

Олег Дубровін, кандидат технічних наук, доцент
Дрогобицького державного педагогічного університету
імені Івана Франка

КОМП'ЮТЕРНА СИСТЕМА ТЕСТУВАННЯ НА БАЗІ XML

У статті розглянуто задачу проектування універсальної комп'ютерної системи тестування на базі технології XML.

Ключові слова: бази даних, файли XML, монітор, Delphi, тестування.

Актуальність теми. В останні роки тестування, як один із способів контролю за знаннями учнів та студентів, набуло широкого поширення завдяки впровадженню модульно-рейтингової системи навчання та 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. У даний час 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 по різних предметах кафедри. При цьому, ряд тестових завдань частково дублюються. Розглянемо тести, які використовуються в конкретних модулях. При здачі заліку, екзамену, екзамену за другим талоном та на комісії частина 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 видом тестування, на наш погляд, є тестування з використанням комп'ютерної техніки. Весь необхідний базовий набір тестів при цьому слід зберігати в спеціальному 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.

Виклад основного матеріалу.

1. Концептуальна модель бази даних

У системі тестування усі інформаційні блоки даних повинні зберігатися у базі даних. Для проектування цієї бази даних 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 по предметам, які викладаються викладачами кафедри для різних спеціальностей (наприклад, "макроекономіка для спеціальності банківський рекет").

3. Існують декілька форм контролю за знаннями студентів:

- поточний модульний контроль;
- залік;
- екзамен;
- підсумковий екзамен по спеціальності бакалавр.

4. Тести поділяються на дві категорії:

- Базові тести, які формуються викладачами для конкретних модулів;
- Комбіновані тести, які формуються на основі ряду базових тестів (наприклад для таких форм контролю як залік та екзамен.

5. Тести містять таку інформацію:

- запитання студентів;
- перший варіант відповіді;
- другий варіант відповіді;
- третій варіант відповіді;
- четвертий варіант відповіді;
- номер правильного варіанту відповіді.

6. Всі студенти, які прийматимуть участь у тестуванні реєструються.

7. Реєстрація містить такі дані про студента:
- логін студента (наприклад, номер студентського білета), за яким він реєструється;
 - пароль студента, за яким він проводить тестування;
 - прізвище студента;
 - ім'я студента;
 - назва групи студента.

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 відносяться до даного предмету.
4. Третій рівень – набір "Запитання", "Відповіді", "Номери правильних відповідей", які відносяться до конкретного модуля.

При реалізації цієї структури за допомогою реляційної модулі, 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 в 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 являє собою складну задачу.
4. Розроблена база даних тестових завдань немає наглядності, яка б допомагала викладачу виключати помилки при формуванні тестів.

Позбутися цих недоліків значною мірою допомагає проектування бази даних за допомогою технології XML [3].

2. База даних, на основі двовірних 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 двовірними 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 не суттєві, 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.

В силу того, що нам потрібно передбачити формування масиву тестової інформації для різних видів іспиту, нашу базу даних ми подамо у виді 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 файлу є необов'язковим"
        _2="Розширення може мати 1-го до 3-х символів"
        _3="Розширення може мати тільки 3 символи"
        _4="Текстові файли 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"?>
<Екзамен Форма="Підсумковий_екзамен">
<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 Запитань, у якому міститься цифра тестових запитань, які стосуються цього модуля.

В документі Examen.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 Examen.xml формуються викладачем для кожного конкретного виду екзамену.

3. 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 викладачу зручний редактор для створення xml-файлів, та формує на основі двох xml-документів – Test.xml та Examen.xml результуючий xml-файл, який і буде використовуватися іншим програмним засобом – *Тестуючим монітором* для проведення тестування. Обидві програми пропонуються 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 дозволяють одержувати інформацію

про об'єкт, маніпулювати значеннями і структурою об'єкта, а також перемішуватися по структурі XML-документа.

Document Object Model подає XML-документ у виді деревовидної структури, яка складається з гілок. Програмні інтерфейси XML DOM дозволяють прикладним програмам перемішуватися 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 xml-документу Examen.xml з документа Test.xml вибирається елемент з рівня Предмет, який відповідає першому предмету у вузлах Examen.xml.

2. У цьому елементі вибирається елемент з рівня "Модуль", який відповідає по імені першому модулю першого предмету у вузлах Examen.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 Examen.xml і з цієї кількості додаються гілки до нового xml-документу CombTest.xml.

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

В 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 1-го до 3-х
символів"
_3="Розширення може мати тільки 3 символи"
_4="Текстові файли мають розширення *.txt
або *.doc"/>

```

```

<Правильна_відповідь N="3"/>
</Питання>
<Питання>
<Запитання/> . . . .
<Відповіді
_1="..."
_2="..."
_3="..."
_4="...""/>
<Правильна_відповідь N=".."/>
</Питання>
. . . . .
</ Комбіновані_Тести >

```

3. Тестовий монітор

Тестовий монітор – це програмний засіб, призначений для того, щоб надати необхідний інтерфейс студенту для проходження тестування. У процесі тестування студентові 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, поданій у форматі xml [4].

Висновки. Аналіз системи тестів, які потрібні для підсумкового екзамєну, мають чітко виражену деревовидну ієрархічну структуру. Використання для інформаційного сховища таких даних клієнт-серверних технологій, які базуються на реляційній моделі має ряд недоліків. Позбутися цих недоліків значною мірою допомагає проектування бази даних за допомогою технології XML. Для роботи з xml-документами в середовищі Delphi зручно використовувати XML DOM – об'єктну модель, яка надає у розпорядження розробника об'єкти для завантаження і обробки XML-файлів.

1. Дубровін О, Дубровін М. Розробка структури універсальної тестуючої системи. //Молодь і ранок. – 2008. – № 1. – С. 45 – 49.

2. Конноли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Второе издание. Вильямс. М., СПб, К. 2001 – 1111 с.

3. Грейвс М. Проектирование баз даних на основе XML. Вильямс. К. 2002. – 639 с.

Стаття надійшла до редакції 12.05.2009

а б с д а б с д а б с д а б с д а б с д а б
v v

?

Джерела мудрості

“... талановитий педагог – запорука успішності суспільства і держави, а основна місія вчителя – прищепити любов не стільки до предмета, скільки до освіти.”

Іван Зязюн

доктор філософських наук, професор, академік АПН України

“Українське суспільство прагне створити системну освіту, яка б відповідала викликам часу і потребам особистості, а тому відшукує нові моделі її розвитку. У найперспективніших моделях випробування часом, в основу покладено ідею формування і динамічного розвитку природного таланту кожної дитини як запоруку її самоствердження на життєвому шляху”.

Василь Мадзігон

академік, перший віце-президент АПН України