

УДК 651.3:518.5

СТАНДАРТ РОЗРОБКИ ТЕСТОВИХ ЗАВДАНЬ В АДАПТИВНІЙ СИСТЕМІ КОНТРОЛЮ ЗНАНЬ

П.І. Федорук

Прикарпатський національний університет ім. Василя Стефаника

e-mail: pavlo@pu.if.ua

1. Вступ

Участь України у Болонському процесі визначило для українських вищих навчальних закладів необхідність забезпечення технологічної, академічної, навчально-методичної, правової та організаційної інтероперабельності навчального процесу. Термінологічно інтероперабельність означає здатність системи до узгодженої взаємодії з іншими системами, тобто працездатність прикладних програм в середовищі будь-яких версій базових системних засобів.

Забезпечення інтероперабельності освітніх систем в умовах відкритої освіти може бути досягнуто за рахунок вирішення комплексу актуальних завдань:

- перехід до технологічно інтероперабельних освітніх систем та розробці відповідних технологій навчання;
- використання об'єктної технології й ідеї інтероперабельності, для розробки стандарту об'єктних баз даних, узгоджених з міжнародними технологічними стандартами;
- розробка розподілених освітніх систем на базі нових загальноприйнятих архітектур;
- створення навчально-методичного забезпечення у відповідності за структурою і якістю міжнародним стандартам;
- забезпечення академічної мобільності, впровадження системи кредитів у відповідності до Болонського процесу;
- розробка навчального контенту з використанням об'єктно-орієнтованої технології підготовки навчально-методичних матеріалів.

Вища освіта гостро потребує розробки і впровадження уніфікованих технологічних рішень. До цього результату веде достатньо тривалий підготовчий процес, що включає впровадження сучасних технологічних систем в освіті, формалізацію процесів взаємодії системних компонентів технологічних освітніх систем, розробку стандартів на інтерфейси, формати, протоколи обміну інформацією з метою забезпечення мобільності, інтероперабельності, стабільності, ефективності навчального процесу в системі відкритої освіти.

У даному дослідженні нами здійснено спробу розробки та запровадження стандарту створення адаптивних тестів.

2. Стандарт розробки адаптивних тестів

Адаптивне тестування – різновид тестування, при якому порядок представлення запитань (або складність) залежить від відповідей того, хто тестується на попередні запитання [1,2]. Доцільність адаптивного контролю впливає з необхідності раціоналізації традиційного тестування. При успішній відповіді наступне завдання комп'ютер підбирає більш важким, при неуспішному – легшим. Зрозуміло, цей алгоритм вимагає попереднього випробування всіх завдань, визначення їх міри складності, а також створення банку завдань і спеціальної програми.

Використання завдань, що відповідають рівню підготовленості, істотно підвищує точність вимірів і мінімізує час індивідуального тестування до 5-10 хвилин. Адаптивне тестування дозволяє забезпечити комп'ютерну видачу завдань на оптимальному, приблизно 50%-му рівні імовірності правильної відповіді для кожного студента.

Вивчення механізму тестування в різних системах навчання показало необхідність використання математичної моделі при розробці системи дистанційного тестування. Зокрема, при складанні логічного зв'язку шарів навчання в СДО використовується мова Unified Modeling Language (UML). Кожен шар розглядається як безліч взаємозв'язаних елементів навчання. Навчальна програма наділяється точками контролю, в яких відбувається розгалуження програми, пов'язане з навчанням на наступному рівні або поверненням учня на перенавчання.

При зберіганні інформаційних ресурсів, наприклад тестів, в системах управління базами даних при проектуванні необхідно враховувати факт майбутньої конвертації даних у файли спеціального формату, зручного для забезпечення передачі цих інформаційних ресурсів в інші СДО. Для забезпечення зв'язку між СДО, які побудовані на різних технологіях і мовах програмування, використовується мова UML. Коректний обмін між системами забезпечується стандартом XML. Проаналізувавши недоліки, врахувавши рекомендації та зразки світових стандартів був розроблений формат Test Standart, який задовільняє вимоги систем дистанційного навчання та адаптивного тестування.

Під форматом TEST STANDART (TST) розуміється:

1. Вимоги до структури і змісту файлу тесту.
2. Вимоги до мови опису тесту.

TST спирається на відкритий стандарт XML, тому також є відкритим.

Мова опису тесту

Файл content.xml є описом документа тесту на мові XML.

Файл повинен починатися із стандартного заголовка XML у вигляді:

```
<?xml version="1.0" encoding="windows-1251"?>
```

Файл може бути записаний в кодуванні UTF-8 або ANSI (windows-1251).

Відповідне значення поточного кодування повинне бути вказане в атрибуті encoding заголовка XML документа.

Елемент test

Кореневим елементом документа є елемент test, тобто, весь опис тесту повинно починатися з відкриваючого тега <test> і закінчуватися закриваючим тегом </test>.

Приклад мінімального вмісту content.xml:

```
<?xml version="1.0" encoding="windows-1251"?>
```

```
<test>
```

```
</test>
```

Елемент test є кореневим елементом документа тесту. Містить решту всіх елементів описи тесту.

```
<!ELEMENT test (name?,author?,abstract?,questions?,selector?,messages?)>
```

Опис тесту складається з наступних частин:

- розділ відомостей про тест;
- розділ опису секцій ;
- розділ опису запитань;
- розділ опису повідомлень.

Розділ відомостей про тест

Розділ відомостей про тест містить назву тесту, відомості про автора документа і текстовий опис документа.

Елемент name містить назву документа.

```
<!ELEMENT name (#PCDATA)>
```

Елемент author містить відомості про автора документа.

```
<!ELEMENT author (#PCDATA)>
```

Елемент abstract містить додаткові відомості про документ.

```
<!ELEMENT abstract (#PCDATA)>
```

Розділ опису секцій

Елемент selector задає розділ секцій тесту. Кожна секція визначається за допомогою елемента <section>.

```
<!ELEMENT selector (section)+>
```

Елемент section описує одну секцію тесту. Кожна секція може складатися з чотирьох списків питань, призначених для різних рівнів складності.

```
<!ELEMENT section (allout?,goodout?,midlout?,badout?)>
```

```
<!ATTLIST section
```

```
id CDATA #IMPLIED>
```

Атрибут id – задає унікальний ідентифікатор секції в межах тесту. Значення може складатися з букв і цифр, але не повинно починатися з цифри.

Елемент allout визначає список питань, які повинні виводитися при будь-якому рівні складності.

```
<!ELEMENT allout EMPTY>
```

```
<!ATTLIST allout %qlistatt;>
```

Елемент goodout визначає список питань, які повинні виводитися при максимальному рівні складності.

```
<!ELEMENT goodout EMPTY>
```

```
<!ATTLIST goodout %qlistatt;>
```

Елемент midlout визначає список питань, які повинні виводитися при середньому рівні складності.

```
<!ELEMENT midlout EMPTY>
```

```
<!ATTLIST midlout %qlistatt;>
```

Елемент badout визначає список питань, які повинні виводитися при мінімальному рівні складності.

```
<!ELEMENT badout EMPTY>
```

```
<!ATTLIST badout %qlistatt;>
```

Атрибути списків питань

```
<!ENTITY % qlistatt "
```

```
count CDATA #IMPLIED
```

```
random (y|n) 'y'
```

```
idref CDATA #REQUIRED
```

```
">
```

idref – визначає список порядкових номерів або ідентифікаторів запитань, які повинні бути включені в даний список. Значення задається у вигляді рядка у який перераховуються ідентифікатори або порядкові номери через пропуск.

random – вказує спосіб вибору питань із списку. Значення "n" – запитання вибираються по-порядку; значення "y" – запитання вибираються випадковим чином.

count – вказує кількість запитань, які повинні бути вибрані із списку.

Значення повинне бути додатним цілим числом. Значення "0" означає, що повинні бути вибрані всі запитання із списку.

Приклад розділу секцій:

```
<selector>
```

```
<section id="a1" allout="5 4 12"/>
```

```
<section id="a2" allout="3 8 6"/>
```

```
</selector>
```

Розділ опису запитань

Елемент questions визначає розділ запитань тесту. Запитання кожного типу задається при допомозі відповідного елементу.

```
<!ELEMENT questions (single|check|number|text|box)+>
```

Елемент single визначає запитання з одиничним вибором відповіді.

```
<!ELEMENT single %Flow;>
```

```
<!ATTLIST single %questatt;>
```

Приклад написання коду запитання з одиничним вибором відповіді:

```
<single random="y" format="custom">
```

Запитання

```
<answer rating="10">Правильна відповідь</answer>
```

```
<answer rating="0">Хибна відповідь</answer>
```

```
<answer rating="0">Хибна відповідь</answer>
```

```
</single>
```

Елемент check визначає питання з множинним вибором відповіді.

```
<!ELEMENT check %Flow;>
```

```
<!ATTLIST check %questatt;>
```

Приклад написання коду запитання з множинним вибором відповіді:

```
<check random="y" format="custom">
```

Запитання

```
<answer rating="10">Правильна відповідь1</answer>
```

```
<answer rating="10">Правильна відповідь2</answer>
```

```
<answer rating="0">Хибна відповідь</answer>
```

```
<answer rating="0"> Хибна відповідь </answer>
```

```
<answer rating="10"> Правильна відповідь3</answer>
```

```
</check>
```

Елемент number визначає запитання з введенням числової відповіді.

```
<!ELEMENT number %Flow;>
```

```
<!ATTLIST number %questatt;>
```

Приклад написання коду запитання з введенням числової відповіді:

```
<number random="y" format="custom">
```

Запитання

```
<answer rating="10" right="Відповідь" />
```

```
</number>
```

Елемент text визначає питання з введенням текстової відповіді.

```
<!ELEMENT text %Flow;>
```

```
<!ATTLIST text %questatt;>
```

Приклад написання коду запитання з введенням текстової відповіді:

```
<text random="n" format="custom">
```

Запитання

```
<answer rating="10" right="Відповідь" />
```

```
</text>
```

Елемент box визначає запитання із зіставленням.

```
<!ELEMENT box %Flow;>
```

```
<!ATTLIST box %questatt;>
```

Атрибути

idref – задає ідентифікатори або порядкові номери питань, для яких призначено повідомлення. Значення задається у вигляді рядка, в якому перераховуються

ідентифікатори або порядкові номери через пропуск. Значення "all" означає, що повідомлення призначене для всіх питань.

idrating – Задає результат виконання питання, по досягненню якого повинно бути виведено дане повідомлення:

- all – будь-який результат;
- bad – мінімальний результат (кількість набраних балів за питання рівна нулю);
- middle – середній результат (кількість набраних балів за питання більше нуля але менше максимально можливого);
- good – максимальний результат (набрана максимально можлива кількість балів).

Приклад розділу повідомлень:

```
<messages>
<sectmsg idrating="all" idref="1">
<b>Закінчена перша секція</b>
</sectmsg>
<questmsg idrating="good" idref="all">
Відмінно!
</questmsg>
<questmsg idrating="middle" idref="all">
Можна краще...
</questmsg>
<questmsg idrating="bad" idref="all">
Погано
</questmsg>
</messages>
```

3. Висновок

Розроблений формат Test Standart (TST) задовільняє вимогам систем дистанційного навчання та адаптивного тестування.

У формат TST визначено:

1. Вимоги до структури і змісту файлу тесту.
2. Вимоги до мови опису тесту.

Даний формат відповідає наступним вимогам:

- відсутність жорсткої прив'язки до якого-небудь предмету;
- можливість використання питань наступних типів: "вибір однієї відповіді з багатьох", "багато з багатьох", "поле введення", "відповідність";
- адаптивний вибір наступного питання залежно від правильності попередніх відповідей студента;
- можливість створення різних завдань з одного набору питань, використання як стандарт зберігання XML.

Література

1. Федорук П.І. Адаптивні тести: статистичні методи аналізу результатів тестового контролю знань // Математичні машини і системи. – 2007. – №3,4. – С. 122-138.
2. Федорук П.І. Моделі і методи діагностики знань з використанням адаптивних тестів // УСиМ. – Київ, 2007. – №5. – С.68-76.