

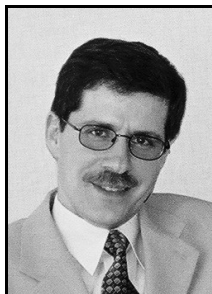
**Artur GRAMACKI, Jacek BIEGANOWSKI**  
 UNIwersytet ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

## Koncepcja systemu komputerowego wspomagającego przeprowadzanie testów wyboru

Dr inż. Artur GRAMACKI

Pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku adiunkta. Interesuje się zagadnieniami związanymi z bazami danych, w szczególności firmy Oracle ale też rozwiązaniami Open Source. Oprócz prowadzenia zajęć dydaktycznych stara się wykorzystywać swoją wiedzę uczestnicząc w różnych projektach informatycznych z tego zakresu. Brał udział w czterech projektach, których celem było przygotowanie systemów wspomagających działalność UZ.

e-mail: a.gramacki@iie.uz.zgora.pl



Mgr inż. Jacek BIEGANOWSKI

Ukończył informatykę na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. Od 2004 roku pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku asystenta. Jego zainteresowania związane są z programowaniem, systemami operacyjnymi, techniką cyfrową oraz algorytmami ewolucyjnymi. W wolnych chwilach lubi słuchać muzyki filmowej oraz jeździć na rowerze.

e-mail: j.bieganowski@iie.uz.zgora.pl



### Streszczenie

W artykule przedstawiono koncepcję systemu komputerowego wspomagającego przeprowadzanie testów wyboru z dowolnej dziedziny. System jest tak zaprojektowany, że umożliwia przeprowadzanie testów zarówno w warunkach laboratoryjnych (gdzie dostępne są komputery), jak i w typowych wieloosobowych salach wykładowych, gdzie studenci rozwiązują test w postaci tradycyjnej (papierowej). W pierwszym przypadku test można przeprowadzić praktycznie całkowicie automatycznie (poza oczywiście przygotowaniem pytań, które to zadanie z oczywistych względów nie poddaje się automatyzacji). W drugim przypadku, dzięki odpowiednim rozwiązaniom, można do minimum uprościć najbardziej czasochłonne i żmudne etapy (głównie drukowanie kart egzaminacyjnych oraz ich sprawdzanie).

**Słowa kluczowe:** test wyboru, egzamin, zaliczenie, aplikacja bazodanowa, XML, LaTeX, OMR.

### The concept of a computer system for carrying out test exams

#### Abstract

The article presents the concept of a computer system for carrying out test exams of any discipline. Such a system is designed to support tests in small laboratory rooms (using computers) as well as in large lecture rooms (using paper forms). In the first case tests can be carried out almost automatically (of course all questions must be prepared manually). In the second case preparing and printing exam sheets and checking answers is much easier corresponding to traditional (manual) way.

**Keywords:** test exam, credits, database system, XML, LaTeX, OMR.

### 1. Wstęp

W artykule przedstawiono koncepcję systemu komputerowego wspomagającego przeprowadzanie testów wyboru z dowolnej dziedziny. Zagadnienie to pozornie wydaje się mało ciekawe. Znane są bowiem mniej lub bardziej udane implementacje systemów komputerowych (zwykle korzystających z jakiegoś systemu baz danych) wspomagających nauczyciela w dość żmudnym i monotonnym procesie przeprowadzania testów.

Wśród komercyjnych rozwiązań tego typu wyróżnić należy produkty firmy Gravic [1] oraz jej oddziału Principia Products [2]. Współpracują one z dedykowanymi skanerami kart egzaminacyjnych. Znany producentem tego typu urządzeń jest firma Sekonic [3]. Rozwiązania te, choć zaawansowane i o dużych możliwościach, są jednak dość drogie (licencja na jednego użytkownika to koszt ok. 800-1000 USD). Jeżeli dodamy do tego cenę specjalizowanego skanera (kolejne 1000 USD) koszty będą naprawdę znaczne. Należy również zauważyć, że wspomniane tu rozwiązania komercyjne są nieprzystosowane do specyfiki polskich studentów i uczniów, dla których ściąganie jest ciągle powszechnie akceptowanym zachowaniem.

Autrzy postawili więc sobie za cel zaprojektowanie podobnego systemu, który miałby zadawalającą funkcjonalność, był łatwy w użyciu, nie wymagał do pracy żadnego dodatkowego komercyjnego sprzętu ani oprogramowania. Ponadto powinien umożliwiać pracę wieloosobową, być przystosowanym do polskich realiów (patrz wyżej) oraz, co najważniejsze, powinien wspierać przeprowadzanie testów w tradycyjnej "papierowej" formie jak i z wykorzystaniem komputerów.

W przypadku, gdy możliwe jest korzystanie z komputerów odpowiednio zbudowana aplikacja generuje pytania, rejestruje udzielane przez studenta odpowiedzi, porównuje je z odpowiedziami wzorcowymi i na końcu automatycznie ocenia test. Oczywiście aby możliwe skutecznie wyeliminować wszechobecne w polskich realiach ściąganie, każdy student widzi na monitorze pytania w odpowiedni sposób przemieszane (kolejność pojawiania się pytań oraz odpowiedzi do wyboru jest losowa). Dobrze jest również, jeżeli odpowiedzi udzielane przez studenta są w jakiś sposób archiwizowane.

W dobie powszechnego dostępu do internetu wydaje się, że jedynym sensownym rozwiązaniem jest aplikacja działająca wyłącznie w środowisku przeglądarki internetowej. Instalowanie bowiem aplikacji klienckich na poszczególnych komputerach jest zajęciem czasochłonnym i nieefektywnym a sama aplikacja jest w gruncie rzeczy bardzo prosta i z powodzeniem działać może w przeglądarce internetowej.

W tak działającym systemie rola nauczyciela sprowadza się w zasadzie do wykonania trzech czynności:

- wprowadzenie zestawów testowych z ustaloną punktacją,
- wprowadzenie danych dotyczących studentów (imię, nazwisko, grupa, numer indeksu),
- zadanie wybranego zestawu testowego wybranym studentom.

Aplikacja o takiej właśnie funkcjonalności jest dość prostym systemem bazodanowym (idealnie nadaje się do tego np. baza MySQL) z interfejsem użytkownika zaprogramowanym w jakimkolwiek języku wspierającym internet (np. PHP, Java).

Problem staje się bardziej złożony, gdy naszym celem jest również system wspierający przeprowadzanie testów w tradycyjnej (papierowej) postaci. Wówczas oczywiście nie możemy korzystać z wszelkich dobrodziejstw jakie dają komputery. Oprócz trzech wymienionych powyżej czynności musimy jeszcze rozwiązać dwa bardzo ważne problemy:

- automatyczne tworzenie gotowych do wydruku arkuszy egzaminacyjnych,
- opracowanie mechanizmów umożliwiających "bezpieczne" sprawdzanie rozwiązanych testów bezpośrednio przez osoby egzaminowane (oszczędność czasu, mniejsze prawdopodobieństwo pomyłek).

Sposób rozwiązania tych problemów podano w dalszej części artykułu.

## 2. Projekt systemu

Podstawowym założeniem jest, aby omawiany system zbudować wyłącznie w oparciu o darmowe i powszechnie dostępne oprogramowanie (Java, Hibernate, XML, LaTeX, MySQL). System ma budowę modułową. Wszystkie istotne do działania dane przechowywane są w relacyjnej bazie danych. Ze względu na użytą technologię (Java oraz środowisko Hibernate) nie jest specjalnie istotne jaki dokładnie będzie to system RDBMS. Ilość przechowywanych danych oraz ich dynamika zmian nie są zbyt wielkie, więc parametry wydajnościowe bazy nie są tu problemem najistotniejszym.

System składa się z siedmiu głównych modułów:

- moduł studenta,
- moduł egzaminatora,
- moduł administratora,
- moduł bazy danych,
- moduł tworzenia arkuszy egzaminacyjnych,
- moduł sprawdzający testy,
- moduł skanujący.

Poniżej bardzo skrótowo omówiono cztery pierwsze moduły. Ich obecność w kompletnym systemie jest oczywista. Omówiono je bardzo pobieżnie, gdyż w kontekście artykułu nie są one najważniejsze. Poza tym są to elementy dość typowe i w gruncie rzeczy proste do wykonania. Trzy ostatnie moduły, stanowiące meritum pracy, omówione są w kolejnych rozdziałach.

Wszystkie moduły działają jako aplikacje internetowe bądź uruchamiane za pośrednictwem internetu. Dzięki temu nie ma potrzeby instalowania na komputerach jakiegokolwiek oprogramowania.

Moduł studenta używany jest przez osoby egzaminowane. Po prawidłowym zalogowaniu się student rozwiązuje w zadanym czasie test. Po zakończeniu test jest automatycznie oceniany. Zaimplementowano mechanizm uniemożliwiający uruchomienie więcej niż jednej instancji aplikacji przez danego studenta w tym samym czasie.

Moduł egzaminatora używany jest przez nauczycieli. Mają oni możliwość tworzenia w nim dowolnych testów i zadawania ich dowolnej grupie studentów.

Zadaniem modułu administratora jest tworzenie kont dostępnych dla nauczycieli, którzy chcą rozpocząć używanie systemu. Każda taka osoba otrzymuje w bazie danych swój własny schemat relacyjny (tabele bazodanowe) wraz z wymaganymi do rozpoczęcia pracy danymi startowymi. Schematy poszczególnych użytkowników są całkowicie od siebie niezależne i w żaden sposób nie powiązane. Dzięki takiemu rozwiązaniu w wielkim stopniu upraszcza się zarządzanie systemem jako całością.

Moduł MBD tworzy zainstalowany oraz skonfigurowany wybrany system RDBMS wraz z wymaganą infrastrukturą sieciową (np. ODBC/JDBC). W testowej aplikacji jest to baza MySQL (choć jak już wspomniano wyżej może to być też inny system wspierany przez środowisko Hibernate).

## 3. Moduł tworzenia arkuszy egzaminacyjnych

Zadaniem omawianego w tym rozdziale modułu jest maksymalne zautomatyzowanie procesu tworzenia arkuszy egzaminacyjnych, które studenci otrzymują przed rozpoczęciem rozwiązywania testu. Moduł ten uruchamiany jest za pośrednictwem modułu ME, jednak może on z powodzeniem działać jako niezależna aplikacja. Efektem jego działania jest automatyczne utworzenie wspomnianych arkuszy egzaminacyjnych zapisanych na dysku komputera w postaci gotowego do wydruku (jednego) pliku pdf.

Jeden przykładowy arkusz (już rozwiązany) pokazano w uproszczonej formie na rysunku 1. Widać na nim cztery pytania – dwa są typu “1 z N” a dwa “N z M”. Zakreślone kółka i kwadraty oznaczają wybranie przez studenta danej odpowiedzi (jednej lub kilku w zależności od rodzaju pytania).

Ponadto, czego nie pokazano na rysunku, jako treść pytania oraz treści odpowiedzi mogą pojawić się pliki graficzne (w najbardziej popularnych formatach, a więc np. bmp, gif, jpg, ps, eps).

Wydaje się, że dzięki temu krąg ew. użytkowników systemu może być znacznie powiększony. Każdy arkusz zawiera te same pytania (wszyscy studenci muszą mieć równe szanse), jednak na każdym arkuszu kolejność pytań oraz kolejność odpowiedzi jest inna. Każdy arkusz otrzymuje losowo wygenerowany numer (tu numer 5433). Ponadto każdej odpowiedzi przypisany zostaje pewien losowo wybrany unikalny numer kontrolny (liczby w nawiasach) z przedziału od 1 do N, gdzie N jest całkowitą ilością odpowiedzi w teście. W przykładowym arkuszu wszystkich odpowiedzi jest 16. Numery te pełnią kluczową rolę przy “bezpiecznym” sprawdzaniu rozwiązanych testów bezpośrednio przez osoby egzaminowane. Jest o tym mowa poniżej.

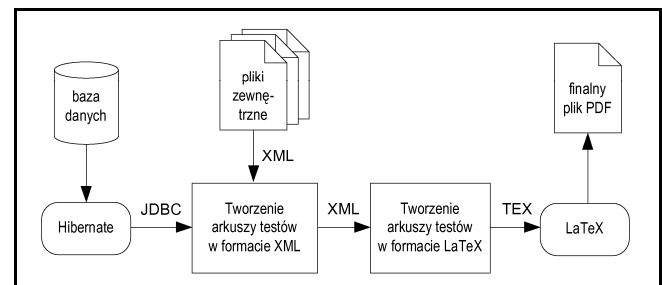
Jan Testowy, 201IDM		Nr testu: 5433	
imię i nazwisko, grupa		Hasło: xvt3	
1. Treść pytania pierwszego		3. Treść pytania trzeciego	
01 <input checked="" type="checkbox"/> odpowiedź 1		05 <input checked="" type="checkbox"/> odpowiedź 1	
07 <input checked="" type="checkbox"/> odpowiedź 2		09 <input checked="" type="checkbox"/> odpowiedź 2	
03 <input type="checkbox"/> odpowiedź 3		13 <input checked="" type="checkbox"/> odpowiedź 3	
11 <input type="checkbox"/> odpowiedź 4		06 <input type="checkbox"/> odpowiedź 4	
2. Treść pytania drugiego		4. Treść pytania czwartego	
08 <input checked="" type="checkbox"/> odpowiedź 1		06 <input type="checkbox"/> odpowiedź 1	
04 <input type="checkbox"/> odpowiedź 2		10 <input checked="" type="checkbox"/> odpowiedź 2	
15 <input type="checkbox"/> odpowiedź 3		12 <input type="checkbox"/> odpowiedź 3	
02 <input type="checkbox"/> odpowiedź 4		14 <input type="checkbox"/> odpowiedź 4	

Rys. 1. Przykładowy arkusz egzaminacyjny  
Fig. 1. A sample test exam sheet

Omawiany w tym miejscu moduł składa się z kilku mniejszych podmodułów, pokazanych schematycznie na rysunku 2. Zdecydowano się na takie a nie inne rozwiązanie, aby zapewnić maksymalne wsparcie dla bardzo popularnych obecnie plików XML. W praktyce oznacza to, że system jest w stanie zaimportować test, który został utworzony zupełnie niezależnie – nawet poza omawianą aplikacją. Pliki te mogą też służyć do wymienia się pytaniami przez nauczycieli.

Na podstawie tych wejściowych plików XML tworzone są arkusze egzaminacyjne dla wszystkich studentów rozwiązujących dany test. Tu również formatem plików jest XML. Powstaje jeden plik XML z opisem wszystkich arkuszy.

Następnie plik ten podlega konwersji do formatu stosowanego w systemie składu tekstu LaTeX (pliki .tex). Jest to również format czysto tekstowy, więc stosunkowo łatwo jest taką konwersję przeprowadzić. System LaTeX wybrano ze względu na jego doskonałe cechy użytkowe.



Rys. 2. Moduł tworzenia arkuszy egzaminacyjnych  
Fig. 2. Module for creating test exam sheets

W ostatnim kroku plik .tex jest automatycznie kompilowany do finalnego pliku w formacie pdf. Praktycznie każda dystrybucja systemu LaTeX (np. TeX Live, MikTeX) posiada standardowo instalowany program (programy) do konwersji plików źródłowych LaTeX-a do formatu pdf.

#### 4. Moduł sprawdzający testy

Po rozwiązaniu testów należy je oczywiście sprawdzić. W przypadku, gdy studenci rozwiązują test przy komputerze (korzystając z modułu studenta – patrz rozdział 2) sprawa jest dość prosta. System komputerowy jest w stanie całkowicie automatycznie sprawdzić test, zarchiwizować udzielone przez studenta odpowiedzi oraz wyświetlić jego wynik (ilość zdobytych punktów).

Problem staje się bardziej złożony, gdy testy rozwiązywane są z wykorzystaniem tradycyjnych (papierowych) arkuszy odpowiedzi. Taka sytuacja będzie w praktyce występowała bardzo często, gdyż nierzadko należy przeprowadzić test (np. w charakterze egzaminu) dla dużych grup studentów (100 lub więcej osób). Praktycznie żadna polska uczelnia lub szkoła nie dysponuje salami z taką liczbą komputerów dostępnych w jednym miejscu. Forma papierowa jest więc koniecznością. Niestety sprawdzanie i opracowanie wyników takich testów przez nauczyciela jest jak wiadomo pracą bardzo żmudną i czasochłonną.

Postanowiono więc zaprojektować system w taki sposób, aby umożliwić zautomatyzowane sprawdzanie i opracowanie wyników testów. Możliwe są dwa scenariusze: sprawdzanie testów przez osoby egzaminowane lub sprawdzanie testów za pomocą odpowiedniego czytnika kart odpowiedzi (ang. *Optical Mark Recognition* – OMR [4]).

##### Scenariusz 1:

1. student otrzymuje arkusz egzaminacyjny (patrz rysunek 2), oraz dwa egzemplarze arkusza odpowiedzi (patrz rysunek 3 oraz opis niżej),
2. student rozwiązuje test w zadanym czasie,
3. student po rozwiązaniu testu wypełnia arkusz odpowiedzi. Zakreśla kratki odpowiadające numerom kontrolnym tych odpowiedzi, które zaznaczył na arkuszu egzaminacyjnym (w przypadku testu z rysunku 1 są to liczby: 1, 5, 7, 8, 9, 10, 13),
4. student po zakończeniu testu oddaje nauczycielowi arkusz egzaminacyjny oraz arkusz odpowiedzi (kopię arkusza odpowiedzi zabiera ze sobą),
5. nauczyciel udostępnia studentom dowolną salę wyposażoną w komputery podłączone do internetu,
6. studenci, mając ze sobą wspomniany wcześniej arkusz odpowiedzi, uruchamiają moduł studenta (patrz rozdział 2) i po zalogowaniu się (login i hasło należy przepisać na arkusz odpowiedzi z arkusza egzaminacyjnego) przenoszą zakreślone liczby do systemu komputerowego,
7. po zakończeniu wprowadzania danych system komputerowy wyświetla wynik testu,
8. student spotyka się z nauczycielem i otrzymuje adekwatną do uzyskanego wyniku ocenę.

##### Scenariusz 2:

1. student otrzymuje arkusz egzaminacyjny (patrz rysunek 2), oraz jeden egzemplarz arkusza odpowiedzi (patrz rysunek 3 oraz opis niżej),
2. student rozwiązuje test w zadanym czasie,
3. student po rozwiązaniu testu wypełnia arkusz odpowiedzi. Zakreśla kratki odpowiadające numerom kontrolnym tych odpowiedzi, które zaznaczył na arkuszu egzaminacyjnym (w przypadku testu z rysunku 1 są to liczby: 1, 5, 7, 8, 9, 10, 13),
4. student po zakończeniu testu oddaje nauczycielowi arkusz egzaminacyjny oraz arkusz odpowiedzi,
5. nauczyciel sprawdza arkusze odpowiedzi za pomocą czytnika OMR (lub ręcznie, jeżeli takiego nie posiada),
6. student spotyka się z nauczycielem i otrzymuje adekwatną do uzyskanego wyniku ocenę.

Numery kontrolne na każdym arkuszu egzaminacyjnym są ustawiane losowo (choć są to zawsze liczby z tego samego przedziału od 1 do N). Student nie ma więc możliwości odkrycia reguły rządzącej przydzielaniem tych liczb. Jedyne błędy jakie

mogą się pojawić to klasyczne pomyłki, które jednak w sytuacjach spornych mogą zostać łatwo zweryfikowane (nauczyciel ma oryginalne arkusze egzaminacyjne).

Student może oczywiście próbować wpisywać do komputera inne liczby niż te zapisane w arkuszu odpowiedzi. Takie działanie ma szansę powodzenia jedynie wtedy, gdy studentowi uda się wykonać kopię arkusza egzaminacyjnego np. za pomocą aparatu fotograficznego wbudowanego w telefon komórkowy. Nauczyciel może jednak w każdej chwili zweryfikować odpowiedzi studenta, ponieważ posiada arkusz egzaminacyjny oraz kopię arkusza odpowiedzi.

Jednostki przeprowadzające dużo testów mogą do sprawdzania wykorzystywać czytnik OMR. Zaletą takiego rozwiązania jest wyeliminowanie pomyłek związanych z ręcznym wprowadzaniem odpowiedzi do systemu. Nie ma też wtedy możliwości nieuprawnionego poprawienia wyniku testu po jego zakończeniu. Wadą tego rozwiązania jest jednak dość wysoki koszt czytnika (około 1000 USD). Czytnik OMR należy wtedy podłączyć do komputera z dostępem do internetu, a odpowiednie oprogramowanie zapewniłoby łączność z modułem bazy danych.

Można również rozważyć samodzielne wykonanie prostego czytnika optycznego. Wydaje się, że z technicznego punktu widzenia nie powinno to być zbyt trudne zadanie. Odpowiednio zaprojektowana, wykonana i oprogramowana matryca z elementów optycznych powinna umożliwiać bezbłędne odczytywanie arkuszy odpowiedzi.

Rys. 3. Przykładowy arkusz odpowiedzi

Fig. 3. A sample answer sheet

#### 5. Podsumowanie

Testy wyboru są popularnym sposobem sprawdzania wiedzy. Przedstawiona w artykule koncepcja przeprowadzania testów wyboru w formie papierowej pozwala wyeliminować powszechne wśród studentów ściąganie (pytania są mieszane losowo), a przy okazji daje możliwość zautomatyzowania czasochłonnego etapu sprawdzania i opracowywania wyników. Zebrane przez system odpowiedzi można również wykorzystać do przygotowania raportów statystycznych dla nauczyciela np. które pytania sprawiają największe trudności lub jaki materiał nie został poprawnie opanowany przez studentów.

Oprócz korzyści po stronie nauczyciela, prezentowane rozwiązanie, jest też korzystne dla studentów, ponieważ czas oczekiwania na wyniki ulega skróceniu w porównaniu do tradycyjnej metody.

#### 6. Literatura

- [1] <http://www.gravic.com/>
- [2] <http://www.principiaproducts.com>
- [3] <http://www.sekonic.co.jp/English/>
- [4] <http://www.omrsolutions.com/>