

АНАЛІЗ ПРОГРАМНИХ ПРОДУКТІВ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ ДИСТАНЦІЙНОЇ ОСВІТИ

О.О. МОЧАЛОВ, О.О. ГАЙША

Наведено аналіз програмного забезпечення, яке може бути застосоване для процесу дистанційної освіти. Розглянуто засоби створення навчальних робіт, а також управління цим процесом. Проведено аналіз загроз процесу дистанційної освіти та вказано шляхи їх нейтралізації.

Дистанційне навчання (ДН) стає все більш популярним перспективним методом освіти. Сама суть ДН передбачає використання сучасних інформаційно-телекомунікаційних технологій. Весь учбовий процес має відбуватися на комп'ютері учня із використанням відповідного учбового та сервісного програмного забезпечення.

При виникненні необхідності у дистанційній освіті (ДО) навчальний заклад має вибрати або розробку власного програмного забезпечення, або використання існуючих програмних продуктів сторонніх розробників. Перший шлях, звичайно, є дуже важким, потребує роботи кваліфікованих програмістів, але може виявитися економічно не вигідним. Щоб вибрати правильний шлях розвитку, слід ознайомитися з програмними засобами, які на даний момент існують для галузі ДО.

У літературі проводиться аналіз переважно якогось одного програмного засобу, що має свою вузьку область застосування, і методик його використання (наприклад, [1], [2]). Значно рідше можна зустріти аналіз двох (чи кількох) приблизно однакових за функціональністю продуктів різних виробників [3, 4].

Безперечно, розглянувши великий масив статей про кожний окремий програмний продукт, можна отримати необхідні знання, але це потребує великих витрат часу. Значно зручнішим було б використання зконцентрованого матеріалу, що дозволяв би швидко дізнатися про призначення, переваги і недоліки розповсюджених програмних засобів ДО.

Мета роботи полягає в аналізі існуючих програмних продуктів для забезпечення процесу дистанційної освіти, управління ним, а також продуктів, які забезпечували б належний рівень його якості (зокрема засобів захисту).

Першою основною проблемою, з якою стикається будь-яка організація, що прагне проводити ДН, є нестача електронного навчально-методичного забезпечення. Справді, для повноцінного ДН його електронне забезпечення повинно мати усі складові традиційного навчального процесу: лекційні, практичні, лабораторні заняття, семінари, контрольні роботи різних типів (рубіжні, поточні, підсумкові тощо). Розглянемо програмні засоби для створення кожної із складових.

Лекції можуть проводитися у двох режимах, відповідно до прийнятого типу ДО. При синхронному режимі викладачеві та учню потрібне якість ко-

мунікаційне середовище (і відповідне програмне забезпечення), що дозволяло б швидко обмінюватися мовною та візуальною інформацією. При синхронній взаємодії учня і викладача — це робота в режимі телеконференції (найліпше відео чи аудіо, гірше — текстової). Відповідно необхідне програмне забезпечення для організації такого зв'язку (це може бути будь-яка програма-чат для відео, аудіо та текстових конференцій, наприклад, Microsoft NetMeeting). Текстові чати можна організовувати як серверними мовами типу PHP/Perl на основі одного із розповсюджених движків, так і за допомогою спеціальних програм типу P2P, наприклад, ICQ або NetMeeting.

В асинхронному режимі лекції в основному проводяться шляхом самостійного опрацювання електронного конспекту, який звичайно зберігається у форматі гіпертекстових сторінок. Для створення гіпертексту потрібне використання мов HTML (із вбудованим JavaScript) та, можливо, PHP, які є відкритими і безплатними. Серверне програмне забезпечення може бути легко побудоване як на платній платформі Microsoft Windows NT, так і на безплатних операційних системах типу Linux. При цьому електронний конспект без будь-яких переробок переноситься з однієї платформи на іншу. Розробляти конспект можна у будь-якому низькорівневому текстовому редакторі (типу Блокнота) чи редакторі WYSIWYG, де є можливість збереження результату в форматі HTML (останній варіант більш зручний для викладача, який звичайно є досвідченим користувачем, але не програмістом). Тут слід уважно віднестися до кінцевого вибору текстового редактора, впевнившись у тому, що код, який ним генерується, відповідає стандарту, запровадженому W3C.

Часто вживаним є варіант створення конспекту лекцій у текстовому процесорі високого рівня типу Microsoft Word із подальшою конвертацією засобами самого Word у формат HTML. Недолік такого підходу — велика кількість зайвого коду, який додає у створювані сторінки текстовий процесор. У той же час при збільшенні швидкості каналів мережі Інтернет даний спосіб переважатиме своєю зручністю та простотою.

Віртуальні лабораторні роботи є надто специфічними складовими навчального процесу і звичайно являють собою окремі завершені програмні продукти, можливо слабо зв'язані з іншим забезпеченням ДО (а саме управляючою системою, розглянутою нижче) кількома внутрішніми операторами чи файловим обміном. У такому випадку вони можуть розроблятися різними програмістами, що є логічним, зважаючи на величезну кількість цих робіт. Їх відокремленість пояснюється специфікою: кожна лабораторна робота, наприклад, з фізики, потребує особливих приладів, елементів управління, різного типу вхідних даних і т.ін. Уніфікація процесу створення всіх лабораторних робіт з усіх дисциплін для створення відповідного універсального редактора практично неможлива.

Слід зазначити, що існують програмні продукти, які призначені для моделювання певних фізичних приладів. Відповідно вони можуть бути застосовані для швидкого і зручного створення лабораторних робіт з деяких предметів. Так, наприклад, для електротехніки та електроніки можна використовувати зручний пакет Electronics WorkBench або P-Cad. Розповсюдженою системою моделювання з широкими можливостями є LabView, однак вона все одно не може бути застосована для створення усіх видів лаборатор-

них робіт з усіх предметів. Тому вибір способу розробки лабораторних робіт залежатиме від конкретного предмету. Слід провести пошук існуючих продуктів для моделювання відповідних об'єктів даної галузі і за відсутності такого розробляти лабораторні роботи у вигляді окремих програм.

Найлегше для розробки віртуальних лабораторних робіт користуватися платними середовищами швидкої візуальної розробки типу Delphi чи Visual C++, хоча можливе і написання програм мовою C для безплатного компілятора GCC, чи у редакторі FreePascal, які не мають графічного інтерфейсу, тому є більш складними у використанні. Ще однією часто вживаною можливістю є використання платної програми Macromedia Flash для створення флеш-роликів, які можуть взаємодіяти із користувачем, тому можуть являти собою повноцінні учбові програми. Зазначимо, що оскільки викладач з дисципліни не повинен бути кваліфікованим програмістом, то до розробки такого електронного забезпечення слід залучати програмістів. Виникає потреба у постійній кропіткій взаємодії між ними, що є важливим організаційним стримуючим фактором для розробки систем ДО.

Практичні роботи також можуть розроблятися у вигляді окремих програм, хоча це не є обов'язковим (те ж саме стосується і контрольних робіт). Справді, у деяких випадках, при розробці навчальних програм певного типу можна позбутися роботи програмістів, наприклад, для розробки тестів скористатися однією із спеціальних програм — конструкторів тестів («Конструктор тестов», «Auto Control», «AVELife TestGold Studio», «Екзаменатор», «SunRav TestOffice» та багато інших). Вони надають широкі можливості у створенні тестів різного типу. При цьому викладачеві не потрібно знати жодної мови програмування, а тільки вводити питання, варіанти відповідей та іншу необхідну інформацію (можна вводити рисунки, аудіо, відео супроводження і т.д.). Програма створює окремий файл-тест, який може програватися на машині студента. Такий спосіб створення робіт є зручним і, зважаючи на можливість роботи у двох режимах (демонстрація розв'язку задачі – навчання та запит відповідей – контроль), такі програмні продукти стають просто необхідними для розробки компонентів систем ДО.

Семінари повинні проводитися у синхронному режимі, тому для їх проведення можливе використання тих самих продуктів, що й для організації online-лекцій (програм типу чатів).

Якщо розробка великої кількості електронних навчальних курсів є загальноприйнятою проблемою, то управління процесом ДО іноді ставлять на другий план. Насправді ж якісна управляюча система є не менш важливою компонентою комплексу ДО, ніж самі навчальні програми. Така система має вести моніторинг освітньої діяльності студента і спрямовувати його на виконання потрібних вправ. При повністю синхронному навчанні такої проблеми принципово не існує, адже викладач сам буде вирішувати і пропонувати студентові, як і коли виконувати необхідні роботи. Вказана проблема у повній мірі проявляється для асинхронних систем навчання. Тому при такій формі ДО необхідно створювати управляючу систему, яка б фіксувала і враховувала поточний рівень виконання учбового плану і в залежності від нього рекомендувала студентові займатися тією чи іншою роботою. Така система в ідеальному випадку повинна мати елементи штучного інтелекту, хоча

в цілому можна обійтися статичним рекомендованим графіком виконання робіт (що дещо зменшує гнучкість і якість ДО).

Зауважимо, що існують готові інтегровані програмні пакети для здійснення ДО, такі, як, наприклад, MOODLE (або IBM Learning Space і багато інших, менш відомих, у тому числі і вітчизняних). Переважно вони являють собою повноцінний веб-сервер із набором готових типових веб-сторінок і скриптів для наповнення їх змістом уроків, завданнями, питаннями і т.д. Існує можливість спілкування через дискусії, форуми, конференції, чати. У цілому такі системи позиціонуються саме як засіб для зручного віртуального спілкування студентів із викладачем, який при середньому рівні комп'ютерної грамотності може легко адмініструвати систему без роботи програміста. Однак MOODLE не пристосований до створення складних навчальних робіт (типу лабораторних) і погано інтегрується із зовнішніми програмами, які такими є. Цей засіб створює зручний інтерфейс, що дозволяє полегшити взаємодію суб'єктів учбового процесу і добре підходить для дистанційного викладання гуманітарних дисциплін, де в основному оперують інформацією у текстовому вигляді. Для створення ж повноцінних курсів з технічних дисциплін необхідно розробляти додаткові програмні продукти, які повинні взаємодіяти з головною системою. Просте розміщення цих робіт у вигляді посилань з системи MOODLE неприпустиме, адже повинен існувати обернений зв'язок, що дозволяв би впевнитися у виконанні цих робіт.

Крім того, MOODLE переважно орієнтований на online-навчання і, хоча встановлення його на жорсткий диск студента для offline-освіти принципово можливе, таке навчання буде мати досить низьку якість, адже студентіві будуть доступними висхідні тексти усіх завдань і навчальних робіт. Відповідно можна сказати, що для проведення навчання у режимі offline у MOODLE (і аналогічних їй серверних систем) немає належної системи захисту.

Для забезпечення високої якості процесу ДО необхідно нейтралізувати загрози, які виникають у таких системах. Проведемо аналіз цих загроз.

По-перше, існує загроза несанкціонованого використання програмних продуктів власної розробки іншими освітніми організаціями (аналогічно до загальної проблеми комп'ютерного піратства). Тож у комплексі засобів ДО необхідно передбачити підсистему захисту від піратства, що найбільш актуально для систем offline-навчання, які особливо вразливі до цієї загрози (докладніше див. [5]).

По-друге, існує небезпека підміни особи, що навчається. Справді, оскільки навчальний процес відбувається віддалено, виконувати навчальні роботи може не сам студент, а його довірений агент. Така загроза є актуальною для усіх систем асинхронної та синхронної взаємодій, крім режиму відеоконференції. Тож слід застосувати якусь, імовірно біометричну поведінкову методику розпізнавання особи, що з достатнім рівнем достовірності забезпечувала б розпізнавання особи студента.

По-третє, існують технічні загрози підробки звітності навчального процесу. Докладно вони були розглянуті у роботі [6]. Цей вид загроз особливо актуальний для систем offline-навчання, хоча і в деяких інших випадках мо-

жливості підробки результатів виконання робіт присутні. Для упередження підробки звітності слід застосувати криптографічний протокол, що унеможливував би вказані дії.

ЛІТЕРАТУРА

1. *Grebenyuk K., Chernishenko S.V.* Internet modeler — the software for teaching simulation // *Методологічні засади дистанційного навчання: Матер. міжнар. наук.-техн. конф.* — Дніпропетровськ: ДНУ, 2005. — С. 115–121.
2. *Хижя А.Л.* Расширение подсистемы интерактивного тестирования в системе MOODLE // *Методологічні засади дистанційного навчання: Матер. міжнар. наук.-техн. конф.* — Дніпропетровськ: ДНУ, 2005. — С. 77–80.
3. *Политикин Б.М., Тендитная Н.В.* Методическое обеспечение пакетов прикладных программ // *Проблеми наскрізної комп'ютерної підготовки у вищій школі: Матер. Всеукр. наук.-метод. конф.* — Миколаїв: НУК, 2005. — С. 45–46.
4. *Соловых Е.К., Жесан Р.В., Аулин В.В.* Задачи, возникающие при организации дистанционного обучения в ВУЗе // *Методика дистанційного навчання фундаментальним та технічним дисциплінам у вищій школі: Матер. Всеукр. наук.-метод. конф.* — Миколаїв: НУК, 2005. — С. 23–29.
5. *Гальчевський Ю.Л., Гайша О.О.* «Логічні» та «фізичні» захисти програмного забезпечення від несанкціонованого копіювання // *Захист інформації.* — 2005. — № 2(23). — С. 34–40.
6. *Мочалов О.О., Гайша О.О.* Проектування системи захисту програмного комплексу дистанційної освіти // *Зб. наук. праць НУК.* — Миколаїв: НУК, 2006. — № 2 (407). — С. 149–156.

Надійшла 15.09.2006