

ДО ПИТАННЯ ЕЛЕКТРОННОГО НАВЧАННЯ ПРОГРАМУВАННЮ

Розглянуто засоби і методи підвищення рівня інтерактивності в колаборативних навчальних середовищах програмування на прикладі створення програмних доповнень до системи MOODLE, спеціалізованого веб-порталу та моделі десяткового навчального процесора з вхідною мовою високого рівня.

Ключові слова: об'єктно-орієнтоване програмування, електронне навчання, веб-портал, система MOODLE.

Говорячи про застосування засобів і методів електронної освіти у вищій школі, не можна обійти увагою предмети, сама природа яких вимагає комп'ютерних технологій. У першу чергу до таких відноситься програмування. Звертаючись до витоків, варто згадати новаторський підхід А. Г. Кушніренка [6], запроваджений на механіко-математичному факультеті Московського державного університету ім. М. В. Ломоносова ще у 1980 р., та створені під його керівництвом навчальні системи програмування, зокрема систему Е-практикум («Е» — на честь А. П. Єршова — автора основоположних ідей навчання інформатиці¹), розвинуту пізніше в систему КуМир (<http://www.niisi.ru/kumir/dev.htm>). Суть підходу полягала у створенні навчальної системи програмування з використанням концепції абстрактного програмного виконавця, вхідна мова якого реалізовувалася спеціалізованим інтерпретатором. Система забезпечувала покрокове виконання команд програмованих виконавців (по суті, об'єктів у їх сучасному розумінні) в режимі діалогу, використовуючи засоби графічного інтерфейсу для відтворення процесу функціонування виконавця. Відомі успішні застосування системи КуМир як на шкільному (виконавці «Робот», «Кресляр» тощо), так і на рівні ВНЗ (виконавці для побудови опуклої оболонки послідовностей точок, компіляції арифметичних формул, побудови зображення поліедра тощо). Завдання для самостійної роботи полягало у програмуванні автономних виконавців або їх компонентів і їх вбудовуванню у попередньо підготовлене програмне ото-

чення, розміщене у локальній мережі, з наступними налагодженням та тестуванням. За своєю суттю тип абстрактного виконавця можна вважати прототипом поняття класу в об'єктному програмуванні.

Схожий підхід застосовано у практиці викладання програмування на механіко-математичному факультеті та факультеті кібернетики Київського національного університету імені Тараса Шевченка у 1980–1990–2000-х роках [7] та децю пізніше на факультеті інформатики Національного університету «Києво-Могилянська академія» з використанням кількох різних платформ, середовищ і мов програмування. Згадані методи навчання по праву можна вважати першими спробами застосування методів електронної освіти в програмуванні, головне досягнення яких полягало у виході навчального процесу за межі аудиторії чи навіть комп'ютерного класу, інтерактивному характері навчання при віддаленому доступі до комп'ютерних ресурсів. Фактично відбувався відчутний зсув парадигми у співпраці викладач-студент.

Але принципово новий етап розвитку електронного навчання починається, як відомо, з появою систем управління навчальним процесом (learning management system — LMS) [2] та наступним створенням колаборативних навчальних середовищ [3]. Розгорнута на факультеті інформатики Національного університету «Києво-Могилянська академія» система електронного навчання MOODLE (<http://www.moodle.com/>), доступна за адресою <http://distedu.ukma.kiev.ua>, дозволила розв'язати проблему як віддаленого доступу до навчальних ресурсів, так і організації навчального процесу (реєстрацію, авторизацію користувачів, запис на курси, налагодження зв'язків між студентами і викладачами, перевірку і рецензування домашніх завдань тощо).

¹ За Єршовим, «продуктивність інформаційних моделей зовнішнього світу надає новий сенс біблійному вислову “На початку було слово”. [Адже]... ми живемо у світі програм, і самі постійно програмуємо, [навіть] не усвідомлюючи цього» [5].

На факультеті інформатики автором створено та розміщено на сайті електронної освіти серію курсів з об'єктно-орієнтованого програмування [1]. Особливе місце в курсах займають програмні проекти, призначені для самостійного виконання. Завдання на проекти розміщуються в місці електронного курсу, відповідному до графіка навчального процесу. Визначається термін, відведений на виконання проекту. Проекти, виконані пізніше зазначеного терміну, не допускаються або приймаються на особливих умовах із зазначенням тривалості запізнення. Особлива увага приділяється якості авторського тестування проекту. Проект перевіряється викладачем. Оцінка і зауваження викладача знову ж розміщуються на порталі курсу, так само як оцінки за контрольні роботи, іспити і підсумкові оцінки. Кожен студент має доступ до власного робочого зошита, в якому зосереджено його доробки і оцінки, що забезпечує повну прозорість проходження курсу.

Беручи до уваги широке розповсюдження системи MOODLE у світі та загалом високу оцінку її можливостей, варто все-таки зазначити, що при майже півтора мільйонній спільноті зареєстрованих учасників на більш ніж 80 тисячах розгорнутих навчальних сайтах залишається чимало нерозв'язаних проблем і місць, відкритих для розвитку системи. Характер MOODLE як системи з відкритим кодом дозволяє долучатися до її розвитку широким колом розробників, а можливість створення власних доповнень (плагінів) дозволяє порівняно просто налагоджувати систему на конкретні застосування залежно від специфіки предметної області навчання, зокрема з метою підвищення ефективності взаємодії учасників навчального процесу.

Аналіз усіх проблемних місць системи MOODLE виходить за рамки цієї роботи, вимагаючи окремого детального дослідження. Зупинимось лише на деяких рішеннях, які, на думку автора, дозволять наблизити систему до потреб навчання програмуванню, хоча деякі з них важливі для інших дисциплін також. Так, наприклад, студентами факультету інформатики було створено і впроваджено модуль взаємного рецензування робіт студентами. Кожен студент, виконавши певне завдання (у нашому випадку програмний проект), розміщує його в системі. Відповідно, викладач перевірить це завдання, оцінить і прорецензує його. Оцінка і рецензія, як і сама робота, доступні для ознайомлення лише студентів-її авторів. Але для майбутнього розробника програмного забезпечення не менш важливо, ніж ознайомитися з рецензією

на власну роботу, вміти прорецензувати роботу колеги, що складає встановлену практику в серйозних софтверних командах. З метою набуття навиків рецензування програмних проектів модулів взаємного рецензування дозволяє викладачеві призначити кожному завданню власного рецензента з числа студентів академічної групи, після чого рецензент одержує доступ до проекту, призначеного йому для рецензування. Виконавши рецензію, рецензент розмістить її на порталі, після чого вона стане доступною викладачеві (для перевірки і оцінювання) та авторові проекту (для ознайомлення). Таким чином викладач оцінює роботу студентів як у ролі авторів проектів, так і рецензентів. Обидві оцінки автоматично накопичуються в підсумковому балі за семестр.

На черзі зараз знаходиться розробка доповнень, покликаних підвищити рівень колаборативності електронного навчання програмуванню, зокрема йдеться про поєднання засобів мультимедіа з середовищами та системами програмування, з метою забезпечення вищого рівня інтерактивності навчального процесу. На даний момент існує кілька успішних прикладів практики колаборативного навчання програмуванню, порталів запитань-відповідей, авторських сайтів підтримки навчання програмуванню, досвід яких значно вплинув на підхід автора. Нагальним видається завдання доповнити навчальні матеріали курсу засобами інтерактивної взаємодії зі студентом. Зокрема важливо перетворити фрагменти програм, наведені у курсі, на складові частини програмних проектів, здатних бути активізованими в процесі перегляду матеріалу студентом. Он-лайн підтримка з боку системи програмування дозволить виконувати проекти, зокрема покроково, забезпечуючи спостереження за перебігом виконання.

На даному етапі ведуться експерименти зі створення окремих компонентів, призначених для вбудовування до колаборативного навчального середовища на базі системи MOODLE. До них відносяться створення порталу об'єктно-орієнтованого програмування, моделювання навчального процесора, обладнаного засобами інтерпретації мови високого рівня для виконання навчальних програм у термінах вхідного коду та інші. Ведуться дослідження архітектур ієрархій візуалізації для розробки графічних інтерфейсів в сукупності із виробленням і застосуванням стратегій об'єктно-орієнтованого проектування.

На сьогодні зусиллями студентів було реалізовано прототип порталу об'єктно-орієнтованого про-

грамування, доступний за адресою <http://oop.in.ua>, покликано розширити можливості наявного середовища електронної освіти та увійти складовою частиною навчального процесу як доповнення до системи MOODLE. Як свідчать результати чисельних опитувань студентів [2], крім цілком зрозумілих потреб у доступі до навчальних матеріалів, перегляді одержаних оцінок і спілкування з учасниками навчального процесу, студенти особливо зацікавлені у можливості самостійної перевірки рівня власних знань. Тому ядро порталу складає система тестування, здатна автоматично перевірити розв'язки задач користувачів. Наразі портал знаходиться на стадії автономних випробувань. З метою заохочення учасників порталу передбачено систему рейтингу користувачів, залежну від активності та досягнень, набутих на порталі.

Як і належить системі електронного навчання, система MOODLE працює за календарним принципом. Студенти записуються на той чи інший курс на певний період. Закінчення періоду навчання призводить до втрати доступу як до курсу, так і до особистого робочого зошита, а також до вилучення із спільноти учасників курсу. Портал дозволяє певною мірою подолати ці обмеження, базуючись не на календарному принципі, а на інтересах учасників. Тому на порталі за вибором його модераторів розміщуються матеріали, вартісні поза часовими обмеженнями. До них відносяться зразки розв'язування екзаменаційних завдань і контрольних робіт попередніх років, кращі презентації з наукових семінарів, кращі курсові та кваліфікаційні роботи, інші програмні проекти, корисні як додаткові навчальні матеріали. Передбачена можливість оцінювання і обговорення розміщених матеріалів, оскільки вміння читати і аналізувати готовий код безперечно є важливою рисою кваліфікації розробника програмного забезпечення.

Ідея моделювання навчального процесора має своїм початком роботу автора [4]. За час використання моделі десятичного процесора в навчальному процесі проявилися значні зрушення

у викладенні перебігу підготовки і виконання програм на реальних комп'ютерах, абстрагуючись від необхідності переходу до двійкової, вісімкової або шістнадцяткової систем числення. Також стало зрозумілим, що рішення про вхідну мову програмування (свого часу це була мова ФОРТРАН) вимагає перегляду. Важливу роль при створенні нового покоління моделі десятичного процесора відіграли також досягнення у візуальному програмуванні, графічних інтерфейсах. Вибір вхідної мови програмування – нею стала мова С – значною мірою визначився, виходячи з тієї ролі, яку ця мова відіграє в структурі сучасних де-факто стандартів мов програмування. Візуальна система програмування дозволяє прослідкувати весь хід підготовки і виконання програми у термінах вхідного коду, акцентуючи увагу на способах реалізації кожної з конструкцій вхідної мови програмування.

Насамкінець автор користується нагодою подякувати студентам – нинішнім і колишнім – факультету інформатики Національного університету «Києво-Могилянська академія» за інтерес до курсів програмування та активну участь у розбудові програмного забезпечення електронної освіти. Особливо вагомими виявилися добробки, до яких долучилися Антон Пець, Анатолій Полухович, Маргарита Сітнер, Андрій Чайка, Андрій Давиденко, котрі по праву можуть вважатися співавторами цієї роботи. На ранніх етапах створення атмосфери електронного навчання програмуванню особливо відзначилися Андрій Апостол, Юлія Божко, Ліанна Болгар, Дарина Броникова, Роман Буняк, Назар Грабовський, Дмитро Івченко, Юрій Рощенко, Ростислав Сліпецький, Антон Шабінський, Володимир Яцевський та багато інших, чий завзятість та ентузіазм суттєво вплинули на становлення електронного навчання на факультеті та сприяли розповсюдженню його ідей як серед академічної спільноти, так і в ширших масштабах.

Детальні звіти про розробку та результати впровадження наведених доповнень передбачається зробити в наступних публікаціях.

Список літератури

1. Бублик В. В. Розвиток колаборативних навчальних середовищ / В. В. Бублик, Н. Ю. Дроздович // Наукові записки НаУКМА. — 2012. — Т. 138. Комп'ютерні науки. — С. 76–80.
2. Бублик В. В. E-Learning: Challenges and Perspectives / В. В. Бублик, В. Гессер // Наукові записки НаУКМА. — 2005. — Т. 36. Комп'ютерні науки. — С. 58–65.
3. Глибовець М. М. Застосування Semantic WEB до створення колаборативного освітнього простору / М. М. Глибовець // Збірник праць ІТ'ягої Міжнародної конференції «Нові інформаційні технології в освіті для всіх: неперервне навчання», 23–25 листопада 2010 р. — К. : Академперіодика, 2010. — С. 179–192.
4. Диалоговий інтерпретатор десятичної одноадресної машини / [В. В. Бублик, М. В. Брук, И. Б. Романенко] Депонировано в ГФАП СССР инв. № 50890000590, 1989. — 36 с.
5. Ершов А. П. Программирование – вторая грамотность [Электронный ресурс] / А. П. Ершов // Архив академика А. П. Ершова – Новосибирск : Институт систем информатики им. А.П. Ершова СО РАН – Режим доступа: http://ershov.iis.nsk.su/russian/second_literacy/article.html. – Назва з екрана.

6. Кушниренко А. Г. Программирование для математиков / А. Г. Кушниренко, Г. В. Лебедев. – Москва : Наука, 1988. – 384 с.
7. Методические рекомендации к выполнению лабораторных работ по разработке программ в режиме диалога / [В. В. Бублик, Б. П. Довгий, А. М. Кириченко та ін.]. – Киев : Изд-во КГУ, 1987. – 96 с.

V. Boublik

TOWARDS E-LEARNING FOR COMPUTER PROGRAMMING

Tools and methods which improves the level of interactivity in collaborative educational environments by creating a software plug-ins to the system MOODLE, development of a specialized web portal and simulation of a decimal processor with high-level source language have been discussed.

Keywords: object-oriented programming, e-learning, web portal, system MOODLE.

Матеріал надійшов 25.06.2013

УДК 681.3

Олецький О. В.

ПРО ЗАСТОСУВАННЯ МАРКОВСЬКИХ ПРОЦЕСІВ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ АВТОМАТИЗОВАНОГО ДОБОРУ НАВЧАЛЬНИХ МАТЕРІАЛІВ У СИСТЕМАХ BLENDED LEARNING

Обговорюється проблема знаходження оптимальної структури гіпертекстових посилань на веб-порталі (тобто оптимізацією навігаційних графів) на основі аналізу та моделювання поведінки відвідувачів. Пропонується підхід до оптимізації навігаційних графів на основі методик навчання з підкріпленням та марковських процесів прийняття рішень. Розглядаються можливості, пов'язані з застосуванням цього підходу в системах e-learning.

Ключові слова: навігаційний граф, навчання з підкріпленням, марковські процеси прийняття рішень, система e-learning.

Вступ

У роботі [4] обговорюється проблема оптимізації структури гіпертекстових посилань у межах окремого веб-сайту. Йдеться про задачу оптимального добору посилань між окремими сторінками веб-сайту з метою максимального сприяння відвідувачеві в задоволенні його потреб і досягненні мети. Ця задача має очевидний зв'язок із загальною проблемою інформаційного пошуку [3], але має і свою специфіку.

Крім класичного підходу до пошуку найбільш релевантних посилань на основі аналізу

мір близькості між сторінками [2; 3], перспективним видається застосування інших підходів, зокрема на основі аналізу поведінки відвідувачів – реальної або промодельованої. Деякі міркування з цього приводу наведено в [5], але проблема потребує подальших досліджень.

Застосування подібних методик видається найбільш перспективним для ресурсів, які характеризуються високою інформаційною зв'язністю, тематичною однорідністю та достатньою якістю інформаційного наповнення. Зокрема, такі властивості притаманні тематичним порталам. Інший перспективний напрямок – системи електронного