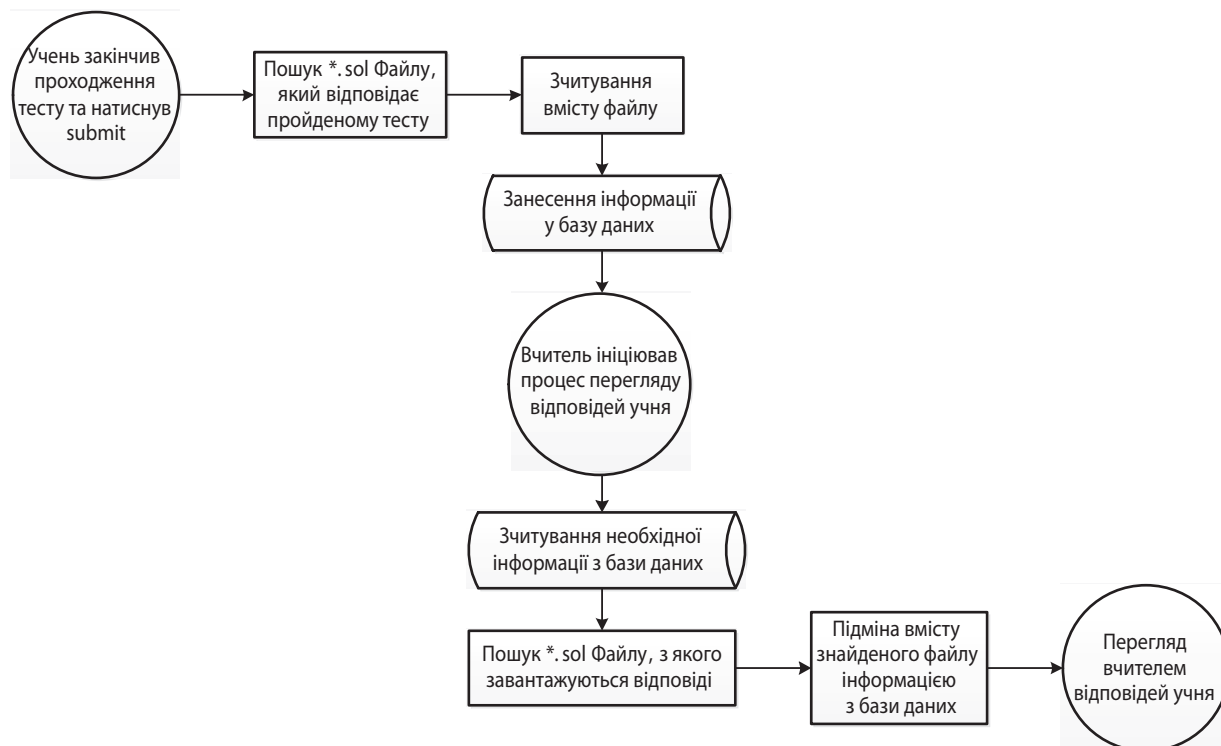


Рис. 3. Алгоритм рішення

плеера. Адреса файлу будується за таким шаблоном: %AppData%\Macromedia\Flash Player\#SharedObjects\<невідомо папка>\<ім'я користувача>_layouts\SharePointLearningKit\Frameset\Content.aspx\0\<id тестування>\<деякий порядковий номер>\quiz.swf

Таким чином, для отримання потрібного файлу було вирішено використати алгоритм рекурентного пошуку всіх *.sol-файлів у папці %AppData%\Macromedia\Flash Player\#SharedObjects\. Слід зазначити, що даний процес відбувається швидко, оскільки файлове дерево даної папки не є важким.

Після довгих міркувань, шляхом спроб і помилок, була отримана відповідь на друге та третє проблемне питання. Для можливості збереження потрібної інформації, буда дороблена схема бази даних SharepointLearningKit, була додана одна колонка до таблиці ActivityAttemptItem із назвою «SOL» та типом varchar(max), оскільки у файлі *.sol генерується значний об'єм розмітки, який залежить лише від розміру самого тесту. Також були внесені корективи у представлення, які пов'язані із таблицею ActivityAttemptItem.

Найвдалішим моментом для пошуку та зчитування *.sol-файлу є момент натиснення кнопки Submit учнем. У коді SLK в цей час виконується метод Microsoft.LearningComponents.Storage.DatabaseNavigator.Save(), у код якого і була додана логіка даного рішення.

Для зчитування інформації з бази даних було обрано метод Microsoft.LearningComponents.Storage.DatabaseNavigator.LoadLearningData Model(), який виконується під час завантаження та підготовки тесту до старту у Flash-плеєрі.

Після того, як вчитель розпочне перегляд відповідей учня, для нього формується пустий *.sol-файл без відповідей, який шукається так само, як і при зчитуванні даних. Як тільки цей файл буде знайдений, його вміст замінюється на збережені у базі даних значення, і таким чином досягається підміна відповідей вчителя відповідями учня для подальшого їх перегляду.

ВИСНОВКИ

За результатами проведеного дослідження було виявлено деякі недоліки у роботі SharePoint Learning Kit, які пов'язані з неможливістю відображення детальної інформації стосовно процесу проходження тестового контролю з використанням навчального електронного контенту створеного на основі Flash. Було запропоновано універсальний підхід, який дозволяє виправити вищезазначений недолік SLK для всіх навчальних матеріалів, побудованих на основі Flash, а також наведено реалізацію даного підходу. Одним із можливих варіантів подальшого розвитку у даному напрямі є розробка спеціалізованого програмного забезпечення по створенню навчального контенту на основі стандарту SCORM, адаптованого для взаємодії з SharePoint Learning Kit через API механізми з боку створюваного SCO, що також є можливим, оскільки даний механізм підтримується SLK для матеріалів створених у форматі Microsoft Class Server (LRM). ■

ЛІТЕРАТУРА

1. Flash Cookie Forensics Object [Електронний ресурс]. – Режим доступу : <http://computer-forensics.sans.org/blog/2009/08/28/flash-cookie-forensics/>
2. Local_Shared_Object [Електронний ресурс]. – Режим доступу : http://en.wikipedia.org/wiki/Local_Shared_Object
3. SharePoint Learning Kit [Електронний ресурс]. – Режим доступу : <http://slk.codeplex.com>
4. SharePoint Learning Kit [Електронний ресурс]. – Режим доступу : <http://www.microsoft.com/rus/education/slk/default.aspx>
5. Willis, R. Debugging the SharePoint Learning Kit Frameset and SCORM Interactions [Електронний ресурс]. – R. Willis. – Режим доступу : <http://blog.salamandersoft.co.uk/index.php/2011/08/debugging-the-sharepoint-learning-kit-frameset-and-scorm-interactions/>
6. Богомолов В. А. Обзор бесплатных систем управления обучением [Електронний ресурс]. – В. А. Богомолов. – Режим доступу : http://ifets.ieee.org/russian/depository/v10_i3/html/9_bogomolov.htm. – International Forum of Educational Technology & Society
7. Стандарт SCORM и его применение [Електронний ресурс]. – Режим доступу : <http://cccp.ifmo.ru/scorm/>