

РЕАЛІЗАЦІЯ ТА ФУНКЦІЇ ІНСТРУМЕНТАЛЬНОЇ СИСТЕМИ РОЗРОБКИ ПРОГРАМ ДИСТАНЦІЙНОГО НАВЧАННЯ

© Семотюк В., 2005

Розглянуто деякі особливості реалізації та основні функції інструментальної системи, призначеної для розробки програм дистанційного навчання.

The approach to design and the main functions of the instrumental distance learning system are discussed.

Вступ

Дистанційне навчання має такі переваги над традиційними способами навчання, як гнучкість, що дає можливість займатися в зручній для себе час, у зручному місці і темпі; модульність – забезпечує можливість із набору незалежних навчальних курсів формувати навчальний план; рівнобіжність – рівнобіжне з професійною діяльністю навчання, тобто без відриву від виробництва; охоплення – одночасне звертання до багатьох джерел навчальної інформації (електронних бібліотек, банків даних, баз знань тощо); економічність – мінімальне використання навчальних площ, технічних засобів, транспортних засобів (у випадку заочного навчання); технологічність – використання новітніх досягнень інформаційних і телекомунікаційних технологій та ін.

Постановка проблеми.

Структуру бази даних розроблено з врахуванням таких аспектів:

до системи можна занести будь-яку кількість навчальних курсів;

zareєстровані користувачі (учні) можуть вивчати будь-які курси, які внесені в систему;

для кожного користувача створено модель учня;

для кожного навчального курсу є можливість створювати будь-яку кількість контрольних робіт, які можуть містити необмежену кількість запитань.

Для створення бази даних (БД) було обрано архітектуру “клієнт–сервер” для сервера MS SQL Server 2000. Він досить простий під час встановлення, налагодження і, головне, адміністрування порівнянно з іншими SQL-серверами, має чудові функціональні можливості. Характерною особливістю архітектури “клієнт–сервер” є перенесення основного обчислювального навантаження на сервер і максимальне розвантаження програми клієнта від обчислювальної роботи, істотного закріплення безпеки даних – як від навмисних, так і від помилкових змін. БД у цьому випадку розміщують на мережному сервері, як і в архітектурі “файл–сервер”, проте прямого доступу до БД з програми клієнта не відбувається. Функції прямого звернення до БД здійснює спеціальна керуюча програма – сервер БД.

Основний матеріал

Взаємодія сервера і програми-клієнта відбувається так. Клієнт формує запит і відсилає його серверу. Сервер, прийнявши запит, опрацьовує його (звертаючись, можливо, до сервера БД) і результат повертає клієнту. В клієнтській програмі в основному інтерпретуються отримані від сервера дані, реалізується інтерфейс з користувачем і вводяться дані.

Для того, щоб серверна частина навчальної системи згенерувала потрібну web-сторінку, клієнтська частина повинна передати на сервер деякі дані, наприклад, назву стану системи (див. рисунок), з якого було передано ці дані, номер лекції, логін і пароль користувача тощо. Тоді сервер визначає наступний стан системи і генерує потрібну html-сторінку.

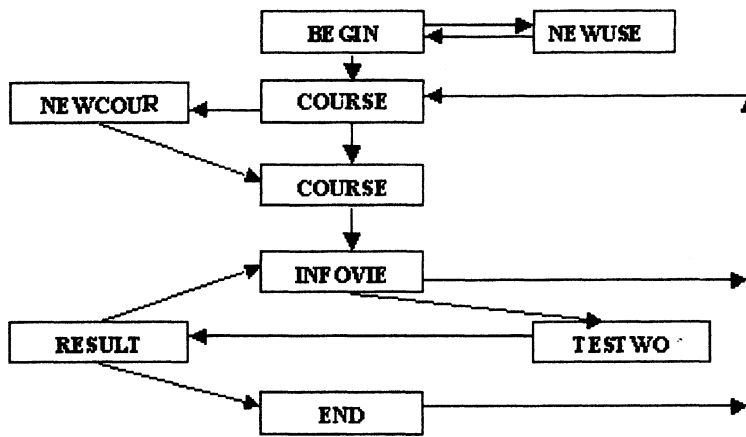


Схема переходів між станами клієнта

Стан **BEGIN** – вхід до системи (перевірка, чи користувач з введеним логіном і паролем зареєстрований у системі). Якщо користувач ще не зареєстрований, вхід у систему не відбудеться.

Реєструючи нового користувача, система переходить у стан **NEWUSER** – створення нового користувача (внесення логіна, імені, прізвища, по батькові, пароля). Якщо користувач зареєстрований у системі, то наступний стан для системи **COURSESEL** – вибір навчального курсу з раніше розпочатих. Якщо учень вибирає новий курс для навчання, то система переходить у стан **NEWCOURSESEL** – вибір нового навчального курсу.

Якщо користувач підтвердив вибір навчального курсу, стан системи – **COURSESHOW** – підтвердження вибору курсу, то учневі пропонується перша лекція навчального матеріалу і стан системи – **INFOVIEW** – показ навчальних матеріалів.

Якщо учень закінчив вивчення лекції і хоче перейти до наступного навчального матеріалу, йому потрібно пройти тестування; стан системи – **TESTWORK** – проходження тестування. Після введення користувачем відповідей на запитання система виводить результат тестів; стан системи – **RESULT** – виведення результатів тестування.

Ці кроки (демонстрація навчальних матеріалів, проходження тестів і виведення результатів) повторюються доти, поки курс не буде закінчено і стан системи не перейде у **END** – успішне закінчення навчального курсу.

У навчальній системі розроблено схему доступу, яка дозволяє зареєстрованим користувачам вибирати курс для навчання, проходити тестові завдання, переглядати матеріали лекцій, які він вже засвоїв. Якщо користувач зареєстрований як експерт системи, то йому надано право змінювати створені раніше курси, створювати нові навчальні курси, вилучати курси.

Схему бази даних системи, що містить одинадцять взаємопов'язаних таблиць, наведено в роботі*. Одна з таблиць – **ModelStructures** – призначена для зберігання назв полів моделі учня. Обов'язковим полем є **"InfoID"**, яке зберігає номер уроку, на котрому перебуває учень. Значення в поля заносять як бали за виконання контрольних робіт. Потім програма шукає рядок в схемі курсу (**SQLSchema**), який задовольняє модель учня (перевіряється кількість балів, набраних учнем для переходу/непереходу на наступну лекцію). У схемі курсу задано, що робити, набравши певну кількість балів (**ActionTypeID**) і на яку лекцію переходити (**ActionText**). Якщо **ActionTypeID=2**, то курс пройдено. Програму діагностування призначено для оцінювання відповідей учня на тестові завдання. **Asp**-файл (серверна частина) передає цій програмі номер завдання й відповідь учня та передає серверу оцінку за це завдання. Щоби визначити, яку кількість балів отримає учень за виконання певного завдання, діагностор шукає в таблиці **Tests** полі **Answer** правильну відповідь і порівнює її з відповіддю учня. Перевірка відбувається шляхом створення тимчасових таблиць (копій з існуючих таблиць із даними). Потім виконується запит користувача (над однією зі створених тимчасових таблиць) та запит, що міститься в правильних відповідях. На основі зіставлення результатів роботи користувачьких і програмних запитів виносяться рішення про

* Семотюк В.М. Інструментальна система розробки програм дистанційного навчання. – Збірник „Комп'ютерні технології друкарства”. 2004. – № 12, С. 93–97.

правильність або неправильність відповіді користувача. Якщо відповідь правильна, програма повертає максимальну кількість балів за це завдання (максимальна кількість балів міститься у таблиці Tests в полі Mark), інакше – нуль.

Програма Editor призначена для редагування курсів. Таблиці у базі даних може редагувати та створювати тільки експерт, тобто доступ для інших користувачів у базу неможливий. Editor має зручний інтерфейс для введення інформації. У ній є багато підказок, а також під час введення можна вибрати інформацію із спадних меню.

Перед початком створення курсу потрібно розробити структуру курсу:
визначити кількість лекцій;
сформувані окремо кожну лекцію;
сформувані окремо питання для контрольних робіт (кожному питанню надати свій порядковий номер);

сформувані контрольні роботи (надати номери питань кожній контрольній роботі).

Для створення нового курсу потрібно внести таку інформацію:

назва курсу (автоматично вноситься до таблиці „Курси” – Courses поле Name);

назва курсу латинськими символами (автоматично вноситься у таблицю „Курси” – Courses поле Location, а також використовується для створення ModelSchema та SchemaTable);

автор створеного курсу (автоматично вноситься у таблицю „Курси” – Courses поле Author);

інформація про структуру курсу:

назва уроку (автоматично вноситься у таблицю „Структура курсу” – CourseStructures поле InfoName);

назва файлу уроку (автоматично вноситься у таблицю „Структура курсу” – CourseStructures поле InfoFile).

Після внесення цієї інформації програма Editor автоматично створює записи у таблицю „Структура моделі учня” – ModelStructures, на основі інформації з таблиці „Структура курсу” – CourseStructures. Потім створюють таблицю „Модель учня” – ModelTable (для конкретного курсу створюють таблицю з відповідною назвою), на основі інформації з таблиці „Структура моделі учня” – ModelStructures. Автоматично створюється таблиця „Схема курсу” – SchemaTable, інформацію в яку вносить експерт за допомогою програми Editor.

У програмі Editor передбачено також створення контрольних питань та формування контрольних робіт для курсу. Для цього потрібно внести таку інформацію:

1. Інформація про питання курсу:

– тип завдання, наприклад, вибірка відповідей із запропонованих (інформацію заносять у таблицю Tests поле TestTypeID);

– максимальна оцінка за відповідь (інформація заноситься у таблицю Tests поле Mark);

– правильна відповідь на питання (інформація заноситься у таблицю Tests поле Answer);

– назва файлу, де міститься питання (інформація заноситься у таблицю Tests поле FileName).

2. Інформація про контрольні роботи після кожної лекції:

– номер завдання в контрольній роботі, який внесений у таблиці TestNumber (інформація заноситься у таблицю ControlWorks).

Таблицю SchemaTable заповнює експерт вручну, попередньо сформувавши схему переходу учня на наступну лекцію (визначається кількістю балів, яку повинен набрати учень для того, щоб перейти/не перейти на наступну лекцію). Для цього потрібно заповнити поля з назвою лекцій. У полях, де стоїть суфікс –min, треба вписати мінімальну кількість балів для переходу на наступну лекцію, а в полях з суфіксом –max – максимальну кількість балів. Таблицю треба заповнювати, враховуючи всі можливі варіанти переходу/непереходу на наступну лекцію. Поля мають такий зміст: InfoID – номер лекції, на якій в даний момент знаходиться учень, поле ActionText вказує, на яку лекцію переходити/не переходити, ActionType номер типу дії при певній комбінації (1 – перехід на наступну лекцію, 2 – закінчення курсу).

Для остаточного створення курсу потрібно виконати такі дії.

1. Кожну лекцію сформувані в окремий html-файл за таким шаблоном: створити текстовий файл, зберегти його як файл html, вставити у файл код програми:

```
<html>
<p align=justify>
{Вставити текст лекції}
</P>
<a target="_top" href=" ../index.asp? source_state=INFOVIEW &source_frame=info&lesson_
number={вставити порядковий номер лекції, наприклад 1}">Розпочати тестування</a>
</html>
```

2. Кожне тестове питання потрібно сформувавши в окремий html-файл за таким шаблоном: створити текстовий файл, зберегти його як файл-html, вставити у файл код програми:

```
<html>
<body>
{Вставити текст тестового питання}
</html>
</body>
```

Цей файл повинен міститись на диску в папці програми Teacher/Client/Courses/{назва папки створеного курсу}/Tests.

Висновок

Розроблена система є універсальною, оскільки дозволяє створювати навчальні курси з будь-якої предметної сфери. Для кожного навчального курсу передбачено створення контрольних робіт, які містять запитання різної складності. Для кожного питання є можливість встановити потрібну кількість балів, залежно від складності питання. Відповідно до набраної кількості балів система дає/не дає можливість користувачеві перейти на іншу лекцію для вивчення.

УДК 004.415

П. Жежнич, А. Пелешин, Ю. Серов, Д. Тарасов
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

АРХІТЕКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛАСИФІКАЦІЇ ТА РАНГУВАННЯ НОВИНИХ ІНТЕРНЕТ-РЕСУРСІВ

© Жежнич П., Пелешин А., Серов Ю., Тарасов Д., 2005

Розглянуто архітектуру та принципи побудови автоматизованої системи класифікації та рангування новинних Інтернет-ресурсів. Призначення системи — автоматизована агрегація новин, поданих у електронній формі, їх класифікація, фільтрація, рангування, впорядкування, зручне подання.

Використання системи значно спрощує роботу користувача новинних каналів у форматі RSS, позбавляє користувача необхідності опрацьовувати нетематичні та малоінформаційні новини.

The paper considers actual problem of intelligent system of information gathering and analysis from news Internet-resources development. The system work is based on news gathering from RSS-channels of news Web-sites. System is intended for new messages fast download, these messages classifying, ranking, ordering and presentation of retrieved messages in proper view.

System consists of 4 components: Crawler, Grabber Analyzer, Representer. The proposed system allows simpler and faster news processing received from Internet-resources.

Постановка проблеми у загальному вигляді

Для забезпечення користувачів Інтернет актуальною інформацією про новини, події, нові технології існує величезна кількість Інтернет-сайтів, котрі надають інформацію про новини в