УДК 37.032

Бублик В.В., Глибовець М.М., Болгар Л.Е.

РОЗРОБКА ТА ІНТЕГРАЦІЯ АВТОМАТИЗОВАНОГО ПРАКТИКУМУ З ПРОГРАМУВАННЯ В ПОРТАЛ ПРОЕКТУ JEP_26195_2005

Робота підбиває підсумки досвіду, отриманого внаслідок участі у низці загальноєвропейських проектів, оплачених програмою ЄС «TEMPUS», у царині поширення IT-освіти для ефективної організації групової роботи студентів над складними програмними проектами. Цей великий міжнародний консорціум, очолений Мінстерським університетом, залучає й інші установи країн-партнерів: Німеччини, Великої Британії, України. Акцентується увага на використані методи і засоби підтримки автоматизованих практикумів, розроблені в рамках виконання проекту «Redesigning Ukrainian IT curricula by introducing team-oriented modules in software engineering Tempus Tacis JEP_26195_2005».

Ключові слова: *автоматизований практикум з програмування, TEMPUS, портал.*

Работа подбивает итоги опыта, полученного в результате участия в ряду общеевропейских проектов, оплаченных программой ЕС «TEMPUS», в области распространения it-образования для эффективной организации групповой работы студентов над сложными программными проектами. Этот большой международный консорциум, возглавленый Минстерским университетом, привлекает и другие учреждения стран-партнеров: Германии, Большой Британии, Украины. Акцентируется внимание на использованные методы и средства поддержки автоматизированных практикумов, разработанные в рамках выполнения проекта «Redesigning Ukrainian IT curricula by introducing team-oriented modules in software engineering Tempus Tacis Jep_26195_2005».

Ключевые слова: *автоматизированный практикум с программирования, Tempus, портал.*

The experience gained in performing the sequence of Joint European Projects sponsored by EU TEMPUS Program in the field of IT for effective organization of students' teamwork on complicated programming projects. This big international consortium lead by Ministry University involves also other institutions of states-partners: Germany, Great Britain, Ukraine. It is accentuated on the used methods and support tools of automatic practical training sessions developed within executing of project «Redesigning Ukrainian IT curricula by introducing team-oriented modules in software engineering Tempus Tacis JEP_26195_2005».

Key words: *practical work is automated from programming, Tempus, portal.*

ВСТУП

Значне поширення WEB-технологій у світі сприяє бурхливому розвитку різноманітних клієнт-серверних архітектур на базі так званого «тонкого» клієнта(***Thin Client***). Такі пріоритети розвитку клієнт-серверних технологій визначаються відносною дешевизною інсталяції та обслуговування клієнтських програм, що є найбільш характерними особливостями технологій на базі тонкого клієнту. Досить успішно зарекомендувала себе дана технологія у випадках оперування серверною програмою великою кількістю так званих «позастанових» або рутинних (таких, поточний стан котрих незалежний від попередньо виконаних дій) клієнтських застосувань. Численні переваги, які надають WEB- застосування, найвизначніші серед яких: незалежність від платформи, економія часу та коштів, як правило затрачуваних на інсталяцію

та підтримку Thick Client застосувань та мала ресурсоємкість – вимагають від розробників сучасного ПЗ якщо не повної інтеграції, то принаймні часткової невідчужуваності новітніх програмних продуктів та сервісів від сучасних WEB-технологій [1-2].

Немало програмних продуктів, які вважалися ПЗ виключно індивідуального використання і вимагали значних системних ресурсів та кропіткої установки засобами ПК, поволі «мігрували» до мережі Інтернет, вирішуючи таким чином чимало стандартних проблем локалізованих застосувань, завдяки чому значно розширили коло власних користувачів. Серед них: блоги, перекладачі, словники, мапи, фотогалереї, мережеві файлові репозитарії. Переважна більшість IT-аналітиків висловлює думку, що з часом особистий робочий простір користувачів комп'ютерами цілковито переміститься у простір глобальних або локальних мереж. Окремим викликом для програмістів у напрямку перенесення локалізованих програмних продуктів до глобальної мережі є емуляція простору розробника ПЗ у мережі, а також on-line компіляція, тестування та відлагодження програм. Складність такої задачі полягає у коректній організації зв'язку між клієнтом та сервером за умови, що від компілятора, який міститься на віддаленому сервері очікується результативна обробка клієнтського програмного коду з урахуванням політик мережевої безпеки.

У 2007 році організація **GCC Work Group** запропонувала розробникам програмного забезпечення першу мережеву версію робочого простору для C++ програмістів **OnlineGCC** на базі стандартизованого набору компіляторів **GCC(GNU Compiler Collection)** [3]. Користувачам ресурсу надається можливість створювати та компілювати власні програмні проекти з будь-якої точки світу, при чому наявність встановленого на персональному комп'ютері Visual Studio чи будь-якого іншого середовища програмування не є обов'язковим. Очевидно, що подібний напрямок Інтернет-технологій є перспективним, у першу чергу, для галузі електронної освіти, особливо для підготовки IT-спеціалістів, оскільки методи автоматизації перевірки учбових практикумів з програмування на сьогодні є нечисленними і у значній мірі обмеженими.

У роботі розглянуто можливості підвищення якості вищої освіти, шляхом впровадження та розвитку засобів інформаційних технологій, які зазвичай об'єднують у функціональний модуль до платформи електронної освіти (e-learning) для підтримки електронного практикуму.

1. КОМПОНЕНТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО НАПИСАННЯ ПРОГРАМНОГО ПРОДУКТУ

1.2. Модуль як категорія функціональності та задача модуляризації програмних систем

Модуль – це виділена за мотивом функціональності частина первинного матеріалу програмного коду. Окрім вихідних текстів програм, до первинних матеріалів належать: тексти вихідних даних, тексти вказівок про атрибути виділених частин програмних матеріалів(автор, задача, що вирішується, дата написання програми), тексти процедур, що застосовуються до програмних матеріалів.

Модуляризацію прийнято розуміти, як розподіл програмного продукту на модулі – обов'язковий атрибут довільного програмного проекту, оскільки із зростом розмірів програми, якість виконання модуляризації відіграє важливу роль у кінцевому успіху проекту. Важливим мотивуючим чинником модуляризації є полегшення процесу подальшого розвитку та розширення функціональності програми. Модуляризація зачіпає усі етапи життєвого циклу створюваної програми(системи): від первинної специфікації майбутньої системи – до зручностей реінжинірінга і супроводу програмного проекту. Наскільки професійно проведена модуляризація майбутньої системи, побудована, а надалі відпрацьована її модель, настільки потім можуть вирости витрати на комплексне відлагодження, дослідницьку експлуатацію та впровадження системи, не кажучи вже про супровід відчужуваного програмного продукту.

Проектований в межах проекту програмний продукт є модулем який володіє чітко визначеною функціональністю і характеризується порівняно високим рівнем стійкості. Функціональної міцності досягнуто завдяки чіткої структуризації внутрішньої організації

платформи, на якій даний модуль був реалізований – e107, а також компонент SMARTY, який дозволяє відділити представлення від реалізації.

1.2. Структурна декомпозиція системи

Структурна декомпозиція «знизу-вгору» – це алгоритмічна декомпозиція (згідно до алгоритму) для боротьби із складністю проблеми, спрощення і прискорення проектування ПЗ згідно до принципів: «розділяй і пануй» – для розбиття складних проблем на множину менших, незалежних задач, легких для розуміння та вирішення та ієрархічного впорядкування для організації складових частин проблеми в ієрархічні структури з додаванням нових деталей на кожному рівні.

Відповідно до методу структурної декомпозиції, було побудовано наступний список властивостей цільового модуля за назвою «Store Space»:

1. Аутентифікація та супровід користувачів: реалізація базових ієрархій та ролей;
   a. Планування та реалізація функцій профіля «Лектор»
   b. Планування та реалізація функцій профіля «Викладач»
   c. Планування та реалізація функцій профіля «Студент»
2. Файловий репозитарій студентських програмних практикумів, підтримка деревовидної структури каталогів;
   a. Структуризація проектів
   b. Програмна реалізація деревовидної структури каталогів
3. Віддалена компіляція C++ проектів: Застосування GCC;
4. Оцінювання та робота з часом.
   a. Оцінювання студентських практикумів
   b. Хронологічний підхід до організації практикуму
      i. Організація контролю над строками здачі проектів
      ii. Автоматизація сортування профілів студентів за часом реєстрації.
5. Задача інтеграції систем «Store Space», «E-folio» та системи підтримки навчальних курсів у мережі.

Наступні розділи буде присвячено висвітленню кожного із пунктів списку характеристик цільової системи.

2. АУТЕНТИФІКАЦІЯ ТА СУПРОВІД КОРИСТУВАЧІВ: РЕАЛІЗАЦІЯ БАЗОВИХ ІЄРАРХІЙ ТА РОЛЕЙ

2.1. Планування та реалізація функцій профіля «Лектор»

Згідно до вимог до цільової системи, користувач із повноваженнями «Лектор» повинен володіти всіма доступними можливостями модуля керування автоматизованими практикумами ***«Store Space»***, серед них:

1. Додавання та видалення нових та вже зареєстрованих користувачів системи.
2. Перегляд комплексного дерева каталогів студентських робіт за всі роки навчання.
3. Управляння всіма студентськими програмними проектами, що були поміщені у базу даних ресурсу (додавання, видалення, перегляд, компіляція).
4. Надання зареєстрованих на ресурсі користувачам повноважень «Викладач».
5. Призначення перездачі проектів.
6. Оцінювання знань всіх підписаних на курс студентів.

Система реалізована так, що підтримує лише одного користувача із повноваженнями «Лектор», він визначається супер-Адміністратором порталу JEP.

2.2. Планування та реалізація функцій профіля «Викладач»

Користувачі із повноваженнями «Викладач» мають подібні права у системі до «Лекторів», за відмінністю, що кількість викладачів у системі необмежена. «Лектор» призначає «Викладачів» із окремого списку зареєстрованих на порталі користувачів. «Викладачам»

надаються наступні повноваження у системі підтримки автоматизованих практикумів з програмування «Store Space»:

Додавання та видалення нових та вже зареєстрованих користувачів системи, за умови, що користувач вказав, що належить до групи даного «Викладача».

1. Перегляд комплексного дерева каталогів студентських робіт за всі роки навчання власної групи:
   а) Управляння всіма студентськими програмними проектами, що були поміщені у базу даних ресурсу студентами окремого «Викладача.
2. Призначення перездачі проектів.
3. Оцінювання знань студентів, що належать до групи даного викладача.

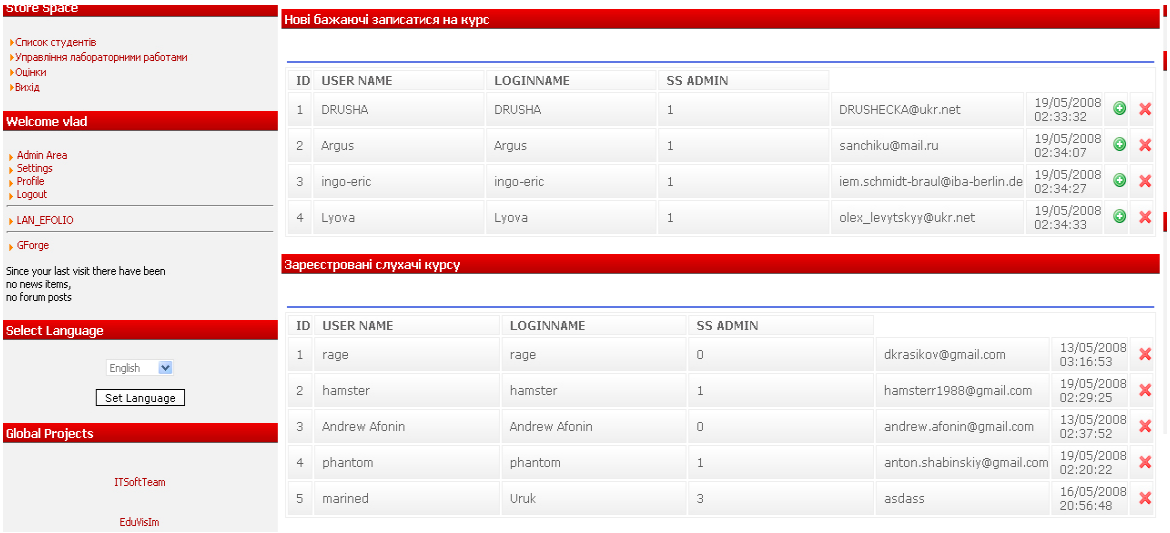

Рис. 1

2.1. Планування та реалізація функцій профіля «Студент»

«Студентам» у системі підтримки автоматизованих практикумів надається найменша кількість повноважень. Серед них:

2. Підпис на курс та доступ до матеріалів через суміжний модуль управляння курсами;
3. Перегляд та компіляція лише власних програмних проектів;
4. Додавання та видалення вмісту виділених даному «Студентові» робочих каталогів;
5. Можливість перегляду балів, отриманих за проекти.

Підписатися на курс може кожен зареєстрований в системі порталу JEP користувач, далі його запит на реєстрацію переглядається користувачем із повноваженням «Лектор», або «Викладач» і у разі його схвальної відповіді, додається до списку записаних на курс слухачів.

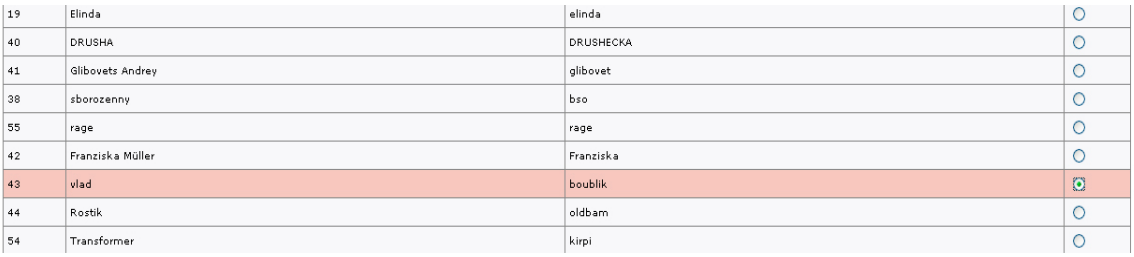

Рис. 2 (а)

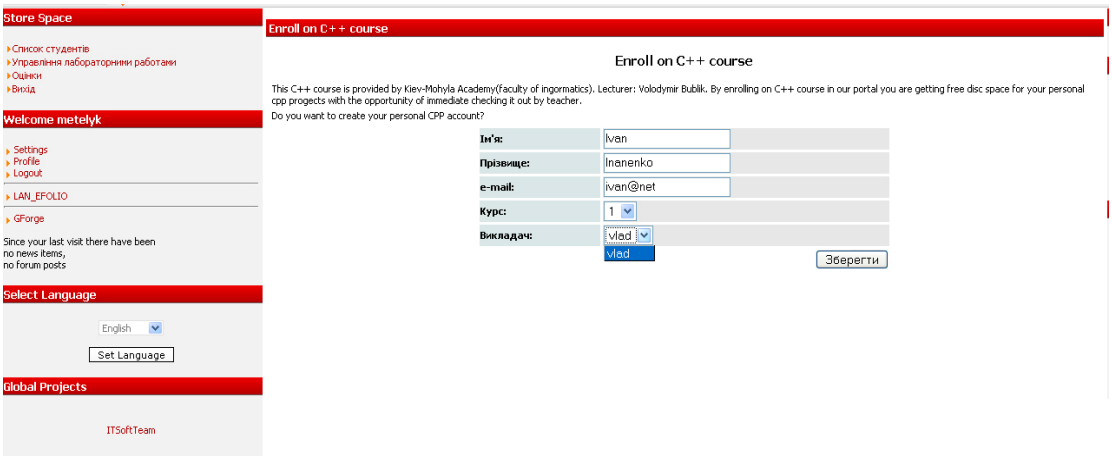

Рис. 2 (б)

3. ФАЙЛОВИЙ РЕПОЗИТАРІЙ СТУДЕНТСЬКИХ ПРОГРАМНИХ ПРАКТИКУМІВ, ПІДТРИМКА ДЕРЕВОВИДНОЇ СТРУКТУРИ КАТАЛОГІВ

Згрупованість та підтримка визначеної структури відіграє вирішальну роль у коректній роботі модуля управляння навчальними практикумами та ефективній контролюючій місії викладача. Каталоги із студентськими проектами структуровані наступним чином:

Рік навчання –> викладач –> номер групи –> ім'я студента (латинськими літерами) –> номер лабораторної роботи –> файли проекту (доступні формати: .h, .cpp)

Кожен студент матиме ту кількість каталогів, яка визначена викладачем у полі «Лабораторні роботи».

Деревовидна структура каталогів найкраще відображає структуру навчальних проектів і попереджує утворення безладу у файловій системі.

Реалізована за допомогою мови Веб-застосувань JavaScript, деревовидна структура каталогів дозволяє легко маніпулювати теками, приховуючи та демонструючи їх в залежності від повноважень поточного користувача системою.

В залежності від повноважень користувача, кожен файл може бути переглянутий у тестовому редакторі, відкомпільований та видалений.

4. ВІДДАЛЕНА КОМПІЛЯЦІЯ C++ ПРОЕКТІВ: ЗАСТОСУВАННЯ GCC

GCC часто вибирається для розробки програмного забезпечення, що повинно працювати на різних апаратних платформах. Розходження між «рідними» для кожної з апаратних платформ компіляторами приводять до труднощів при розробці коду, який би коректно компілювався різними компіляторами. При використанні GCC для компіляції коду під різні платформи використовується один і той же синтаксичний аналізатор. Тому якщо вдалося зібрати програму для однієї із цільових платформ, те велика ймовірність, що програма нормально збереться й для інших платформ.

Зовнішній інтерфейс GCC є стандартом для компіляторів на платформі Unix. Користувач викликає керуючу програму, що називається *gcc*. Вона інтерпретує аргументи командного рядка, визначає й запускає для кожного вхідного файлу свої компілятори потрібної мови, запускає, якщо необхідно, асемблер і компонувальник.

У практичній частині даної роботи для генерування команд для виконання компіляції на сервері була написана функція ***CompilatorCPP***:

```
function CompilatorCPP($dir)
{
 $ CompilatorCPP = ‘‘;
 $dhandle = opendir($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'))
                                        $ CompilatorCPP.= "{$dir}/{$fname} ";
                    }
 }
 closedir($dhandle);
}
    $CompilatorCPP = 'g++ '.$ CompilatorCPP.' -o '.$dir.'/project.exe';
    return $CompilatorCPP;
}
```

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

5. ОЦІНЮВАННЯ ТА РОБОТА З ЧАСОМ

Студентські практичні роботи з програмування, подібно до робіт із інших дисциплін, можуть бути переглянуті та оцінені користувачем із повноваженнями «Лектор», або «Викладач». За основу взята найпростіша система оцінювання якості виконання практичної роботи з програмування – компіляція проекту із подальшим його переглядом та виведенням оцінки, виходячи із якості виконаної роботи. Система оцінювання представлена у вигляді таблиці із стовпцями, що відповідають за номер лабораторної роботи і рядками, в які внесені прізвища слухачів курсу. Передбачена можливість викладачеві власноруч заповнювати відповідні позиції таблиці балами.

Окрім якості написання коду та відповідності результатів компіляції програмного проекту визначеному зразку, система управляння навчальними практикумами надає можливість узгоджувати час здачі окремих проектів із передбаченими для виконання кожного з них строками. Проекти, що були здані із запізненням, маркуються відповідно у списку зданих лабораторних робіт.

Невчасно здані проекти можуть бути відхилені викладачем. Також існує можливість реабілітації проектів минулих років на вимогу студента-автора з метою демонстрації, або аналізу результатів роботи програм.

Застарілі та неактуальні списки студентів – проблема, яку доводиться вирішувати більшості сучасних LMS в межах e-Learning. Неможливість структурування та відсіювання неактуальних даних стосовно складу студентських груп згідно до періода навчання нерідко призводить до захаращування та втрати чіткої структури каталогів. З метою запобігти цьому було розроблено програмний фільтр, що сортує студентів за вказаним ними роком навчання і виводить лише ті каталоги, які належать студентам, що прослуховують курс C++ на поточний момент часу, решта каталогів переводиться системою у стан тимчасової неактивності.

6. ЗАДАЧА ІНТЕГРАЦІЇ ФУНКЦІОНАЛЬНИХ МОДУЛІВ «STORE SPACE», «EFOLIO» ТА СИСТЕМИ ПІДТРИМКИ НАВЧАЛЬНИХ КУРСІВ У МЕРЕЖІ

Окрім модуля підтримки електронних практикумів з програмування, який було розроблено як результат теоретичних досліджень в межах теми даної кваліфікаційної роботи, у напрямку розширення функціональності міжнародного освітнього порталу JEP також розроблялися суміжні модулі підтримки електронних курсів та студентські електронні портфоліо. Дані модулі контекстуально пов'язані і доповнюють функціональність одне одного. Так, модулі підтримки електронних практикумів з програмування та електронних курсів поділяють доступ до спільної таблиці lab_tabl, що містить список лабораторних робіт, доступних для виконання у межах даного курсу. Такий вид зчеплення модулів називають ***зчепленням по спільній області пам'яті***. Модуль, що реалізує студентські електронні портфоліо, пов'язаний із «Store Space» найбільш вдалим з точки зору компонентно-орієнтованого програмування видом зчеплення – ***параметричним зчепленням***, оскільки модулі викликають методи одне одного і активно обмінюються параметрами.

Незважаючи на те, що вихідна система має окремі вади та неточності, у тому числі неможливість генерації тестових випадків для програмних проектів, та відсутності профіля «Зовнішнього рецензента» програмних робіт, вихідна система являє собою комплексний функціональний продукт, який, як і багато інших, покликаний оптимізувати систему електронної освіти та суттєво зменшити незручності, що виникають в результаті несумісності компіляції одних і тих самих мовних структур програмного коду на різних платформах.

7. ІНСТРУМЕНТАРІЙ, ВИКОРИСТАНИЙ У РОЗРОБЦІ

У процесі розробки програмного модуля «Store Space» були застосовані наступні технічні засоби:

Скорочена назва технічного засобу	Розгорнута назва технічного засобу	Короткий опис
Smarty	–	Система шаблонів для Веб, написана мовою PHP. Smarty позиціонується як засіб, що дозволяє відділяти програмне представлення від реалізації.
GCC	GNU Compiler Collection	Набір компіляторів для різних мов програмування, розроблений у рамках проекту GNU
PHP		Мова програмування, створена для генерації HTML-сторінок на веб-сервері та роботи з базами даних. На сьогодні підтримується переважною більшістю хостинг-провайдерів.
MySQL	–	відкрита система керування базами даних (СУБД).
CSS	Cascading Style Sheets	технологія опису зовнішнього вигляду документа, написаного мовою розмітки. Переважно використається як засіб оформлення веб-сторінок у форматі HTML й XHTML
JavaScript	–	скриптова мова, що найчастіше використовується при створенні сценаріїв поведінки браузера, які вбудовуються у веб-сторінки. Є однією з реалізацій мови ECMAScript.

HTML	Hypertext Markup Language	Стандартна мова розмітки документів у мережі Інтернет.
XML	**Extensible Markup Language**	Загальна специфікація для створення користувацьких розміток.

ВИСНОВКИ

Поступово виникають новітні технічні засоби та програмні рішення [4, 5], які дозволяють не лише ефективно проводити та оцінювати студентські практикуми з програмування без особистісної взаємодії «викладач – студент», але й, базуючись на архітектурі тонкого клієнта, не вимагають для успішної компіляції навчальних проектів кропіткої установки коштовного та ресурсоємкого програмного забезпечення на локальному комп'ютері користувача на зразок середовищ програмування Visual C++, C++ Builder. Системи на зразок On-line GCC вже сьогодні дозволяють створювати власні учбові та науково-дослідницькі проекти у мережі Інтернет та успішно компілювати їх на віддаленому сервері з будь-якої точки світу, маючи лише пристрій доступу до всесвітньої мережі (ПК/мобільний пристрій) та веб-браузер. Незавжаючи на технологічну та функціональну недосконалість подібних систем насьогодні, варто відмітити, що перший крок у напрямку перенесення локального робочого простору програміста у глобальну мережу було зроблено, і враховуючи бурхливий розвиток Веб-технологій та поступове поширення ПЗ на базі архітектури тонкого клієнта, серед розробників програмного забезпечення інтерес до он-лайн середовищ з часом лише зростає.

Результатом досліджень деяких із технологічних особливостей набору компіляторів GCC та методів електронної освіти, як webfolio та окремих функціональних компонентів платформи електронної освіти Moodle, стала система підтримки автоматизованих навчальних практикумів з ООП. Процес проектування та розробка цільової системи входив у рамки компонентно-орієнтованого програмування, оскільки об'єкт розробки являв собою спеціалізований функціональний модуль, приєднуваний до порталу міжнародного освітнього проекту JEP(Joint European Projects) [6] і певним чином інтегрований із суміжними модулями efolio та системою підтримки навчальних курсів. До списку властивостей спроектованого модуля підтримки автоматизованих практикумів входять: збереження та компіляція програмних проектів на сервері, хронологічна систематизація практикумів та учасників навчального процесу.

Однією із найважливіших цілей створення даної системи можна назвати задачу інтеграції систем «Store Space», «Efolio» та системи підтримки навчальних курсів у мережі.

Разом три зазначені модулі утворюють комплексну повнофункціональну систему підтримки електронного навчання, яку можна охарактеризувати наступним чином:

Конвент (модуль підтримки навчальних курсів та лекційних матеріалів) + зворотній зв'язок (модуль підтримки автоматизованих практикумів) + історія навчання (модуль Efolio)

Незважаючи на те, що описана система має окремі недоліки, пов'язані із недосконалістю систем електронного захисту від несанкціонованого доступу та не найкращими методами зчеплення модулів, проектування подібних програмних проектів є важливим кроком на шляху розвитку електронної освіти вцілому.

Існує велика імовірність, що з часом проектування он-лайн робочих середовищ вийде на рівень з соціальними мережами і стане одним із найбільш перспективних напрямків розвитку Інтернет-технологій.

ЛІТЕРАТУРА

1. V Boublik, W. Hesser, (2005). *E-learning: Challenges and perspectives*. (Періодичне видання) Наукові записки НаУКМА, том 36, 2005 (1).[Україна].
2. Valiathan Purnima, (2005). *Blended Learning Models* – [www.learningcircuits.org/2002/aug2002/vilathan.html]
3. http://gcc.gnu.org/ – офіційний сайт набору компіляторів GNU

4. Karrer, T., (2006) *What is eLearning 2.0?* [www.elearningtech.blogspot.com/2006/02/what-is-elearning-20.html]
5. http://www.online-gcc.com/sections/description_sections.php – офіційний сайт інтернет-проекту Online-GCC
6. http://jep.ukma.kiev.ua/ – офіційний сайт міжнародного освітнього порталу JEP(Joint European Projects)

Рецензенти: д.т.н., проф. КондратенкоЮ.П.
д.т.н., проф. Фісун М.Т.

© Бублик В.В., Глибовець М.М.,
Болгар Л.Е., 2009

Стаття надійшла до редколегії 10.03.09