

Р.М. Дубан

Національний авіаційний університет

АРХІТЕКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІД- ТРИМКИ ТЕСТОВОГО КОНТРОЛЮ ЗНАТЬ «LOGIT»

Показано варіант побудови інформаційної системи підтримки тестового контролю знань. Особливість підходу у відкритій моделі формування тесту, сплайн-моделі профілів та програмно-апаратній підтримці тестування. Очікуваним результатом є отримання інструментарію, який дасть можливість супроводжувати тест на всіх етапах його розробки, до створення закінченого інструменту для вимірювання знань.

Ключові слова: інформаційна система, тест, контроль знань.

Представлен вариант построения информационной системы поддержки тестового контроля знаний. Особенность подхода в открытой модели формирования теста, сплайн-модели профилей и программно-аппаратной поддержке тестирования. Ожидаемым результатом является получение инструментария, который позволит сопровождать тест на всех этапах его разработки, до создания законченного инструмента для измерения знаний.

Ключевые слова: информационная система, тест, контроль знаний.

This paper presents a way of developing an information system to support the test control knowledge. Feature of the approach in an open model of the formation test, the spline model profiles, software and hardware support for testing. The expected result is to provide tools that will accompany the test at all stages of its development, to create a complete tool for measuring knowledge.

Key words: information system, test, control knowledge.

Вступ. Серед десяти пріоритетних національних проєктів в Україні є проєкт інноваційного підходу до навчання «Відкритий світ». Проєкт передбачає широке впровадження інформаційних технологій у навчання [1]. Важливе місце серед інформаційних освітніх технологій займає тестовий контроль.

Аналіз останніх досліджень. Відомими перевагами тестового контролю знань є об'єктивність, простота процесу перевірки, низька вартість процесу тестування, можливість дистанційного проведення. Однак створення якісних тестів є складною задачею, що потребує знач-

них ресурсів. Основні вимоги до тестів та процедура їх створення і перевірки викладені в комплексі нормативних документів МОН [2], монографіях та навчальних посібниках [3]. Питання оцінювання якості тестів розглядає IRT [4], у рамках якої найчастіше застосовують в якості моделі профілю тестового запитання функції Раша. На жаль, у переважній більшості випадків, при тестуванні застосовуються тести, що не пройшли належної перевірки і мають невизначені властивості, як інструмент вимірювання знань. Ймовірною причиною такого положення є відсутність практично зорієнтованих методів, алгоритмів і програмного забезпечення для оцінювання тестів.

Існує велика кількість програмного забезпечення стосовно тестового контролю, як комерційного так і вільного, проте переважна більшість із них охоплює лише окремі аспекти тестового контролю. Найбільш відоме та вживане вільне програмне забезпечення - це Moodle, iTest, та OpenTEST2, що дозволяють створювати бази тестів та проводити тестування з використанням комп'ютерної техніки, а в OpenTEST2 і проведення статистичного аналізу результатів. Комерційні програмні продукти забезпечують технічну підтримку у вирішенні проблем та мають покращений вигляд інтерфейсу та вдосконалений функціонал. Але сам підхід до створення тестів у цих та інших програмних засобах має закритий характер, автор чи група авторів розробляють тест і використовують його для вимірювання знань без попереднього рецензування та випробування. Інформаційні системи які охоплюють весь життєвий цикл тесту від створення до використання невідомі.

Постановка задачі. Робота розглядає задачу проектування інформаційної системи підтримки тестового контролю, що охоплює весь життєвий цикл тесту. Тест обов'язково, до його використання в якості вимірювального інструмента, має пройти стадії створення, рецензування, багаторазового випробування та вдосконалення. Лише після отримання тесту з нормованими характеристиками якості, його можна рекомендувати до використання. Така система має бути максимально відкритою і безкоштовною, створеною на засадах вільного програмного забезпечення. Основною метою її є спрощення процедур тестового контролю і одночасно підвищення їхньої якості.

Основний матеріал. Функціональна структура елементами якої є основні модулі інформаційної системи, а зв'язками між елементами виступають потоки інформації, що циркулює у системі зображена на рисунку 1.

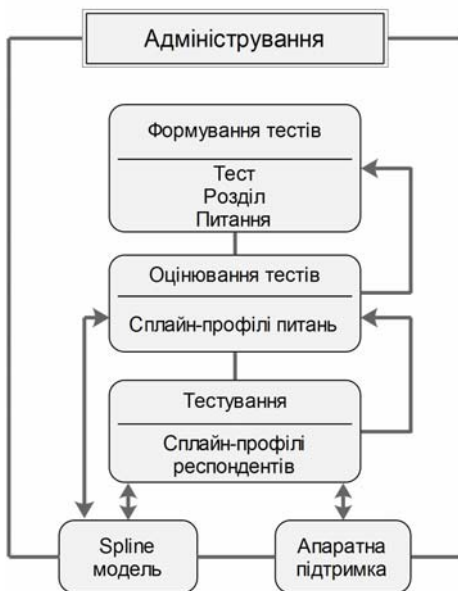


Рис. 1. Модульна структура системи

Модуль формування тестів використовується для створення нового та редагування вже існуючих тестів. Обов'язковими складовими тесту є розділи з яких він складається. Кожний розділ має містити близько п'ятдесяти питань. Кожне питання повинно містити від чотирьох до десяти варіантів відповідей з однією правильною. Після завершення наповнення тесту здійснюється його рецензування та оцінка.

Модуль оцінювання тестів використовується під час рецензування тесту та його питань, а також оцінювання характеристик якості на основі слайн-профілів питань. Процес рецензування починається разом із наповненням тесту питаннями, та не припиняється під час багаторазового вдосконалення тесту. Рецензування виконують спеціалісти у предметній області тесту на предмет коректності, правильності формулювання питань та відповідності меті. Слайн-профілі питань дають інформацію про характер якості та складності кожного питання і дозволяють оцінити характеристики якості тесту в цілому.

Модуль тестування застосовується під час випробування тесту на групах респондентів для отримання матриці результатів відповідей, а після рекомендації тесту для перевірки знань.

Spline-модель – є спеціальним математичним модулем системи, що дозволяє на базі матриці відповідей побудувати профілі питань та рес-

пондентів. Профілі будуються за емпіричними даними як сплайн-функції. Ці профілі використовуються для визначення характеристик якості тесту без значних затрат ресурсів. Тест вдосконалюється, проводяться нові випробування на групах доки не буде досягнутий достатній рівень його якості. Після того як вдосконалення тесту завершено і його можна рекомендувати використовувати для перевірки рівня знань, тоді модуль тестування забезпечує процес адаптивного тестування.

Модуль апаратної підтримки тестування забезпечить швидкість та зручність проведення тестового контролю, як на етапі випробувань на групах, так і на етапі тестування для оцінки рівня знань. Апаратна підтримка повинна мати вигляд простих й дешевих спеціальних технічних засобів, що поєднуюватимуться із комп'ютером викладача і забезпечать потік інформації від респондента до бази даних підчас тестування. Це дозволить проводити тестування в будь-якій аудиторії, а не лише комп'ютеризованій, і відразу отримувати результати.

Модуль адміністрування має доступ до всіх інших модулів, що дає змогу адміністраторам та менеджерам інформаційної системи здійснювати контроль, технічну та функціональну підтримку на всіх етапах створення та використання тесту. В модулі адміністрування здійснюється розподіл прав доступу для користувачів системи завдяки використанню списків контролю доступу.

З огляду на те, що інформаційна система підтримки тестового контролю знань має бути максимально відкритою та безкоштовною, технології що використовуються для її створення доцільно обирати серед вільного програмного забезпечення. Для забезпечення спрощеного доступу до системи планується використовувати web-технології. В якості HTTP-сервер обрано популярний Apache, що забезпечує стабільну роботу та має велику кількість налаштувань добре документованих, підтримується майже всіма операційними системами. Для створення серверного програмного коду обрано мову програмування PHP5, що є однією із лідерів серед мов web-програмування, та використовує моделі об'єктно-орієнтованого програмування. Дані, що використовуються системою, будуть зберігатись у базі даних. В якості сервера баз даних обрано MySQL5, обмін даними виконується за допомогою запитів на мові SQL. Таким чином, за допомогою обраних технологій, реалізується трьох-рівнева клієнт серверна архітектура інформаційної системи (рис. 2).

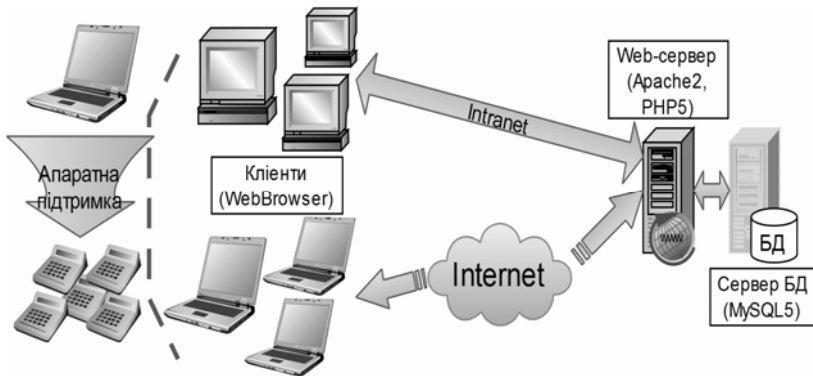


Рис. 2. Архітектура інформаційної системи

Для написання коду, що виконується на стороні сервера використовується набір бібліотек ZendFramework, що забезпечує стандарт написання коду. Завдяки використанню набору бібліотек та використання патерну MVC (модель-вигляд-контролер) - швидкість розробки підвищується, а подальша підтримка буде легшою.

Доступ до інформаційної системи здійснюється за HTTP-протоколом через локальну або глобальну Internet мережу. Для клієнтів використовувати якесь спеціальне програмне забезпечення непотрібно, достатньо звичайного сучасного браузера. Після виконання запиту сервер виконує скрипти, за необхідності звертається до бази даних, та формує HTML документ що надсилається клієнту. Разом із HTML-розміткою документа браузер клієнта отримує посилання на таблиці стилів CSS, для оформлення документа. Також для підвищення інтерактивності HTML-документу та зменшення навантаження на сервер використовується програмний код на мові JavaScript з використанням бібліотеки JQuery, що виконується на стороні клієнта. Значне зменшення навантаження на сервер досягається шляхом використання асинхронних запитів до сервера, що реалізується технологією Ajax.

На перших етапах проектування інформаційної системи та написання програмного коду з використанням об'єктно-орієнтованої моделі будуватиметься діаграма класів для визначення основних класів та їх властивостей. Укрупнена структура системи у вигляді UML діаграми класів, що відображає основні класи системи та взаємозв'язки між цими класами, зображено на рисунку 3.

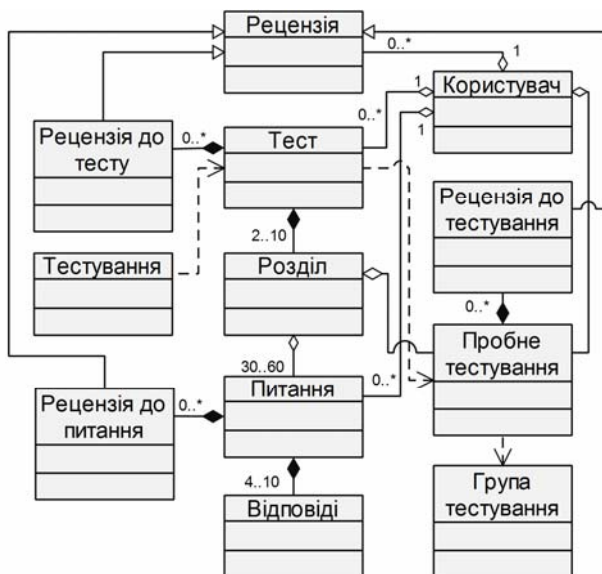


Рис. 3. Діаграма класів системи

Кожний клас інформаційної системи має характерні йому властивості та методи, що забезпечать функціонування системи. «Тест» є центральним об'єктом системи. Властивості характерні класу «тест»: область знань (наприклад, за УДК), назва тесту, дисципліна, анотація, автор тесту, адміністратор тесту, стадія, дата створення, дата редагування, якість, статус. Клас «Розділ» знаходиться із класом «Тест» у відношеннях композиції, має наступні властивості: тест з яким пов'язаний та назва розділу. Розділів в тесті має бути від двох до десяти. Між класами «Питання» та «Розділ» установлене відношення агрегації, бо питання можна переміщати в інші розділи. Властивості об'єкта класу «Питання» наступні: тест та розділ до якого він належить, дата створення, дата редагування, профіль, складність, диференціююча здатність, текст питання, статус, автор питання. Питань по кожному розділу має бути близько тридцятип'ятдесяти. У відношенні композиції до класу «Питання» знаходиться клас «Відповіді» і має наступні властивості: питання до якого належить, текст відповіді, правильність відповіді. Відповідей по кожному питанню має бути від чотирьох до десяти.

Головними суб'єктами системи є користувачі, їх властивості описані в класі «Користувач»: назва користувача, пароль, прізвище, ім'я, по батькові, e-mail адреса, ісд номер, скуре, рівень доступу, статус тестувальника, статус автора, статус рецензента, науковий ступінь, вчене звання,

вид діяльності, анкета, дата реєстрації, дата останніх змін у профілі, дата останнього візиту, статус користувача, ідентифікатор сесії, кількість спроб входу з неправильним паролем. Усі дії в системі виконуються користувачами, тому клас «Користувач» майже з усіма класами знаходиться у відношенні агрегації. Користувачі можуть мати різні набори статусів та рівнів доступу.

На всіх етапах формування тесту відбувається процес рецензування що забезпечується класами: «Рецензія до тесту», «Рецензія до питання», «Рецензія до тестування», похідними від класу «Рецензія». Властивості цих класів: об'єкт рецензування, автор рецензії, текст рецензії, дата рецензії. Рецензії мають вигляд обговорення-коментування об'єкта з яким вона пов'язана відношенням композиції.

Підчас випробування тесту тестування відбувається з використанням класу «Пробне тестування». Результати пробних тестувань дають змогу побудувати профілі питань та визначити характеристики якості тесту. Властивості класу «Пробне тестування» включають: тест та розділ за яким проводиться пробне тестування, дата проведення, тестувальник що проводив тестування, групи респондентів, результати тестування.

На результати пробних тестувань впливають характеристики груп респондентів на яких проводилось тестування. Властивості класу «Група тестування» наступні: назва групи, характеристика групи. Після закінчення вдосконалення тесту, характеристики його якості впливають на процес тестування.

У зв'язку з тим, що до процесу розробки тесту має залучатись група користувачів системи з різними правами доступу, система обов'язково повинна мати можливість авторизації користувачів та розподілу прав. Адміністратор інформаційної системи - технічний спеціаліст, що володіє знаннями про функціонування системи та виконує її обслуговування, допомагає у вирішенні питань всім користувачам, виправляє помилки виявлені в програмі та розвиває її функціонал. Менеджери – найбільш досвідчені та поважні користувачі системи, що розуміються на розробці тестів та, здебільшого, мають науковий ступінь, вся відповідальність за просування тесту на наступну стадію лежить на них. Звичайні зареєстровані користувачі інформаційної системи можуть мати певні ролі або їх комбінації, а саме: тестувальник, рецензент, автор. Менеджер має всі можливі з перерахованих ролей користувачів та може надавати їх будь якому користувачу. Гості є незареєстрованими користувачами системи, що можуть лише переглядати списки тестів та, можливо, проходити тестування. Роль тестувальника надає можливість користувачу проводити тестування на групі респондентів на етапі випробування тесту, після чого він має внести опис групи та результати. Роль рецензента

надає можливість користувачу переглядати цікаві йому тести, їх розділи та питання, залишати рецензії на них у вигляді зауважень або коментарів. Автор має змогу створювати нові тести, визначати їх розділи та наповнювати їх питаннями, публікувати чи знімати з публікації питання, відповідати на рецензії. Він має змогу запропонувати іншому автору, в цікавий йому тест, своє питання, але опублікувати це питання має змогу лише автор тесту.

Новий тест створений автором має містити тему та опис. Будь-який менеджер, краще якщо спеціаліст по темі тесту, вирішує чи потрібен такий тест і публікує його, призначивши відповідальним за контроль над тестом менеджера – «адміністратора тесту». Задача адміністратора тесту полягає у відповідальності за переведення тесту на наступну стадію «життєвого циклу». Першою стадією є «розробка», на яку тест переходить відразу після його публікації і призначення менеджером адміністратора. На першій стадії автор тесту визначає розділи та наповнює їх питаннями, інші автори можуть запропонувати свої питання до певних розділів тесту, автор тесту може їх або прийняти із збереженням авторства або відхилити, рецензенти мають переглядати питання та робити зауваження і коментарі до тесту цілому і до окремих питань. Після того, як тест має достатню кількість питань, не менш як тридцять на кожний розділ, та прийняті до уваги всі зауваження рецензентів, або подані пояснення до них, автор може зробити запит до адміністратора тесту на перехід до наступної стадії. Адміністратор тесту перевірявши виконання необхідних умов переводить тест на наступну стадію – «випробування», або висловлює свої зауваження. На етапі «випробування» проводиться тестування користувачами тестувальниками на групах респондентів у звичайному паперовому вигляді, або з використанням комп'ютерної техніки. Важливим на цьому етапі є використання усіх питань тесту, для цього зручнішим буде проведення тестування за розділами. Після отримання перших результатів тестування починається процес статистично-математичного аналізу по певним групам респондентів. З надходженням результатів наступних тестувань розрахунки будуть уточнюватись а їх достовірність збільшуватиметься. Отримані показники мають використовуватись рецензентами та самим автором для роботи над вдосконаленням тесту.

Є значна кількість наукових досліджень присвячених теорії тестування з погляду теорії ймовірності та статистики. Однією з центральних проблем досліджень є побудова адекватної моделі умовної ймовірності правильної відповіді на запитання. Найбільш відомими моделями є однопараметрична функція Г. Раша, двох параметрична та трьох параметрична функції А. Бірнбаума. Ці моделі є основою, за допомогою якої

досліджуються оцінки характеристик тесту. Практично важливою є побудова вказаних моделей за результатами тестування. Однак серйозною складністю є нелінійна залежність вказаних моделей від параметрів.

Застосування функції Раша пов'язане із труднощами оцінки параметрів, які входять до рівняння нелінійно. Вибір профілюючої функції, значною мірою, довільний і ґрунтується на інтуїції й досвіді оператора, а це утруднює автоматизацію процедури. Однак існують функції, які мають гарну адаптивну здатність у сполученні із простотою розрахунків і лінійною залежністю від параметрів - це сплайни. У практиці обробки даних найбільш відомі інтерполяційні кубічні сплайни із двома неперервними похідними. Для їхнього розрахунку необхідно вирішити систему інтерполяційних рівнянь. Зручніше користуватися кубічними ермітовими сплайнами, які мають неперервність лише першої похідної і локальні. При інтерполяції з їхньою допомогою немає необхідності вирішувати систему рівнянь. Зміст параметрів моделі співпадає із змістом вхідних даних: точок спостережень [5]. Іншим важливим аспектом моделі є компактне подання моделі профілю тестових питань. Уся інформація про моделі міститься в значеннях вузлових точок сплайна. Вірогідність моделі описується шириною довірчих інтервалів у точках стикування. Представлені алгоритми оцінювання характеристик питань досить загальні й не враховують деяких особливостей оброблюваних даних. Їхнє подальше вдосконалення полягає в застосуванні зваженого МНК і врахуванні обмежень рішення.

Оцінка профілів тестових питань вимагає значного числа респондентів з різним рівнем знань. Тому практично доцільною є ітераційна процедура оцінки характеристик тесту. Спочатку виявляють найбільш очевидні недоліки й устанавлюються загальні характеристики на деякій контрольній групі респондентів. Для прискорення уточнення характеристик тесту й зниження вартості цієї процедури доцільно залучити до процесу широке коло тестувальників. Процедура аналогічна бета тестуванню програмного забезпечення.

При досягненні необхідного рівня вірогідності оцінок профілів тестових питань виконується оптимізація тесту (наприклад, вилучення малоінформативних питань і питань зі статистично близькими профілями). Формується паспорт тесту, що включає профілі питань. Наявність цих характеристик дає можливість більш об'єктивно оцінювати рівень знань респондентів. Із цього моменту тест можна вважати завершеним інформаційним продуктом з нормованими характеристиками. Тест переводиться на стадію «застосування», а зміни до нього не вносяться, за винятком умов, що потребують перегляду тесту. Тобто, одержуємо своєрідний вимірювальний прилад, що вимірює знання у певній області.

Висновки. Такий підхід до формування, випробування й застосування тестів здатен гарантувати необхідний (або заданий) рівень вірогідності тестування, перетворивши тест у вимірювальний інструмент із нормованими характеристиками щодо похибок. Забезпечення функціональних можливостей для цього процесу планується реалізувати на веб-платформі, що дозволить оперативно обмінюватись інформацією між учасниками. Результати поточних розробок інформаційної системи «Logit» подані на сайті за адресою <http://kdpu.edu.ua/logit/>.

Бібліографічні посилання

1. **Семиноженко В.** Національний проект «Відкритий світ» – це інноваційні підходи до навчання / В. Семиноженко // Прес-служба Державного Агентства України з питань науки, інновацій та інформації. http://www.kmu.gov.ua/control/publish/article?art_id=244099166
2. Комплекс нормативних документів для розробки складових систем стандартів вищої освіти. Додаток 1 до наказу Міносвіти N 285 від 31 липня 1998 р. – К. 1998. – 115 с.
3. **Чельшкова М. Б.** Теория и практика конструирования педагогических тестов : учебное пособие / М. Б. Чельшкова. – М., 2002. – 432 с.
4. **Baker F. B.** The Basics of Item Response Theory. – ERIC, 2001. – 172 p
5. Сплайни в цифровій обробці даних і сигналів / І. В. Шелевицький, М. О. Шутко, В. М. Шутко, О. О. Колганова. – Кривий Ріг, 2008. – 232 с.

Надійшла до редколегії 19.05.11