

УДК 519.63

Розробка комп'ютерної системи тестування на базі XML

Дубровін О. Ф.

`informatyka@drohobych.net`

Дрогобицький державний педагогічний університет імені Івана Франка, кафедра інформаційних систем і технологій, вул. Стрийська 3, м. Дрогобич, 82100

У статті розглядається методика розробки бази даних тестувальної системи на основі технології XML.

Вступ

В останні роки тестування як один із способів контролю за знаннями учнів та студентів набуло широкого поширення, зокрема, завдяки впровадженню модульно-рейтингової системи навчання і проведенню незалежного тестування при вступі до вищих навчальних закладів. На наш погляд, уся система тестування повинна бути комп'ютеризованою, що допоможе як прискорити сам процес тестування, так і повністю виключити суб'єктивний фактор при його проведенні. Сьогодні існує велика кількість систем тестування, яким, на наш погляд, притаманна одна методологічна помилка. Розглянемо її на прикладі. На кожній кафедрі вищого навчального закладу розробляються десятки і навіть сотні різних тестових завдань для оцінки рівня підготовки студентів різних спеціальностей з різних предметів кафедри. При цьому чимало тестових завдань певною мірою дублюються. Як приклад розглянемо тести, які використовуються в конкретних модулях. При здачі заліку, екзамену, екзамену за другим талоном та на комісії частини цих тестів знову включається до опитування. І, нарешті, при підсумковому (бакалаврському) екзамені, який проводиться з профільних предметів спеціальності, тести з окремих модулів (назвемо їх базовими) входять до складу результуючого тесту.

В описаному вище випадку найбільш доцільним, на наш погляд, є тестування з використанням комп'ютерної техніки. Весь необхідний базовий набір тестів при цьому слід зберігати у спеціальним чином організованій базі даних, а всі похідні тести формувати викладачам за допомогою спеціально розробленого інтерфейсу. Саме тому розробка методичних та програмних засобів, які полегшують створення комп'ютерних систем тестування, є актуальним завданням.

Мета статті. Показати ефективність зберігання бази даних тестових завдань у виді XML-файлів та запропонувати способи розробки інтерфейсу для роботи з такою базою даних.

Концептуальна модель бази даних

У системі тестування всі інформаційні блоки даних повинні зберігатися у базі даних. Для її проектування спочатку необхідно побудувати концептуальну модель. У результаті попереднього дослідження системи тестування було виявлено такі концептуальні твердження:

1. На кафедрі проводиться тестування студентів.
2. Тестування проводиться з предметів, які викладаються викладачами кафедри для різних спеціальностей (наприклад, "макроекономіка для спеціальності банківський рекет").
3. Існують декілька форм контролю за знаннями студентів:
 - a) поточний модульний контроль;
 - b) залік;
 - c) екзамен;
 - d) підсумковий державний екзамен зі спеціальності.
4. Тести поділяються на дві категорії:
 - a) базові тести, які формуються викладачами для конкретних модулів;
 - b) комбіновані тести, які формуються на основі низки базових тестів (наприклад, для таких форм контролю, як залік та екзамен).
5. Тести містять таку інформацію:
 - a) запитання студентів;
 - b) перший варіант відповіді;
 - c) другий варіант відповіді;
 - d) третій варіант відповіді;
 - e) четвертий варіант відповіді;
 - f) номер правильного варіанту відповіді.
6. Усі студенти, які братимуть участь у тестуванні, реєструються.
7. Реєстрація містить такі дані про студента:
 - a) логін студента (наприклад, номер студентського квитка), за яким він реєструється;
 - b) пароль студента, за яким він проводить тестування;
 - c) прізвище студента;
 - d) ім'я студента;

е) назва групи.

Попередній аналіз показує дані, які потрібні для підсумкового екзамєну, мають чітко виражену деревовидну ієрархічну структуру. На верхньому рівні цієї структури (корінь дерева) знаходиться вузол "Підсумковий екзамєн". Всю структуру можна описати так:

1. Нульовий рівень – "Підсумковий екзамєн" – кореневий вузол.
2. Перший рівень – набір вузлів "Предмети", які входять до підсумкового екзамєну.
3. Другий рівень – набір вузлів "Модулі", які стосуються до даного предмету.
4. Третій рівень – набір "Запитання", "Відповіді", "Номери правильних відповідей", які стосуються до конкретного модуля.

При реалізації цієї структури за допомогою реляційної моделі, на етапі логічного проектування одержимо п'ять взаємопов'язаних таблиць [1–2], з подальшою фізичною реалізацією бази даних на базі клієнт-серверної СУБД "Fire Bird 1.5", яка досить добре описана, без ліцензійна і поширюється через Internet.

Подібна стратегія розробки бази даних тестування має ряд недоліків:

1. Використання клієнт-серверної бази даних, розміщеної разом з СУБД на комп'ютері викладача, не завжди забезпечує надійність тестування через можливі збої у комп'ютерній мережі.
2. Використання клієнт-серверної бази даних, розміщеної разом з СУБД на комп'ютері студента, поглиблює проблему адміністрування такої системи даних.
3. Розробка інтерфейсу користувача, призначеного для формування тестових завдань для всіх можливих варіантів заліків та екзамєнів є складним завданням.
4. Розроблена база даних тестових завдань не має наглядності, яка б допомагала викладачу виключати помилки при формуванні тестів.

Позбутися цих недоліків значною мірою можна при проектуванні бази даних за допомогою технології XML [3].

База даних на основі двомірних файлів XML

Найпростіший спосіб зберігання даних XML полягає у використанні одного файлу, в якому зберігається весь документ XML. До таких даних можна звертатися за допомогою різноманітних текстових редакторів та спеціальних інструментальних засобів XML. До файлу легко звернутися за допомогою довільної мови програмування, і він доступний для усіх синтаксичних аналізаторів XML.

Реляційні та об'єктні бази даних мають переваги перед двомірними файлами XML у тому, що, використовуючи механізм індексації, забезпечують значно вищу швидкість пошуку інформації і більш ефективно використовують пам'ять при

дуже великих масивах інформації. Для нашого випадку ці переваги несуттєві, оскільки в нашій базі даних у процесі тестування пошук не здійснюють, а проводиться лише послідовне зчитування даних. Об'єм даних, необхідних для тестування, також не можна назвати великим відносно апаратної бази сучасного комп'ютера.

У силу того, що нам потрібно передбачити формування масиву тестової інформації для різних видів іспиту, нашу базу даних ми подамо у вигляді двох XML-документів: файлу Test.xml

```
<?xml version="1.0" encoding="windows-1251"?>
<Тести>
<Предмет Назва="Операційна система MSDOS" >
<Модуль Назва="Файлова система" Запитань="25" >
<Запитання/> Яке з вказаних тверджень є хибним?
<Відповіді
_1="Розширення файла є обов'язковим"
_2="Розширення може мати 1-го до 3-х символів"
_3="Розширення може мати тільки 3 символи"
_4="Текстові файли в основному мають розширення *.txt або *.doc"/>
<Правильна_відповідь N="3"/>
</Модуль>
<Модуль Назва="Основні команди MSDOS" Запитань="25" >
<Запитання/> Яка команда знищує підкаталоги?
</Модуль>
<Модуль Назва="Пакетні файли" Запитань="25" >
<Запитання/> Яке розширення мають пакетні файли MSDOS?
</Модуль>
</Предмет>
<Предмет Назва="Операційна система Windows" >
<Модуль/>
</Предмет>
</Тести>
```

та файлу Examen.xml

```
<?xml version="1.0" encoding="windows-1251"?>
<Екзамєн Форма="Підсумковий_екзамєн" >
<Предмет Назва="Операційна система MSDOS" >
<Модуль Назва="Пакетні файли" Запитань="10"/>
<Модуль Назва="Основні команди MSDOS" Запитань="10"/>
<Модуль Назва="Файлова система" Запитань="10"/>
```

```

</Предмет>
<Предмет Назва="Операційна система Windows">
<Модуль/>
</Предмет>
</Екзамен>

```

Документ Test.xml містить повний набір тестів для всіх предметів та їх модулів. У кожного модуля, крім атрибуту Назва, є атрибут Запитання, у якому міститься цифра тестових запитань, які стосуються цього модуля.

У документі Екзамен.xml міститься керуюча інформація про те, з яких предметів, яких модулів і в якій кількості з документа Test.xml формується набір тестів для даного типу іспиту (залік, екзамен чи підсумковий екзамен). Таким чином, документ Test.xml створюється в єдиному екземплярі, а документи Екзамен.xml формуються викладачем для кожного конкретного екзамену.

Монітор викладача

Монітор викладача – це програмний засіб, призначений для того, щоб полегшити роботу викладача в процесі розробки тестових завдань. Він виконує дві основні задачі – надає викладачеві зручний редактор для створення xml-файлів, та формує на основі двох xml-документів – Test.xml та Екзамен.xml результуючий xml-файлів, який і буде використовуватися іншим програмним засобом – *Тестуючим монітором* для проведення тестування. Обидві програми пропонується реалізувати засобами Delphi.

Для роботи з xml-документами Delphi використовує XML DOM – об'єктну модель, яка надає у розпорядження розробника об'єкти для завантаження і обробки XML-файлів. Об'єктна модель складається з таких основних об'єктів: XMLDOMDocument, XMLDOMNodeList, XMLDOMNode, XMLDOMNamedNodeMap і XMLDOMParseError. Кожен з цих об'єктів (крім XMLDOMParseError) містить властивості і методи, які дають змогу одержувати інформацію про об'єкт, маніпулювати його значеннями і структурою, а також переміщатися по структурі XML-документа.

Document Object Model подає XML-документ в виді деревовидної структури, яка складається з гілок. Програмні інтерфейси XML DOM дозволяють прикладним програмам переміщуватися по дереву документа і маніпулювати його гілками. Кожна гілка може мати специфічний тип (DOMNodeType), згідно якого визначаються батьківська та дочірні гілки. В більшості XML-документів можна зустріти гілки типу element, attribute і text. Атрибути (attribute) являють собою особливий вид гілки і не є дочірніми гілками. Для керування атрибутами використовуються

спеціальні методи, які надаються об'єктами XML DOM.

Алгоритм формування комбінованого тесту із системи базових тестів має вид:

1. Згідно з xml-документом Екзамен.xml з документа Test.xml вибирається елемент з рівня Предмет, який відповідає першому предмету у вузлах Екзамен.xml.

2. У цьому елементі вибирається елемент з рівня Модуль, який відповідає по імені першому модулю першого предмету у вузлах Екзамен.xml.

3. Усі дочірні елементи цього вузла заносяться до масиву, компонентами якого є запис, поля якого по імені співпадають з цими вузлами (Запитання, Відповіді та ін).

4. Після цього всі елементи масиву сортуються випадковим чином, з них вибирається кількість елементів, яка задана атрибутом Запитань з поточного вузла Екзамен.xml і з цієї кількості додаються гілки до нового xml-документа CombTest.xml.

5. Далі вибирається наступний елемент з рівня Предмет, який відповідає першому предмету у вузлах Екзамен.xml, і вся процедура повністю повторюється.

У результаті одержуємо xml-документ CombTest.xml, який має структуру.

```

<?xml version="1.0" encoding="windows-1251"?>
<Комбіновані_Тести>
<Питання>
<Запитання/>Яке з вказаних тверджень є хибним?
<Відповіді
_1="Розширення файлу є необов'язковим"
_2="Розширення може мати від 1-го до 3-х символів"
_3="Розширення може мати тільки 3 символи"
_4="Текстові файли в основному мають розширення *.txt або *.doc"/>
<Правильна_відповідь N="3"/>
</Питання>
<Питання>
<Запитання/> . . . .
<Відповіді
_1=". . . ."
_2=". . . ."
_3=". . . ."
_4=". . . ."/>
<Правильна_відповідь N=".."/>
</Питання>
. . . . .
</ Комбіновані_Тести >

```

Тестовий монітор

Тестовий монітор – це програмний засіб, призначений для того, щоб надати необхідний інтерфейс студенту для проходження тестування. У

процесі тестування студентіві послідовно видаються запитання та набір відповідей, з яких необхідно вибрати правильну. Тестування проходить у певних часових рамках, які відміряються таймером. Результати тестування заносяться до спеціального xml-файлу, який утворює базу даних результатів.

Інформація, необхідна для тестування, подається у моніторі файлом CombTest.xml. Робота xml-файлами у тестовому моніторі організується за допомогою компонентів Delphi dbExpress, які надають можливість працювати з файловою базою даних MyBase, поданих у форматі xml [4].

Бібліографія

1. Дубровін О, Дубровін М. Розробка структури універсальної тестуючої системи // Молодь і ринок. – 2008. – № 1. – С. 45–49.
2. Конноли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Второе издание. – М., С.Пб, К.: Вильямс, 2001 – 1111 с.
3. Грейвс М. Проектирование баз даних на основе XML. – К.: Вильямс, 2002 – 639 с.
4. Кенто М. Delphi 2005. – С.Пб.: Питер, 2001 – 905 с.