

О.В. Конюшенко, Д.О. Івченко

Побудова інтерактивного електронного навчального посібника в системі управління контентом E107

Рассмотрено построение интерактивного электронного учебного пособия для изучения языков программирования (объектно-ориентированного программирования). Описаны специализированный веб-инструментарий объединения разноформатных учебных материалов в единое интерактивное пособие, основные этапы проектирования и концепция архитектуры системы.

An interactive electronic textbook construction for the programming languages study (Object Oriented Programming) is considered. A specialized web-toolkit that combines multiformat learning materials into a single interactive guide, the basic stages of the design and the architecture system concept are described.

Розглянуто побудову інтерактивного електронного навчального посібника для вивчення мов програмування (об'єктно-орієнтованого програмування). Описано спеціалізований веб-інструментарій об'єднання різноформатних навчальних матеріалів в єдиний інтерактивний посібник, основні етапи проектування і концепція архітектури системи.

Вступ. Специфіка сучасного електронного навчання – це стрімке впровадження та поширення комп'ютерних технологій у всіх його галузях [1]. Але попри всі позитивні сторони цього процесу з'являються такі нові проблеми, як збільшення обсягів інформації і витрат на обробку та аналіз даних, проблеми структуризації матеріалу тощо. Особливо це стосується навчання, пов'язаного з інформаційними технологіями. Тому виникає потреба у створенні спеціалізованого інструментарію, який дозволяв би ефективно оперувати навчальними матеріалами, давати можливість організувати їх за власним бажанням, створювати навчальні посібники з елементами інтерактивності і разом з цим мати зручний, простий та інтуїтивно зрозумілий інтерфейс.

Існують різні системи управління контентом, зокрема системи створення та керування електронними курсами або портфоліо. Але з огляду на специфіку, пов'язану з програмуванням, вони не відповідають повною мірою необхідним вимогам: відсутність орієнтації на специфічні формати матеріалів з конкретної галузі; відсутність інтерактивності у роботі з матеріалами з різних мов програмування; складна система адміністрування та підтримки навчального процесу.

Опишемо основні можливості та переваги розробленої системи: повне керування створеними курсами, розміщення в них будь-яких матеріалів, оперування цими матеріалами; зручне подання презентацій та програмного коду – важ-

ливих складових навчання з об'єктно-орієнтованого програмування (ООП); інтерактивність роботи з програмними проектами – динамічне виконання та тестування; можливість захистити матеріали від несанкціонованого доступу шляхом використання технології *Flash* (захист авторських прав); незалежність у демонстрації матеріалів від будь-якої платформи; динамічний зв'язок між розділами *LCMS* «Навчальні курси», автоматизованих практикумів «Практичні роботи» та електронних портфоліо «Портфоліо».

Концепція робочого зошиту викладача

Електронне навчання природно застосовується в сфері дистанційного навчання, але також може широко застосовуватись не тільки дистанційно, а й за безпосереднього контакту між викладачем та студентами, що можна охарактеризувати як змішане навчання. Значною мірою система організації навчальних матеріалів з ООП орієнтована на такий змішаний тип навчання, тобто як дистанційний інструмент для навчання, так і за безпосереднього контакту учасників навчального процесу.

В сфері вищої освіти поширюється тенденція до створення віртуальних навчальних середовищ [2], які іноді об'єднуються з адміністративною інформаційною системою для створення керованого навчання. Крім того, деякі університети пропонують мережеву студентську службу підтримки, таку як он-лайн повідомлення, електронні рекомендації, студентські управління тощо.

Термін *електронне навчання 2.0* наближає процес навчання до мережних технологій Веб 2.0. Колишній підхід до електронного навчання передбачав певні навчальні пакети в межах інтернет-технологій, що використовувались студентами. Роль студента полягала у вивченні прочитаного і підготовці завдань, які оцінювались викладачем. Новий підхід надає більше значення соціальному навчанню, зокрема використанню соціального програмного забезпечення [3]. Саме в такому напрямі працює створений електронний підручник, який інтегрується з електронними портфоліо та системою підтримки автоматизованих практикумів, створюючи у такий спосіб соціальну навчальну систему, де забезпечується інтерактивність, соціалізація, взаємодія будь-яких суб'єктів навчання. Причому дана система передбачає можливість подальшого розвитку, налаштування нових сервісів, ще більше інтеграції з новими ресурсами.

Соціальне навчання (на відміну від традиційного картезіанського підходу) змінює підхід до самого об'єкта – знання, яке тепер стало соціально конструйованим в обговорюваннях контенту та у взаємодії при вирішенні певних проблем. Дійсно, найкращий спосіб щось вивчити, це почати цьому вчити інших.

Для управління навчальним процесом зазвичай використовуються *LMS*-системи. Системи управління навчанням поступово розвиваються в системи управління навчальним контентом (мультикористувацьке середовище, де можна створювати, зберігати, застосовувати нове використання (*reuse*), управляти електронними матеріалами електронного навчання, зосередженими в певному центральному репозитарії [4]. Зараз часто використовують узагальнене поняття *CLCIMS* (*Computer Learning Content Information Management System*) для визначення навчальних середовищ.

Все ж у більшості всі згадані системи в основному спрямовані на організацію взаємодії між викладачем та студентом (завдяки тематичним форумам, формам контакту, організації здачі домашніх завдань тощо). Також іноді використовуються портфоліо, основна задача

яких – представлення/збереження матеріалів певного суб'єкта процесу навчання.

Одним із рішень є *віртуальний клас*. Він є користувацьким ядром освітнього ІТ-середовища і являє собою комплексну розподілену систему. В неї зазвичай входять інфраструктурні програмні й технічні компоненти, що віртуально поєднують робочі місця викладача та студентів у навчальну групу, яка працює в мережі (локальній або глобальній). Проте віртуальний клас також не вирішує проблему представлення навчальних матеріалів з ООП, він орієнтований на інтерактивну взаємодію учасників навчального процесу.

Робочий зошит викладача – це те середовище, де буде зосереджено всі навчальні матеріали, які викладач забажає розмістити, причому це не просто репозитарій, а певна динамічна структура, яка повинна мати належну функціональність, адже цей зошит використовуватимуть і викладачі, і студенти як на лекціях та практичних заняттях, так і при «віддаленому навчанні». Тому на нього накладаються такі вимоги: має бути суцільним середовищем; обов'язкова можливість швидкої демонстрації виконання програм, без потреби «збирати» проект на різних локальних машинах; можливість працювати як з наочними матеріалами (демонстраційні програми), так і з матеріалами-завданнями для студентів; керування своїм власним робочим простором; підтримка віртуальних класів.

Дослідження свідчать, що студенти надають перевагу курсам змішаного типу (гібридним), які використовуються в 40 процентах університетів, розглянутих у [5], і число пропозицій зростає більше ніж на 10 процентів щорічно.

Розробка системи управління навчальними матеріалами в робочому зошиті викладача

При розробці структури проекту основна увага була зосереджена на максимально просто для користувача, і водночас здатному задовольнити вимоги викладачів інструменті. На рис. 1 подано вигляд сторінки при перегляді матеріалів курсу.

Навчальні курси

Навчальний курс: Object Oriented Programming with C++
 Автор (вкладач): DmytroVtchchenko (Переглянути всі курси автора)
 Концепції сучасного програмування в рамках парадигми провідно-орієнтованого програмування, абстрактних типів даних (об'єктне програмування) об'єктно-орієнтованого (баричного) у їх розвитку / взаємозв'язку, Основу вивчення складають поняття типу, функції, в тому числі узагальненої, об'єкта і класу, в тому числі параметризованого. Проблеми розширення області визначення функції, спеціалізація класів, побудова кращих об'єктів і класів, збереження поліморфізму, улаштування інтерфейсів / реалізації.



Рис. 1

Як видно з зображення, при перегляді курсу можна продивлятися самі матеріали, код лабораторних робіт, запускати ці лабораторні роботи на виконання і переглядати результат цього виконання, читати коментарі викладача. Також можна працювати зі звичайними вкладеннями. Викладач в свою чергу може змінювати код програм, і одразу запускати їх на виконання, маючи таким чином змогу ефективно демонструвати поставлені задачі на лекції/практичному занятті. Викладач може також контролювати хід навчального процесу, активуючи або дезактивуючи відповідні матеріали. Ця ж сторінка при перегляді викладачем (автором курсу) виглядає як на рис. 2.

Редагування контенту

Навчальний курс: Object Oriented Programming with C++
 Концепції сучасного програмування в рамках парадигми провідно-орієнтованого програмування, абстрактних типів даних (об'єктне програмування) об'єктно-орієнтованого (баричного) у їх розвитку / взаємозв'язку, Основу вивчення складають поняття типу, функції, в тому числі узагальненої, об'єкта і класу, в тому числі параметризованого. Проблеми розширення області визначення функції, спеціалізація класів, побудова кращих об'єктів і класів, збереження поліморфізму, улаштування інтерфейсів / реалізації.



Рис. 2.

Наочно показано, що для викладача реалізовано згадані задачі, а також набір стандартних операцій, таких як редагування, вилучення, реорганізація структури дерева навчальних матеріалів.

База даних застосування містить чотири таблиці:

```
e107_coursecreator_attachment,  
e107_coursecreator_courses,  
e107_coursecreator_program,  
e107_coursecreator_uploadunit.
```

Наприклад, структура таблиць e107_coursecreator_courses така:

```
`ID` int(11) NOT NULL auto_increment,  
`abbr` varchar(20) NOT NULL default "",  
`name` varchar(255) NOT NULL default "",  
`description` text NOT NULL,  
`owner` int(11) NOT NULL default '0',  
`sort` int(11) default NULL,  
`active` enum('0','1') NOT NULL default '1'.
```

Структуру системи подано на рис. 3.

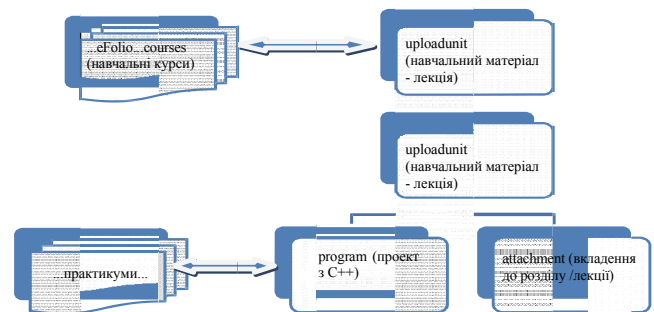


Рис. 3

В правій частині зображено схему представлення даних в LCMS, стрілками показано динамічні зв'язки з системою електронного портфоліо та студентських автоматизованих практикумів. Така схема водночас робить процес ознайомлення з новим веб-інструментом максімально інтуїтивним, забезпечує відповідну функціональність та дає змогу розвивати цю систему в майбутньому без перешкод. Це також стосується он-лайн компіляції не тільки проектів з C++, а будь-яких інших практикумів, написаних на мові, що підтримується бібліотекою GCC, але про це – в наступних розділах.

Процес додавання контенту реалізовано у такий спосіб: викладач створює курс, до якого додає матеріал у вигляді веб-сторінки або флеш-презентації, до цього матеріалу можна додавати проекти з C++ (можливо також з інших мов програмування) або будь-які вкладення. Всі результати цих дій видно як на головній сторінці

плагіну, так і на сторінці редагування у автора курсу.

Веб-сторінки, які додає автор курсу – це файли, отримані шляхом конвертації документів (зазвичай *Microsoft Word*) в формат *HTML*. Автори статті цей процес поклали на плечі авторів курсу, адже це запобігає проблеми сумісності форматів при автоматичній конвертації, а також робить систему створення інтерактивних навчальних матеріалів універсальною, тобто відкритою до створення навчальних матеріалів з будь-яких наук, дозволяє додавання будь-яких документів, що конвертуються в *HTML* або *Flash*. Це дозволяє розширити застосування даного проекту за межі об'єктно-орієнтованого програмування.

Навчальні матеріали завантажуються на сервер в основному в вигляді архівів. Тому в даному проекті використано бібліотеку *pclzip* [6], за допомогою якої можна проводити багато операцій над архівами формату *.zip*: створювати, розархівовувати, вилучати, зливати в один, дублювати тощо. Цей уривок коду ілюструє одне з використань даної бібліотеки в практичній частині роботи:

```
require_once('unzip/pclzip.lib.php');
$archive = new PclZip($target);
if($archive->extract(PCLZIP_OPT_PATH,
$target_extract) == 0)
{
    die("Помилка : ".$archive->errorInfo(true));
}
```

Реалізація інтерактивності навчання проведена застосуванням динамічного оновлення та компілювання. Динамічність полягає в можливості переглядати виконання проекту одразу після внесення змін в програмний код. Всі дії відбуваються на сервері через веб-інтерфейс браузера. Для проектів, що потребують введення даних в програму вже після процесу компіляції і лінування, передбачено також можливість завантажити виконавчий файл на комп'ютер і вже на ньому виконувати програму. Крім того, при помилках в програмному коді автор може бачити, які повідомлення про помилки та попередження генерує компілятор *GCC* на сервері.

Кожен викладач має повний контроль над своїми матеріалами, і ніяк не може дістатися матеріалів інших авторів. Ці налаштування пов'язані з інтегруванням в систему авторизації (та аутентифікації) користувачів в системі управління контентом *E107*. На головній сторінці відображаються матеріали всіх авторів, однак можна вибрати конкретного автора і передивлятися тільки його матеріали. Адміністратор у свою чергу має доступ до матеріалів всіх викладачів і може контролювати наповнення. Але він не може редагувати їх, може тільки активувати/деактивувати або видаляти.

Налаштування роботи *GNU Compilers Collection* під *OC FreeBSD*

GNU Compiler Collection (GCC) – набір компіляторів для різних мов програмування, розроблений в межах проекту *GNU* [7], який є відкритим програмним забезпеченням і поширюється на умовах ліцензій *GNU GPL* і *GNU LGPL*, використовується як стандартний компілятор для відкритих *Unix*-подібних операційних систем, а також для *Apple Mac OS X*. Крім того, що *GCC* є офіційним компілятором системи *GNU*, включаючи варіації системи, побудовані на ядрі *Linux (GNU/Linux)*, *GCC* також є головним компілятором для зборки інших операційних систем, серед яких: різні варіанти *BSD*, *ReactOS*, *Mac OS X*, *NeXTSTEP* і *BeOS*.

GCC часто обирається для розробки програмного забезпечення, що повинно працювати на різних апаратних платформах. Різниця між «рідними» для кожної з апаратних платформ компіляторами створюють труднощі при розробці коду, який би коректно компілювався різними компіляторами. При використанні *GCC* для компіляції коду під різні платформи застосовується один і той же синтаксичний аналізатор. Тому якщо вдалося зібрати програму для однієї з цільових платформ, то є велика ймовірність, що програма нормально збереться й для інших платформ.

Зовнішній інтерфейс *GCC* є стандартом для компіляторів на платформі *Unix*. Користувач викликає керуючу програму *gcc*. Вона інтерпретує аргументи командного рядка, визначає й запускає для кожного вхідного файлу свої

компілятори потрібної мови, запускає, якщо необхідно, асемблер і компоувальник [8].

Для генерування команд виконання компіляції на сервері була написана функція *CompileStringCPP*:

```
function CompileStringCPP($dir)
{
    $CompileString = "";
    $dhandle = opendir($dir); if ($dhandle)
    {
        while (false !== ($fname = readdir($dhandle)))
        {
            if (is_file( "{$dir}/{$fname}" ))
            {
                $path_parts = pathinfo($fname);
                $ext = $path_parts['extension'];
                if (!strcasecmp($ext, 'cpp'))
                $CompileString .= "{$dir}/{$fname} ";
            }
        }
        closedir($dhandle);
    }
    $CompileString = 'g++ '.$CompileString.' -o '.$dir.'/project.exe'; return $CompileString;
}
```

Команда *G++* викликає компілятор для програм написаних на *C++*. При подальшій розробці проекту є можливість розробити універсальне середовище, що буде компілювати, лінкувати та генерувати виконавчі файли для таких мов як *C*, *Objective-C*, *Java*, *Fortran* та *Ada*, і тому можна буде повною мірою реалізувати роботу з набагато більшою кількістю курсів з об'єктно-орієнтованого програмування.

Особливості реалізації

В основу реалізації покладено використання платформи *e107* [20] – набір скриптів, написаних на *PHP*, які взаємодіють з *MySQL*, що в цілому забезпечує відмінну готову порталну систему. З її допомогою можна створювати як величезні порталні системи, так і невеликі сайти. В нашому випадку на цій системі базується навчальний портал *Joint European Projects* (<http://jep.ukma.kiev.ua>), і оскільки *e107* є модульною системою, орієнтованою на роботу з функціональними розширеннями та модулями, розроблений інструмент з оперування навчальними матеріалами був реалізований у вигляді плагіну.

Основними особливостями *E107* є високо-ефективне файлово-орієнтоване кешування, інтегрована системи новин та *RSS*, спрощений процес створення та використання шаблонів, валідний *XHTML*-код, потужна система підтримки форумів, зручний підхід до створення нових модулів, зручна система адміністрування.

Спільна інтеграція з проектом підтримки автоматизованих практикумів та системою електронних портфоліо впроваджувалася для можливості викладачу швидко переходити від перегляду свого робочого зошиту (система створення курсів і наповнення їх контентом) до перегляду робіт студентів, що, наприклад, виконують лабораторну роботу, і навпаки. Для того, щоб забезпечити таку функціональність, для зберігання даних в *LCMS* та практикумах про лабораторні роботи, задані викладачем, використовується спільна таблиця *e107_course_creator_program*. Практичні роботи, збережені в цій таблиці, відображаються як навчальні матеріали певного розділу певного навчального курсу. Студентські практикуми прив'язані до певних робіт, створених викладачем. Отже, викладач при перегляді курсу може орієнтуватися на посилання і бачити, скільки студентів беруть участь, що зробили з конкретної лабораторної роботи. Він може повернутись на свій робочий зошит, адже в робочому зошиті студента відомо, до якої лабораторної роботи і до якого курсу «прив'язаний» його практикум.

Існує також можливість подібної навігації між системою створення навчальних курсів та між системою електронних портфоліо. При перегляді курсів можна переходити по посиланню на особисте електронне портфоліо автора курсу або переглянути всі курси даного автора. В свою чергу через інтерфейс *eFolio* можна бачити інформацію про матеріали конкретного викладача (автора курсу).

Висновки. Створена *LCMS* дозволяє пришвидшити і покращити процес навчання, зробити його максимально наочним, внести елемент інтерактивності у роботу з матеріалами мов програмування, уникнути проблеми несумісності платформ, централізувати розрізнений навчальний матеріал. В контексті стрімкого розвитку

веб-технологій це дозволяє наблизитись до якісно нового рівня організації навчання як електронного, так і змішаного типу в царині дисциплін, пов'язаних з програмуванням.

1. *Програмні засоби створення і супроводу розподіленого навчального середовища* / І.В. Сергієнко, М.М. Глибовець, С.С. Гороховський та ін. // Нац. ун-т «Києво-Могилянська академія». – К.: НаУКМА, – 2012. – 709 с.
2. *Virtual learning environment*. – http://en.wikipedia.org/wiki/Virtual_learning_environment
3. *Downes S. E-Learning 2.0*. – <http://www.downes.ca/post/31741>. – Oct. 17, 2005.

4. *Greenberg L. LMS and LCMS: What's the Difference?* – Dec. 9, 2002. – http://scripts.cac.psu.edu/staff/g/m/gms/fa07/IST-440W/LMS%20and%20LCMS_%20What's%20the%20Difference_.pdf
5. *Katz Richard N. Signifying a lot: what is really happening with IT in Higher education?* CAUBO Annual Conf., Saskatoon, Saskatchewan, June 14, 2004.
6. *PclZip Overview* – <http://www.phpconcept.net/pclzip/>
7. *GCC, the GNU Compiler Collection*. – <http://gcc.gnu.org/>
8. *GNU Compiler Collection* – http://en.wikipedia.org/wiki/GNU_Compiler_Collection
9. *E107* – <http://uk.wikipedia.org/wiki/E107>

Поступила 04.09.2014

Тел. для справок: +38 044 465-6064, 095 576-6069 (Київ)

E-mail: okonyushenko@gmail.com, ivchd@ukma.kiev.ua

© А.В. Конюшенко, Д.А. Івченко, 2014

А.В. Конюшенко, Д.А. Івченко

Построение интерактивного электронного учебного пособия в системе управления контентом E107

Введение. Специфика современного электронного обучения – это стремительное внедрение и использование компьютерных технологий во всех его отраслях [1]. Но, несмотря на все положительные стороны этого процесса, появляются такие новые проблемы, как увеличение объемов информации и затрат на обработку и анализ данных, проблемы структуризации материала и т.д. Особенно это касается обучения, связанного с информационными технологиями. Поэтому возникает потребность в создании специализированного инструментария, который позволял бы эффективно оперировать учебными материалами, давать возможность организовывать их по собственному желанию, создавать учебные пособия с элементами интерактивности и вместе с этим иметь удобный, простой и интуитивно понятный интерфейс.

Существуют различные системы управления контентом, в частности системы создания и управления электронными курсами или портфолио. Но, учитывая специфику, связанную с программированием, они не соответствуют в полной мере необходимым требованиям: отсутствие ориентации на специфические форматы материалов по конкретной отрасли; отсутствие интерактивности в работе с материалами на различных языках программирования; сложная система администрирования и поддержки учебного процесса.

Опишем основные возможности и преимущества разработанной системы: полное управление созданными курсами, размещение в них каких-либо материалов, оперирование этими материалами; удобное представление презентаций и программного кода – важных составляющих обучения с объектно-ориентированным программированием (ООП); интерактивность работы с программными проектами – динамичное выполнение и тес-

тирование; возможность защитить материалы от несанкционированного доступа путем использования технологии *Flash* (защита авторских прав); независимость в демонстрации материалов от любой платформы; динамическую связь между разделами *LCMS* «Учебные курсы», автоматизированных практикумов «Практические работы» и электронных портфолио «Портфолио».

Концепция рабочей тетради преподавателя

Электронное обучение естественно применяется в дистанционном обучении, но может широко применяться не только дистанционно, но и при непосредственном контакте между преподавателем и студентами, что можно охарактеризовать как смешанное обучение. Во многом система организации учебных материалов по ООП ориентирована именно на смешанный тип обучения, т.е. как дистанционный инструмент для обучения, так и в аудитории при непосредственном контакте участников учебного процесса.

В сфере высшего образования наблюдается тенденция к созданию виртуальных учебных сред [2], которые иногда объединяются с административной информационной системой для создания управляемого обучения. Кроме того, некоторые университеты предлагают сетевую студенческую службу поддержки, такую как онлайн сообщения, электронные рекомендации, студенческие управления и пр.

Термин *электронное обучение 2.0* приближает процесс обучения к сетевым технологиям *Web 2.0*. Прежний подход к электронному обучению предусматривал опеределенные учебные пакеты в пределах интернет-технологий, используемых студентами. Роль студента заключалась в изучении прочитанного и подготовке заданий, оцениваемых преподавателем. Новый подход придает большее значение социальному обучению, в част-

ности использованию социального программного обеспечения [3]. Именно в этом направлении работает созданный электронный учебник, интегрируясь с электронными портфолио и системой поддержки автоматизированных практикумов, создавая социальную учебную систему, где обеспечивается интерактивность, социализация, взаимодействие любых субъектов обучения. Причем данная система предусматривает возможность дальнейшего развития, настройки новых сервисов, еще больше интеграции с новыми ресурсами.

Социальное обучение (в отличие от традиционного картезианского подхода) меняет подход к самому объекту – знанию, которое теперь стало социально конструируемым в обсуждениях контента и во взаимодействии при решении определенных проблем. Действительно, лучший способ что-то выучить, – начать этому учить других.

Для управления учебным процессом обычно используются LMS-системы. Системы управления обучением постепенно развиваются в системы управления учебным контентом (мультипользовательские среды, где можно создавать, хранить, применять новое использование (*reuse*), управлять электронными материалами электронного обучения, сосредоточенными в определенном центральном репозитории [4]. В настоящее время часто используют обобщенное понятие CLCIMS (*Computer Learning Content Information Management System*) для определения учебных сред.

Все упомянутые системы в основном все же направлены на организацию взаимодействия между преподавателем и студентом (благодаря тематическим форумам, формам контакта, организации сдачи домашних заданий и др.). Иногда также используются портфолио, основная задача которых – представление/хранение материалов определенного субъекта процесса обучения.

Одно из решений – это виртуальный класс. Он служит пользовательским ядром образовательной ИТ-среды и представляет собой комплексную распределенную систему, куда обычно входят инфраструктурные программные и технические компоненты, виртуально объединяющие рабочие места преподавателя и студентов в учебную группу, которая работает в сети (локальной или глобальной). Тем не менее виртуальный класс также не решает проблему представления учебных материалов с ООП, он ориентирован на интерактивное взаимодействие участников учебного процесса.

Рабочая тетрадь преподавателя – это та среда, где будут сосредоточены все учебные материалы, которые преподаватель пожелает разместить, причем это не просто репозиторий, а определенная динамическая структура, которая должна иметь надлежащую функциональность, поскольку эту тетрадь используют и преподаватели, и студенты как на лекциях и практических занятиях, так и при «удаленном обучении». Поэтому к тетради предъявляются требования: должна быть сплошной средой; обязательна возможность быстрой демонстрации выполнения программ без необходимости «собирать» проект на разных локальных машинах; возможность

работать как с демонстрационными материалами (демонстрационные программы), так и материалами – задачами для студентов; управление собственным рабочим пространством; поддержка виртуальных классов.

Исследования подтверждают, что студенты предпочитают курсы смешанного типа (гибридные). Гибридные курсы используются в 40 процентах университетов, рассмотренных в [5], и число предложений растет более чем на 10 процентов ежегодно.

Разработка системы управления учебными материалами в рабочей тетради преподавателя

При разработке структуры проекта основное внимание было сосредоточено на максимально простой для пользователя, и одновременно способной удовлетворить требования преподавателей инструмент. На рис. 1 изображена страница при просмотре материалов курса.



Рис. 1

Как видно из рисунка, при просмотре курса можно видеть материалы, а также просматривать код лабораторных работ, запускать эти работы на выполнение и просматривать результат выполнения, читать комментарии преподавателя. Также можно работать с обычными вложениями. Преподаватель в свою очередь может изменять код программ, и сразу запускать их на выполнение, таким образом имея возможность эффективно демонстрировать поставленные задачи на лекции/практическом занятии. Помимо того преподаватель может контролировать ход учебного процесса, активируя/деактивируя соответствующие материалы. Эта же страница при просмотре преподавателем (автором курса) выглядит уже иначе (рис. 2).

Очевидно, что для преподавателя реализованы упомянутые задачи, плюс набор стандартных операций – редактирование, удаление, реорганизация структуры дерева учебных материалов.

База данных применения содержит четыре таблицы:
e107_coursecreator_attachment,
e107_coursecreator_courses,
e107_coursecreator_program,
e107_coursecreator_uploadunit.

Например, структура таблиц *e107_coursecreator_courses* такова:

```
`ID` int(11) NOT NULL auto_increment,
```

```
'abbr' varchar(20) NOT NULL default '',
'name' varchar(255) NOT NULL default '',
'description' text NOT NULL,
'owner' int(11) NOT NULL default '0',
'sort' int(11) default NULL,
'active' enum('0','1') NOT NULL default '1'.
```

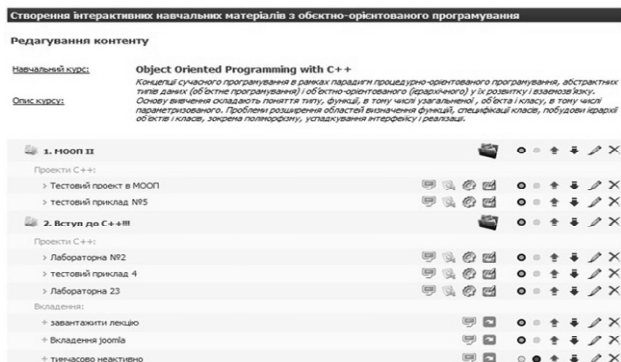


Рис. 2

Структура системи показана на рис. 3.

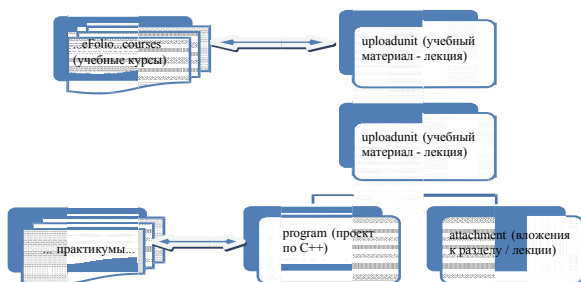


Рис. 3

В правой части изображена схема представления данных в *LCMS*, стрелками показаны динамические связи с системой электронного портфолио и студенческих автоматизированных практикумов. Такая схема одновременно делает процесс ознакомления с новым веб-инструментом максимально интуитивным, обеспечивает соответствующую функциональность и позволяет развивать эту систему беспрепятственно. Это также касается он-лайн компиляции не только проектов по *C++*, а любых других практикумов, написанных на языке, который поддерживается библиотекой *GCC*, но об этом – в следующих разделах.

Процесс добавления контента реализован следующим образом: преподаватель создает курс, к созданному курсу добавляет материал в виде веб-страницы или флеш-презентации, к этому материалу можно добавлять проекты с *C++* (возможно также с другими языками программирования) или вложения. Все результаты этих действий отображаются как на главной странице плагина, так и на странице редактирования у автора курса.

Веб-страницы, которые добавляет автор курса – это файлы, полученные путем конвертации документов (обычно *Microsoft Word*) в формат *HTML*. Авторы умышленно этот процесс возложили на авторов курса,

так как это предотвращает проблемы совместимости форматов при автоматической конвертации, а также делает систему создания интерактивных учебных материалов универсальной, т.е. открытой к созданию учебных материалов по любым наукам, позволяет добавление каких-либо документов, конвертируемых в *HTML* или *Flash*. Это позволяет расширить рамки применения данного проекта за пределы объектно-ориентированного программирования.

Учебные материалы загружаются на сервер в основном в виде архивов. Поэтому в данном проекте использована библиотека *pclzip* [6], что позволяет проводить многие операции над архивами формата *.zip*: создавать, разархивировать, изымать, сливать в один, дублировать и пр. Этот отрывок кода иллюстрирует одно из использований данной библиотеки в практической части работы:

```
require_once('unzip/pclzip.lib.php');
$archive = new PclZip($target);
if ($archive->extract(PCLZIP_OPT_PATH, $target_extract) == 0)
{
    die("Ошибка : ".$archive->errorInfo(true));
}
```

Реализация интерактивности обучения проведена путем применения динамического обновления и компилирования. Динамичность заключается в возможности просматривать выполнение проекта непосредственно после внесения изменений в программный код. Все действия происходят на сервере через веб-интерфейс браузера. Для проектов, требующих ввода данных в программу уже после процесса компиляции и линкования, предусмотрена возможность загрузить выполняемый файл на компьютер и уже на нем выполнять программу. При ошибках в программном коде автор может также видеть, какие сообщения об ошибках и предупреждения генерирует компилятор *GCC* на сервере.

Каждый преподаватель имеет полный контроль над своими материалами, и никак не может получить доступ к материалам других авторов. Эти настройки связаны с интегрированием в систему авторизации (и аутентификации) пользователей в системе управления контентом *E107*. На главной странице отображаются материалы всех авторов, однако можно выбрать конкретного автора и просматривать только его материалы. Администратор в свою очередь имеет доступ к материалам всех преподавателей и может контролировать наполнение. Но он не может редактировать материалы, может только активировать/деактивировать или удалять.

Настройка работы *GNU Compilers Collection* под *OC FreeBSD*

GNU Compiler Collection (GCC) – набор компиляторов для различных языков программирования, разработанный в рамках проекта *GNU* [7], который есть открытым программным обеспечением, распространяется на условиях лицензий *GNU GPL* и *GNU LGPL*, используется как стандартный компилятор для открытых *Unix*-подобных операционных систем, а также для *Apple Mac*

OS X. Кроме того, что *GCC* – официальный компилятор системы *GNU*, включая вариации системы, построенные на ядре *Linux* (*GNU/Linux*), он также служит главным компилятором для сборки ряда других операционных систем, среди которых: различные варианты *BSD*, *ReactOS*, *Mac OS X*, *NeXTSTEP* и *BeOS*.

GCC часто выбирается для разработки программного обеспечения, которое должно работать на различных аппаратных платформах. Различие между «родными» для каждой из аппаратных платформ компиляторами создает трудности при разработке кода, который корректно компилировался бы различными компиляторами. При использовании *GCC* для компиляции кода под различные платформы применяется один и тот же синтаксический анализатор. Поэтому если удалось собрать программу для одной из целевых платформ, то велика вероятность, что программа нормально соберется и для других платформ.

Внешний интерфейс *GCC* есть стандарт для компиляторов на платформе *Unix*. Пользователь вызывает управляющую программу *gcc*. Она интерпретирует аргументы командной строки, определяет и запускает для каждого входного файла свои компиляторы нужного языка, запускает, если необходимо, ассемблер и компоновщик [8].

Для генерирования команд выполнения компиляции на сервере была написана функция *CompileStringCPP*:

```
function CompileStringCPP($dir)
{
    $CompileString = "";
    $dhandle = opendir($dir); if ($dhandle)
    {
        while (false !== ($fname = readdir($dhandle)))
        {
            if (is_file("{ $dir } / { $fname }"))
            {
                $path_parts = pathinfo($fname);
                $ext = $path_parts['extension'];
                if (!strcasecmp($ext, 'cpp'))
                $CompileString .= "{ $dir } / { $fname } ";
            }
        }
        closedir($dhandle);
    }
    $CompileString = 'g++ '.$CompileString.' -o '.$dir.'/project.exe'; return $CompileString;
}
```

Команда *G++* вызывает компилятор для программ, написанных на *C++*. При дальнейшей разработке проекта есть возможность разработать универсальную среду, которая будет компилировать, линковать и генерировать файлы для таких языков, как *C*, *Objective-C*, *Java*, *Fortran* и *Ada*, и поэтому можно будет в полной мере реализовать работу с гораздо большим количеством курсов по объектно-ориентированному программированию.

Особенности реализации

В основу реализации положено использование платформы *e107* [9] – набор скриптов, написанных на *PHP*, взаимодействующих с *MySQL*, что в целом обеспечивает

отличную готовую порталную систему. С ее помощью можно создавать как огромные порталные системы, так и просто небольшие сайты. В нашем случае на этой системе базируется учебный портал *Joint European Projects* (<http://jep.ukma.kiev.ua>), и поскольку *e107* – модульная система, ориентированная на работу с функциональными расширениями, модулями, разработанный инструмент по оперированию учебными материалами был реализован в виде плагина.

Основные особенности *E107* – высокоэффективное файло-ориентированное кэширование, интегрированная система новостей и *RSS*, упрощенный процесс создания и использования шаблонов, валидный *XHTML*-код, мощная система поддержки форумов, удобный подход к созданию новых модулей, удобная система администрирования.

Совместная интеграция с проектом поддержки автоматизированных практикумов и системой электронных портфолио внедрялась для возможности преподавателю быстро переходить от просмотра своей рабочей тетради (система создания курсов и наполнения их контентом) к просмотру работ студентов, которые, например, выполняют лабораторную работу, и наоборот. Для того, чтобы обеспечить такую функциональность, для хранения данных в *LCMS* и практикумах о лабораторных работах, заданные преподавателем, используется общая таблица *e107_coursecreator_program*. Практические работы, сохраненные в этой таблице, отображаются как учебные материалы определенного раздела определенного учебного курса. Студенческие практикумы привязаны к определенным работам, созданным преподавателем. Таким образом преподаватель при просмотре курса может ориентироваться по ссылкам и видеть сколько студентов участвуют, что выполнили по конкретной лабораторной работе. Он может затем вернуться на свою рабочую тетрадь, так как находясь в рабочей тетради студента, известно, к какой лабораторной работе и к какому курсу «привязан» его практикум.

Существует также возможность подобной навигации между системой создания учебных курсов и между системой электронных портфолио. При просмотре курсов можно переходить по ссылке на личное электронное портфолио автора курса или просмотреть все курсы данного автора. В свою очередь через интерфейс *eFolio* можно видеть информацию о материалах конкретного преподавателя (автора курса).

Заключение. Созданная *LCMS* позволяет ускорить и улучшить процесс обучения, сделать его максимально наглядным, внести элемент интерактивности при работе с материалами языков программирования, избежать проблемы несовместимости платформ, централизовать разрозненный учебный материал. В контексте стремительного развития веб-технологий это позволяет приблизиться к качественно новому уровню организации обучения, причем как электронного, так и смешанного типа в области дисциплин, связанных с программированием.