

Komputerowe metody testowania wiedzy

3 maja 2003

Spis treści

Podziękowania	5
Wstęp	6
1 Klasyfikacja komputerowych systemów wspomagających weryfikację wiedzy	8
1.1 Podstawy rozwoju komputerowych systemów wspomagających edukację	8
1.2 Rodzaje technologii wspierających edukację – definicje	10
1.3 Klasyfikacja systemów CBA z punktu widzenia technologii i architektury	13
2 Teoretyczne podstawy metod testowania wiedzy	16
2.1 Pomiar dydaktyczny – teoretyczna podstawa testowania	16
2.1.1 Zadania otwarte	17
2.1.2 Zadania zamknięte	22
2.1.3 Podsumowanie	25
2.2 Formy pytań możliwe do wykorzystania w aplikacjach egzaminujących	25
2.3 Porównanie efektywności egzaminowania za pomocą komputera z egzaminami tradycyjnymi	26
3 Moduł testujący komercyjnego systemu szkoleniowego – Oracle iLearning	31

3.1	Wprowadzenie	31
3.2	Zakres funkcjonalności modułu testującego	31
3.3	Scenariusz przykładowego egzaminu komputerowego	33
3.4	Technologia modułu testującego	38
3.5	Podsumowanie	38
4	Profesjonalny system do przeprowadzania egzaminów certyfikacyjnych	
	– system Prometric	40
4.1	Informacje ogólne o systemie	40
4.2	Zakres funkcjonalności systemu	41
4.3	Scenariusz egzaminu	42
4.4	Wymagania sprzętowe centrum egzaminowania	43
4.5	Podsumowanie	45
5	Komputerowy System Egzaminowania (KSE) – opracowany w SGH	46
5.1	Cele projektu KSE	46
5.2	Historia powstania projektu KSE	49
5.3	Zakres funkcjonalności systemu KSE	50
5.4	Architektura aplikacji wchodzących w skład projektu KSE	54
5.4.1	System Zarządzania Bazą Danych	55
5.4.2	Panel administracyjny – serwer aplikacji	55
5.4.3	Serwer sesji	56
5.4.4	Oprogramowanie stanowiska egzaminującego	58
5.5	Przykładowy scenariusz przeprowadzania egzaminu z wykorzystaniem KSE	59
5.5.1	ETAP 1	60
5.5.2	ETAP 2	60
5.5.3	ETAP 3	63
5.5.4	ETAP 4	63

5.5.5	ETAP 5	65
5.5.6	ETAP 6	65
5.5.7	ETAP 7	67
5.5.8	ETAP 8	67
5.5.9	ETAP 9	68
5.6	Nowe perspektywy badawcze	68
6	Systemy weryfikacji wiedzy analizujące wypowiedzi otwarte	70
6.1	Wprowadzenie	70
6.2	Atrybuty systemu analizującego tekst otwarty	72
6.3	Project Essay Grade – pierwszy system analizujący wypowiedzi otwarte	73
6.4	Intelligent Essay Assessor – przykład implementacji metody LSA do oceny wypowiedzi otwartych	75
6.4.1	Ogólny opis działania metody LSA	76
6.4.2	Singular Value Decomposition jako technika redukcji liczby wymiarów przestrzeni semantycznej	78
6.4.3	Sposób wykorzystania metody LSA w aplikacji IEA	80
6.5	Podstawy metodologiczne budowy systemów analizujących wypowiedzi otwarte opartych na osiągnięciach językoznawstwa komputerowego	82
6.5.1	Segmentacja	83
6.5.2	Analiza morfologiczna	83
6.5.3	Analiza syntaktyczna	85
6.5.4	Analiza semantyczna	85
6.5.5	Model przestrzeni wektorowej	86
6.6	System Electronic Essay Rater – przykład implementacji wykorzystującej osiągnięcia językoznawstwa komputerowego	90
6.6.1	Analiza struktury syntaktycznej	91
6.6.2	Analiza struktury retorycznej	91
6.6.3	Analiza zawartości tematycznej	92

6.6.4 Ocena końcowa	93
6.7 Perspektywy systemu analizującego wypowiedzi otwarte w języku polskim . . .	94
Zakończenie	96
Bibliografia	97
Spis rysunków	104

Podziękowania

Pragnę złożyć podziękowania na ręce osób, których pomoc okazała się dla mnie szczególnie ważna. Przede wszystkim chciałbym podziękować Pani dr Annie Rokickiej-Broniatowskiej, promotorowi, bez której zainteresowania i wsparcia oraz naukowych konsultacji nie byłoby możliwe powstanie tej pracy oraz rozwijanie projektu Komputerowego Systemu Egzaminowania (KSE), który opisuję w jednym z rozdziałów. Do podjęcia trudu stworzenia tego systemu zainspirował nas Pan dr Andrzej Kobyliński, któremu z tego tytułu należą się podziękowania. Nie mógłbym oczywiście nie wspomnieć kolegów Szymona Pury oraz Krzysztofa Darmetko, z którymi ten projekt rozwijam. Z całą pewnością nie byłbym w stanie opisać działania systemu certyfikującego firmy Prometric bez pomocy Pani Barbary Baszuń, administratora jednego z warszawskich Autoryzowanych Centrów Egzaminacyjnych Prometric. Osobą, której należą się ogromne podziękowania jest też Pan dr Adam Przepiórkowski, którego konsultacje umożliwiły mi zajęcie się tematyką podstaw metodologicznych systemów analizujących wypowiedzi otwarte, wykorzystujących dorobek językoznawstwa komputerowego. Na końcu pragnę podziękować rodzicom oraz przyjaciółom, którzy w trakcie powstawania niniejszej pracy zmuszeni byli wysłuchiwać moich wymówek oraz narzeków.

Wstęp

„I have learned to use the word 'impossible' with the greatest caution.”

/ Wernher von Braun /¹

Do podjęcia problematyki systemów wspomagających weryfikację wiedzy przyczyniła się sytuacja, która zaistniała nieco ponad rok temu w Katedrze Informatyki Gospodarczej SGH. Zaczęto wówczas myśleć o odciążeniu wykładowców od mechanicznej pracy, jaką muszą oni wkladać w przygotowanie i przeprowadzanie egzaminów dla kolejnych roczników studentów SGH. Rezultatem tych przemyśleń było rozpoczęcie przez trzech studentów stażystów prac nad systemem, który nadawałby się do wdrożenia w warunkach naszej uczelni. Od początku zaangażowałem się w ten projekt, którego efekty chciałbym przedstawić poniżej.

W początkowej części pracy zarysowane zostały przyczyny rozwoju aplikacji wspomagających weryfikację wiedzy. Rozdział pierwszy zawiera ponadto definicje typów komputerowych systemów egzaminujących oraz klasyfikację z punktu widzenia ich budowy. Następny rozdział opisuje teoretyczno-metodologiczną podstawę egzaminowania – stanowi więc bazę dla wszystkich, opisywanych w dalszych częściach pracy, implementacji. W kolejnych rozdziałach starałem się pokazać, jak działają systemy testujące, należące do dwóch kategorii – po

¹Wernher von Braun (1912-77), niemiecki specjalista techniki raketowej i kosmicznej. Od 1930 roku do końca II wojny światowej aktywnie uczestniczył w badaniach nad konstrukcją rakiet militarnych w Niemczech. Od 1945 roku pracował w USA m.in. dla NASA, gdzie wydatnie przyczynił się do opracowania serii amerykańskich pocisków balistycznych oraz układów napędowych do statków kosmicznych. Kierował m.in. opracowaniem napędu raketowego Jupiter C, za pomocą którego w 1958 roku został wyniesiony na orbitę pierwszy amerykański satelita Ziemi. Później był także członkiem kierownictwa programu kosmicznego Apollo [ENCY95]. Dzięki jego badaniom nad trajektorią lotu rakiety kosmicznej stało się możliwe zrealizowanie programu załogowych lotów na Księżyc, Apollo. Cytat pochodzi ze zbiorów internetowych, dostępnych pod adresem: <http://www.quotationspage.com>

pierwsze, do kategorii systemów ukierunkowanych na prowadzenie szkoleń oraz, po drugie, do kategorii aplikacji, wspomagających przeprowadzanie egzaminów certyfikatowych. W dalszej części – w ramach przykładu dużego systemu, przeznaczonego do szkolenia pracowników, przybliżona została specyfika modułu testującego wiedzę, będącego częścią produktu firmy Oracle o nazwie iLearning. Natomiast w rozdziale czwartym, przedstawiono działanie centrów egzaminujących firmy Prometric. Istnienie tego typu centrów egzaminacyjnych na świecie jest sygnałem upowszechniania idei weryfikacji różnego rodzaju kompetencji, wiedzy akademickiej i zawodowej w oparciu o metody komputerowe. Tendencja ta stanowi jedną z zalecanych przez Komisję Europejską dróg ku społeczeństwu informacyjnemu². Stąd wydaje się niezwykle ciekawe zwrócenie uwagi na metodologię i rozwiązania technologiczne w takich, swego rodzaju globalnych, systemach testowania wiedzy.

W rozdziale piątym przedstawiono nieco węższe podejście, wychodzące naprzeciw oczekiwaniom lokalnych systemów testowania wiedzy, w ramach ośrodków akademickich lub wewnętrznych systemów egzaminowania kompetencji pracowniczych, w ramach różnych instytucji. Istotą tego podejścia stanowi Komputerowy System Egzaminowania (KSE) – projekt, którego jestem współautorem i który należy do aplikacji wspomagających testowanie w lokalnym Uczelnianym Centrum Egzaminowania.

Ostatnia część mojej pracy poświęcona jest najbardziej zaawansowanym systemom wspomagającym weryfikację wiedzy, mianowicie aplikacjom, potrafiącym oceniać wypowiedzi otwarte.

Mam nadzieję, że przedstawione w pracy zagadnienia, będące wynikiem różnych przemysłów, poszukiwań badawczych i prób ich implementowania, przyczynią się do upowszechnienia nowych technologii informacyjnych w ramach systemów weryfikacji wiedzy.

²Zalecenia te zawarto w raporcie „Biała Księga Kształcenia i Doskonalenia. Nauczanie i uczenie się. Na drodze do uczącego się społeczeństwa” [BIAL97]

Rozdział 1

Klasyfikacja komputerowych systemów wspomagających weryfikację wiedzy

1.1 Podstawy rozwoju komputerowych systemów wspomagających edukację

„Ipsa Scientia Potestas Est”

/ Sir Francis Bacon /¹

Świat, w którym żyjemy, zdaje się kręcić z dnia na dzień coraz szybciej. Każdego poranka dowiadujemy się o nowych odkryciach, o nowych trendach w modzie, kulturze a nawet nauce. Coraz częściej prawdy, które do niedawna uważane były za nienaruszalne, są podważane lub negowane. Zmienia się także nasze życie zawodowe – to gdzie pracujemy, czym dokładnie się zajmujemy, jakich narzędzi używamy podczas pracy. Coraz częściej naszym biurem staje się nasz dom. Dzięki rozwojowi technik telekomunikacji, a zwłaszcza Internetu, wszystko czego nam potrzeba do pracy staje się mobilne, dostępne z dowolnego punktu na ziemi, podłączonego do globalnej sieci. „Zmierzamy w kierunku tworzenia społeczeństwa informacyjnego, którego (...) jedną z cech jest (...) powstawanie ’technologii intelektualnych’” [NAJP03]. Nowe

¹ „Wiedza jest władzą” (tłum. autora) – Religious Meditations, Of Heresies, 1597

narzędzia, jakimi się posługujemy, wymagają ciągłego doksztalcania się. Dzisiaj nie tylko „czas to pieniądz” ale także „informacja to pieniądz”. Właściwa informacja stanowi podstawę wiedzy, bez której nie można się rozwijać, nie da się podążać za światem. Każdego dnia uczymy się czegoś nowego. Wszystko to sprawia, że stara filozoficzna prawda mówiąca o tym, że człowiek uczy się przez całe życie, staje się nam coraz bliższa. Coraz częściej mówi się o (...) „przygotowaniu absolwentów do samokształcenia i uczestniczenia w procesie edukacji obejmującej całe życie” [NAJP03]. Zmieniająca się gospodarka ciągle się restrukturyzuje, niektóre zawody giną bezpowrotnie, rodzą się nowe. Ogromne rzesze ludzi są zmuszane do zmiany swoich kwalifikacji, lub do zdobywania nowych, które nagle stają się konieczne do wykonywania swoich dawnych zawodów. Jeszcze dwadzieścia lat temu sekretarka musiała potrafić pisać na maszynie, korzystać z telefonu, robić kawę. Dzisiaj, aby znaleźć pracę musi posiadać certyfikat ukończenia kursu obsługi komputera, certyfikat potwierdzający znajomość co najmniej jednego języka obcego i być może jeszcze inne.

Już dzisiaj mamy zatem do czynienia z ciągłą koniecznością zdobywania nowych umiejętności i dodatkowej wiedzy, której nie można otrzymać w wyniku przejścia przez standardowy proces edukacji. Coraz bardziej popularne stają się więc korzystanie z prywatnych kursów i szkoleń, które finansowane są najczęściej przez osoby zainteresowane lub ich opiekunów. „Społeczeństwo jutra będzie inwestować w wiedzę i stanie się społeczeństwem uczenia się i nauczania, w którym każdy będzie tworzył swoje kwalifikacje. Innymi słowy powstanie **społeczeństwo uczące się**” [BIAL97, s.5]. Oznacza to zwrot w kierunku świadomego samokształcenia się jednostek, którym jednak należy dać solidną podstawę wiedzy ogólnej, pozwalającej na elastyczną zmianę kwalifikacji w trakcie życia zawodowego. Taki obrót sprawy oznacza nieuchronny rozwój rynku edukacyjnego, a zwłaszcza rynku nowych technologii wspierających i przyspieszających poszerzanie kwalifikacji. Wraz ze wzrostem zapotrzebowania na wiedzę konieczną do pracy, wzrasta także potrzeba obiektywnego weryfikowania tejże wiedzy, gdyż (...) „jednostki tradycyjnie podążają drogą zdobywania kwalifikacji formalnych, 'papierkowych'” [BIAL97, s.7]. Wymienione wyżej argumenty świadczą o konieczności budowy aplikacji wspomagających system szkoleń oraz testowania wiedzy. Obecnie powstaje

wiele firm specjalizujących się w przeprowadzaniu szkoleń oraz wydawaniu certyfikatów. To właśnie one stanowią silną grupę klientów dla producentów oprogramowania edukacyjnego.

Z drugiej strony (...) „powstaje nowy model zdobywania wiedzy i tworzenia *know-how*, łączący wysoką specjalizację z kreatywnością. Specjalizacja pociąga za sobą konieczność pogłębienia relacji między nauką a przedsiębiorstwami. Przemysł w coraz większym stopniu odwołuje się do wiedzy (...)” [BIAL97, s.24]. Wobec rosnącego tempa postępu technicznego, nowych technik multimedialnych oraz konieczności odnowy i zmiany kwalifikacji zawodowych przez pracujących niezbędne staje się organizowanie kształcenia ustawicznego przez szkoły i uczelnie w miarę potrzeb i w porozumieniu z podmiotami gospodarczymi [NAJP03]. „Szkola i przedsiębiorstwo są miejscami zdobywania wiedzy uzupełniającymi się nawzajem, dlatego należy je koniecznie zbliżyć” [BIAL97, s.61]. Istotnym przejawem takiego zbliżenia jest rosnąca liczba przeprowadzanych w przedsiębiorstwach szkoleń dla pracowników. Dla ułatwienia doksztalcania kadry, firmy sięgają często po zdobycze nowych technologii edukacyjnych. Zastosowanie znajdują tutaj systemy e-learning, poprzez które rozumie się technologie przeprowadzania szkoleń wykorzystujące Internet, a zwłaszcza Web w procesie nauczania zdalnego.

Można więc reasumując powiedzieć, że mamy do czynienia z dwoma głównymi grupami odbiorców systemów wspomagających nauczanie. Są to z jednej strony instytucje – centra szkoleniowe, z drugiej natomiast same firmy, którym ta technologia przydaje się do prowadzenia wewnętrznego kształcenia pracowników. Potrzeby jednych i drugich, zwłaszcza w zakresie weryfikacji wiedzy, są dosyć różne, dlatego inna jest także budowa wykorzystywanych przez nich aplikacji.

1.2 Rodzaje technologii wspierających edukację – definicje

Definiując różne formy weryfikacji wiedzy, chciałbym pokazać podstawowe różnice, jakie występują pomiędzy egzaminami „tradycyjnymi” a tymi, które wspomagane są technologią kom-

puterową. Na wstępie należy wziąć pod uwagę fakt, że interesują nas wyłącznie tzw. „egzaminy pisemne”. Przeprowadzenie tradycyjnego egzaminu pisemnego sprowadza się z grubsza rzecz ujmując do wykonania następujących czynności:

- opracowanie zestawu pytań i odpowiedzi,
- powielenie wydrukowanego zestawu pytań,
- właściwe egzaminowanie:
rozdanie zestawów, nadzorowanie zdających, zebranie wypełnionych formularzy lub kartek z odpowiedziami,
- sprawdzanie prac i wystawianie ocen,
- wypełnianie protokołu i/lub innych dokumentów.

W przeprowadzanym w ten sposób procesie egzaminowania, komputer jest traktowany jako narzędzie pomocnicze, co więcej można ten proces przeprowadzić nie korzystając w ogóle z komputera. Jego rola sprowadza się do pomocy przy drukowaniu zestawu pytań. Dodatkowo, jeśli w danej instytucji funkcjonuje elektroniczny system zarządzający danymi studentów, taki jak np. Wirtualny Dziekanat w SGH, należy za pomocą komputera wypełnić protokół z ocenami lub dodatkowo dokonać transferu wyników egzaminu do odpowiedniego systemu.

Funkcją systemów wspomagających procesy weryfikacji wiedzy jest automatyzacja niektórych faz procesu testowania. Podstawowym kryterium dzielącym te systemy na różne kategorie jest zakres ich działania.

W polskiej literaturze naukowej nie doczekaliśmy się jeszcze jednoznacznych nazw poszczególnych kategorii systemów egzaminujących. Dlatego spróbuję teraz zdefiniować kilka pojęć zaczerpniętych z języka angielskiego.

Computer assisted assessment (CAA) – pod tym pojęciem kryją się wszelkiego typu programy wspierające weryfikację wiedzy. Oznacza to, że mogą to być w szczególności systemy wykorzystujące techniki optycznego rozpoznawania znaków (ang. *Optical Mark Recognition OMR*) do wczytywania specjalnie przygotowanych, a następnie wypełnionych formularzy z odpowiedziami. Dzięki możliwości odczytania udzielonych odpowiedzi, oraz wcześniejszemu zaprogramowaniu odpowiedzi prawidłowych, mogą one sprawdzać prace i wystawiać oceny. W tym przypadku mamy jednak do czynienia z procesem egzaminowania, który nie jest w całości przeprowadzany elektronicznie [CHAR98, s.3].

Computer-based assessment (CBA) – pod tą nazwą kryją się systemy wspierające cały proces egzaminowania. Oznacza to, że w tym przypadku studenci udzielają odpowiedzi za pośrednictwem terminali. Jest to więc pojęcie węższe niż CAA, choć często bywa błędnie używane zamiennie. Na CBA składa się wiele środków i metod, które często nie dają się łatwo przełożyć na tradycyjne egzaminy pisemne. Są to między innymi nowe formy pytań oraz nowe modele oceniania – tzw. **testowanie adaptywne** (ang. *Computerized Adaptive Testing*) [DICT].

Computerized adaptive testing (CAT) – metoda przeprowadzania testów, podczas których kolejne pytania wybierane są przez komputer, w zależności od odpowiedzi udzielonej przez zdającego na pytanie poprzednie. Dzięki stopniowaniu trudności pytań daje się w dosyć dużym stopniu wyeliminować wpływ stresu oraz motywacji zdającego na jego ocenę końcową². Teoretyczną podstawą budowania systemów CAT jest teoria *Item Response Theory* [BAKE01]. Problematyka CAT jest dosyć szeroko opisana w literaturze z zakresu psychologii oraz pedagogiki, natomiast w niniejszej pracy nie będzie ona szerzej poruszana.

Wszystkie opisywane w tej pracy systemy wspierające egzaminowanie należą do klasy systemów CBA.

²Omawiane zależności opisane zostały po raz pierwszy w czasopiśmie pt. "Journal of Educational Psychology", nr 77 z roku 1985 przez T. Rocklina i J.M. Thompsona. [DICT]

1.3 Klasyfikacja systemów CBA z punktu widzenia technologii i architektury

Na wstępie rozważań sygnalizowano, że systemy CBA można podzielić na dwie duże grupy w zależności od zastosowania. Obecnie przedstawię różnice technologiczne występujące pomiędzy różnymi rodzajami tych systemów.

Dla porządku należy wyróżnić systemy jedno- i wielostanowiskowe. W praktyce, dzięki upowszechnieniu się sieci komputerowych, te pierwsze, mające formę aplikacji wykonywalnej oddzielnie dla każdego studenta, zostały zastąpione bardziej złożonymi programami, należącymi do drugiej grupy. Prawie wszystkie obecnie budowane systemy opierają się na architekturze klient-serwer.

Kolejnym czynnikiem odróżniającym CBA jest występowanie lub brak centralnej bazy danych. Prostsze systemy pozbawione są ogromnych możliwości raportowania i zarządzania egzaminami przez wielu egzaminatorów/wykładowców, jakie daje zastosowanie bazy danych do przechowywania treści pytań, odpowiedzi, ustawień systemowych oraz wyników. Zaletą systemów pozbawionych bazy danych jest natomiast najczęściej dużo prostsza instalacja, oraz brak konieczności posiadania odpowiednio skonfigurowanego i ogólnodostępnego serwera bazy danych.

Ostatni – najbardziej skomplikowany podział, to kwestia interfejsu użytkownika. Najstarsze i najprostsze systemy wykorzystują interfejs aplikacji pracującej w trybie tekstowym. Ich zaletą są niskie wymagania sprzętowe oraz duża wydajność, wadą natomiast jest brak możliwości wykorzystywania rysunków, schematów itp. w treści pytań oraz często mało intuicyjna obsługa. Bardziej skomplikowane systemy wykorzystują „tryb okienkowy” lub strony WWW, jako interfejs dla zdających. Pośród tych systemów należy wyróżnić dwa typy rozwiązań. Są to tzw. architektury „cienki klient” i „gruby klient”.

Do pierwszej kategorii należą systemy określane często mianem *web-based* lub *on-line assessment* [CHAR98, s.3]. Wykorzystują one centralny serwer WWW, praktycznie zawsze współpracujący z Systemem Zarządzania Bazą Danych. Studenci zdają egzaminy za pośrednictwem przeglądarki internetowej. Tego typu systemy CBA doskonale sprawdzają się w zastosowaniach, jako programy testujące w systemach e-learning, zainstalowanych w przedsiębiorstwach, a mających na celu dokształcanie pracowników. Mogą one również z powodzeniem być stosowane, jako narzędzia umożliwiające samotestowanie w trakcie kursu. Wadą, z punktu widzenia organizacyjnego, która często powoduje, że nie mogą one być używane podczas poważnych egzaminów certyfikacyjnych, jest niewielka kontrola jaką oferują, nad tym co dodatkowo robią osoby, zdające na komputerach, w trakcie odpowiadania na pytania. Z punktu widzenia technicznego największą zaletą tych rozwiązań są minimalne wymagania sprzętowe definiowane od strony zdającego, brak konieczności instalowania jakiegokolwiek oprogramowania na maszynach studentów, oraz najczęściej pełne uniezależnienie od środowiska systemu operacyjnego. Wadą są natomiast spore wymagania dla serwera, które na dodatek w bardzo dużym stopniu zależą od liczby jednocześnie zdających osób. Ponadto, ze względu na synchroniczną wymianę danych, odbywającą się w trakcie egzaminu między serwerem a przeglądarką, są one w dużym stopniu zależne od szybkości i jakości funkcjonowania połączeń sieciowych.

Drugą grupę stanowią systemy CBA, budowane w architekturze „grubego klienta”. Stosowane są tu dedykowane dla tych systemów aplikacje stanowisk egzaminacyjnych, które mogą mieć pełną kontrolę nad stacjami dydaktycznymi zdających. Oznacza to w szczególności możliwość zablokowania korzystania z innych aplikacji w trakcie odpowiadania na testowe pytania. Co za tym idzie, możemy tutaj także ograniczyć porozumiewanie się zdających między sobą za pomocą komputerów oraz korzystanie z innych źródeł informacji np. Internetu. Aby skorzystać z tych zabezpieczeń, trzeba jednak zainstalować i uruchomić aplikację na każdym komputerze studenta. Od strony technicznej, podstawową zaletą tego rozwiązania jest rozłożenie obciążenia pomiędzy serwer a maszyny studentów. Daje to szczególnie duże efekty podczas organizowania

egzaminów dla wielu osób jednocześnie. Ponadto w przypadku tego typu budowy CBA mniejsze są wymagania odnośnie przepustowości sieci, gdyż transfer pytań oraz wyników pomiędzy serwerem a aplikacją zdającego odbywa się asynchronicznie. Wadę rozwiązania stanowi konieczność uwzględnienia, podczas budowy aplikacji klienckiej, rodzaju sprzętu oraz oprogramowania systemowego komputerów, pełniących role stanowisk egzaminacyjnych.

Rozdział 2

Teoretyczne podstawy metod testowania wiedzy

„Our progress as a nation can be no swifter than our progress in education. The human mind is our fundamental resource.”

/ John F. Kennedy /¹

2.1 Pomiar dydaktyczny – teoretyczna podstawa testowania

Zanim zajmiemy się komputerowymi aplikacjami wspomagającymi procesy egzaminowania, warto najpierw zapoznać się z aktualnymi poglądami naukowymi, na temat samego procesu egzaminowania. Niniejsza część mojej pracy oparta jest na materiałach pochodzących z książki profesora Bolesława Niemierko pt.: „Pomiar wyników kształcenia” [NIEM99].

Proces egzaminowania nazywany jest w naukach pedagogicznych **pomiarem dydaktycznym**. „Pod tym pojęciem rozumie się przyporządkowanie symboli (ocen) uczniom w taki sposób, by relacje między symbolami odpowiadały relacjom między uczniami ze względu na określone osiągnięcia. Zasady przyporządkowania powinny być ustalone i możliwie do-

¹Cytat pochodzi ze zbiorów internetowych, dostępnych pod adresem: <http://www.quotationspage.com>

kładnie przestrzegane, a proces pomiarowy powinien podlegać obiektywnej kontroli” [NIEM99].

To, co w potocznym rozumieniu nazywamy egzaminem, pedagodzy określają jako **test osiągnięć szkolnych**. Profesor Niemierko podaje następującą definicję: „Test osiągnięć szkolnych jest zbiorem zadań przeznaczonych do rozwiązywania w toku jednego zajęcia szkolnego, reprezentujących wybrany zakres treści kształcenia w taki sposób, by z ich wyników można było wnioskować o poziomie opanowania tej treści” [NIEM99].

Testy osiągnięć szkolnych zawierają różne zadania. W podstawowym podziale wyróżniamy zadania otwarte oraz zamknięte. Obie te formy mogą przyjmować różne szczegółowe postacie, które omówię poniżej.

2.1.1 Zadania otwarte

Pisemne zadania otwarte mają pod pewnymi względami przewagę nad zadaniami praktycznymi. Charakteryzują się nie tylko wyższą sprawnością w dostarczaniu nauczycielom szczegółowej informacji o wielu uczniach w ograniczonym czasie, lecz także dokładniej odwzorowują umiejętność kodowania informacji na poziomie wyobrażeń, pojęć i struktur teoretycznych oraz umiejętność samodzielnego przetwarzania informacji na tych poziomach.

W pisemnych zadaniach otwartych uczeń lub student samodzielnie formułuje i zapisuje odpowiedzi. Ta samodzielność stanowi wielką zaletę zadań otwartych, jest jednak także ich słabością, gdyż wielu uczniom zredagowanie odpowiedzi pisemnej, nawet bardzo krótkiej, sprawia większą trudność i zabiera więcej czasu niż samo rozwiązanie danego zadania od strony merytorycznej.

Wśród zadań otwartych ze względu na ich długość oraz złożoność wyróżniamy następujące formy [NIEM99]:

1. *Zadanie rozszerzonej odpowiedzi*

Forma ta określana angielskim terminem *extended-response item*, wymaga od ucznia rozwiniętej odpowiedzi pisemnej. Bywa ona także nazywana zadaniem-rozprawką (ang. *essay item*). Może obejmować nie tylko wypracowania werbalne, lecz także dłuższe wypowiedzi matematyczne, techniczne (z rysunkami i wyliczeniami), a nawet artystyczne (np. plastyczne).

2. *Zadanie krótkiej odpowiedzi*

Znane jest jako termin angielski *short-answer item*, wymagające od ucznia udzielenia odpowiedzi w postaci jednego słowa, liczby, symbolu, pełnego zdania, wyrażenia matematycznego lub najwyżej 2-3 zdań. W testach nauczycielskich jest to najbardziej popularna forma zadań.

3. *Zadania z luką*

W angielskiej terminologii znane jako *completion item*, wymagające od ucznia uzupełnienia zwrotu, zdania, fragmentu tekstu, wyrażenia matematycznego lub rysunku. Ze względu na gotowy kontekst wypowiedzi ucznia jest to zadanie rozwiązywane najszybciej, lecz najbardziej powierzchownie.

Poniżej opisuję krótko każdą formę zadań otwartych.

Zadanie rozszerzonej odpowiedzi

Zadanie rozszerzonej odpowiedzi to najobszerniejsza forma zadania. Ze względu na subiektywizm oraz różnorodność odpowiedzi, zadania te stanowią poważny problem dla osób oceniających. Aby wspomóc nauczycieli podczas oceniania, pedagodzy opracowali listy cech wypowiedzi tekstowych, które należy brać pod uwagę podczas szczegółowej ewaluacji tekstu. W

programie Nowa Matura, profesor Kuczyńska [KUCZ97, s.148-149] przedstawiła następujący zbiór kryteriów i schemat punktowania wypracowań [NIEM99]:

1. *Wnikliwość omówienia tematu.*
2. *Porównanie ilościowe tekstu i tematu.*
3. *Warstwa faktograficzna (erudycyjna).*
4. *Operowanie informacjami, poziom ogólności.*
5. *Układ treści, kompozycja wypowiedzi.*
6. *Spójność logiczna tekstu.*
7. *Sprawność językowa.*
8. *Ład stylistyczny.*
9. *Grafia (ortografia, interpunkcja).*

Dzięki powyższej liście kryteriów, można zwiększyć precyzję oceny zadań rozszerzonej odpowiedzi poprzez zastosowanie osobnego punktowania dla każdego kryterium, a następnie obliczania sumarycznego wyniku końcowego.

Wśród pedagogów toczy się dyskusja dotycząca stosowania tego typu zadań w egzaminowaniu. Zadania rozszerzonej odpowiedzi są bardzo często przeciwstawiane zadaniom wielokrotnego wyboru. Poniżej zamieszczono opinie amerykańskich pedagogów, na które profesor Niemirko zwraca szczególną uwagę.

„Najważniejszymi cechami zadań rozszerzonej odpowiedzi jest swoboda wypowiedzi dana egzaminowanym oraz fakt, iż nie tylko żadna pojedyncza odpowiedź nie może być uznana za prawidłową i kompletną, a następnie podyktowana mniej kompetentnym pomocnikom w celu punktowania zadań, lecz nawet ekspert nie może jednoznacznie sklasyfikować pewnych

odpowiedzi jako prawidłowych lub błędnych. Istnieją natomiast różne poziomy jakości lub wartości odpowiedzi, które mogą być rozpoznane” [STAL51].

„Powinno się pamiętać, że typowy nauczyciel jest rzadko zainteresowany czystym pomiarem. Angażuje się natomiast w ciągły proces interakcji z uczniami i w nieustanne ocenianie osiągnięć ucznia na podstawie tych interakcji. Egzaminy, jakie prowadzi, są tylko jednym ze źródeł informacji w kontekście danych zgromadzonych wcześniej. Nauczyciel, który podjął się budowania zadań zamkniętych, najpewniej przeżyje rozczarowanie(…)” [COFF71].

Poniżej wypunktowane zostały najważniejsze pozytywne, a następnie negatywne cechy zadań rozszerzonej odpowiedzi. Wśród zalet wymienia się [NIEM99]:

1. *Wszechstronność.*

Można uzyskać opisy, wyjaśnienia, oceny i projekty działania.

2. *Pozytywny wpływ na uczenie się.*

Takie zadania skłaniają uczniów do solidnego przygotowywania się do egzaminów, czego nie można powiedzieć o zadaniach zamkniętych, zachęcających do posługiwania się sprytem testowym (ang. *test-wiseness*).

3. *Związek z działaniem praktycznym.*

Jeżeli dalsze kształcenie i zawód, do których uczeń się przygotowuje, obejmują pisemne opisy, analizy, projekty i raporty, zadanie rozszerzonej odpowiedzi może być równoważne z odpowiednią próbą pracy.

4. *Łatwość opanowania tej formy ewaluacji osiągnięć szkół przez nauczycieli.*

Z kolei najważniejsze wady to:

1. *Wybiórcze reprezentowanie treści kształcenia.*

Uczeń może trafić na „swój” temat lub nie – w obu przypadkach wnioski o ogólnym poziomie jego osiągnięć będą niedokładne.

2. *Wyniki zadań rozszerzonej odpowiedzi są silnie uzależnione od zdolności (językowych, rysunkowych, matematycznych) uczniów, niekoniecznie popartych systematyczną wiedzą.*
3. *Punktowanie zadań rozszerzonej odpowiedzi jest mało obiektywne.*
4. *Punktowanie zadań rozszerzonej odpowiedzi jest czasochłonne, i to tym bardziej, im silniej dążymy do obiektywizmu.*

Zadanie krótkiej odpowiedzi i zadanie z luką

Mniejszą, z punktu widzenia objętości formą zadań otwartych, jest zadanie krótkiej odpowiedzi. W tych zadaniach zwraca się uwagę jedynie na poprawność merytoryczną wypowiedzi ucznia, nie zaś na jej rozwinięcie, układ i styl. Dzięki temu pozbawione jest ono części wad zadań rozszerzonej odpowiedzi wynikających z braku obiektywizmu w ocenie.

Zadanie z luką jest najbardziej uproszczoną formą zadania otwartego. W tym zadaniu, ze względu na zwięzłość odpowiedzi, która przyjmuje formę słowa lub wyrażenia złożonego z kilku słów, nie ocenia się stylu, układu ani sformułowania treści, a jedynie jej merytoryczną zawartość. Zadania tego typu są z reguły bardziej obiektywne niż zadania krótkiej i rozszerzonej odpowiedzi.

„Luka jest zazwyczaj zaznaczona kropkowaniem lub ramką, ale może też nie być zaznaczona i wtedy uczeń musi sam ją odnaleźć” [NIEM99]. „Taki wariant zadania z luką nazywamy zadaniem typu korekta” [NIEM99]. Obydwa warianty zadań z luką znajdują bardzo szerokie zastosowanie w testach językowych.

Poniżej zamieszczam zalety zadań krótkiej odpowiedzi oraz zadań z luką [NIEM99]:

1. *Zadawalająca reprezentatywność zbioru zadań dla stosunkowo szerokiego zakresu treści kształcenia.*

2. *Bardziej obiektywne punktowanie w stosunku do zadań rozszerzonej odpowiedzi.*
3. *Niepodatność na zgadywanie odpowiedzi przez uczniów.*

W porównaniu z zadaniami zamkniętymi możliwości odgadnięcia prawidłowej odpowiedzi są bardzo małe.

4. *Łatwość konstrukcji.*

Wśród najważniejszych wad tego typu zadań pedagodzy wymieniają:

1. *Rozdrobnienie treści kształcenia na pojedyncze pytania i jednowyrazowe lub jednozdaniowe wypowiedzi.*
2. *Ciążenie ku niskim kategoriom celów kształcenia, w szczególności ku pomiarowi zapamiętania pojedynczych wiadomości (nazw, liczb, znaków) i najprostszych umiejętności.*
3. *Niepełny obiektywizm punktowania, uniemożliwiający automatyczne szczytywanie wyników testowania.*

Istotną przeszkodą są tutaj częste błędy ortograficzne popełniane przez uczniów, które w przypadku oceny maszynowej dyskwalifikują często właściwą merytorycznie odpowiedź.

2.1.2 Zadania zamknięte

Zadania zamknięte uważane są za najbardziej obiektywną formę egzaminowania. Są one szczególnie chętnie wykorzystywane w krajach anglosaskich. Z punktu widzenia prezentowanej pracy, największą zaletą zadań zamkniętych jest łatwość wykorzystania ich w aplikacjach egzaminujących.

Zadania zamknięte dzielą się na kilka kategorii. Są to [NIEM99]:

1. *Zadania wyboru wielokrotnego (ang. multiple-choice items), w szczególnym przypadku mogą to być zadania jednokrotnego wyboru.*

Obecnie dąży się do możliwie szerokiego stosowania zadań praktycznych, ale zadanie wyboru wielokrotnego pozostało formą najbardziej zaawansowaną teoretycznie i konstrukcyjnie, a także najbardziej podatną na analizy statystyczne.

2. *Zadanie na dobieranie* (ang. *matching items*), w którym odpowiedzi do wyboru tworzą listę wspólną dla grupy zadań.

Zadanie na dobieranie występuje wyłącznie w tzw. układzie wiązkowym. W tym przypadku chodzi o prezentowanie, wspólnej dla kilku zadań, listy słów lub wyrażeń kluczowych, które uczeń powinien połączyć, lub w inny sposób zaznaczyć relacje między nimi zachodzące. Mamy więc do czynienia z sytuacją, w której odpowiedzi są wspólne dla całej wiązki, a trzon zadań jest częściowo wspólny, a częściowo rozdzielony na hasła kolejnych zadań. Podane odpowiedzi mogą być na zmianę prawidłowe i błędne.

3. *Zadanie typu prawda-fałsz*

Zadanie typu prawda-fałsz jest najprostszą formą zadania zamkniętego.

Poniżej prezentuję wypunktowane zalety oraz wady zadań zamkniętych. Do zalet tych zalicza się:

1. *Szeroki zakres zastosowań.*

Zadania zamknięte, zwłaszcza zadanie wyboru wielokrotnego, są uniwersalne w tym sensie, że można nimi ująć każdy zakres treści kształcenia od strony logicznej.

2. *Obiektywne punktowanie wyników.*

3. *Sprawność pomiarowa.*

W toku pracy nad testem złożonym z zadań zamkniętych uczeń „czyta i myśli”, a nie „myśli i pisze”, co znacznie przyspiesza rozwiązywanie typowych zadań zamkniętych i zapobiega jałowemu „pływaniu” uczniów wokół tematu.

4. *Wdrażanie do podejmowania decyzji.*

Zadania zamknięte tworzą sytuację decyzyjną, uczą pokonywania wątpliwości i ponoszenia ryzyka, a te umiejętności mają rosnące znaczenie w kształceniu ogólnym i zawodowym na całym świecie.

Wśród wad zadań zamkniętych pedagodzy wymieniają:

1. *Niemożliwość tworzenia syntez przez uczniów.* Zadania zamknięte nie zostawiają uczniom miejsca na samodzielne formułowanie hipotez i projektowanie eksperymentów oraz na wyrażanie oryginalnych poglądów.
2. *Fałszywy obraz świata i wiedzy ludzkiej jako zamkniętych systemów o stałych i czytelnym prawidłowościach.*
3. *Przewagę formy zadań nad treścią kształcenia.*

Skomplikowana forma zadań powoduje, że wielu konstruktorów dopasowuje do niej wybrane, nie zawsze istotne, elementy treści kształcenia, inni zaś jednostronnie akcentują zależności logiczne w obrębie tej treści ze szkodą dla umiejętności praktycznych uczniów. Ponadto zadania zamknięte są podatne na spryt testowy.

4. *Podatność na zgadywanie odpowiedzi przez uczniów.*

Teoretycznie uczeń może uzyskać średnio od 20% (przy pięciu odpowiedziach do wyboru) do 50% (w zadaniach prawda-fałsz) maksymalnej liczby punktów przez czyste zgadywanie, to jest losowy wybór (ang. *blind guessing*).

5. *Trudność konstruowania.*

Tylko wybitni nauczyciele spontanicznie ujmują materiał kształcenia strukturalnie, to jest w sposób umożliwiający wykorzystanie walorów zadań zamkniętych.

2.1.3 Podsumowanie

Reasumując powyżej przedstawione wiadomości teoretyczne możemy powiedzieć, że zadania otwarte mogą dać generalnie bardziej dokładny obraz umiejętności i wiedzy ucznia. Są za to mało obiektywne i to zarówno, ze względu na zróżnicowany odbiór danego tekstu przez różnych oceniających, jak też z powodu trudności jakie stwarzają, mniej uzdolnionym w pisaniu, uczniom. Z drugiej strony mamy zadania zamknięte, które są podatne na spryt testowy i trudne w konstrukcji, lecz dzięki klarownym kryteriom oceny, dają bardzo obiektywny obraz znajomości zawartej w pytaniach treści.

2.2 Formy pytań możliwe do wykorzystania w aplikacjach egzaminujących

Biorąc pod uwagę wnioski wynikające z poprzedniego podrozdziału, oraz analizę praktycznych aplikacji systemów wspomagających przeprowadzanie egzaminów, chciałbym przedstawić poniżej rodzaje pytań, dających się wykorzystać w komputerowych systemach egzaminujących. Zawarta tutaj lista typów pytań została opracowana na podstawie artykułu Colleen McKenna et al pt.: „Designing effective objective test questions: an introductory workshop” [MCKE99, s.2].

- Pytania jednokrotnego wyboru.
- Pytania wielokrotnego wyboru.
- Pytania typu: prawda/fałsz.
- Pytania łączące elementy pytań wyboru z oceną uzasadnienia typu prawda/fałsz.
- Graficzne wskazywanie obszaru schematu – zaawansowane wersje mogą dodatkowo zawierać wprowadzanie etykiet lub nawet budowanie pytań. Pytania tego typu są znacznie trudniejsze i bardziej kosztowne do implementacji niż pytania wyboru.

- Krótkie odpowiedzi otwarte – wprowadzanie tekstu lub liczby za pomocą klawiatury. Analiza odpowiedzi na takie pytania polega na porównywaniu tekstu podanego przez studenta z jednym z zaprogramowanych przez egzaminatora. Trudno na razie mówić o systemach, które byłyby w stanie, w sposób niezawodny, oceniać dłuższe wypowiedzi oraz przeprowadzać analizę znaczeniową tekstu odpowiedzi.
- Dobieranie elementów – graficzne łączenie elementów z dwóch lub więcej grup ze sobą.
- Pytania, w których tekście należy znaleźć nie pasujące słowo (lub fragment) i je zaznaczyć lub wpisać za pomocą klawiatury.
- Rangowanie wyrażen – polega na ustaleniu relacji między obiektami znajdującymi się w jednej kolumnie. Tego typu pytania są szczególnie przydatne do sprawdzania znajomości poziomów gradacji oraz kolejności zdarzeń.
- Ustawianie w kolejności – tego typu pytania wymagają od studenta, aby ustawił tekst albo obiekty graficzne w odpowiedniej kolejności. Są one szczególnie przydatne do sprawdzania metodologii.
- Pytania symulacyjne – symulują prawdziwe zadania. W praktyce są bardzo trudne do zaimplementowania i rzadko spotykane.

Przedstawione powyżej typy stosowanych pytań, pomogą nam w ocenie i porównaniu od strony funkcjonalnej, opisywanych w kolejnych rozdziałach, implementacji różnych systemów wspomagających proces egzaminowania.

2.3 Porównanie efektywności egzaminowania za pomocą komputera z egzaminami tradycyjnymi

W 1995 roku przeprowadzono w Advanced Learning Laboratory (ALL School of Worcester), mieszczącej się w USA w stanie Massachusetts, badanie porównujące efekty egzaminu, który

został zorganizowany przy użyciu tego samego materiału, w dwóch formach: tradycyjnej – studenci odpowiadali na pytania pisząc na kartkach papieru, oraz komputerowej. To co jest jeszcze bardziej istotne, badany egzamin zawierał również dłuższą wypowiedź, którą jedna grupa uczniów formułowała na papierze, a druga wpisywała do komputera. Efekty tego badania są zadziwiające.

Badanie to przeprowadzono dwuetapowo. Poniżej prezentuję sposób przeprowadzenia oraz wnioski wynikające z pierwszego etapu tego badania.

W celu zapewnienia porównywalności wyników egzaminów obu grup dołożono wszelkich starań, by wyeliminować wszelkie czynniki, które mogłyby wpłynąć na łatwość odpowiadania na pytania w którejkolwiek z badanych form. Przy konwertowaniu egzaminu papierowego na formę elektroniczną zwrócono uwagę na kolejność prezentacji odpowiedzi, czytelność czcionki wyświetlanej na monitorze (w komputerach użyto trochę większej czcionki, aby zrekompensować zmęczeniu wzroku, wynikającym z konieczności czytania na komputerze), starano się ograniczyć tekst wprowadzający do pytań tak, aby jego czytanie na monitorze nie było męczące dla uczniów, a także zachowano taki sam układ tekstu oraz grafiki prezentowanej na każdej stronie egzaminu w obydwu formach. Z drugiej strony, aby wyeliminować wpływ oceniających na wyniki egzaminów w każdej grupie, wprowadzono do komputera wszystkie odpowiedzi tekstowe, z wyjątkiem wypowiedzi otwartych, które pisane były przez wszystkich uczniów na kartkach papieru, a następnie wymieszano tak, aby nauczyciele nie byli w stanie rozpoznać, w jakiej formie dana odpowiedź została udzielona. Wypowiedzi tekstowe oceniane były przez trzech niezależnych nauczycieli, a końcowe wyniki stanowiły średnie z wystawionych przez nich ocen.

Egzamin składał się z pięciu części:

- Wypowiedź otwarta (uczniowie odpowiadali na kartkach).
- Język Angielski (pytania wyboru i krótkie odpowiedzi tekstowe).
- Nauki ścisłe (pytania wyboru i krótkie odpowiedzi tekstowe).

- Matematyka (pytania wyboru i krótkie odpowiedzi tekstowe).
- Wypracowanie (dwie grupy uczniów pisały odpowiednio na kartkach i komputerze).

W poniższej tabeli prezentuję średnie wyniki z poszczególnych części egzaminu dla obydwu grup.

		Kartka			Komputer				
	Skala	n	Średnia	Od.St.	n	Średnia	Od.St.	Stystyka t	Poziom <i>a</i>
Wypowiedź otwarta	0-14	68	7.62	3.14	46	8.24	2.66	1.10	0.27
Język Angielski	0-15	42	9.24	3.96	44	10.41	3.58	1.44	0.15
Nauki ścisłe	0-23	42	8.67	4.17	44	10.68	4.39	2.18	0.03
Matematyka	0-18	42	6.00	3.30	44	6.41	3.51	0.56	0.58
Wypracowanie	1-4	46	2.30	0.55	40	2.81	0.59	4.16	<0.0001**

** Statystycznie istotne przy poziomie istotności 0.01

Tablica 2.1: Wyniki badania porównawczego

Źródło: Opracowanie własne na podstawie [RUSS97, tabele: 2, 4 i 5]

Jak wynika z tabeli 2.1 statystycznie istotne różnice pomiędzy dwiema rozpatrywanymi formami zdawania egzaminu pojawiły się przy ocenie wypracowań. Jest to bardzo ważne, gdyż w większości wcześniejszych prac porównawczych, w których skupiano się na testach wielokrotnego wyboru okazywało się, że forma weryfikacji wiedzy nie wpływa w sposób znaczący na ocenę [RUSS97, s.12]. Obliczona przez autorów badania wielkość zauważonego efektu wynosząca 0,94 oznacza, że w przypadku wypracowania ocena przeciętnego ucznia należącego do grupy egzaminowanych za pomocą komputera, jest wyższa od 83% ocen osób zdających egzamin w formie tradycyjnej [RUSS97, s.12]. Wnikliwa analiza wyników omawianego badania pokazuje dodatkowo, że uczniowie, którzy zdawali egzamin za pośrednictwem komputera, pisali prawie dwukrotnie więcej oraz formułowali swoje wypowiedzi w większej liczbie akapitów [RUSS97, s.13]. Autorzy badania wykluczyli wpływ motywacji jako czynnika, który mógłby

podwyższać oceny osób egzaminowanych komputerowo. Posłużyli się oni dwoma argumentami. Po pierwsze: jeśli motywacja miałaby wpływać na wyniki, to prawdopodobnie musiałaby również poprawić rezultaty testów wyboru. W tym jednak badaniu nie stwierdzono statystycznie istotnych różnic dotyczących tych części egzaminu. Po drugie w przeprowadzonych przez O'Neill, Sugrue & Baker² w 1996 roku badaniach porównywano wpływ motywacji na wyniki egzaminów. W tym przypadku jednak wielkość efektu różnicującego oceny wahała się od 0,16 do 0,24 a więc była kilkakrotnie niższa niż w opisywanym badaniu [RUSS97, s.14].

Dwa lata po ukazaniu się artykułu opisującego powyższe badanie M. Russell, jeden z jego autorów, przeprowadził kolejną, głębszą analizę, w której skupił się na wypowiedziach otwartych.

Opisane poniżej badanie zostało opracowane na podstawie artykułu Micheala Russella pt. „*Testing On Computers: A Follow-up Study Comparing Performance On Computer and On Paper*” z 1999 roku [RUSS99].

W drugim etapie badania, uczniowie losowo podzieleni, analogicznie, na dwie grupy odpowiadali w sposób swobodny na pytania z zakresu matematyki, umiejętności pisania, czytania oraz nauk ścisłych. Autor wprowadził pięć istotnych poprawek. Po pierwsze, zwiększona została próba uczniów w taki sposób, aby pokrywała osoby o różnym doświadczeniu w korzystaniu z komputera. Po drugie zebrano informacje dotyczące uprzednich doświadczeń uczniów w pracy z komputerem, ich preferencji odnośnie formy wypowiedzania się – na papierze lub na komputerze, oraz szybkości pisania na klawiaturze. Po trzecie zastosowano więcej zadań otwartych, a po czwarte zadania te nie dotyczyły wyłącznie umiejętności pisania, ale także innych zakresów wiedzy. Ostatnia zmiana polegała na tym, że tym razem wykorzystano wyłącznie sprawdzone zadania, stosowane wcześniej podczas egzaminów państwowych.

²O'Neil, H. F. Jr., Sugrue, B. & Baker, E. L. (1996). *Effects of motivational interventions on the National Assessment of Educational Progress mathematics performance*. Educational Assessment, 3(2), 135-157.

W późniejszym badaniu zwrócono szczególną uwagę na trzy sprawy. Po pierwsze, badano czy, zaobserwowana w pierwszym etapie zależność, utrzymuje się przy zmieniających się doświadczeniach w korzystaniu z komputera oraz szybkości pisania na klawiaturze; po drugie, czy siła wpływu formy zdawania egzaminu zależy od testowanego przedmiotu oraz czy uczniowie, deklarujący swoją preferencję w wypowiedzianiu się w jednej z proponowanych form, uzyskują lepsze wyniki, gdy dana im jest możliwość zdawania w ulubionej formie.

Wyniki tego badania pokazują, że rezultaty tylko niektórych zadań zależą od szybkości pisania na klawiaturze. Okazało się, że jedynie w przypadku zadania z nauk ścisłych, grupa uczniów, udzielających odpowiedzi na komputerze, wypadła, w sposób statystycznie istotny, lepiej niż grupa pisząca na papierze. Ciekawe natomiast, że w zadaniach matematycznych lepsze oceny udało się uzyskać osobom odpowiadającym na papierze, choć zaobserwowane różnice nie były, przy przyjętym współczynniku ufności, statystycznie istotne. Podobny wniosek można wysunąć, jeśli chodzi o wpływ preferowanej formy egzaminowania na ocenę. Badanie pokazało, że jedynie w przypadku zadań z nauk ścisłych, udzielanie odpowiedzi w preferowanej formie, owocowało wyższą oceną.

Myślę, że wyniki tych badań należy potraktować poważnie. W odróżnieniu od wielu wcześniejszych badań, tym razem okazało się, że forma w jakiej uczniowie udzielają odpowiedzi na pytania, zwłaszcza w przypadku zadań otwartych, może mieć istotne znaczenie dla jakości tych odpowiedzi. Implikacją zauważonego tutaj faktu powinno być wprowadzanie egzaminowania wykorzystującego komputery także w przypadku sprawdzianów otwartych.

Przedstawione podstawy teoretyczne metod testowania odniesiemy obecnie do przykładowego ich zaimplementowania w rozwiązaniu proponowanym przez firmę Oracle w ramach systemu iLearning. Z tego tytułu przyjrzymy się bliżej jednemu z modułów tego systemu – modułowi testującemu wiedzę.

Rozdział 3

Moduł testujący komercyjnego systemu szkoleniowego – Oracle iLearning

3.1 Wprowadzenie

W niniejszym rozdziale opisano funkcjonalność oraz budowę komercyjnego systemu informatycznego, wspomagającego przeprowadzanie szkoleń. Rozwiązaniem tego typu jest produkt firmy Oracle o nazwie iLearning. Produkt ten należy do kategorii profesjonalnych systemów nauczania zdalnego (ang. *e-learning*). Umożliwia on przygotowywanie materiałów szkoleniowych, zarządzanie szkoleniami, przygotowywanie testów oraz ich przeprowadzanie. Dodatkową funkcją jest obsługa płatności związanych ze szkoleniami. Ze względu na ogromne możliwości tego systemu oraz tematykę tej pracy, postaram się ograniczyć do opisu modułu służącego do weryfikacji wiedzy.

3.2 Zakres funkcjonalności modułu testującego

W prezentowanym podrozdziale pracy prezentuję jedynie wypunktowane funkcje systemu iLearning, związane z testowaniem wiedzy. Szerszy opis, tłumaczący wypisane cechy programu, został zawarty w następnym podrozdziale, przy okazji przedstawienia scenariusza przeprowadza-

nia egzaminu. Do głównych funkcji modułu testującego systemu iLearning należy:

1. *Przeprowadzanie testów: otwartych z kilkoma słowami kluczowymi lub zamkniętych pojedynczego i wielokrotnego wyboru oraz typu prawda - fałsz.*
2. *Natychmiastowe sprawdzanie poprawności odpowiedzi oraz wystawianie ocen (proces sprawdzania przeprowadzany komputerowo, z pominięciem czynnika ludzkiego).*
3. *Zgodność ze standardami systemów szkoleniowych – m. in. ze standardem SCORM.¹*
4. *Archiwizacja danych osób zdających i udzielonych odpowiedzi.*
5. *Możliwość wielokrotnego wykorzystywania pytań i konstruowania różnych testów w oparciu o ten sam materiał tematyczny.*
6. *Możliwość losowego konstruowania pojedynczego zestawu pytań egzaminacyjnych.*
7. *Import pytań z / eksport pytań do pojedynczego pliku zewnętrznego.*

iLearning wykorzystuje format XML do przechowywania konfiguracji egzaminów oraz treści pytań.
8. *Możliwość jednoczesnej obsługi wielu egzaminów z wielu przedmiotów.*

Rozbudowana struktura szkoleń oraz obsługa osobnych serwisów (z ang. *site*) na jednej instalacji.
9. *Możliwość konstruowania egzaminów próbnych – nieocenianych.*
10. *Możliwość ustalania czasu dostępności egzaminu.*
11. *Możliwość prowadzenia szkoleń w różnych stylach.*

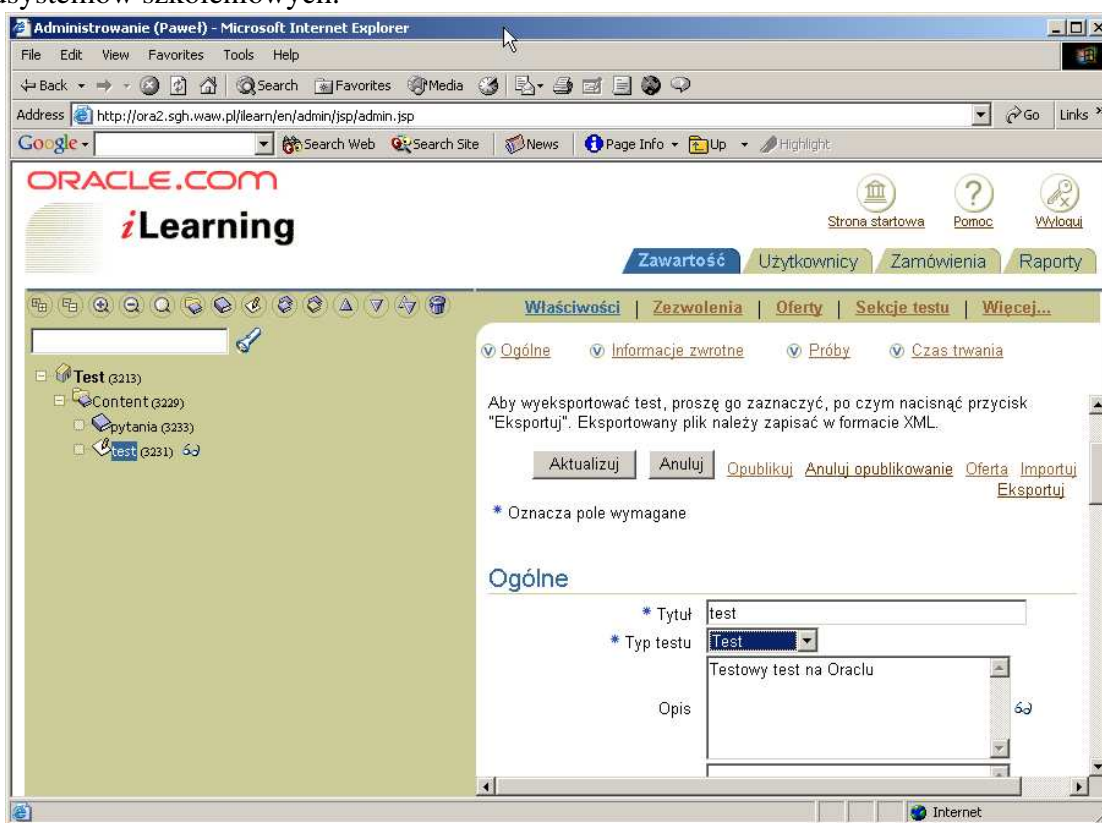
¹SCORM – Sharable Content Object Reference Model –projekt prowadzony przez inicjatywę ADL (ang. *Advanced Distributed Learning*) Departamentu Obrony Stanów Zjednoczonych. Standard SCORM pozwala treść e-learningu opracowaną zgodnie z jego regułami, rozmieszczać w dowolnym systemie typu LMS (ang. *Learning Management System*) i łączyć z innymi treściami w celu zaprojektowania kursu, który będzie spełniał odpowiednie wymagania [KOMP03, s. 25-26].

Style dzielą się na dwie podgrupy: pierwsza to szkolenia zaplanowane – wszyscy uczestnicy zdają egzamin w tym samym czasie, a druga to – praca we własnym tempie. Dodatkowo w każdej podgrupie możemy wyróżnić style: eStudy, eSeminar, eClass, oraz inClass.

12. *Możliwość swobodnego nadawania uprawnień innym administratorom egzaminu – współpraca przy konstruowaniu testów.*

3.3 Scenariusz przykładowego egzaminu komputerowego

Pracę z systemem iLearning należy rozpocząć od zalogowania się za pomocą przeglądarki internetowej do systemu na odpowiedni serwis (z ang. *site*). Konieczność wyboru serwisu wynika z faktu, że program umożliwia tworzenie na jednej instalacji wielu zupełnie niezależnych podsystemów szkoleniowych.



Rys. 3.1: Panel administracyjny systemu iLearning

Po zalogowaniu wybieramy ikonę „Administrator” i przechodzimy do panelu administracyjnego (rys. 3.1). Po lewej stronie mamy drzewo przedstawiające wszystkie stworzone dotąd elementy, a na górze menu z ikonami służącymi do tworzenia i usuwania obiektów. Na początku należy stworzyć nowy folder. Następnie klikamy myszką na stworzony folder i wybieramy opcję „Bank testów”. Tutaj powinniśmy dodać nowy „bank elementów testów”. Wybierając tę opcję, podajemy nazwę oraz opis. W kolejnym kroku wchodzimy w edycję nowo utworzonego „banku elementów testów” i wybieramy z listy pozycję „Elementy testu”. Teraz można przystąpić do dodawania kolejnych pytań (czyli elementów testu). Opcja ta jest aktywowana za pośrednictwem przycisku „Dodaj element testu”. Wprowadzanie pytań do systemu rozpoczynamy od wyboru typu pytania. Dostępne są następujące możliwości:

- Proszę wypełnić puste (odpowiedź tekstowa).
- Proszę wypełnić puste (odpowiedź liczbowa).
- Prawda lub fałsz.
- Wiele opcji (jedna poprawna odpowiedź).
- Wiele opcji (wiele poprawnych odpowiedzi).

W zależności od wybranego typu pytania otrzymujemy różne formularze. Wspólne dla wszystkich są pola: „Tekst pytania”, „Informacja zwrotna o poprawnej odpowiedzi”, „Informacja zwrotna o niepoprawnej odpowiedzi” oraz pole wyboru kolejności wyświetlania odpowiedzi (stała lub losowa). Ponadto dla pierwszych dwóch typów pytań można podać od jednej do sześciu prawidłowych odpowiedzi, z którymi będzie porównywana odpowiedź udzielona przez zdającego. Dla trzeciego typu pytania możemy wybrać, czy podany tekst pytania jest prawdą czy fałszem, natomiast dla „wielu opcji...” podajemy tekst możliwych do wyboru odpowiedzi oraz zaznaczamy właściwą (lub właściwe). Po wprowadzeniu w ten sposób wszystkich pytań do programu, publikujemy bank elementów testu, umożliwiając tym samym wykorzystanie go w teście.

Kolejny etap to przygotowanie samego egzaminu. Należy wrócić do menu, znajdującego się po lewej stronie i dodać nowy test. Dla każdego testu określamy:

- Typ – może to być test lub sprawdzian (ten pierwszy nie jest oceniany).
- Opis, charakteryzujący dany egzamin.
- Cele. Będą one wyświetlane obok opisu i pomogą studentom wybrać właściwy dla nich egzamin.
- Słowa kluczowe. Dzięki nim studenci będą mogli wyszukać odpowiedni dla nich egzamin.
- Język. Dostępne tutaj opcje zależą od konkretnej instalacji systemu. Zawsze jednak można użyć języka angielskiego.
- Źródło elementów – elementy mogą być wstępnie wybrane, lub są za każdym razem losowane przez system z dostępnej puli.
- Liczbę elementów wyświetlanych na jednej stronie.
- Kolejność wyświetlanych sekcji – losowa, lub ustalona.
- Opcje oceny – brak oceny, procentowo, suma poszczególnych wyników.
- Mistrzowski wynik – będzie on wyświetlany studentowi w podsumowaniu.
- Instrukcje odnośnie zdawania testu dla studentów.
- Typ informacji zwrotnych – czy mają one być wyświetlane po każdej stronie, na koniec, lub nie mają być w ogóle prezentowane.
- Tekst wyświetlany pod poprawną oraz pod niepoprawną odpowiedzią.
- Tekst wyświetlany na zakończenie.
- Czas trwania egzaminu.

- Tekst przypominający o dobiegającym do końca czasie egzaminu oraz moment jego wyświetlenia.

Mając określone parametry testu przechodzimy do sekcji testu i dodajemy nową pozycję. Przy każdej sekcji można podać opis, informacje zwrotne oraz określić, czy elementy będą pokazywane losowo, czy w ustalonej kolejności. Następnie dokonujemy edycji nowopowstałej sekcji i wybieramy elementy testu. W tym przypadku pytania importujemy z „banku elementów testu”, który uprzednio stworzyliśmy. Do każdej sekcji możemy dodawać pytania z różnych banków elementów.

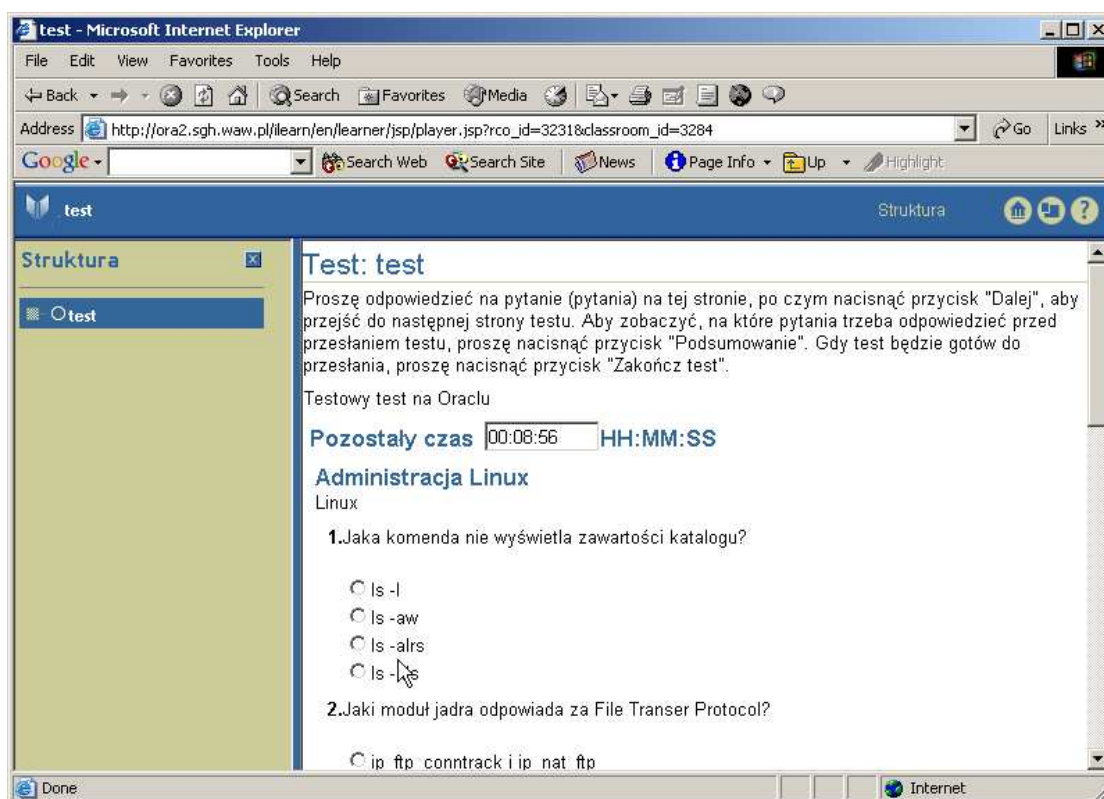
iLearning pozwala dodatkowo umieszczać odnośniki referencyjne do testów, oraz wymogi wstępne. Dzięki tej ostatniej funkcji wykładowca może wybrać inne testy, których zaliczenie jest konieczne do tego, aby student mógł podchodzić do danego sprawdzianu.

W ten sposób stworzony egzamin, może zostać za pomocą zakładki „właściwości” opublikowany, a tym samym udostępniony administratorom do dodania do oferty.

Ostatni etap przygotowania testu polega na wprowadzeniu go do odpowiedniej oferty, a co za tym idzie uczynieniu go widocznym dla zdających. Należy zmodyfikować właściwości testu i opcję „oferta”. Obiekt zostaje w tym momencie dodany do oferty z domyślnymi parametrami. Należy wybrać, z menu znajdującego się na górze po prawej stronie, opcję „oferty”, a następnie dokonać edycji odpowiedniej pozycji. Znajdują się tutaj opcje związane ze stylem nauczania (lista możliwych do wyboru stylów została zaprezentowana w podrozdziale 3.2 w podpunkcie 11), w którym ma ten test być udostępniony. Można ponadto zaplanować, kiedy test ma być dostępny, czy i ile razy może być powtarzany, a także wprowadzić ograniczenia zgłoszenia udziału. Dodatkowo można konfigurować odtwarzacz a więc np. dostępne przyciski (czat, E-mail, wstecz itp.).

Po dokonaniu powyższych czynności mamy przygotowany oraz udostępniony test dla

zdających. Każdy student zapisany na dany kurs może po zalogowaniu się na swoje konto systemu iLearning, wyszukać test i zgłosić w nim swój udział. Następnie określonego dnia lub od razu – w zależności od typu kursu (we własnym tempie lub zaplanowany) – może on uruchomić zdawanie (odtwarzanie) egzaminu (rys. 3.2).



Rys. 3.2: Okno zdającego podczas odpowiadania na pytania

Po rozpoczęciu egzaminu wyświetlane są formularze z pytaniami, pole tekstowe z odliczającym czas zegarem oraz przyciski: podsumowania oraz zakończenia. Na koniec, w zależności od ustawień, system prezentuje podsumowanie zdobytych punktów, procentową ocenę oraz ewentualnie tablicę ze szczegółowymi wynikami poszczególnych pytań.

3.4 Technologia modułu testującego

System iLearning jest typowym przedstawicielem klasy systemów tzw. *on-line based*, a więc zbudowanych w architekturze „cienki” klient – serwer.

Od strony osób chcących skorzystać ze szkoleń, za pomocą tego systemu, wymagane jest jedynie posiadanie komputera podłączonego do sieci Internet / Intranet, z zainstalowaną przeglądarką graficzną, obsługującą język Javascript oraz potrafiącą uruchamiać aplety Javy. Do tego celu najlepiej nadają się programy Microsoft Internet Explorer w wersji 5.0 lub nowszej oraz Netscape Navigator od wersji 4.7.

Po stronie serwera wymagania są znacznie większe. Obecnie najnowsza wersja 4.0 systemu iLearning pracuje na serwerze aplikacji Oracle 9iAS (Application Server), natomiast do przechowywania wszelkich danych potrzebny jest Relacyjny System Zarządzania Bazą Danych (RDBMS) Oracle w wersji 8.1.7 Enterprise Server lub nowszej. W kwestii sprzętu Oracle zaleca serwery firmy Sun Microsystems pracujące pod kontrolą systemów Sun Sparc Solaris. Ich szczegółowe konfiguracje są uzależnione od liczby potencjalnych użytkowników. W przypadku większych instalacji (dla więcej niż 100 użytkowników) zaleca się zakup trzech osobnych maszyn: jednej dla bazy danych Oracle, drugiej dla serwera aplikacji oraz trzeciej służącej do obsługi serwera zarządzającego treścią szkoleń oraz egzaminów.

3.5 Podsumowanie

Przedstawiona charakterystyka modułu pokazuje jak wiele znanych z teorii testowania typów zadań (zamkniętych: wielo-, jednokrotnego wyboru itp.) jest stosowanych w praktyce. Taka możliwość przenoszenia rozwiązań teoretycznych na implementację komputerowych systemów wspierających egzaminowanie świadczy o dużej wartości dydaktycznej omawianej aplikacji.

System Oracle iLearning jest profesjonalną, bardzo złożoną i rozbudowaną aplikacją,

służącą do przeprowadzania szkoleń. Jego specyfika wskazuje na to, że najlepiej nadaje się do zastosowania jako system wspomagający doksztalcenie kadry przedsiębiorstw. Jego wysokie wymagania sprzętowe (od strony serwera), konieczność stosowania dodatkowego komercyjnego oprogramowania, a także chyba przede wszystkim jego cena sprawiają, że przeznaczony jest on w głównej mierze dla firm dużych lub średnich.

Powyższe argumenty nie wykluczają używania go w instytucjach edukacyjnych. Należy jednak zauważyć, że jego moduł testujący nie powinien być stosowany jako jedyny program, służący do obiektywnej weryfikacji wiedzy, w placówkach nakierowanych na wydawanie certyfikatów. Z drugiej jednak strony dzięki wbudowanemu modułowi zarządzającemu stroną organizacyjną oraz finansową, związaną z prowadzeniem szkoleń, może on spełniać rolę systemu umożliwiającego zdalne nauczanie w firmach oraz instytucjach edukacyjnych. Systemy certyfikacyjne wymagają dodatkowo całościowej polityki i szerszej kontroli zdających, z tego tytułu obecnie poświęcimy im więcej uwagi, przedstawiając poniżej system egzaminujący firmy Prometric.

Rozdział 4

Profesjonalny system do przeprowadzania egzaminów certyfikacyjnych – system Prometric

4.1 Informacje ogólne o systemie

Firma Prometric jest aktualnie największą na świecie firmą pośredniczącą w przeprowadzaniu egzaminów certyfikacyjnych dla firm z branży informatycznej (IT Certification). Została założona w 1990 roku, od tego czasu była kilkakrotnie wykupywana, a obecnie należy do korporacji Thomson. Według danych z roku 2000, Prometric operował na terenie około 100 krajów i przeprowadził około siedmiu milionów egzaminów, uzyskując tym sposobem pierwsze miejsce na rynku [THOM03]. Aktualna oferta egzaminów zawiera około 1000 pozycji, spośród których znajdziemy testy dotyczące oprogramowania i sprzętu produkowanego przez wielu firm - gigantów świata informatyki. Wśród firm oferujących egzaminy za pośrednictwem Prometrica znajdziemy takie korporacje jak: IBM, Sun Microsystems, Oracle, Cisco, Microsoft, Novell oraz wiele innych. Egzaminy przeprowadzane są w przeszło 3500 autoryzowanych centrach egzaminacyjnych (ang. *Authorized Prometric Testing Center*). W Polsce funkcjonuje obecnie około 35 centrów, reprezentujących firmę [PROM03]. APTC oferuje możliwość weryfikacji

wiedzy informatycznej według światowych standardów. Dzięki zdaniu odpowiednich egzaminów można np. uzyskać jeden z prestiżowych dyplomów: *Microsoft Certified System Engineer* oraz *Certified Novell Engineer* [KOLB02, s. 164].

Opis funkcjonalny oraz scenariusz egzaminowania opracowałem, na podstawie rozmowy z administratorem jednego z warszawskich autoryzowanych centrów egzaminowania firmy Prometric.¹

4.2 Zakres funkcjonalności systemu

Do najważniejszych cech funkcjonalnych systemu egzaminacyjnego firmy Prometric można zaliczyć:

- Komputerowe przeprowadzanie egzaminów certyfikatowych.
- Automatyczne, natychmiastowe sprawdzanie poprawności udzielonych odpowiedzi i natychmiastowy wydruk zaświadczenia o zdanym egzaminie.
- Zabezpieczenie przed korzystaniem z innych aplikacji w czasie trwania egzaminu. Aplikacja zdającego blokuje możliwość przełączenia się do innych programów, a pasek zawierający przycisk „start” jest ukryty i nie może być uaktywniony bez podania hasła administratora APTC.
- Obsługiwane typy pytań: pytania jedno- i wielokrotnego wyboru, pytania z luką, pytania zawierające interaktywną grafikę, pytania polegające na dobieraniu elementów, pytania symulacyjne – mogą wymagać np. dokonania konfiguracji routera za pomocą symulatora konsoli urządzenia.
- Obsługa testów adaptacyjnych (CAT).

¹Za poświęconą mi czas bardzo dziękuję pani Barbarze Baszuń.

- W zależności od ustawień dostawcy egzaminu istnieje możliwość cofania się do poprzedniego pytania. Jeśli umożliwiające jest cofanie się do wcześniejszych pytań, pod koniec egzaminu prezentowana jest dodatkowo lista wszystkich zadań (ang. *Review list*), pozwalająca na „przeskoczenie” do dowolnego pytania.

Wymienione wyżej funkcje zostały bardziej szczegółowo opisane, przy okazji prezentowania scenariusza przeprowadzania egzaminu, z wykorzystaniem oprogramowania firmy Prometric.

4.3 Scenariusz egzaminu

Pierwszym etapem, związanym ze zdawaniem egzaminu za pośrednictwem sieci Prometric, jest dokonanie zgłoszenia. Klient może zgłosić się bezpośrednio do firmy Prometric za pośrednictwem serwisu WWW lub do wybranego centrum egzaminacyjnego. Drugi sposób jest wygodniejszy, gdyż w tym przypadku APTC przejmuje na siebie dopełnienie wszystkich niezbędnych formalności. W opisywanym przypadku założymy więc, że klient skontaktował się z APTC celem zgłoszenia chęci zdawania wybranego przez siebie egzaminu. Po odebraniu zgłoszenia od klienta, administrator centrum egzaminacyjnego loguje się do aplikacji administracyjnej i zamawia określony egzamin, podając kod egzaminu oraz dane osobowe zdającego. Po wypełnieniu formularza, serwer firmy Prometric, w sposób całkowicie automatyczny i transparentny dla administratora, przesyła do aplikacji administracyjnej (na serwer centrum egzaminacyjnego) gotowy plik, zawierający cały egzamin, przeznaczony do uruchomienia dla konkretnego zdającego. Ściągnięty w ten sposób plik może być uruchomiony w aplikacji stanowiska egzaminującego jedynie po wprowadzeniu numeru identyfikacyjnego osoby, dla której został zamówiony. Nie ma więc możliwości, aby APTC zmieniło dane osobowe zdającego po dokonaniu zamówienia w serwisie firmy Prometric. Połączenie APTC z serwerem Prometric jest realizowane za pośrednictwem modemu analogowego, lub w nowszych rozwiązaniach, poprzez sieć Internet.

Umówionego dnia, o ustalonej z APTC godzinie, klient przychodzi do centrum egzamina-

cyjnego, siada przed komputerem pełniącym rolę stanowiska egzaminującego i wybiera spośród dostępnych, przygotowanych wcześniej przez administratora, egzaminów ten, na który się zapisał, a następnie wprowadza swój, unikalny numer identyfikacyjny. W pierwszym kroku system prezentuje dane osobowe zdającego, a następnie umożliwia uruchomienie, składającego się z 12 pytań, egzaminu próbnego. Aktualnie są to pytania z dziedziny astronomii. Podczas tego testu zdający zapoznawany jest ze wszystkimi możliwymi typami pytań. Po zakończeniu lub pominięciu testu próbnego, zdający uruchamia test właściwy. W trakcie trwania egzaminu nie może on przełączyć się do żadnego innego programu, gdyż aplikacja egzaminująca przejmuje pełną kontrolę nad systemem operacyjnym. Po zakończeniu zdawania, program oblicza wynik i najczęściej prezentuje go zdającemu. W zależności od dostawcy, może to być jedynie wynik typu „zdał” „nie zdał” (tak jest w przypadku większości egzaminów firmy Microsoft) lub szczegółowy raport, zawierający podział na kategorie pytań i procentowe osiągnięcia zdającego w każdej dziedzinie (tak jest np. w przypadku egzaminów firm: Sun Microsystems i Cisco). Jednocześnie w sposób automatyczny drukowane jest oświadczenie o zdaniu lub niezdaniu testu. Wyniki odbytego egzaminu trafiają na serwer centrum egzaminacyjnego, skąd są, za pośrednictwem panelu administracyjnego, przesyłane przez administratora do centrali firmy Prometric. Zazwyczaj, po około trzech dniach, zdający może sprawdzić wyniki egzaminu w Internecie, po zalogowaniu się do serwisu dostawcy egzaminu. Po określonym przez dostawcę czasie klientowi dostarczany jest certyfikat potwierdzający zdanie odpowiedniego egzaminu.

4.4 Wymagania sprzętowe centrum egzaminowania

Autoryzowane centrum egzaminowania Prometric musi być wyposażone w specjalną salę egzaminacyjną, w której znajdują się maszyny – stanowiska egzaminacyjne. Wszystkie komputery muszą być połączone za pośrednictwem sieci komputerowej z serwerem centrum egzaminacyjnego. Firma Prometric zaleca następującą minimalną konfigurację sprzętową dla serwera APTC:

- Komputer typu IBM PC minimalnie z procesorem Pentium II, 266 MHz.
- 128 MB RAM.

- Dysk twardy o dostępnej dla aplikacji Prometric pojemności 2GB.
- CD ROM minimum 8-krotnej prędkości.
- Monitor.
- Kartę sieciową.
- System operacyjny: Microsoft NT 4.0 z Service Pack 4, lub Microsoft 2000 Server, albo Novell NetWare 4.11.
- Internet Explorer 5.5 Service Pack 2.
- Modem analogowy z podłączeniem do linii telefonicznej oraz Internetu.
- Oprogramowanie antywirusowe.

Wymagania sprzętowe dla stacji administratora oraz stanowisk egzaminacyjnych są zbliżone, dlatego prezentuję poniżej jedynie specyfikacje komputerów – stanowisk egzaminacyjnych.

- Komputer typu IBM PC minimalnie z procesorem Pentium II, 266 MHz zawierający:
- 128 MB RAM.
- Dysk twardy o pojemności 2GB.
- CD ROM minimum 8-krotnej prędkości.
- Monitor kolorowy SVGA o minimalnej rozdzielczości 800 na 600 pikseli przy 16 bitowej głębi kolorów.
- Kartę sieciową.
- System operacyjny: Microsoft Windows NT 4.0 Workstation, Microsoft Windows 2000 Professional, lub Microsoft Windows 98 Drugiej Edycji.
- Internet Explorer 5.5 Service Pack 2.
- Oprogramowanie antywirusowe.

4.5 Podsumowanie

System egzaminowania firmy Prometric jest bardzo dobrze dopasowany do potrzeb centrów egzaminacyjnych tej firmy. Dzięki bardzo dużej elastyczności form pytań, możliwe jest dostosowanie egzaminów do potrzeb firm zlecających, dostarczających i przygotowujących treść merytoryczną zestawów pytań. Aplikacja stanowiska egzaminacyjnego dobrze zabezpiecza komputer zdającego, przed możliwościami porozumiewania się osoby zdającej z osobami z zewnątrz, innymi zdającymi lub korzystania z nieautoryzowanych źródeł za pomocą Internetu, lub sieci komputerowej. Z drugiej strony, dzięki temu, że do lokalnego APTC trafia gotowy plik, zawierający wszystkie dane potrzebne do przeprowadzenia testu, system jest także dobrze chroniony, przed dostępem do treści pytań i danych egzaminów, ze strony osób pracujących w centrach egzaminacyjnych.

Istnieje jednak możliwość wdrożenia węższych koncepcji systemów testujących dla potrzeb środowiska akademickiego w ramach lokalnego Uczelnianego Centrum Egzaminacyjnego. Przykład takiego rozwiązania stanowi opisany poniżej autorski system KSE, opracowany w Szkole Głównej Handlowej w Warszawie.

Rozdział 5

Komputerowy System Egzaminowania (KSE) – opracowany w SGH

5.1 Cele projektu KSE

Komputerowy System Egzaminowania to system typu CBA (ang. *Computer-based assessment*), rozwijany w Katedrze Informatyki Gospodarczej Szkoły Głównej Handlowej przez grupę studentów: Szymona Purę, Krzysztofa Darmetko i Pawła Rubacha, pod kierownictwem dr Anny Rokickiej-Broniatowskiej. Prace nad projektem rozpoczęły się w kwietniu 2002 roku. Na początku grudnia 2002 roku udało się przeprowadzić pierwszy egzamin z wykorzystaniem jeszcze nie w pełni funkcjonalnego, ale już działającego systemu.

Podczas opracowywania projektu założono zbudowanie systemu informatycznego, który usprawniłby proces weryfikacji wiedzy i odciążył wykładowców od pracy związanej z egzaminowaniem studentów. Powstały system zakłada realizację następujących celów:

1. *Automatyzacja procesu przeprowadzania egzaminu*

Funkcja ta ma na celu zautomatyzowanie wszystkich aspektów egzaminowania. Oznacza to, że wykładowca, który korzysta z systemu, podczas całego procesu egzaminowania ma możliwość wykonania jedynie trzech czynności:

- przygotowania pytań,
- ustawienia opcji i uruchomienia egzaminu,
- sprawdzenia i zatwierdzenia wyników za pośrednictwem sieci komputerowej.

Oznacza to, że w zestawieniu z tradycyjną metodą przeprowadzania egzaminu pisemnego można ominąć etap powielania kart z pytaniami, sprawdzania prac oraz wypełniania protokołu. Powyższe czynności może realizować system informatyczny.

2. *Archiwizowanie szczegółowych wyników egzaminu*

Po przeprowadzonym egzaminie wszystkie udzielone odpowiedzi powinny zostać zapisane tak, żeby wykładowca mógł w każdej chwili sprawdzić kto, na jakie pytanie i jak odpowiedział. Ta funkcjonalność może być zapewniona dzięki zastosowaniu centralnej bazy danych pytań i odpowiedzi.

3. *Wyeliminowanie czynnika ludzkiego z procesu sprawdzania prac*

Podczas przeprowadzania tradycyjnego egzaminu pisemnego istnieje ryzyko, że osoba sprawdzająca przeoczy właściwą odpowiedź lub np. pomyli się przy sumowaniu wyniku. Zastosowanie maszynowego oceniania prac powinno wyeliminować wpływ błędu ludzkiego na oceny osób zdających.

4. *Skrócenie czasu oczekiwania na ocenę z egzaminu*

Dzięki maszynowemu ocenianiu prac możliwe jest zaprezentowanie wyników tuż po zakończeniu odpowiadania na pytania. Powinno to zaowocować ogromną oszczędnością czasu zarówno wykładowcy jak również studentów, gdyż nie będzie konieczne ponowne spotkanie się z wykładowcą celem uzyskania wpisu do indeksu.

5. *Ograniczenie lub nawet wyeliminowanie problemu ściągania podczas egzaminu*

Poprzez zastosowanie algorytmu losującego pytania ze zbioru wszystkich pytań przygotowanych na dany egzamin przez egzaminatora, a także zastosowanie losowej kolejności

odpowiedzi, można zmniejszyć prawdopodobieństwo otrzymania przez osoby egzaminowane, podobnych zestawów pytań. W celu wyeliminowania możliwości porozumiewania się osób zdających, za pośrednictwem sieci komputerowej lub Internetu, aplikacja – pełniąca rolę stanowiska egzaminacyjnego, powinna uniemożliwiać korzystanie z przeglądarki internetowej oraz innych aplikacji zainstalowanych na komputerze zdającego, przez cały czas trwania egzaminu.

6. *Zapewnienie poufności danych osobowych oraz treści pytań i odpowiedzi*

Dzięki zastosowaniu odpowiednich metod autoryzacji i autentykacji system powinien być zabezpieczony przed wszelkimi próbami włamania. Ponadto, aby zapobiec przechwyceciu cennych informacji podczas przesyłania danych przez sieć, wykorzystywane protokoły transmisji powinny być szyfrowane.

7. *Łatwość i intuicyjność w obsłudze*

Zbudowany system powinien być na tyle przyjazny dla użytkownika, aby nie wymagał przeprowadzania żadnych szkoleń, zarówno dla egzaminatorów, jak i również dla egzaminowanych.

8. *Brak konieczności zakupu dodatkowych licencji na oprogramowanie*

We wstępnych założeniach przyjęto, że system powinien w miarę możliwości wykorzystywać jedynie oprogramowanie darmowe tak, żeby nie narażać instytucji pragnącej wykorzystać KSE na koszty zakupu licencji na oprogramowanie firm trzecich.

9. *Skalowalność oraz bezawaryjność pracy systemu*

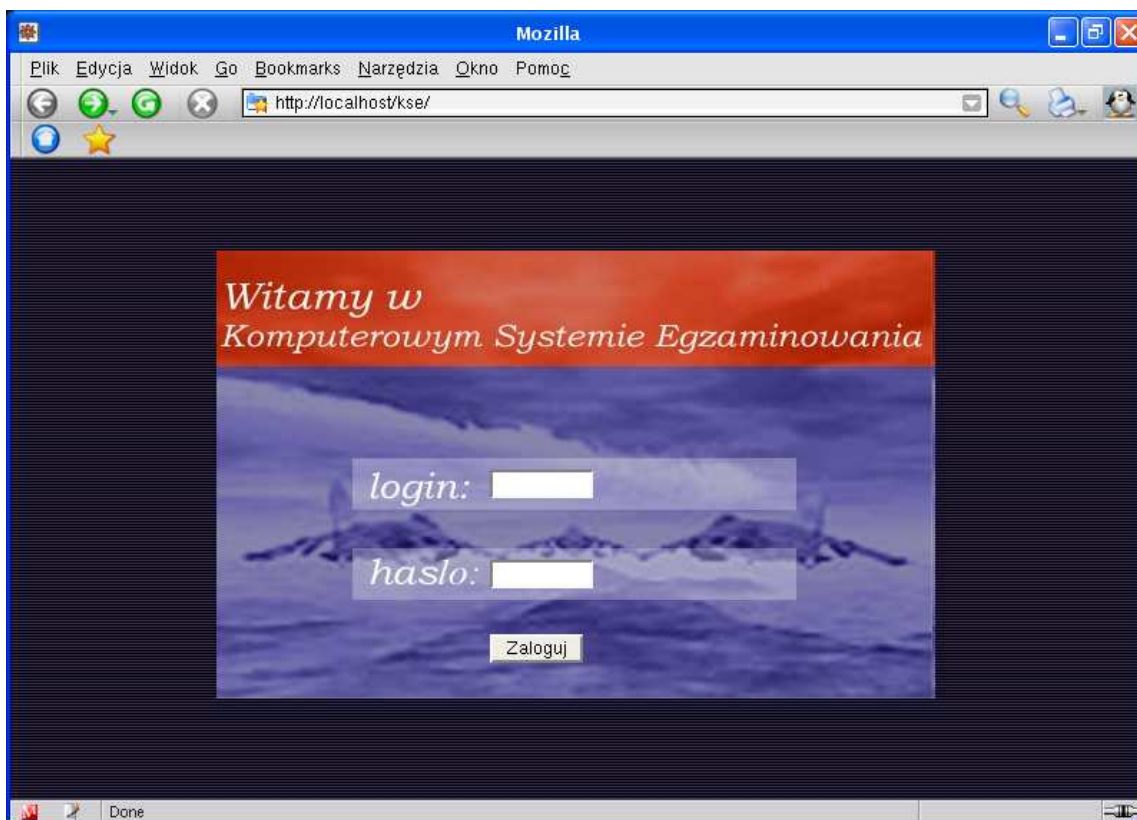
W rozwiązaniach systemu starano się położyć duży nacisk na wysoką niezawodność aplikacji oraz na rozłożenie obciążenia pracy procesora pomiędzy serwer a maszyny, pełniące rolę stanowisk egzaminacyjnych tak, aby umożliwić przeprowadzanie egzaminów nawet dla kilkuset osób jednocześnie.

10. *Uniwersalność*

Przyjęto, że system powinien być tak skonstruowany, aby umożliwić łatwe przeniesienie do zupełnie innego środowiska niż to, dla którego został zaprojektowany. Taką uniwersalność powinny zapewnić: budowa modułowa oraz szerokie możliwości konfiguracji, zarówno samego egzaminu, jak również całego środowiska informatycznego.

5.2 Historia powstania projektu KSE

Powstanie projektu KSE wiąże się ściśle z pojawieniem się potrzeby wdrożenia systemu wspomagającego proces egzaminowania wśród pracowników Katedry Informatyki Gospodarczej. Bezpośrednią odpowiedzią na tę potrzebę był poprzednik KSE, projekt aplikacji egzaminującej nazwanej odpowiednio „Egzamin – serwer” i „Egzamin – klient”. Obydwa programy zostały napisane samodzielnie przez Szymona Purę, na potrzeby egzaminu końcowego z przedmiotu „Inżynieria Oprogramowania”. Były to aplikacje napisane w języku C++, wykorzystujące interfejs środowiska MS Windows oraz zwykły plik tekstowy do przechowywania zarówno pytań, odpowiedzi, jak również wyników przeprowadzonego egzaminu. Projekt KSE stanowi rozwinięcie koncepcji systemu „Egzamin”. Dzięki połączeniu doświadczeń z budowy aplikacji w architekturze klient – serwer dla środowiska okienkowego MS Windows, z doświadczeniami w zakresie konstruowania aplikacji internetowych współpracujących z bazą danych typu SQL, powstał złożony system, który doskonale nadaje się do zastosowania w dużych instytucjach, takich jak uczelnie wyższe. Jego wyższość nad prototypowym systemem „Egzamin” polega przede wszystkim na wykorzystaniu bazy danych do przechowywania pytań, wszelkich ustawień a także wyników. Ponadto dzięki zastosowaniu technologii budowy aplikacji internetowych do stworzenia panelu administracyjnego (rysunek 5.1 znajdujący się na następnej stronie pokazuje okno logowania do panelu) możliwe jest przygotowywanie egzaminu, wprowadzanie pytań oraz sprawdzanie wyników bez konieczności instalacji jakiegokolwiek oprogramowania na stacji roboczej.



Rys. 5.1: Okno logowania do panelu administracyjnego systemu KSE

5.3 Zakres funkcjonalności systemu KSE

Po zarysowaniu celów, jakie postawiono jeszcze przed rozpoczęciem projektu, oraz naszkicowaniu krótkiej historii zdarzeń, która doprowadziła do rozpoczęcia prac, chciałbym teraz szczegółowo opisać od strony funkcjonalnej efekt tych prac. W punktach poniżej wymienione są szczegółowe funkcje KSE.

1. *Umożliwienie przeprowadzania testów zamkniętych pojedynczego lub wielokrotnego wyboru.*

W danej chwili do systemu możemy wprowadzać pytania testowe zamknięte, zawierające od dwóch do pięciu odpowiedzi, z czego dowolna liczba odpowiedzi może być prawidłowa. W przypadku pytań wielokrotnego wyboru istnieje ponadto możliwość oceniania

każdej odpowiedzi osobno. Daje to odpowiednik pytań, w których zarysowuje się problem, a następnie podaje kilka zdań, przy których należy podać odpowiedź TAK lub NIE. Obecnie zarówno pytania jak również odpowiedzi, mogą mieć jedynie formę tekstową. Jednym z pomysłów na rozszerzenie funkcjonalności KSE jest dodanie możliwości prezentowania rysunków, wykresów oraz grafiki do treści pytań. Kolejny sposób formułowania pytań i odpowiedzi, który mógłby zostać dodany pod postacią modułu do projektu, omówiony jest pod koniec niniejszej pracy. Jest to problem automatycznej analizy i oceny odpowiedzi otwartych pisanych w języku naturalnym, czyli np. po polsku.

2. *Natychmiastowe sprawdzanie poprawności odpowiedzi oraz wystawianie ocen (proces sprawdzania przeprowadzany komputerowo, z pominięciem czynnika ludzkiego) .*

Wykładowca konstruując egzamin może zdecydować, czy wynik punktowy oraz ocena mają być automatycznie prezentowane osobie zdającej, czy też mają być wyświetlane jedynie na ekranie wykładowcy i dostępne później za pomocą panelu administracyjnego. Z wystawianiem ocen wiąże się problem ustalenia progów punktowych dla ocen. Wykładowca może dowolnie konfigurować skalę ocen podczas tworzenia zestawu pytań.

3. *Archiwizacja danych osób zdających i udzielonych odpowiedzi .*

Wszystkie udzielone odpowiedzi, a także czas spędzony przez studenta nad każdym z pytań jest zapisywany do bazy danych w chwili zakończenia przeprowadzania egzaminu. Od tego momentu wykładowca może wyświetlić wszystkie te dane korzystając z panelu administracyjnego.

4. *Możliwość wielokrotnego wykorzystywania zestawów pytań.*

Raz skonstruowany zestaw pytań, składający się z poszczególnych pytań pogrupowanych w kategorie, może być wielokrotnie wykorzystywanych podczas kolejnych egzaminów w ramach jednego przedmiotu.

5. *Możliwość losowego konstruowania pojedynczego zestawu pytań egzaminacyjnych .*

Podczas konstruowania zestawu pytań wykładowca dzieli pytania na kategorie i ustala

liczbę pytań, które mają być wylosowane z każdej kategorii. Dodatkowo, podczas wprowadzania pytań egzaminator decyduje, czy poszczególne odpowiedzi mają być wyświetlane osobie zdającej w losowej kolejności.

6. *Import pytań z pliku pojedynczego pliku zewnętrznego,*

Osoba egzaminująca może, zamiast wprowadzania pytań na bieżąco za pośrednictwem sieci komputerowej (Intranetu), pobrać szablon w postaci pliku MS Excel, następnie uzupełnić go korzystając ze swojego komputera np. w domu oraz załadować na serwer bazy danych poprzez panel administracyjny.

7. *Możliwość wprowadzania ocen z ćwiczeń i automatycznego wystawiania ocen końcowych z przedmiotu.*

KSE posiada panel administracyjny dla osób prowadzących ćwiczenia. Za pomocą tego narzędzia można wprowadzić oceny z ćwiczeń, których system może użyć do zweryfikowania, czy dany student jest dopuszczony do egzaminu. Dodatkowo wykładowca definiuje procentową wagę oceny z ćwiczeń i egzaminu, na podstawie której KSE wylicza ocenę końcową z przedmiotu.

8. *Możliwość integracji z istniejącym systemem (np. z Wirtualnym Dziekanatem w przypadku SGH) w celu pobierania listy uprawnionych do zdawania osób i przysyłania wyników egzaminu.*

W krótkim czasie możliwe jest zbudowanie interfejsu umożliwiającego wymianę informacji pomiędzy bazą danych KSE a systemem informatycznym obsługującym dziekanat. Konkretnie rozwiązanie zależy od typu bazy danych pracującej w systemie uczelnianym oraz od organizacji danych w tejże bazie danych. Integracja np. z Wirtualnym Dziekanatem (w SGH) pozwoliłaby na zaimportowanie listy uprawnionych do zdawania danego egzaminu studentów. Dzięki temu możliwa byłaby weryfikacja za pomocą systemu, czy osoba zdająca widnieje w protokole. Ponadto zaoszczędziłoby to pracy przy wprowadzaniu ocen z ćwiczeń, gdyż ćwiczeniowiec nie musiałby wpisywać imienia i nazwiska, a jedynie numer albumu, za pomocą którego KSE identyfikowałby osobę.

9. *Możliwość przeprowadzania centralnych egzaminów z przedmiotów, prowadzonych przez wielu wykładowców jednocześnie.*

W KSE wyodrębniliśmy role wykładowcy i koordynatora przedmiotu. Koordynator ma uprawnienia konstruowania zestawów pytań i egzaminów, wykładowca natomiast posiada jedynie możliwość wglądu w wyniki przeprowadzonego uprzednio egzaminu. Taka właśnie konstrukcja umożliwia organizację przy pomocy KSE centralnych egzaminów z przedmiotów prowadzonych przez wielu wykładowców.

10. *Możliwość jednoczesnej obsługi wielu egzaminów z wielu przedmiotów – system użytkowników i haseł dostępu dla koordynatorów egzaminu.*

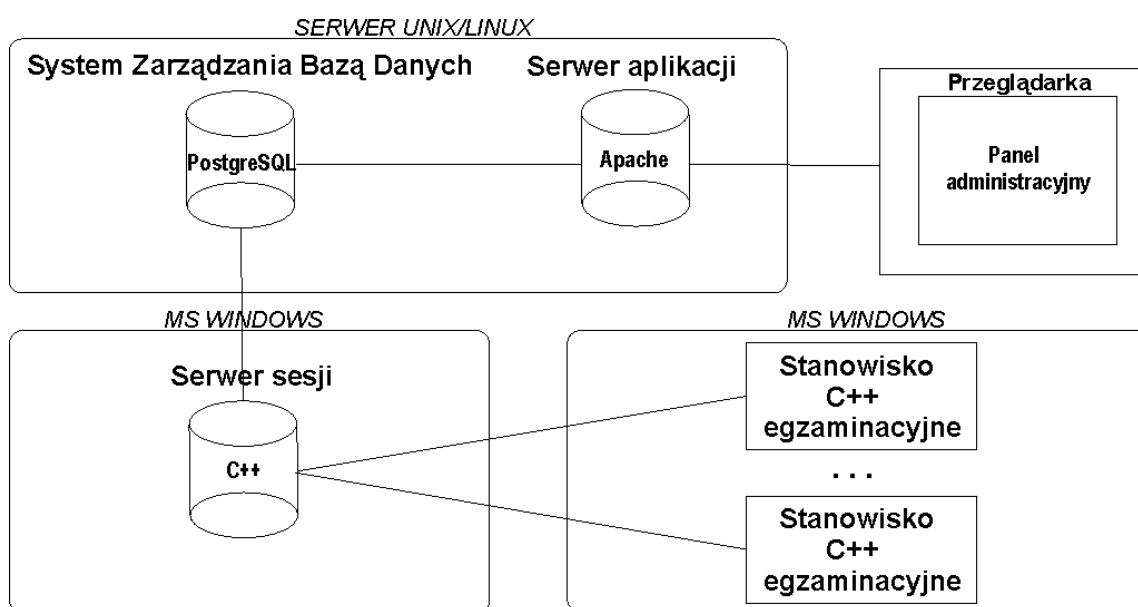
Modułowa budowa oraz skalowalność uzyskana dzięki zastosowaniu architektury wielowarstwowej umożliwia uruchamianie w tej samej sieci komputerowej wielu egzaminów z różnych przedmiotów. Dla każdego przedmiotu określony jest koordynator, który zarządza treścią pytań, a także tworzeniem kont dla użytkowników pełniących role wykładowcy oraz ćwiczeniowca. Nad wszystkimi użytkownikami systemu czuwa administrator, którego zadaniem jest tworzenie przedmiotów oraz kont koordynatorów. To on także przypisuje przedmioty poszczególnym koordynatorom.

11. *Możliwość analizy statystycznej wyników egzaminu; także badanie statystyczne poziomu trudności poszczególnych pytań w zestawie.*

Dzięki temu, że wszystkie odpowiedzi trafiają od razu do bazy danych można obliczyć różnego rodzaju statystyki, dzięki którym wykładowca może się dowiedzieć, które pytania stanowiły największą trudność, które były najłatwiejsze, jaki był średni czas odpowiadania na pytania a także zdobyć wiele innych, cennych informacji, które pomogą mu w skonstruowaniu, następnym razem, bardziej odpowiedniego zestawu pytań. Obecnie zaimplementowany raport statystyczny zawiera rozkład ocen, łącznie ze średnią oceną, jej odchyleniem standardowym i średnią liczbę oraz odchylenie standardowe, uzyskanych przez studentów punktów. Możliwości rozbudowy raportu są bardzo duże.

5.4 Architektura aplikacji wchodzących w skład projektu KSE

System KSE ma wielowarstwową budowę, co widać na zamieszczonym poniżej rysunku 5.2. Poniżej zawarto wyjaśnienie, dlaczego taką architekturę wybrano, a także jakie czynniki zadecydowały o wyborze narzędzi programowania, Systemu Zarządzania Bazą Danych oraz serwera aplikacji.



Rys. 5.2: Architektura systemu KSE

KSE zbudowany jest z czterech warstw oprogramowania. Centralne miejsce w systemie zajmuje System Zarządzania Bazą Danych (SZBD), z którym po jednej stronie połączony jest panel administracyjny, działający na serwerze aplikacji internetowych, z drugiej zaś strony tzw. serwer sesji – aplikacja uruchamiana na czas trwania egzaminu przez egzaminatora. Ostatnią najbardziej „wierzchnią” warstwę stanowi oprogramowanie stanowiska egzaminacyjnego – samodzielna aplikacja komunikująca się z serwerem sesji. Do komunikacji między poszczegól-

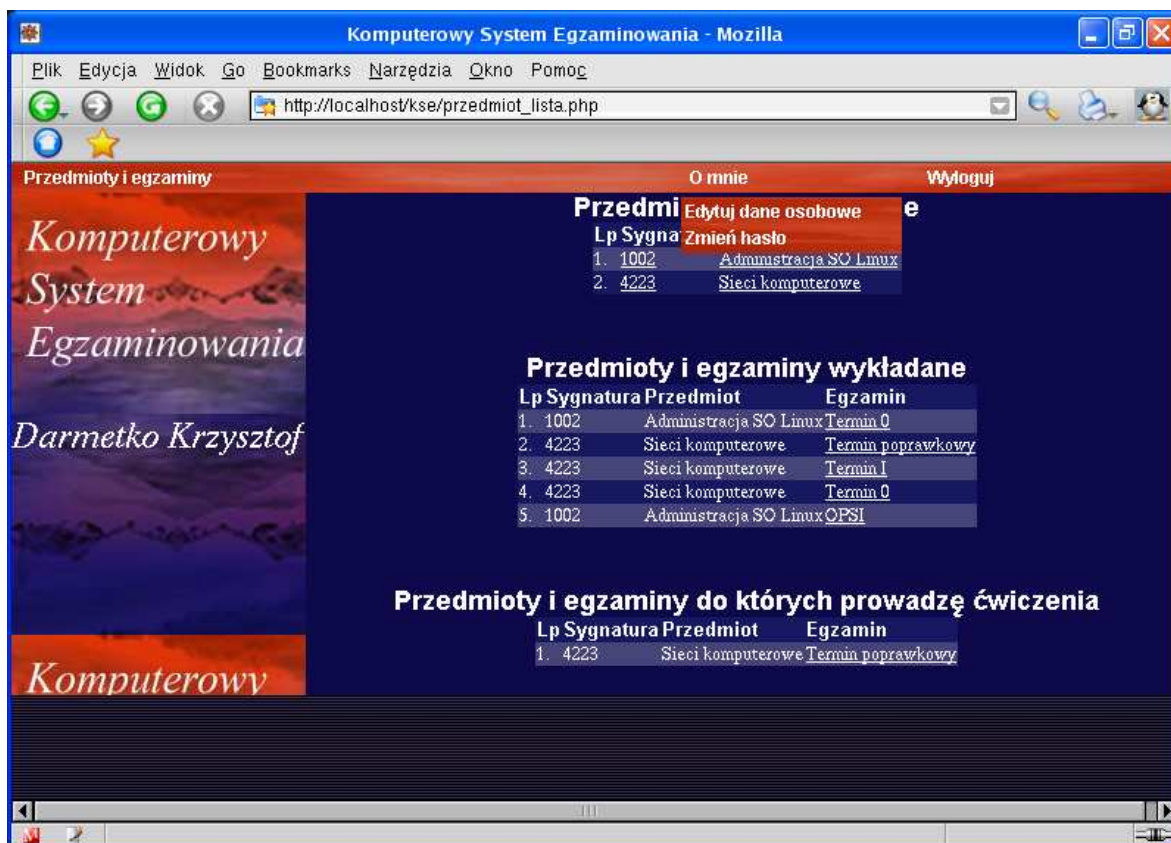
nymi warstwami systemu używana jest sieć komputerowa oparta na protokole TCP/IP (Transfer Control Protocol / Internet Protocol).

5.4.1 System Zarządzania Bazą Danych

W celu przechowywania ustawień i treści pytań oraz odpowiedzi wykorzystano relacyjną bazę danych, zgodną ze standardem SQL. Istnieje wiele komercyjnych SZBD spełniających ten warunek, takich jak choćby Oracle, DB2, Informix, Sybase czy MSSQL Server. Ponadto dostępnych jest kilka darmowych implementacji. Spośród nich najbardziej popularne to MySQL i PostgreSQL. W KSE postanowiliśmy ograniczyć wykorzystywanie oprogramowania komercyjnego do niezbędnego minimum. Dlatego wybraliśmy jeden z darmowych SZBD – PostgreSQL. Wśród argumentów, które przeważyły na jego korzyść należy wymienić: możliwość wykonywania zapytań zagnieżdżonych, bardziej klarowny – naszym zdaniem – system praw dostępu do danych, obsługę języka PL/SQL umożliwiającego tworzenie skryptów przechowywanych i wykonywanych w serwerze bazy danych. Dodatkowo, ze względu na wymianę danych z systemami zewnętrznymi, dosyć istotnym argumentem okazał się fakt, iż działający w Szkole Głównej Handlowej Wirtualny Dziekanat wykorzystuje właśnie ten SZBD. Poniżej znajdują się podrozdziały opisujące kolejne warstwy projektu.

5.4.2 Panel administracyjny – serwer aplikacji

Panel administracyjny (rys. 5.3 na następnej stronie) służy do zdalnego administrowania systemem poprzez stronę intranetową. Do jego napisania wykorzystano bardzo powszechny język skryptowy: Personal Hypertext Preprocessor (PHP), który należy do języków interpretowalnych, wykonywanych po stronie serwera WWW. Spośród podobnych technologii, takich jak: Perl, Java Server Pages (JSP) i Active Server Pages (ASP) wybraliśmy właśnie PHP ze względu na większą wydajność (niż np. JSP), a także brak koniecznych opłat licencyjnych na serwer WWW (technologia ASP wymaga serwera Internet Information Server (IIS) firmy Microsoft, który z kolei działa jedynie na serwerach Windows NT/2000/XP). Tak więc panel potrzebuje serwera WWW, obsługującego język skryptowy PHP.

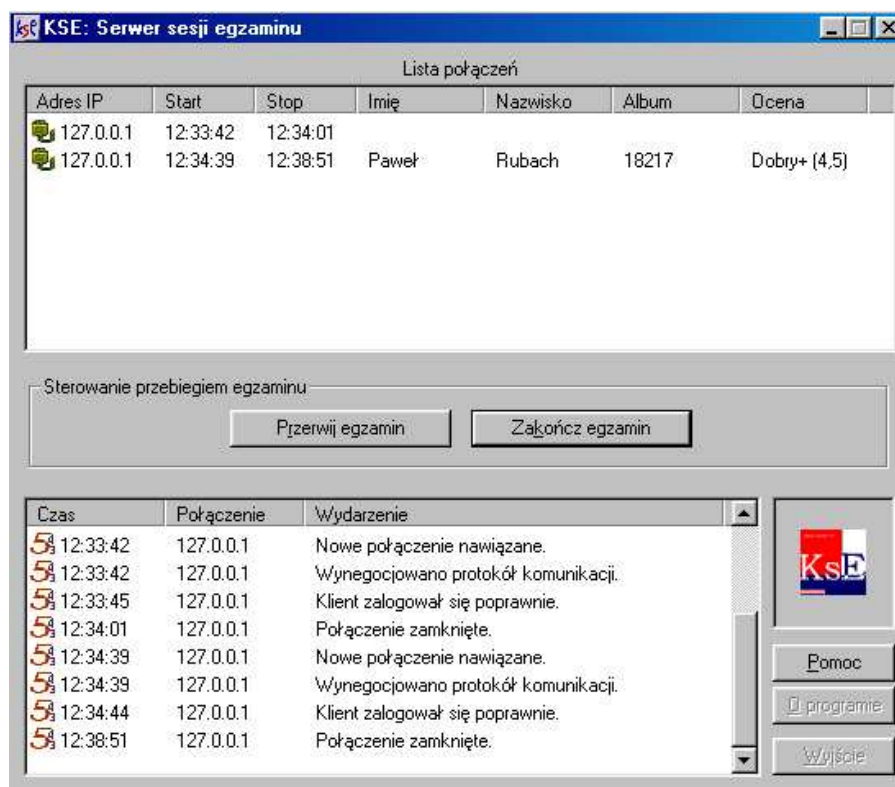


Rys. 5.3: Panel administracyjny – okno wyboru przedmiotu

Najlepszym przykładem takiego programu jest najbardziej powszechny serwer, występujący w wersjach dla bardzo wielu systemów operacyjnych, publicznie dostępny Apache. Całe oprogramowanie tej warstwy, a także relacyjna baza danych PostgreSQL, może funkcjonować na dowolnym systemie operacyjnym. Ze względu na konieczne w procesie egzaminowania stabilność i bezpieczeństwo, najlepszym rozwiązaniem wydaje się być platforma UNIX/Linux.

5.4.3 Serwer sesji

Jest to aplikacja uruchamiana przez egzaminatora oddzielnie dla każdego egzaminu, tylko na czas jego trwania. Stanowi ona warstwę pośrednią pomiędzy bazą danych, a oprogramowaniem stanowiska egzaminacyjnego.



Rys. 5.4: Okno serwera sesji przedstawiające trwający egzamin

Uruchamiając ją egzaminator decyduje z jakiego przedmiotu i który egzamin może być aktualnie przeprowadzany. W trakcie działania, program pobiera treści pytań oraz parametry egzaminu z bazy danych, a następnie udostępnia je aplikacjom – klientom, przy których studenci zdają egzamin. Wszystkie połączenia ze stanowiskami egzaminacyjnymi są na bieżąco monitorowane. Na ekranie egzaminatora pojawiają się dane osobowe oraz oceny osób (rys. 5.4), które zdecydowały się zakończyć odpowiadanie na pytania. Po zakończeniu egzaminu aplikacja przesyła szczegółowe wyniki do centralnej bazy danych, czyniąc je w ten sposób dostępnymi dla wykładowców, za pośrednictwem panelu administracyjnego. Dzięki opcji zmiany portu, na którym nasłuchuje serwer sesji, możliwe jest uruchomienie wielu serwerów sesji jednocześnie, a co za tym idzie, można przeprowadzać wiele egzaminów w tym samym czasie. Aplikacja została napisana w języku C++ i działa w środowisku Windows 98/NT/2000/XP.

5.4.4 Oprogramowanie stanowiska egzaminującego

Oprogramowanie stanowiska egzaminacyjnego to aplikacja służąca studentom do odpowiadania na pytania. W celu uruchomienia programu, potrzebna jest znajomość tzw. hasła sesji – ciągu liter zaprogramowanego w trakcie uruchomienia serwera sesji przez egzaminatora. Hasło sesji zabezpiecza przed tym, aby ktoś z zewnątrz, nawet po wejściu w posiadanie aplikacji klienckiej, nie mógł wystartować programu i zdawać egzaminu poza kontrolą osób prowadzących. Po wprowadzeniu hasła sesji program uruchamia się w trybie pełnoekranowym, uniemożliwiając korzystanie z żadnych innych zainstalowanych na danej maszynie aplikacji (rys. 5.5).

The screenshot shows a software interface for an exam station. At the top left, the user's name and ID are displayed: "Paweł Rubach (18217)". At the top right, a timer shows "0:03:41". The main area is titled "Pytanie 5/5" and contains the question "Kto napisał jądro Linuxa?". Below the question, a instruction reads: "Wskaż dokładnie jedną poprawną odpowiedź spośród A-D:". There are four radio button options: "Linus Torvalds", "Sahra Linux", "Marco Linus", and "Linus Franc". The first option, "Linus Torvalds", is selected. At the bottom, there are navigation buttons: "<< Poprzednie", "Następne >>", and "Koniec". A progress bar is located between the first and second buttons.

Rys. 5.5: Okno stanowiska egzaminacyjnego w trakcie trwania egzaminu

Po zakończeniu udzielania odpowiedzi przez studenta program wyświetla wynik egzaminu, a następnie przesyła szczegółowe odpowiedzi do serwera sesji. Aplikacja została napisana, podobnie jak serwer sesji, w języku C++. Najlepiej pracuje ona w środowisku Windows 98/NT/2000/XP. Jako rozwiązanie przyszłościowe należy traktować możliwość uruchamiania stanowiska klienckiego na maszynach UNIX/Linux za pośrednictwem darmowego emulatora MS Windows o nazwie WINE. Wykorzystanie emulatora umożliwiłoby zaoszczędzenie na kosztach zakupu licencji na systemy operacyjne firmy Microsoft.

Z przedstawionego powyżej opisu widać, że struktura KSE jest znacznie bardziej skomplikowana niż architektura typowych systemów klasy CBA, opartych na technologii „cienkiego” klienta, a więc wykorzystujących, po stronie studenta i wykładowcy, jedynie przeglądarkę internetową. Naszym zdaniem taka właśnie architektura, biorąc pod uwagę specyfikę zastosowań systemu, spełnia najlepiej swoje zadanie. Dodatkowa warstwa serwera sesji zapewnia bowiem zmniejszenie obciążenia sieci komputerowej, oraz daje większą kontrolę nad przeprowadzanym egzaminem. Dzięki temu, że wykorzystywane aplikacje pracują na różnych maszynach, zmniejsza się obciążenie serwera bazy danych, oraz serwera aplikacji. Wpływa to także pozytywnie na stabilność systemu. KSE jest ponadto odporny na chwilowe przerwy w połączeniach sieciowych, ponieważ najważniejsza jego część, a więc aplikacje serwer sesji oraz stanowisko egzaminacyjne, wykorzystują transmisję asynchroniczną.

5.5 Przykładowy scenariusz przeprowadzania egzaminu z wykorzystaniem KSE

Przeprowadzanie egzaminu drogą elektroniczną z wykorzystaniem Komputerowego Systemu Egzaminowania składa się z ośmiu lub dziewięciu etapów. Liczba etapów jest uzależniona od tego, czy dany przedmiot prowadzony jest w formie wykładu oraz osobnych ćwiczeń, czy też przyjmuje on tylko jedną z tych form. W poniższym opisie posłużymy się przykładem przedmiotu, wykładanego jednocześnie przez wielu wykładowców, do którego prowadzi się odrębne

ćwiczenia i który posiada koordynatora, odpowiedzialnego za przygotowanie i przeprowadzenie egzaminu dla wszystkich studentów. W tym wypadku wyróżniamy następujące etapy pracy z systemem.

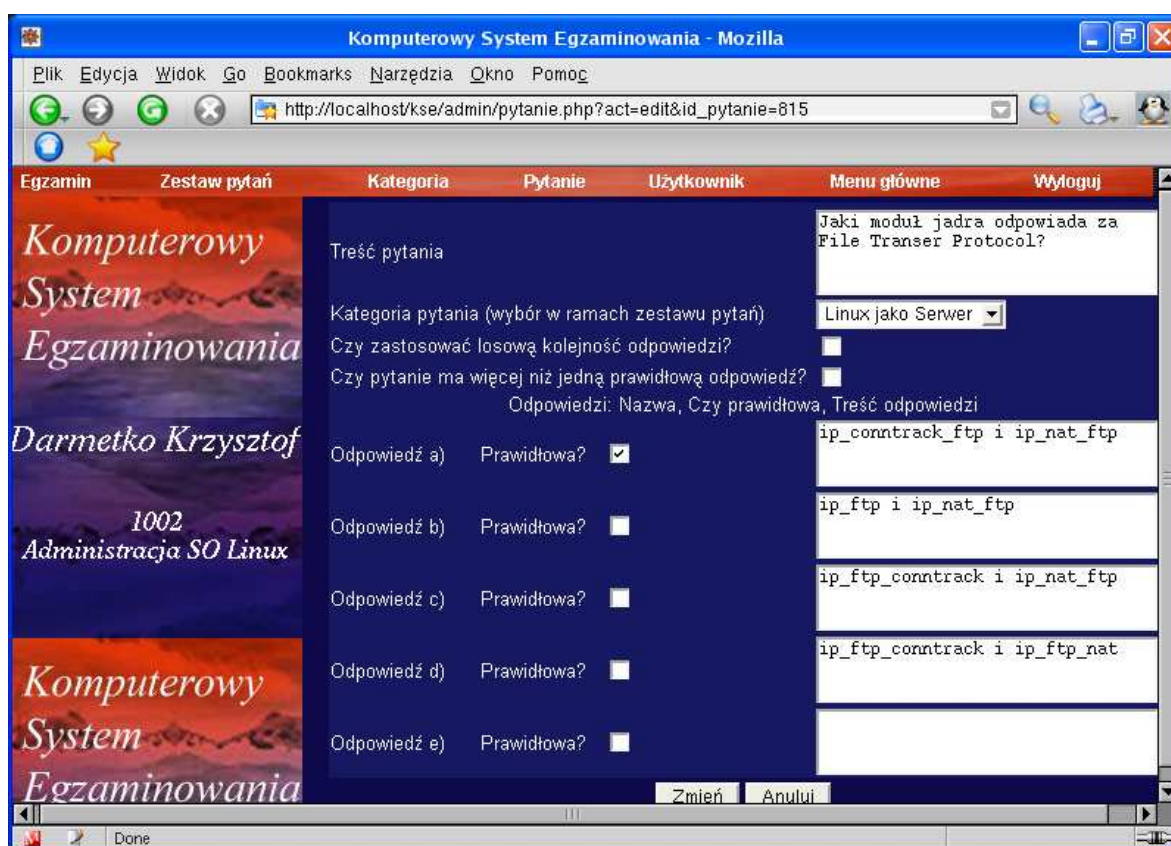
5.5.1 ETAP 1

Zaczynamy od wprowadzenia do bazy danych nazwy przedmiotu, oraz numeru jego sygnatury, następnie tworzymy konto dla użytkownika, który ma zostać koordynatorem tego przedmiotu i przypisujemy temu użytkownikowi uprawnienie koordynatora. W przypadku gdy wybrany przez nas użytkownik już widnieje w bazie danych, wystarczy dokonać edycji jego uprawnień. Powyższe czynności dokonujemy po zalogowaniu się przez stronę WWW do systemu na konto administratora. System umożliwia nadanie kompetencji administratora każdemu użytkownikowi, a także pozwala na czasowe dezaktywowanie konta dowolnie wybranego użytkownika.

5.5.2 ETAP 2

Kolejny etap rozpoczynamy od zalogowania się do systemu jako koordynator przedmiotu. Podobnie jak poprzednio pracujemy w aplikacji internetowej. Zaraz po zalogowaniu ujrzymy listę dostępnych przedmiotów z zaznaczeniem uprawnień, jakimi dysponujemy w odniesieniu do każdego z nich. Spośród listy wybieramy ten, który chcemy edytować. W celu rozpoczęcia wprowadzania pytań konieczne jest utworzenie tzw. zestawu pytań. Zestaw pytań grupuje kategorie pytań, które z kolei zawierają poszczególne pytania egzaminacyjne. Dzięki możliwości utworzenia wielu zestawów pytań, jesteśmy w stanie wprowadzić do bazy danych pytania dla różnych terminów egzaminu, co za tym idzie, możemy utworzyć zestaw „0” - dla terminu zerowego, zestaw „1” - dla terminu pierwszego itp. Dodatkowo w każdej chwili możemy przesuwać kategorie pytań pomiędzy zestawami pytań oraz poszczególne pytania pomiędzy kategoriami pytań. Na tym etapie tworzenia zestawu pytań definiujemy oceny, jakie studenci mogą uzyskać z koordynowanego przez nas przedmiotu, a także procentową skalę tychże ocen oraz limit czasu trwania egzaminu. Dodatkowo możemy tutaj zdecydować, czy system ma pozwolić na cofanie się do poprzedniego pytania, a także możemy zdefiniować bonusy punktowe

za zakończenie egzaminu przed czasem. Mając stworzony zestaw pytań, przystępujemy do wprowadzania do bazy danych kategorii pytań. Dzięki podziałowi na kategorie, koordynator ma możliwość pogrupowania pytań według tematu, stopnia trudności oraz rodzaju; pytania z jedną odpowiedzią prawidłową lub wieloma, taki podział daje w konsekwencji możliwość ustalania osobnej punktacji dla każdej kategorii oraz gwarantuje, że każdy student otrzyma w wyniku losowania tę samą, ustaloną liczbę pytań z poszczególnych kategorii. Zatem ogólna liczba pytań, które otrzyma każdy student w czasie egzaminu jest sumą ustalonych przez nas liczb pytań, jakie system ma losować z każdej kategorii w wybranym zestawie pytań.



Rys. 5.6: Panel administracyjny – okno edycji pytań

Tworząc więc kategorie możemy zdefiniować liczbę pytań, które system będzie losował oraz liczbę punktów przyznawanych za dobrą lub złą odpowiedź, a także za brak odpowiedzi. Istnieje jeszcze jeden parametr – „Czy system ma oceniać każdą odpowiedź osobno?” – ustawienie

tego parametru oznacza, że wszystkie pytania wielokrotnego wyboru stają się pytaniami typu prawda-fałsz, gdyż każdy podpunkt jest traktowany osobno.

Po utworzeniu kategorii pozostaje jedynie wprowadzenie do bazy danych poszczególnych pytań i odpowiedzi (rys. 5.6 na poprzedniej stronie). Każde pytanie może mieć do pięciu możliwych odpowiedzi. Także tutaj na poziomie pytania możemy określić, czy odpowiedzi mają się ukazywać w kolejności, w jakiej zostały wprowadzone do bazy, czy też w kolejności losowej.

W momencie, gdy mamy gotowy pełen zestaw pytań z podzielonymi na kategorie pytaniami, możemy stworzyć egzamin.

Egzamin – oznacza w systemie każdą sesję związaną z konkretnym terminem zdawania. Mamy więc do czynienia np.: z egzaminem–terminem „0” w sesji letniej, egzaminem–terminem pierwszym w sesji zimowej itp. Wprowadzając do bazy danych egzamin wybieramy wcześniej stworzony zestaw pytań oraz określamy, czy system ma wyświetlać wynik egzaminu, zaraz po zakończeniu zdawania. Podajemy także procentowy wpływ oceny z ćwiczeń i egzaminu na ocenę końcową z przedmiotu. Dodatkowo ustawiamy parametr decydujący o tym, czy zezwalamy na zdawanie egzaminu osobom nie widniejącym w protokole, a także jeśli się na to zdecydujemy, czy do dopuszczenia takiego studenta do egzaminu potrzebne jest, dla zapewnienia większej kontroli, wprowadzenie hasła sesji przez osobę pilnującą laboratorium, w którym się odbywa egzamin. W ten sposób stworzony egzamin jest gotowy do uruchomienia.

W gestii koordynatora pozostają jeszcze dwie sprawy. Po pierwsze powinien on utworzyć konta lub jedynie nadać uprawnienia wykładowcom i osobom prowadzącym ćwiczenia z danego przedmiotu – egzaminu. Celowo użyto tutaj pojęcia przedmiot – egzamin, gdyż w odróżnieniu od koordynatora, który odpowiada za cały przedmiot i wszystkie egzaminy, jakie w ramach tego przedmiotu są organizowane, konta wykładowców oraz „ćwiczeniowców” posiadają uprawnienia odnoszące się do konkretnych egzaminów. Przyjęto takie rozwiązanie, gdyż zestaw

nazwisk pracowników dydaktycznych ulega często zmianie z semestru na semestr, a ponadto każdy wykładowca może chcieć zorganizować osobny termin egzaminu dla swojej grupy wykładowej w ramach danego przedmiotu.

Drugą kwestią jest „zaimportowanie” protokołu studentów z bazy dziekanatowej do bazy danych KSE. Protokół jest powiązany z jednym z wcześniej zdefiniowanych egzaminów. Import danych studentów, zapisanych na dany przedmiot z bazy danych wirtualnego dziekanatu, wpłynie na przyspieszenie wprowadzania ocen z ćwiczeń w kolejnym kroku oraz pozwoli na dodatkową weryfikację danych osób zdających egzamin.

5.5.3 ETAP 3

Trzeci etap to wprowadzanie ocen z ćwiczeń. Z założenia więc realizowany jest on przez pracowników prowadzących ćwiczenia. Rozpoczyna się analogicznie do poprzednich etapów, od zalogowania do KSE za pośrednictwem strony internetowej na konto „ćwiczeniowca”. Po zalogowaniu wybieramy właściwy przedmiot, a następnie w ramach dostępnego menu, wchodzimy we wprowadzenie ocen z ćwiczeń. Po wpisaniu numeru albumu system sprawdza, czy osoba o tym numerze jest na protokole pobranym z dziekanatu. Jeżeli znajdzie tę osobę, wyświetla jej imię i nazwisko i umożliwia podanie oceny, w przeciwnym przypadku do wpisania oprócz oceny pozostają dane osobowe (imię i nazwisko) studenta. Jeśli przy wprowadzaniu powyższych danych do bazy, nastąpiła pomyłka, osoba, która je wpisywała może zawsze dokonać korekty. Ten etap kończy prace przygotowujące system do przeprowadzenia egzaminu.

5.5.4 ETAP 4

Czwarty etap to uruchomienie aplikacji egzaminującej – serwera sesji. W dotychczasowych etapach pracowaliśmy z interfejsem WWW, teraz mamy za zadanie uruchomić w środowisku Windows aplikację, która przeprowadzi egzamin.



Rys. 5.7: Okno powitalne kreatora sesji egzaminacyjnej

Podczas uruchamiania aplikacja, mająca formę tzw. kreatora zadaje kolejne pytania (rys. 5.7). Na początku podajemy adres komputera, na którym pracuje baza danych. Jeśli już wcześniej korzystaliśmy z tej aplikacji, komputer automatycznie zasugeruje użycie tej samej nazwy serwera. Teraz możemy określić port, na którym ma pracować serwer sesji. Dzięki możliwości zmiany numeru portu jesteśmy w stanie uruchomić w tej samej sieci wiele serwerów sesji, przeprowadzających różne egzaminy jednocześnie. Następnie logujemy się na konto koordynatora przedmiotu, lub na konto laboranta centrum egzaminowania. Po prawidłowym zalogowaniu wybieramy z listy kolejno najpierw przedmiot, a później egzamin. W następnym kroku możemy założyć hasło sesji umożliwiające uruchomienie stanowiska egzaminującego. Hasło może zostać pominięte, lecz wtedy istnieje niebezpieczeństwo, że ktoś posiadający dostęp do aplikacji – stanowiska egzaminującego może uruchomić egzamin i zdawać go nie przebywając w sali, w której się on odbywa, a jedynie będąc podłączonym do sieci wewnętrznej uczelni. Jeżeli w momencie tworzenia egzaminu ustalono, że aby zezwolić na zdawanie studentowi, który nie widnieje w protokole, potrzebne jest potwierdzenie osoby pilnującej laboratorium, to do wyżej wspomnianego potwierdzenia potrzebne będzie również omawiane hasło. Po obejrzeniu w kolejnym kroku

„resume danych” na temat zdefiniowanego w poprzednim etapie egzaminu, uruchamiamy serwer sesji. W tym momencie na ekranie pojawia się okno, w którym będą pokazywać się kolejne połączenia ze stanowiskami egzaminacyjnymi, z podanymi imionami i nazwiskami studentów, zdających na nich egzamin, czasem rozpoczęcia oraz po zakończeniu zdawania, także czasem zakończenia egzaminu oraz jego wynikiem.

5.5.5 ETAP 5

Ten etap to przygotowanie stanowiska egzaminującego. Osoba pilnująca laboratorium komputerowego wywołuje aplikację stanowiska egzaminującego, następnie wprowadza adres komputera, na którym pracuje serwer sesji dla danego egzaminu i podając (jeśli zostało ono ustawione) hasło sesji uruchamia program egzaminujący. W tym momencie przy komputerze może usiąść student i rozpocząć zdawanie egzaminu. Podobnie jak przy uruchamianiu serwera sesji, system pamięta poprzednie uruchomienie i jeżeli adres komputera i port, na którym pracuje serwer sesji, nie zostały zmienione, nie trzeba wprowadzać tych informacji ponownie.

5.5.6 ETAP 6

Student rozpoczyna egzamin od podania swojego imienia, nazwiska, numeru albumu oraz adresu e-mail. Jeżeli wcześniej wprowadzone zostały oceny z ćwiczeń, lub została wgrana lista zdających z bazy dziekanatu, istnieje możliwość weryfikacji danych podawanych przez studentów z danymi wcześniej wprowadzonymi. Oznacza to, że jeśli dany student nie widnieje w protokole, osoba pilnująca może być poproszona o wprowadzenie wcześniej ustawionego hasła sesji, aby umożliwić takiemu studentowi zdawanie egzaminu. Po zatwierdzeniu danych system uniemożliwia studentowi cofnięcie się i ponowne wprowadzenie danych osobowych – taka blokada ma na celu zapobieżenie temu, aby studenci mogli po weryfikacji danych osobowych KSE z rzeczywistymi, cofnąć się i rozpocząć zdawanie egzaminu jako inna osoba. W ostatnim kroku kreatora student dowiaduje się, jakie są reguły gry – skala ocen, liczba punktów itp.

Po kliknięciu myszką otwiera się okno stanowiska egzaminującego, w którym obok pytań

i możliwych do wybrania odpowiedzi pojawiają się: klawisze nawigacji między pytaniami (aktywne jeśli na etapie tworzenia zestawu pytań umożliwiono cofanie się do poprzedniego pytania), imię, nazwisko i numer albumu zdającego oraz odmierzany czas do zakończenia egzaminu. Po wyświetleniu ostatniego pytania pokazuje się przycisk zakończenia egzaminu. Student może zakończyć zdawanie przed czasem, jeśli tego nie zrobi, po upływie ustalonego czasu program przerwie zdawanie egzaminu.



Rys. 5.8: Wynik końcowy egzaminu, wyświetlany po zakończeniu odpowiadania na pytania

Jeżeli wcześniej aktywowano opcję wyników – wyświetlone zostaje teraz okno z wynikiem i oceną z egzaminu (rys. 5.8). Po kliknięciu myszką kreator powraca do pozycji, w której następny student może rozpoczynać zdawanie egzaminu – nie ma potrzeby, po raz kolejny wprowadzać hasło sesji, celem uruchomienia stanowiska egzaminacyjnego. Przez cały czas trwania egzaminu aplikacja stanowiska egzaminującego blokuje możliwość uruchomienia na tej samej maszynie innych programów, dzięki temu student nie może korzystać z pomocy Internetu, porozumiewać się z innymi, czy też korzystając ze specjalnych narzędzi podglądać pulpit innego zdającego.

5.5.7 ETAP 7

Jeśli wszyscy studenci zakończyli zdawanie egzaminu koordynator może zamknąć serwer sesji. Po zamknięciu aplikacji wyniki przeprowadzonego egzaminu zostają przesłane do bazy danych. Od tej chwili wykładowcy mogą poprzez interfejs WWW sprawdzić szczegółowe wyniki egzaminu.

5.5.8 ETAP 8

W celu sprawdzenia wyniku przeprowadzonego elektronicznie egzaminu logujemy się do KSE na konto wykładowcy, następnie wybieramy interesujący nas egzamin i wchodzimy w prezentację wyników (rys. 5.9). Ukazuje się nam lista studentów z ocenami z ćwiczeń, egzaminu oraz z oceną końcową.

The screenshot shows a web browser window titled 'Komputerowy System Egzaminowania - Mozilla'. The address bar shows the URL: `http://localhost/kse/wykladowca/wyniki.php?act=dokl&id_egzamin=30&id_zdajacy=60`. The page has a navigation bar with 'Wyniki', 'Statystyki', 'Menu główne', and 'Wyloguj'. The main content area is titled 'Szczegółowe wyniki dla: 43127 Jaś Miłowski'. It displays personal information, exam session details, and a table of exam questions and answers.

Nr pytania	Treść pytania	Odpowiedź	Liczba punktów	Czas odpowiedzi(sek.)	
1.	Jaki moduł jądra odpowiada za File Transer Protocol?	Błędna	0	15	Szczegół
2.	Jak nazywa się serwer odpowiedzialny za udostępnianie katalogów dla MS Windows.	Prawidłowa	1	0	Szczegół
3.	Kto napisał jądro Linuxa?	Prawidłowa	1	8	Szczegół
4.	Jaka jest budowa jądra systemu linuxa?	Prawidłowa	1	27	Szczegół
5.	Jaka komenda można sprawdzić czas jaki upłynął od uruchomienia systemu?	Prawidłowa	1	35	Szczegół

Buttons: 'Powrót' and 'Szczegół' (repeated for each row).

Rys. 5.9: Panel administracyjny wykładowcy – szczegółowe wyniki egzaminu

Jeśli chcemy obejrzeć dokładniejsze informacje, klikamy myszką na konkretnego zdającego i otrzymujemy listę wylosowanych przez niego pytań oraz dane dotyczące czasu, kiedy został przeprowadzony egzamin, czasu trwania egzaminu, oraz informację, czy student widniał wcześniej w protokole. Dodatkowo możemy obejrzeć szczegółowe odpowiedzi, jakich udzielił zdający, oraz czas odpowiedzi na poszczególne pytania. Dzięki zebraniu wszystkich tych danych w systemie, możemy dokonać analizy statystycznej, aby dowiedzieć się, które pytania sprawiły największą trudność, na które odpowiadano najszybciej itp. Wykładowca może wydrukować protokół z ocenami swoich studentów, oraz zatwierdzić wyniki egzaminu. Ponadto w sposób automatyczny może on przesłać ostateczną ocenę wraz z krótką informacją np. o terminie wpisów do wszystkich zdających.

5.5.9 ETAP 9

Jeśli wszyscy wykładowcy zatwierdzili oceny z egzaminu, koordynator może w sposób automatyczny wygenerować skrypt eksportujący wyniki egzaminu z bazy danych KSE do bazy danych dziekanatu. W ten sposób kończy się egzamin przeprowadzony całkowicie elektronicznie.

5.6 Nowe perspektywy badawcze

Przedstawione dotąd rozdziały zawierają opis trzech różnych systemów wspomagających proces egzaminowania. Wspomniane aplikacje w odmienny sposób i w różnym stopniu stanowią implementację teorii metod testowania wiedzy, opisanej w rozdziale drugim. Przedstawione programy typu CBA skupiają się na stosowaniu zadań zamkniętych, choć dwa pierwsze systemy umożliwiają także stosowanie zadań otwartych w postaci zadań z luką. Żadna z wymienionych aplikacji nie potrafi jednak analizować pytań otwartych.

Niezwykłe inspirujący dla merytorycznych rozwiązań systemów testowania wiedzy jest kierunek badań związany ze stosowaniem tzw. **wypowiedzi otwartych**. Kolejny rozdział poświęcono zatem szczegółowemu przedstawieniu teorii oraz implementacji systemów testowania wiedzy, analizujących wypowiedzi otwarte, które stanowią naturalne rozwinięcie systemów klasy CBA, choć z dzisiejszej perspektywy wdrożenia – jeszcze dosyć odległe.

Rozdział 6

Systemy weryfikacji wiedzy analizujące wypowiedzi otwarte

„Examinations are formidable even to the best prepared, for the greatest fool may ask more than the wisest man can answer.”

/ Charles Caleb Colton /¹

6.1 Wprowadzenie

Nauczanie formułowania wypowiedzi pisemnych jest bardzo ważną częścią naszej edukacji. Jednak ze względu na ogromną pracochłonność, a co za tym idzie wysokie koszty procesu sprawdzania prac, wielu nauczycieli i wykładowców rezygnuje z tej formy weryfikacji wiedzy i umiejętności studentów. Wprowadzenie efektywnej technologii komputerowej, pozwalającej oceniać prace pisemne, umożliwiłoby zwiększenie wykorzystania tej formy egzaminowania i przyczyniłoby się zapewne do polepszenia naszych umiejętności pisania oraz podniesienia jakości tworzonych przez nas tekstów.

¹Cytat pochodzi ze zbiorów internetowych, dostępnych pod adresem: <http://www.quotationspage.com>

Komputerowe systemy, potrafiące analizować i oceniać wypowiedzi otwarte, należą do grupy najbardziej zaawansowanych aplikacji wspomagających nauczanie. Choć ich rozwój rozpoczął już w drugiej połowie lat sześćdziesiątych Ellis Batten Page, nadal daleko im do doskonałości² [POWE01, s.1]. Funkcjonujące obecnie programy potrafią oceniać niezbyt długie i co najważniejsze, napisane w języku angielskim teksty.

W nauce toczy się dyskusja nad efektywnością oraz celowością stosowania tego typu systemów. Nie brakuje sceptyków. Donald E. Powers i inni przytaczają argumenty z artykułu „*Art Young, personal communication*” z 11 stycznia 2000 roku, mówiące o tym, że pojęcie komputerowo wspomaganej analizy nie jest zgodne z aktualnymi koncepcjami prac nad nauczaniem pisania, które to podkreślają zdolności piszącego do „nawiązania łączności z wybranymi czytelnikami”. Autorom tego artykułu wtórują Deloughry [DELO95] i Mitchell [MITC98] pisząc, że w odróżnieniu od ludzi, komputery nie są w stanie, według niektórych krytyków, odróżniać wyjątkowych, inspirujących prac od tych, które są wprawdzie z technicznego punktu widzenia poprawne, ale niezbyt górnolotne [POWE01, s.1]. Z drugiej strony, podkreśla się potencjał do prawdziwego, złożonego i naturalnego oceniania, jaki tkwi w tego typu technologiach [FRAS, s.2]. Prawdziwy entuzjasta i prekursor w tej dziedzinie mówi wręcz, że komputerowa ocena wypowiedzi otwartych wkroczyła w nową fazę (...) [PAGE95, s.1].

Dalsze rozważania poświęcone zostaną naszkicowaniu bieżącej sytuacji, dotyczącej systemów oceniających prace pisemne, oraz ukazaniu perspektyw rozwoju tego typu aplikacji. Rozpocznę więc od opisanie warunków, jakie muszą być spełnione podczas tworzenia tego typu programów. Następnie przedstawię dostępne implementacje tych systemów, ze szczególnym uwzględnieniem trzech aplikacji: Project Essay Grade, Electronic Essay Rater oraz Intelligent Essay Assessor, na temat których z racji tego, że są rozwijane przez ośrodki akademickie, istnieją materiały umożliwiające pokazanie metod ich działania. W podsumowaniu tej części problemów spróbuję zastanowić się nad możliwością implementacji aplikacji, analizującej wypowiedzi for-

²Ich rozwój był wcześniej przewidywany przez Ellisa Battena Page’a w pracach pt.: „*The imminence of grading essays by computer*” (1966) oraz „*Analyzing student essays by computer. International Review of Education*” (1968).

mułowane w języku polskim.

6.2 Atrybuty systemu analizującego tekst otwarty

Automatyczna analiza tekstów otwartych może być dokonowywana różnymi metodami. O zdecydowanej większości obecnie stosowanych sposobów mówi się, że należą do grupy tzw. **parsowania płytkiego** lub **powierzchniowego** (ang. *shallow parsing*). Parsowanie oznacza według Jurafsky'ego i innych [JURA00, s.57], analizę danych wejściowych poprzez konstruowanie struktury tych danych. Podstawą większości, opisywanych w tej pracy, metod oceny tekstów otwartych, są nie lingwistyczne a statystyczne cechy tekstu. Dlatego też bardzo ważne jest, aby aplikacje oceniające spełniały kryteria warunkujące wiarygodność wystawianych przez nie ocen. W raporcie z sierpnia 1998 roku [KAPL98, s.2-3] jednej z czołowych instytucji zajmujących się tą problematyką, Educational Testing Service (ETS), założonej przy uniwersytecie w Princeton, autorzy wymieniają cztery główne kryteria.

1. *Przejrzystość oceny.*

Każda ocena wystawiona przez system powinna dać się uzasadnić. Innymi słowy aplikacja nie może zachowywać się jak czarna skrzynka, do której wrzucamy wypowiedź, a następnie wyciągamy wynik. Każdy wygenerowany wynik powinien być, w sposób racjonalny, wytłumaczony za pomocą analizy jego powstania.

2. *Precyzja.*

Każda procedura oceny (zarówno ręczna, jak i również automatyczna) powinna charakteryzować się wysoką precyzją. Dokładność oceny, będąca istotnym miernikiem efektywności danej aplikacji, daje się zazwyczaj zbadać poprzez porównanie wyników oceniania przykładowego zestawu tekstów przez wykwalifikowane osoby oceniające oraz komputer. Im wyższy jest stopień korelacji ocen wystawionych manualnie i maszynowo, tym wyższa precyzja danej aplikacji.

3. *Nieprzezroczystość procedury oceniającej.*

Aplikacja nie powinna dawać się oszukiwać. Wyobraźmy sobie scenariusz, w którym procedura oceniająca bierze pod uwagę jedynie długość wypowiedzi. W tym przypadku zdający może w prosty sposób manipulować wynikiem, gdyż przy jego obliczaniu nie ma znaczenia ani treść, ani jej forma. Opisany tutaj scenariusz to oczywiście przypadek ekstremalny, jest on jednak przykładem sytuacji absolutnie nie do zaakceptowania. Procedura oceniająca powinna być zatem złożona i na tyle nieprzezroczysta, aby nie mogła zostać wykryta przez studentów.

4. *Koszty.*

Aplikacje oceniające mają służyć obniżeniu kosztów weryfikacji tekstów otwartych, dlatego koszty stosowania procedury oceniającej nie powinny przekraczać z góry ustalonych, akceptowalnych limitów. Z drugiej strony także koszty instalacji i przygotowania programu oceniającego, w tym „trenowanie aplikacji” przy użyciu danych wzorcowych, mogą znacznie podnieść całkowite koszty stosowania tej technologii i dlatego powinny zostać uwzględnione podczas oceny procedur testujących.

Myślę, że wymienione tutaj podstawowe kryteria są na tyle uniwersalne, że powinny być stosowane przez wszystkich twórców tego typu systemów.

6.3 Project Essay Grade – pierwszy system analizujący wypowiedzi otwarte

Project Essay Grade (PEG) to implementacja systemu komputerowego potrafiącego analizować wypowiedzi otwarte. Prace nad tego typu systemem rozpoczął w drugiej połowie lat 60. Ellis Batten Page. Początki były bardzo trudne i żmudne, gdyż każdy tekst musiał być wprowadzany do komputera za pomocą kart perforowanych. Page badał wypowiedzi używając kilkudziesięciu zmiennych dotarczających statystycznych informacji na temat tekstu. Prace nad praktycznym zastosowaniem badań Page’a zostały zintensyfikowane na początku lat 90., czego

rezultatem było badanie przeprowadzone w 1994 przez ETS, wykorzystujące PEG do oceny wypowiedzi otwartych.

Podstawą działania systemu PEG są dwa słowa kluczowe zaprezentowane przez autora: *trin* i *prox*. Słowo *trin* pochodzi od angielskiego wyrażenia *intrinsic variables of interest* i oznacza wewnętrzne zmienne, opisujące tekst otwarty, takie jak: dykcja, płynność, gramatyka, interpunkcja i wiele innych [PAGE95, s.7]. Trudno zmierzyć te zmienne, dlatego w praktyce system analizuje tekst stosując tzw. *prox* czyli *approximations*, a więc przybliżenia zmiennych *trin* [PAGE95, s.7]. Pośród listy kilkudziesięciu stosowanych zmiennych *prox* można znaleźć między innymi: liczbę słów w wypowiedzi, średnią długość słowa, liczbę średników, przecinków itp.

Metodologia stosowania systemu PEG polega na:

- dokonaniu oceny losowo wybranej partii tekstów przez ekspertów,
- wczytaniu tych wypowiedzi i wystawionych ocen do systemu PEG,
- zbudowaniu, przy wykorzystaniu regresji liniowej, modelu wypowiedzi, dla każdej z możliwych do wygenerowania ocen.

Tak zbudowany model jest używany w kolejnym kroku podczas oceny pozostałej części prac.

W badaniu przeprowadzonym przez ETS analizowano około 1300 wypowiedzi. Wystawione przez aplikację PEG oceny porównywano z ocenami przyznanymi przez ekspertów, a następnie obliczono współczynniki korelacji między tymi ocenami. Uzyskana precyzja oceniania przez oprogramowanie wahała się, w zależności od poszczególnych kryteriów jakie przyjęto. Stosunkowo najgorsze wyniki obserwowano podczas oceny całokształtu wypowiedzi oraz przy ewaluacji stylu pisma, w tych przypadkach współczynnik korelacji ocen wystawionych maszynowo był mniej więcej równy współczynnikowi korelacji ocen wystawionych przez dwóch lub trzech ekspertów. Najlepszą zgodność oceny automatycznej z manualną uzyskano

podczas testowania treści merytorycznej oraz kreatywności piszących. Dla tych kryteriów system osiągał skuteczność porównywalną z oceną sześciu egzaminatorów [PAGE95, s.13].

Dzisiaj, mimo dosyć dobrych wyników zgodności ocen wystawianych przez system PEG z ocenami ekspertów, program, który bazuje praktycznie wyłącznie na statystycznej analizie tekstów, trudno uznać za w pełni wiarygodny. Nie spełnia on bowiem pierwszego z czterech przedstawionych w podrozdziale 6.2 warunków, a mianowicie kryterium przejrzystości oceny. Trzeba jednak wziąć pod uwagę, że wyżej wymienione warunki zostały opublikowane w 1998 roku, a więc cztery lata po przeprowadzeniu opisanego tutaj badania, wykorzystującego aplikację PEG. Reasumując należy powiedzieć, że mimo braku pełnego zaufania co do precyzji i adekwatności ocen wystawianych przez ten system, trzeba uznać ogromny wkład twórcy systemu PEG, Ellisa Battena Page'a, w badania nad automatyczną oceną wypowiedzi otwartych.

Wersja demonstracyjna aplikacji oceniającej, bazującej na systemie PEG dostępna jest w Internecie³.

6.4 Intelligent Essay Assessor – przykład implementacji metody LSA do oceny wypowiedzi otwartych

Kolejnym przykładem systemu zdolnego do analizy wypowiedzi otwartych jest aplikacja Intelligent Essay Assessor (IEA) proponowana przez firmę Knowledge Analysis Technologies ⁴. W odróżnieniu od pierwszego opisywanego systemu – aplikacji PEG, oprogramowanie to bazuje głównie na analizie zawartości merytorycznej badanych wypowiedzi, nie uwzględnia przy tym czysto statystycznych miar tekstu. Aplikacja ta, do badania porównawczego treści wypowiedzi, korzysta z metody zwanej *Latent Semantic Analysis* (LSA), która stanowi rozwinięcie podejścia *Latent Semantic Indexing* (LSI), opatentowanego w 1989 roku, jako metoda kategoryzacji i pozyskiwania informacji.

³pod adresem <http://134.68.49.185/pegdemo/>

⁴Więcej informacji o firmie można znaleźć w Internecie na stronie <http://www.knowledge-technologies.com/>

6.4.1 Ogólny opis działania metody LSA

LSA to statystyczno-matematyczna technika reprezentacji znaczenia słów i większych fragmentów tekstu, takich jak zdania lub akapity, polegająca na analizie korpusu tekstu [FRAS, s.13]. Metoda LSA jest blisko związana ze sztucznymi sieciami neuronowymi, ale opiera się na tzw. *Singular Value Decomposition* (SVD), matematycznej technice redukcji wymiaru macierzy podobnej do analizy czynnikowej⁵, za pomocą której bada się teksty, zawierające w swoich treściach wiedzę na zadany temat zbliżoną, z punktu widzenia objętości, do wiedzy posiadanej przez ludzi o odpowiednich kwalifikacjach [LAND98, s.3]. Dzięki zastosowaniu metody LSA możliwe jest symulowanie ludzkiej oceny bliskości znaczeniowej słów, lub fragmentów tekstu. Należy przy tym zaznaczyć, że nie chodzi tutaj jedynie o proste badanie sąsiedowania i współwystępowania słów, ale o wyrafinowaną analizę matematyczną, która jest w stanie oddać znacznie głębsze zależności między wyrazami. Dodatkowo, w celu zwiększenia precyzji działania, omawiana metoda przetwarza wstępnie tekst, analizując rozkład znaczeń poszczególnych słów w zależności od kontekstu, w jakim zostały one użyte. Metoda LSA może mierzyć odległość semantyczną dwóch fragmentów tekstu w „przestrzeni semantycznej”, która jest zbudowana w oparciu o tekst wejściowy [STEI01, s.6].

W celu skonstruowania takiej przestrzeni tworzy się macierz, w której komórkach zapisane są liczby występowania poszczególnych słów w badanym tekście. Możemy tutaj rozpatrzeć przykład zdania „Trzy lwy zjadły trzy konie.” Stwórzmy więc sobie macierz, w której rząd i będzie reprezentować opisywany fragment tekstu, a kolejne kolumny, poszczególne słowa użyte w tym zdaniu. W ten sposób, jeśli przyjmiemy, że kolumna j odpowiada słowu „trzy”, to w komórce ij znajdzie się liczba 2, gdyż słowo „trzy” występuje dwa razy w badanym tekście, natomiast pozostałe komórki w rzędzie i -tym, odpowiadające innym, występującym w omawianym zdaniu, słowom, będą zawierały liczby 1. Jeżeli teraz w analogiczny sposób przeanalizujemy wiele dokumentów, związanych z wybraną tematyką, otrzymamy ogromną

⁵Analiza czynnikowa stanowi zespół metod i procedur statystycznych pozwalających na zredukowanie dużej liczby zmiennych do kilku wzajemnie nieskorelowanych czynników. Zachowują one stosunkowo dużą część informacji tkwiących w zmiennych pierwotnych, a jednocześnie każda z nich jest nośnikiem innych treści merytorycznych [PANE03, Analiza czynnikowa].

macierz, której kolumny odpowiadać będą poszczególnym słowom występującym w tych tekstach, wypełnioną w większości zerami. Ze względu na tę ostatnią cechę tego typu konstrukcję nazywamy **macierzą rzadką** (ang. *sparse matrix*).

Macierz występująca w tej postaci nie może być jednak, z kilku powodów, wykorzystana bezpośrednio do obliczania podobieństwa znaczeniowego słów. Po pierwsze jej wymiar, sięgający w praktyce nawet kilkadziesiąt tysięcy, uniemożliwia, przy dzisiaj dostępnym sprzęcie, dokonywanie skomplikowanych obliczeń, które są potrzebne do badania zależności słów. Drugi, znacznie ważniejszy, powód to duża niedokładność takiej aproksymacji znaczenia wyrazów, wynikająca z „rzadkości” wykorzystywanej macierzy oraz z faktu, że w prawdziwym języku, słowa, o prawie synonimicznym znaczeniu, rzadko występują razem w przeciętnej wypowiedzi [LAND97, s.9]. Autorzy tekstów mają przeważnie swoje preferencje odnośnie używanego słownictwa, co w praktyce oznacza, że wykorzystują tylko niektóre z wielu możliwych terminów, oznaczających dane zjawisko. Z drugiej strony, gdybyśmy mieli do czynienia z macierzą, której wymiar byłby dokładnie równy liczbie różnych kontekstów, w jakich występuje dany wyraz, mielibyśmy sytuację, w której podczas analizy każdego wystąpienia badanego słowa uważalibyśmy, że ma ono zupełnie inne znaczenie [LAND97, s.9]. Z wyżej przedstawionych powodów wynika, że najlepszym rozwiązaniem jest redukcja wymiarów opisywanej macierzy do pewnego, optymalnego poziomu.

Przed przystąpieniem do redukcji liczby wymiarów macierzy aplikacja dokonuje dwustopniowej transformacji obliczonych wartości w macierzy. W pierwszej fazie wartości częstości występowania poszczególnych wyrazów w badanych fragmentach tekstów są przekształcane do logarytmów naturalnych tych wartości. Druga faza polega natomiast na przemnożeniu poszczególnych wartości w komórkach przez wagi, obliczone za pomocą statystycznej miary zwanej **entropią** (ang. *entropy*), czyli teoretyczną zawartością informacyjną każdego słowa.

Każda waga stanowi iloraz entropii danego słowa i entropii obliczonej dla wszystkich słów w danym rzędzie macierzy. Efektem takiej transformacji jest skorygowanie częstości występowania poszczególnych wyrazów o miarę ich ważności dla badanego fragmentu tekstu oraz, odwrotnie

proporcjonalnie, o stopień w jakim wiedza, dotycząca faktu wystąpienia danego słowa, wpływa na znajomość fragmentu, w którym ono wystąpiło [LAND98, s.17].

6.4.2 Singular Value Decomposition jako technika redukcji liczby wymiarów przestrzeni semantycznej

Po etapie wstępnej konstrukcji macierzy można przejść do kolejnego kroku, czyli zmniejszenia wymiaru przestrzeni. W sposób naturalny nasuwają się dwa pytania. Pierwsze – w jaki sposób zredukować wymiar macierzy tak, aby w jak najmniejszym stopniu utracić informacje zawarte w jej komórkach? Drugie pytanie – jak wyznaczyć optymalny wymiar macierzy?

W odpowiedzi na pierwsze pytanie, autorzy systemu IEA wykorzystują **statystyczną technikę SVD**, której metodę działania można porównać do budowy prostej, ale dosyć dużej sztucznej sieci neuronowej o trzech warstwach neuronów [LAND97, s.12]. W takiej, hipotetycznej sieci każdy neuron pierwszej warstwy odpowiada każdemu analizowanemu słowu, natomiast każdy neuron trzeciej warstwy – poszczególnym, badanym tekstom. Druga warstwa natomiast stanowi ogniwo obliczeniowe łączące neurony warstwy pierwszej z trzecią i vice-versa. Tego typu sieć jest symetryczna, a wszystkie obliczenia, dokonywane w niej – liniowe [LAND97, s.12]. W praktyce metoda SVD jest algebraiczną metodą liniowego rozkładu macierzy na niezależne główne składowe. Metoda SVD jest uogólnieniem analizy czynnikowej, która jako przypadek szczególny wymaga, aby macierz podlegająca obliczeniom była kwadratowa, a wartości reprezentowane przez kolumny oraz rzędy, wyrażone były w tych samych jednostkach [LAND97, s.13]. Szczegółowe obliczenia wykonywane przy zastosowaniu metody SVD zostały opisane w dwóch artykułach Thomasa Landauera i innych pt.: „*A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge*” [LAND97] oraz „*Introduction to Latent Semantic Analysis. Discourse Processes*” [LAND98].

Wiedząc, jak należy dokonywać transformacji macierzy, aby uzyskać większą skuteczność porównywania znaczenia słów oraz fragmentów tekstu, trzeba odpowiedzieć na drugie, postawione wyżej pytanie, o optymalną liczbę wymiarów ostatecznej przestrzeni semantycznej.

Według autorów aplikacji optymalna liczba wymiarów jest kwestią empiryczną [LAND98, s.16]. Należy wziąć pod uwagę zewnętrzne czynniki, takie jak wyniki testów synonimiczności albo przewidywania brakujących słów we fragmentach tekstu [LAND98, s.16]. W praktyce przeprowadzone przez autorów programu badania wykazały, że najlepsze efekty można uzyskać stosując liczbę między 200 a 400 wymiarów [STEI01, s.7].

W zbudowanej w powyższy sposób przestrzeni, dokumenty o zbliżonej zawartości tematycznej ułożone są w niedużej odległości od siebie. Dzięki zastosowaniu redukcji wymiaru macierzy synonimy, które rzadko występują obok siebie, są także zgrupowane blisko. Powyższe cechy tłumaczą nazwę badanej przestrzeni, która jest **przestrzenią semantyczną** [STEI01, s.7]. W celu porównania podobieństwa znaczeniowego słów, fragmentów tekstu lub całych dokumentów należy wziąć pod uwagę wektory, reprezentujące daną wypowiedź w przestrzeni semantycznej. W przypadku dokumentów, które służyły do budowy tej przestrzeni, wystarczy odczytać odpowiednie wartości wektorów, natomiast dla pozostałych wypowiedzi trzeba obliczyć odpowiednie wektory wypadkowe poprzez zsumowanie wektorów dla poszczególnych słów, zawartych w badanym tekście [STEI01, s.8]. Mając do dyspozycji dwa wektory reprezentujące porównywane słowa lub wypowiedzi, należy obliczyć cosinus kąta pomiędzy nimi oraz ich długości. Im wyższa jest wartość cosinusa kąta pomiędzy tymi wektorami, tym bardziej zbliżone jest znaczenie badanych fragmentów tekstu. Długość badanych wektorów świadczy natomiast o wiedzy, jaką metoda LSA „posiada” na temat zawarty w wypowiedziach, a co za tym idzie im dłuższy jest dany wektor, tym więcej informacji zgromadzonej jest w przestrzeni semantycznej na badany temat [STEI01, s.8].

6.4.3 Sposób wykorzystania metody LSA w aplikacji IEA

Po przedstawieniu sposobu działania metody LSA należy omówić postępowanie związane z wykorzystaniem aplikacji IEA do oceny wypowiedzi otwartych. Przed przystąpieniem do oceniania zadań otwartych program musi zostać „przetrenowany” przy użyciu słownictwa związanego tematycznie z zadaniami, które mają być sprawdzane. Źródłem wiedzy specjalistycznej mogą być podręczniki w formie elektronicznej, artykuły naukowe, lub wzorcowe wypowiedzi tekstowe, przygotowane przez wykwalifikowanych dydaktyków. Na podstawie tych tekstów system buduje przestrzeń semantyczną, na której następnie operuje porównując wektory obliczone dla badanych wypowiedzi z wektorami odpowiadającymi tekstom wzorcowym o znanej wartości merytorycznej [FOLT99, s.2]. Dzięki zastosowaniu metody LSA aplikacja nie koncentruje się wyłącznie na porównywaniu tekstów za pomocą współwystępowania słów kluczowych, lecz dokonuje głębszej, pojęciowej analizy treści wypowiedzi [FOLT99, s.2]. Co za tym idzie, bardzo podobne według IEA teksty, wcale nie muszą wykorzystywać identycznego słownictwa.

Istotnym czynnikiem mającym wpływ na skuteczność działania aplikacji jest dobór zestawu tekstów z jakimi program porównuje badane wypowiedzi. Jednym z podejść jest technika oceny holistycznej, polegająca na porównywaniu zadania z kilkoma (zazwyczaj dziesięcioma), najbardziej podobnymi według metody LSA fragmentami tekstów. Wyniki jej zastosowania są zbliżone do efektów pracy ekspertów, którzy częściej oceniają wypowiedź tekstową jako całość niż biorą pod uwagę jej poszczególne elementy składowe [FOLT99, s.2]. W przeprowadzonych przez Educational Testing Service badaniach porównujących skuteczność oceniania holistycznego przez IEA z oceną ekspertów brano pod uwagę fragmenty egzaminów typu Graduate Management Admissions Test (GMAT)⁶. Każdy egzamin był oceniany przez dwóch ekspertów oraz przez aplikację IEA. Średnie wyniki uzyskane podczas oceny 1205 wypowiedzi pokazują, że współczynnik korelacji ocen wystawionych przez egzaminatorów,

⁶Egzaminy GMAT są nadzorowane przez Graduate Management Admissions Council, zajmujące się weryfikowaniem wiedzy studentów podyplomowych programów nauczania zarządzania (przede wszystkim MBA). Więcej informacji można znaleźć na stronach poświęconych egzaminom GMAT pod adresem <http://www.gmat.org>

wynoszący 0,707 jest tylko nieznacznie wyższy od współczynnika korelacji ocen wystawionych przez program IEA z efektami pracy jednego z ekspertów i wynosi 0,701 [FOLT99, s.2].

Inna metoda stosowania oprogramowania IEA polega na porównywaniu wypowiedzi zdających z wybranymi tekstami wzorcowymi. W tym przypadku egzaminujący przygotowuje tekst „idealny”, który jest następnie wprowadzany do aplikacji IEA. System wystawia oceny na podstawie analizy podobieństwa wypowiedzi egzaminowanego z tekstem eksperta [FOLT99, s.3].

Istotą jeszcze innego podejścia jest porównywanie fragmentów badanej wypowiedzi z częściami tekstów zaczerpniętych z podręczników lub artykułów naukowych. Opisywana tutaj metoda porównawcza znajduje szczególne zastosowanie przy sprawdzaniu znajomości poszczególnych wycinków tematyki, będącej przedmiotem wypowiedzi [FOLT99, s.3]. Może być ona pomocna dla zdających, jako metoda „samosprawdzania” wiedzy na zadany temat.

Sposób działania aplikacji IEA, polegający na porównywaniu różnych tekstów ze sobą, powoduje, że wypowiedzi, charakteryzujące się nietypową konstrukcją lub rzadkim słownictwem, mogą być niewłaściwie oceniane przez system komputerowy. Z punktu widzenia metody LSA prawdopodobne jest, że merytorycznie dobry, lecz w sposób oryginalny sformułowany, tekst może zostać oceniony na równi z wypowiedzią, której treść przedstawia co najwyżej przeciętną wartość. Dlatego aplikację wyposażono w metodę rozpoznawania nietypowych wypowiedzi, które są oznaczane przez system i przesyłane do oceny przez ekspertów [FOLT99, s.4].

Reasumując wiadomości dotyczące aplikacji IEA należy zwrócić uwagę na najważniejsze wady oraz zalety tego programu.

Do wad tego rozwiązania należy zaliczyć fakt, że metoda LSA nie bierze pod uwagę kolejności ulokowania poszczególnych słów w badanych zdaniach. Kolejnym mankamentem jest traktowanie każdego, z punktu widzenia ortografii, różnego słowa jako zupełnie osobną jed-

nostkę znaczeniową. Jeszcze inną wadą są ogromne wymagania sprzętowe, dotyczące zwłaszcza wymaganej wielkości pamięci operacyjnej komputera (autorzy jako minimum określają ok. 1GB), na którym pracuje IEA. Z drugiej strony, wysokie oceny skuteczności aplikacji oraz wbudowane mechanizmy wykrywania nietypowych prac, powodują, że oprogramowanie firmy Knowledge Analysis Technologies jest znacznie bardziej wiarygodne niż omawiany wcześniej system PEG.

Zarówno działanie metody LSA, jak i również całej aplikacji IEA, można także sprawdzić korzystając z demonstracyjnego serwisu internetowego dostępnego pod podanym w przypisie adresem.⁷

Obecnie przedstawimy inne podejście, wykorzystujące, obok statystycznej analizy tekstu, metody zaczerpnięte z językoznawstwa komputerowego. Zanim jednak będziemy mogli pokazać przykład implementacji takiego systemu, zatrzymamy się na podstawach metodologicznych, umożliwiających budowę tego typu aplikacji.

6.5 Podstawy metodologiczne budowy systemów analizujących wypowiedzi otwarte opartych na osiągnięciach językoznawstwa komputerowego

Systemy analizujące zadania otwarte mogą być budowane w różny sposób. Pierwsze tego typu aplikacje zajmowały się jedynie wartością statystyczną tekstów. Obecnie tworzone implementacje dokonują głębszej analizy, polegającej w głównej mierze na porównywaniu zawartości tekstu wzorcowego z fragmentem ocenianym. W celu wykonania takiego porównania trzeba jednak odpowiednio przygotować tekst podawany na wejściu. Metody wstępnego przetwarzania, jak i również porównywania zawartości znaczeniowej tekstu, stanowią przedmiot badań **języko-**

⁷<http://LSA.colorado.edu> i <http://psych.nmsu.edu/essay/>.

znawstwa komputerowego (ang. *computational linguistics*). Lingwiści zajmujący się problematyką komputerowego badania tekstu wymieniają kilka etapów analizy, potrzebnych do dokonania oceny wypowiedzi otwartej. Etapy te opisuję poniżej wykorzystując informacje przekazane mi przez Adama Przepiórkowskiego, pracownika Instytutu Podstaw Informatyki Polskiej Akademii Nauk, współautora m.in. książki pt.: „Formalny opis języka polskiego” [PRZE02].

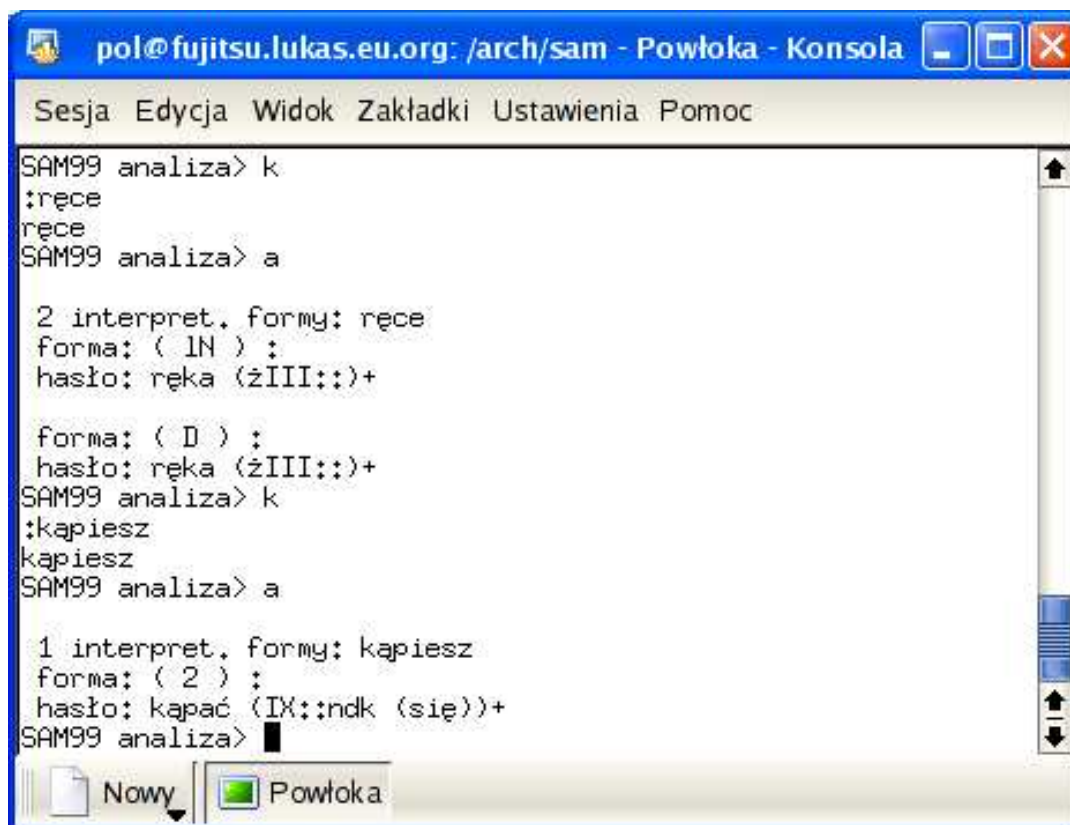
6.5.1 Segmentacja

Pierwszym etapem komputerowego badania wypowiedzi otwartych jest **segmentacja**. Polega ona na podziale tekstu na **segmenty** (tokeny) na podstawie zapisu ortograficznego. Choć wydawać by się mogło, że jest to zadanie dosyć proste, istnieją jednak sytuacje, w których poszczególne ciągi liter pomiędzy spacjami reprezentują więcej niż jeden segment. Przykładem, ilustrującym ten problem, zaczerpniętym z języka polskiego, jest następujące wyrażenie: *dlugom szedł*. Wyraz *dlugom* składa się z dwóch segmentów: *dlugo* i *m*. Aby sprawdzić, że tak jest wystarczy przekształcić powyższą formę w równoznaczne zdanie: *dlugo szedłem*. W praktyce, segmentacji dokonują zazwyczaj **analizatory morfologiczne**, przy czym większość z nich stosuje bardzo uproszczoną metodę, polegającą na podziale tekstu na segmenty na podstawie rozmieszczenia spacji oraz znaków interpunkcyjnych.

6.5.2 Analiza morfologiczna

Kolejnym etapem badania tekstu jest **analiza morfologiczna**. Jej istota polega na identyfikacji poszczególnych typów wyrazów oraz na ewentualnej próbie podziału ich na **morfemy**. Jako przykład takiego podziału można wskazać wyraz *narodowość*, który składa się z trzech morfemów: morfemu leksykalnego *naród*, morfemu słotwórczego *ow* tworzącego przymiotniki oraz morfemu słotwórczego *ość* tworzącego rzeczowniki. Jednak większość aplikacji, zwanych **analizatorami morfologicznymi**, pomija szczegółowy podział na morfemy, skupiając się na prostszych metodach identyfikacji części mowy wykorzystujących wbudowany słownik. Niektóre z nich zawierają dodatkowo tzw. **słownik a tergo**, który składa się z wyrazów zapisanych od końca w porządku alfabetycznym. Dzięki temu słownikowi analizatory morfologiczne

mogą rozpoznawać formy nieznanych im wyrazów na podstawie badania ich końcówek. Przykład działania prostego analizatora morfologicznego pokazany został na rysunku 6.1.



Rys. 6.1: Przykład działania analizatora morfologicznego SAM, autorstwa Krzysztofa Szafrana

Niestety zdarza się często, że na podstawie prostego badania zapisu danego wyrazu nie daje się ustalić kategorii gramatycznej wyrazu, gdyż istnieje kilka możliwych wariantów postępowania. Tak więc może się zdarzyć, że nie uda się określić np. czy dany wyraz to rzeczownik, czy czasownik. Problem ten ilustruje wyraz *piec*, który może być czasownikiem (*piec ciasto*) lub rzeczownikiem (*piec hutniczy*). W takich przypadkach trzeba skorzystać z **tagera** zwanego też **dezambiguatorem analiz morfoskładniowych** (ang. *tagger*), który poprzez analizę kontekstu dokonuje wyboru jednej z wcześniej znalezionych identyfikacji właściwej formy badanego wyrazu. Większość **tagerów** można podzielić na dwie kategorie. Są to programy oparte na **wnioskowaniu regułowym** oraz programy **wnioskujące stochastycznie** [JURA00, s.300]. Istnieje także

oprogramowanie wykorzystujące do klasyfikacji wyrazów zarówno reguły, jak również stochastyczne zależności. Zostało ono opracowane przez Brilla (1995) i stąd nosi nazwę **klasyfikatora Brilla** (ang. *Brill tagger*) lub klasyfikatora opartego na transformacji (ang. *transformation-based tagger*) [JURA00, s.300]. Klasyfikator Brilla łączy w sobie nie tylko cechy obydwu kategorii tagerów, ale także samego analizatora morfologicznego.

6.5.3 Analiza syntaktyczna

Następnym problemem, który porusza językoznawstwo komputerowe jest **analiza syntaktyczna** tekstu, która zajmuje się badaniem odwzorowania wypowiedzi od strony gramatycznej. Najbardziej znane podejścia to tzw. gramatyki bezkontekstowe (ang. *context-free grammars*) oraz gramatyki regularne (ang. *regular grammars*). Z punktu widzenia aplikacji oceniających zadania otwarte istotny jest fakt, że analiza syntaktyczna może dostarczać informacji dotyczących stopnia skomplikowania budowy zdań oraz stosowanych rodzajów zdań w badanych wypowiedziach.

6.5.4 Analiza semantyczna

Ostatnim etapem oceny tekstu otwartego jest **analiza semantyczna** wypowiedzi. Maszynowe badanie znaczenia tekstu przysparza wiele trudności. Pierwszą z nich jest wieloznaczność wyrazów. W językoznawstwie komputerowym określa się to mianem **ujednoznaczniania** lub **dezambiguacją** (ang. *word sense disambiguation*). Pojęcie to jasno definiuje Jurafsky pisząc, że „zadaniem ujednoznaczniania jest badanie współwystępowania wyrazów w kontekście w celu określenia dokładnego sensu, w jakim został użyty dany wyraz w danym miejscu” [JURA00, s.631].

Kolejny problem analizy semantycznej to tzw. **rozwiązywanie zaimków** (ang. *pronoun resolution*), które ma za zadanie zidentyfikowanie rzeczowników, zastępowanych w danym tekście przez zaimki.

Jeszcze inny aspekt badania semantycznego to tzw. **utożsamianie nazw własnych** (ang. *named entity recognition*) polegające na znajdowaniu w analizowanej wypowiedzi rzeczowników, które w danym kontekście są równoznaczne i powinny być traktowane jako jedna jednostka leksykalna. Przykładem ilustrującym taką sytuację jest mógłby być tekst opowiadający o losach prezydenta Kennedy’ego, w którym autor, używając wyrazów *prezydent, John, JFK, Kennedy* itp., miałby na myśli zawsze tę samą osobę.

W celu ułatwienia znajdowania podobieństwa między wyrazami tworzy się tzw. **ontologie**, czyli zbiory pojęć powiązane relacjami. Ontologie mają formę drzew, lub innych podobnych struktur, grupujących i klasyfikujących pojęcia według ich znaczenia. Wykorzystanie drzew umożliwia oddanie hierarchicznej struktury semantycznej, w której słowa uszeregowane są od tych – mających najszerze znaczenie, po te – najbardziej specjalistyczne.

Najważniejszą techniką umożliwiającą dokonywanie porównania semantycznego słów, lub nawet całych fragmentów tekstu, jest zastosowanie modelu przestrzeni wektorowej. Ze względu na ogromną wartość tej metody, przedstawieniu szczegółów jej działania poświęcona jest poniższa część tego podrozdziału.

6.5.5 Model przestrzeni wektorowej

W 1971 roku Gerard Salton opublikował pracę pt.: „*The SMART Retrieval System Experiments in Automatic Document Processing*” [SALT71], w której zawarł opis **modelu przestrzeni wektorowej** (ang. *vector space model*) jako metody pozyskiwania i kategoryzacji dokumentów. Model przestrzeni wektorowej jest obecnie jedną z najpopularniejszych metod automatycznego przetwarzania tekstów. Może on być stosowany zarówno w rozmaitych wyszukiwarkach i programach klastrujących, jak i również w systemach oceniających wypowiedzi otwarte. Jest on narzędziem umożliwiające porównywanie zawartości merytorycznej tekstów. Wiele twórców aplikacji analizujących zadania otwarte modyfikuje podstawowy model opracowany przez Saltona, dostosowując go do własnych potrzeb. We wcześniejszym podrozdziale, podczas

zagłębiania się w szczegóły działania metody LSA, zetknęliśmy się *de facto* z opisywanym modelem, a właściwie jego modyfikacją, choć nazwa *vector space model* nie została nigdzie wyraźnie zaznaczona. Poniżej zamieszczono oryginalną postać modelu opracowaną na podstawie książki pt.: „Speech and Language Processing” [JURA00, s.647].

Jurafsky cytuje definicję modelu opracowaną przez Saltona: „W modelu przestrzeni wektorowej, dotyczącym pozyskiwania informacji, dokumenty i zapytania są reprezentowane jako **wektory atrybutów** reprezentujących wyrażenia, które w nich występują” [SALT71] (tłum. autora). Wyjaśniając należy dodać, że prezentowane wektory atrybutów zawierają informacje odnośnie wystąpienia lub braku danego atrybutu w badanym dokumencie. Ponieważ metoda modelu przestrzeni wektorowej została przez Saltona przedstawiona jako algorytm pozyskiwania informacji, operujemy tutaj pojęciem dokumentu oraz zapytania. Dla naszych potrzeb, a więc w celu porównywania wypowiedzi otwartych, możemy przyjąć, że opisywane zapytanie reprezentuje analizowaną wypowiedź studenta. W sposób formalny powyższe wektory zapisuje się następująco:

$$\vec{d}_j = (t_{1,j}, t_{2,j}, \dots, t_{N,j}) \quad (6.1)$$

$$\vec{z}_k = (t_{1,k}, t_{2,k}, \dots, t_{N,k}) \quad (6.2)$$

Wektory \vec{d}_j oraz \vec{z}_k oznaczają odpowiednio: dokument oraz zapytanie, natomiast poszczególne atrybuty t reprezentują N wyrażen, które występują w badanych tekstach. Dla uproszczenia, założmy, że atrybuty mogą przyjmować jedynie wartości zero-jedynkowe, oznaczające brak, lub wystąpienie danego wyrażenia w badanym dokumencie lub zapytaniu. Korzystając z tych założeń możemy wyznaczyć miarę podobieństwa dokumentu oraz zapytania. Zapisujemy ją formalnie jako:

$$sim(\vec{z}_k, \vec{d}_j) = \sum_{t=1}^N t_{i,k} \times t_{i,j} \quad (6.3)$$

Równanie 6.3 możemy interpretuje się następująco: miarą podobieństwa dokumentu i

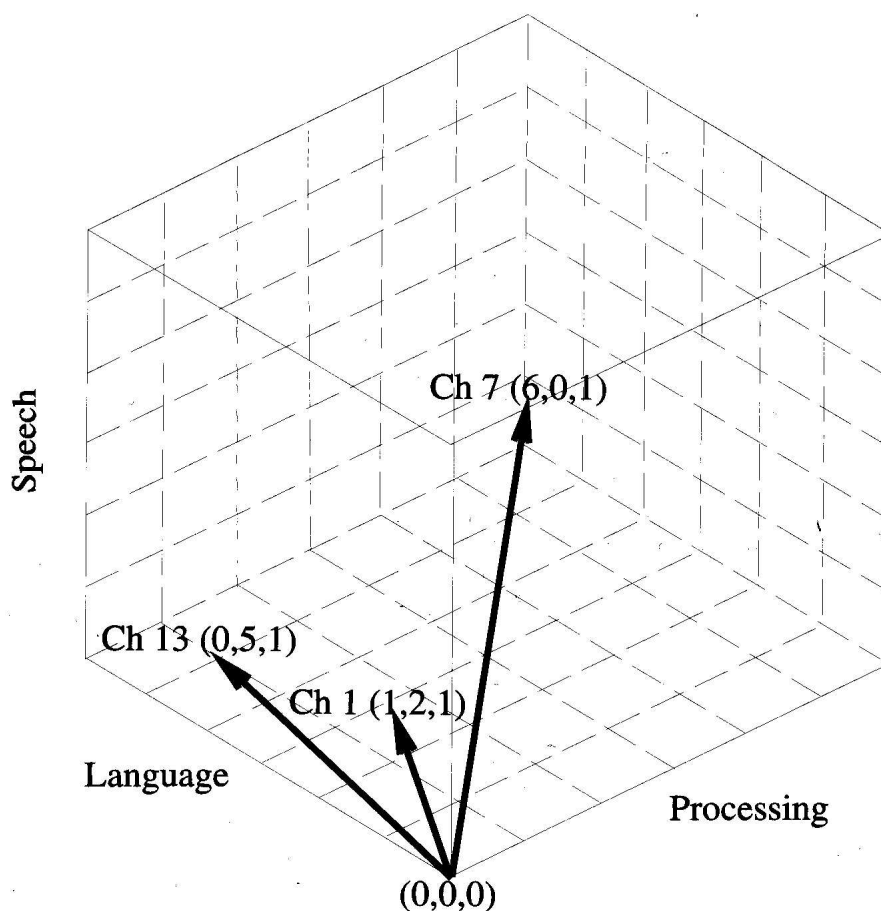
zapytania jest liczba wspólnych wyrażeń.

Wykorzystywanie zmiennych zero-jedynkowych do reprezentacji występowania wyrażeń nie oddaje hierarchii ważności tychże wyrażeń w badanych tekstach, dlatego w praktyce lepiej zastąpić zmienne t wagami w . Wówczas otrzymujemy następującą ogólną postać równań wektorów:

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{N,j}) \quad (6.4)$$

$$\vec{z}_k = (w_{1,k}, w_{2,k}, \dots, w_{N,k}) \quad (6.5)$$

Na zestaw badanych tekstów możemy zatem spojrzeć jak na macierz wag, w której każda wartość w_{ij} oznacza wagę wyrażenia i dla dokumentu j .



Rysunek 6.2: Wektory atrybutów, przykład zaczerpnięty z [JURA00, s.649]

Tak zdefiniowana macierz nazywana jest **macierzą wyrażen i dokumentów** (ang. *terms-by-document matrix*). W powyższy sposób zdefiniowane wektory atrybutów nie dają się jednak precyzyjnie porównywać. Jak widać na rysunku 6.2 wektory *ch1*, *ch2* i *ch3* mają różne długości, czego efektem są błędy w obliczeniach porównawczych.

Rozwiązaniem umożliwiającym dokładne porównanie dokumentów i zapytań reprezentowanych przez wektory atrybutów jest dokonanie normalizacji tych wektorów. Ostatecznie otrzymujemy więc następujące równanie:

$$sim(\vec{z}_k, \vec{d}_j) = \frac{\sum_{t=1}^N w_{i,k} \times w_{i,j}}{\sqrt{\sum_{t=1}^N w_{i,k}^2} \times \sqrt{\sum_{t=1}^N w_{i,j}^2}} \quad (6.6)$$

Równanie 6.6 ma bardzo praktyczną interpretację. Miarą podobieństwa dokumentu *d* i zapytania *z* jest **cosinus kąta między wektorami atrybutów** \vec{d}_j oraz \vec{z}_k reprezentującymi dokument *d* i zapytanie *z*. Funkcja *cosinus* osiąga maksimum wynoszące 1 dla kąta 0°, a więc wtedy, gdy znormalizowane wektory reprezentujące dokument *d* oraz zapytanie *z* są identyczne. Z drugiej strony, ponieważ operujemy na wartościach nieujemnych, minimum funkcji *cosinus* wynosi 0. Ostatecznie miara podobieństwa dokumentu i zapytania w modelu przestrzeni wektorowej $sim(\vec{z}_k, \vec{d}_j)$ przyjmuje wartości z przedziału <0,1>, przy czym im większa jest jej wartość, tym bardziej podobne są: dokument *d* i zapytanie *z*.

Część z wyżej opisanych metod analizy wypowiedzi otwartych została zaimplementowana przez twórców oprogramowania testującego. Poniżej zamieszczono opis jednego z takich systemów, korzystających z dorobku językoznawstwa komputerowego

6.6 System Electronic Essay Rater

– przykład implementacji wykorzystującej osiągnięcia językoznawstwa komputerowego

System o nazwie Electronic Essay Rater (E-rater), analizujący odpowiedzi otwarte, jest rozwijany przez Educational Testing Service, pod kierownictwem Jill Burstein. Stanowi on prototypową implementację badań nad automatycznym ocenianiem zadań otwartych, prowadzonych przez ten ośrodek od początku drugiej połowy lat 90-tych. Program E-rater, w odróżnieniu od wcześniejszych tego typu aplikacji, które mierzyły przede wszystkim czysto statystyczne wartości tekstu, lub zadowalały się jedynie całościową analizą porównawczą treści wypowiedzi, został zbudowany w oparciu o metody językoznawstwa komputerowego. Zalicza się go do grupy systemów przetwarzania języka naturalnego (ang. *Natural Language Processing*).

Autorzy programu wykorzystują w pracach nad tym systemem doświadczenia organizacji ETS w przeprowadzaniu egzaminów typu GMAT. E-rater ma wbudowany sześciostopniowy system oceniania, zgodny ze skalą stosowaną przy ocenie egzaminów GMAT. Aplikacja została tak opracowana, aby przy ocenianiu wykorzystywała kryteria możliwie najbardziej zbliżone do używanych przez ekspertów, oceniających egzaminy GMAT [BURS98]. E-rater używa przeszło 60 zmiennych, które mogą stanowić materiały dowodowe podczas oceny wypowiedzi. Dzięki zastosowaniu takiego podejścia opisywany system spełnia kryterium przejrzystości oceny wymienione w podrozdziale 6.2 w punkcie 1.

Kryteria oceny podzielono na trzy części:

- syntaktyczne,
- retoryczne,
- tematyczne.

Przeprowadzono pomiary występowania wymienionych cech na podstawie zestawu „egzaminów do trenowania” programu, a następnie, wykorzystując regresję liniową, obliczono wagi dla poszczególnych kryteriów oceny tak, aby odpowiadały one przewidywanym wynikom oceny przez ekspertów. Wyniki oceny automatycznej i ręcznej uznawano za zgodne, jeśli wystawione oceny nie różniły się więcej niż jednym punktem. Takie rozumowanie jest stosowane także przy ocenie ręcznej. W przypadku, gdy oceny dwóch ekspertów różnią się więcej niż jednym punktem, praca jest oddawana trzeciemu ekspertowi do oceny [BURS98].

6.6.1 Analiza struktury syntaktycznej

Analiza struktury syntaktycznej stanowi bardzo ważną grupę kryteriów, przyczyniających się do oceny wypowiedzi tekstowych. Umożliwia ona sprawdzenie złożoności syntaktycznej, a więc stopnia skomplikowania budowy zdań (zdania proste lub złożone), typów struktur zależnych i wielu innych cech opisujących techniczną stronę zapisu wypowiedzi. E-rater używa do analizy struktury syntaktycznej narzędzia firmy Microsoft o nazwie *Microsoft Natural Language Processing* (MSNLP). Na podstawie rezultatów przetwarzania tekstu przez ten parser, inne programy wchodzące w skład systemu E-rater obliczają liczbę zdań dopełniających, podrzędnych, bezokolicznikowych oraz zależnych, a także liczbę występowania czasowników modalnych w trybie łączącym (takich jak: *would, could, should, might* i *may*) [BURS98].

6.6.2 Analiza struktury retorycznej

Analiza struktury retorycznej bada zawarte w ocenianej wypowiedzi argumenty, za pomocą których zdający uzasadnia swoją, lub wymienioną w temacie pracy, tezę. E-rater dokonuje obliczeń dotyczących organizacji pracy na podstawie struktury argumentów w wypowiedzi [BURS98]. Autorzy programu wykorzystali ponadto występowanie paralelizmów oraz kontrastu do pomiaru struktury retorycznej. Burstein i inni. [BURS98] podkreśla, że Cohen [COHE84] i innymi badaczami, że zależności retoryczne dają się zidentyfikować za pomocą obserwacji występowania słów kluczowych i specjalnych struktur syntaktycznych. Jako przykład twórcy oprogramowania wykorzystali zależności koniunktywne opisane przez Quirk et al. [QUIR85], który

zauważył, że wyrażenia angielskie „*In summary*” i „*In conclusion*” występują najczęściej na początku podsumowania wypowiedzi, natomiast słowa „*perhaps*” i „*possibly*” są słowami kluczowymi oznaczającymi wyrażanie własnych przekonań. System E-rater zawiera słownik słów kluczowych, będących wyróżnikami struktury retorycznej, a także służących do tego samego celu, reguł heurystycznych, opartych na analizie syntaktycznej oraz badaniu rozkładu występowania słów kluczowych [BURS98]. Z wyżej wymienionych metod korzysta program APA (ang. *argument partitioning and annotation*), dzielący wypowiedź na fragmenty, według występujących w niej argumentów merytorycznych [BURS98].

6.6.3 Analiza zawartości tematycznej

Nieodłączną cechą dobrych wypowiedzi tekstowych jest właściwe skupienie się na zadanym temacie pracy oraz wykorzystywanie odpowiedniego, precyzyjnego słownictwa. Dlatego można oczekiwać, że dobre wypowiedzi będą zawierały podobne sformułowania do tych, stosowanych w dobrych wypowiedziach wzorcowych, a gorsze będą miały słownictwo podobne do słabo ocenionych tekstów przykładowych [BURS98]. E-rater oblicza dwie miary podobieństwa słownictwa. Pierwsza z nich, nazywana przez autorów *EssayContent*, obejmuje porównanie słownictwa całych wypowiedzi, druga, zwana *ArgContent*, skupia się na poszczególnych „jednostkach argumentowych”, wyznaczonych przez moduł APA.

W celu obliczenia wartości *EssayContent* obliczany jest najpierw „superwektor” dla każdej możliwej oceny (od 1 do 6), na podstawie wszystkich wzorcowych wypowiedzi tekstowych. Następnie aplikacja bada cosinus kąta pomiędzy wektorem obliczonym w oparciu o zawartość badanej wypowiedzi, a „superwektorami” i wystawia ocenę taką, jaką reprezentuje najbliższy „superwektor” [BURS98].

Wartość *ArgContent* obliczana jest w analogiczny sposób jak miara *EssayContent* z tym, że w tym przypadku brane są pod uwagę wartości częstotliwości ważonej. Burstein i inni powołują się na metodę odwrotnego badania częstotliwości dokumentu (ang. *inverse document frequency weighting*), opisaną przez Saltona (1988) przy wyznaczaniu wartości częstotliwości ważonej. W wyniku obliczeń otrzymujemy ocenę dla każdego wymienionego w tekście argumentu. Z ocen

wszystkich argumentów obliczana jest średnia ważona, która uwzględnia liczbę wymienionych w tekście argumentów. Autorzy do obliczenia wartości *ArgContent* stosują następujący wzór:

$$ArgContent = (arg + n)/(n + 1) \quad (6.7)$$

gdzie:

arg – średnia ocena dla wszystkich argumentów

n – liczba argumentów.

6.6.4 Ocena końcowa

E-rater oblicza ocenę końcową jako średnią ważoną, z uzyskanych wyników dla każdej ze zmierzonych 67 cech wypowiedzi, uwzględniającą wyprowadzone za pomocą badań empirycznych wagi. Wartości tych wag obliczone zostały za pomocą regresji liniowej, zastosowanej do próbki wzorcowych wypowiedzi, ocenionych wcześniej przez co najmniej dwóch ekspertów.

Ostateczne wyniki porównujące precyzję wystawianych automatycznie przez system E-rater ocen, podają, że średnia zgodność oceny aplikacji z ocenami wystawionymi przez ekspertów wynosiła między 87% a 94%. Tak wysokie wartości precyzji oceny systemu E-rater pozwalają na zastosowanie tego oprogramowania, jako drugiego eksperta przy sprawdzaniu egzaminów, złożonych ze swobodnych wypowiedzi tekstowych.

System E-rater jest przykładem implementacji narzędzia do oceny wypowiedzi otwartych, które korzysta z osiągnięć językoznawstwa komputerowego. Podczas parsowania wykorzystuje on miary dotyczące zarówno treści merytorycznej tekstu, zawartych w nim argumentów, jak i również formy gramatycznej i stylistycznej w jakiej został napisany. W metodzie działania oprogramowanie E-rater łączy częściowo cechy systemu PEG z podejściem LSA, dokonując zarówno pomiarów statystycznych, jak również porównując zawartość merytoryczną prac. Tak szczegółowa analiza wypowiedzi, przeprowadzana przez tę aplikację, pozwala wierzyć w wiarygodność

ocen, które są przez nią generowane.

6.7 Perspektywy systemu analizującego wypowiedzi otwarte w języku polskim

Przedstawione w tej pracy przykładowe implementacje systemów weryfikacji wiedzy analizujących wypowiedzi otwarte, to aplikacje oceniające tylko i wyłącznie teksty angielskie. Dla czytelnika polskiego zainteresowanego tą tematyką nasuwa się pytanie o możliwości stworzenia podobnego oprogramowania dla tekstów pisanych w języku polskim. Zdaniem niektórych językoznawców komputerowych⁸ język polski nie stanowi pod tym względem dużo większego wyzwania niż język angielski, dlatego w niedalekiej przyszłości powinno być możliwe skonstruowanie tego typu aplikacji. Przyjrzyjmy się więc bliżej dzisiejszemu stopniowi zaawansowania prac nad implementacjami rozwiązań poszczególnych problemów, związanych z analizą wypowiedzi otwartych.

Jedną z pierwszych faz maszynowego badania tekstu jest analiza morfologiczna. Pod tym względem, jak pokazuje raport badający dostępne analizatory morfologiczne [HAJN01], istnieje wiele funkcjonujących implementacji, zajmujących się morfologią wypowiedzi otwartych. Jednak żadna z testowanych przez autorki tego raportu aplikacji nie dokonuje dezambiguacji morfoskładniowej. Z drugiej strony w Instytucie Podstaw Informatyki Polskiej Akademii Nauk (IPI PAN) prowadzone są obecnie badania, mające na celu stworzenie tego typu złożonego analizatora morfologicznego, zawierającego tager.

Kolejnym etapem badania zadań otwartych jest analiza syntaktyczna. Pracę prowadzącą do stworzenia aplikacji analizującej gramatykę języka polskiego rozpoczął jeszcze w połowie lat 70. Stanisław Szpakowicz (Uniwersytet Warszawski). Skorzystał on z podejścia znanego w literaturze angielskiej jako *Definite Clause Grammar* i stworzył opis formalny gramatyki,

⁸Duża część wyrażonych w tym podrozdziale poglądów została zaczerpnięta z konsultacji Adama Przepiórkowskiego.

używając do tego języka *Prolog*. Jego prace były kontynuowane przez Marka Świdzińskiego (Uniwersytet Warszawski), który w swojej pracy habilitacyjnej, opublikowanej w 1992 roku [ŚWID92], w sposób teoretyczny rozwinął model stworzony przez Szpakowicza. Obecnie trwają prace nad implementacją wyżej wspomnianego teoretycznego modelu, które prowadzi Marcin Woliński (IPI PAN). Niezależnie od tego podejścia zespół badawczy z IPI PAN wprowadza w życie teorię tzw. *Head-driven Phrase Structure Grammar (HPSG)*, czego dowodem jest ukazanie się w zeszłym roku książki pt.: „Formalny opis języka polskiego” [PRZE02]. W Polsce są jeszcze dwie osoby zajmujące się komputerowym przetwarzaniem gramatyki języka polskiego. Jest to Zygmunt Vetulani (Uniwersytet Adama Mickiewicza), który pracuje nad systemem POLINT i korzysta z podejścia *Definite Clause Grammar*, dokonując jednak analizy gramatycznej w oparciu o słownik leksykalny. Drugą osobą jest współpracownik Vetulaniego, Tomasz Obrębski, który w obronionej niedawno pracy doktorskiej stworzył model gramatyki polskiej w oparciu o jeszcze inne podejście, znane jako gramatyka zależnościowa (ang. *Dependency Grammar*).

Z opisu przedstawionego powyżej widać, że na temat maszynowej analizy morfologicznej i syntaktycznej języka polskiego prowadzonych jest wiele prac. Pomimo tego, jak z moich informacji wynika, nikt dotąd nie zajął się implementacją komputerowego systemu potrafiącego oceniać wypowiedzi otwarte formułowane w języku polskim. Można jednak mieć nadzieję, że ogromny wysiłek jaki wkładany jest w badania nad maszynową analizą tekstów tworzonych po polsku pozwoli w niedługim czasie na stworzenie tego typu aplikacji.

Zakończenie

Przedstawiona w ostatnim rozdziale tematyka systemów weryfikacji wiedzy, analizujących wypowiedzi otwarte pokazuje, jak skomplikowanych metod, z punktu widzenia teorii i implementacji, potrzeba do zbudowania tego typu aplikacji, a także jak wiele zaawansowanej pracy badawczej jest koniecznej do ich wdrożenia. Z pewnością rynek edukacyjny oczekuje przede wszystkim niedrogich produktów, które mogłyby zostać wdrożone w niedługim czasie. Dlatego nie można wymagać, żeby oprogramowanie stanowiące odpowiedź na tę potrzebę rynku wykorzystywało najnowocześniejsze i najbardziej zaawansowane technologie takie jak np. metody przetwarzania języka naturalnego. Istotne więc jest, ażeby opracowywać systemy prostsze i tańsze. Cel ten przyświecał moim kolegom i mnie w tworzeniu systemu KSE przedstawionego w niniejszej pracy. Mam nadzieję, że na obecnym etapie badawczym, system ten okaże się wartościowym i przydatnym narzędziem wspomagania testowania wiedzy w Szkole Głównej Handlowej w Warszawie, a także być może w innych środowiskach akademickich, czy instytucjach szkolących własnych pracowników.

Bibliografia

- [BAKE01] Baker F. B., The Basics of Item Response Theory, ERIC Clearinghouse on Assessment and Evaluation, University of Wisconsin, 2001.
- [BIAL97] Biała księga. Nauczanie się i uczenie się. Na drodze do uczącego się społeczeństwa. Komisja Europejska, Wyższa Szkoła Pedagogiczna TWP, Warszawa, 1997.
- [BURS98] Burstein J., Kukich K., Wolff S., Chi Lu, Chodorow M., Computer Analysis of Essays, NCME Symposium on Automated Scoring, kwiecień 1998, <http://www.ets.org>.
- [BURS99] Burstein J., Wolff S., Chi Lu., Using Lexical Semantic Techniques to Classify Free-Responses, Kluwer Academic Press, 1999.
- [BURS00] Burstein J., Marcu., Benefits of Modularity in an Automated Scoring System. Proceedings of the Workshop on Using Toolsets and Architectures to Build NLP Systems, 18th International Conference on Computational Linguistics, Luxembourg, sierpień 2000.
- [BURS00a] Burstein, J., Marcu D., Towards Using Text Summarization for Essay-Based Feedback., Le 7e Conference Annuelle sur Le Traitement Automatique des Langues Naturelles TALN'2000, Lausanne, Szwajcaria, październik 2000.
- [BURS01] Burstein J., Marcu D., Andreyev S., Chodorow M., Towards Automatic Classification of Discourse Elements in Essays, Annual Meeting of the Association for Computational Linguistics, Toulouse, France, lipiec 2001.

- [CHAR98] Charman D., Elmes A., Computer Based Assessment (Volume 1); A guide to good practice, University of Plymouth, 1998.
- [COFF71] Coffman, Essay Examinations Thorndike Educational Measurement, Second Edition, Washington 1971.
- [COHE84] Cohen R., A computational theory of the function of clue words in argument understanding, International Computational Linguistics Conference, Stanford, s. 251-255, 1984.
- [COOP84] Cooper P. L., The Assessment of Writing Ability, A Review of Research, GRE Board Report No. 82-15R, Educational Testing Service, Princeton, maj 1998.
- [DELO95] DeLoughry, T. J., Duke professor pushes concept of grading essays by computer. Chronicle of Higher Education, s. A24-25, 20 października 1995.
- [DICT] Dictionary of Education Pedagogical Language Usage Server, słownik internetowy prowadzony przez American Educational Research Association (AERA), Frederick L. Goodman, School of Education University of Michigan,
<http://dictionary.soe.umich.edu/plus/>.
- [ENCY95] Nowa Encyklopedia Powszechna, tom 1, Wydawnictwa Naukowe PWN, Warszawa 1995.
- [FOLT98] Foltz P. W., Kintsch W., Landauer T. K. The Measurement of Textual Coherence with Latent Semantic Analysis. Discourse Processes 25, s. 285-307, 1998,
<http://www.knowledge-technologies.com/>.
- [FOLT99] Foltz P. W., Laham D., Landauer T. K., The Intelligent Essay Assessor: Applications to Educational Technology, Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, Volume 1, Number 2, październik 1999, Wake Forest University,
<http://imej.wfu.edu>.

- [FRAS] Frase, Lawrence T., Seven Technologies For Assessment, George Mason University.
<http://mason.gmu.edu/~amelmed/Frase.htm>.
- [GLAS94] Glaser R., Silver E., Assessment, Testing, and Instruction: Retrospect and Prospect, CRESST / Learning Research and Development Center, University of Pittsburgh, Los Angeles, 1994.
- [HAJN01] Hajnicz E., Kupść A., Przegląd analizatorów morfologicznych dla języka polskiego, Instytut Podstaw Informatyki Polskiej Akademii Nauk, Warszawa, grudzień 2001.
- [JURA00] Jurafsky D., Martin J.H., Speech and Language Processing, Prentice Hall, Upper Saddle River, 2000.
- [KAPL98] Kaplan R. M., Wolff S., Burstein J. C., Chi Lu, Rock D., Kaplan B., Scoring Essays Automatically Using Surface Features, GRE Board Report No. 94-21P, Educational Testing Service, Princeton, sierpień 1998.
- [KOBY02] Kobyliński A., Darmetko K., Pura Sz., Rubach P., Szlanta G., Piecko M., Komputerowy System Egzaminowania i Internetowy System Egzaminowania, Materiały z konferencji pt.: Dydaktyka Informatyki Ekonomicznej – Kształcenie Dla Społeczeństwa Informacyjnego, NTIE, Kołobrzeg, wrzesień 2002.
- [KOLB02] Kolbusz J., Skoczylas A., „e-Uniwersytet” – nowoczesne metody zdobywania wiedzy, Materiały z konferencji pt.: e-Uniwersytet, metody i narzędzia, Wyższej Szkole Informatyki i Zarządzania w Rzeszowie, 15-17 września 2002
- [KOMP03] Kompendium wiedzy o sieciach nr 5, NetWorld Akademia, Warszawa, marzec 2003.
- [KUCZ97] Kuczyńska E., Kryterium oceny pisemnych prac maturalnych z języka polskiego, Polonistyka 1997 nr 3.
- [KUKI98] Kukich K., Computer Analysis of Essays, NCME Symposium on Automated Scoring, kwiecień 1998.

- [LAHA97] Laham D., Latent Semantic Analysis Approaches to Categorization., Proceedings of the 19th annual meeting of the Cognitive Science Society (s. 979), Mahwah, Erlbaum, 1997.
- [LAND97] Landauer T. K., Dumais, S. T., A solution to Plato's problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge, Psychological Review, 104, 1997.
<http://www.knowledge-technologies.com/>.
- [LAND98] Landauer T. K., Foltz P. W., Laham D., Introduction to Latent Semantic Analysis. Discourse Processes, 25, 1998.
<http://www.knowledge-technologies.com/>.
- [MCKE99] pod red. McKenna C., Bull J., Designing effective objective test questions: an introductory workshop, CAA Centre, Loughborough University, 1999.
- [MITC98] Mitchell, J. S., Komentarz: SATs don't get you in. Education Week on the Web., 27 maja 1998
<http://www.edweek.org/ew/current/3mitch.h17>.
- [NAJP03] Najpilniejsze problemy do rozwiązania w latach 2004-2015 w systemie edukacji – referat prof. Czesława Banacha i prof. Antoniego Rajkiewicza, 17 marca 2003,
<http://www.prezydent.pl>.
- [NIEM99] Niemierko B., Pomiar wyników kształcenia, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1999.
- [ONEI96] O'Neil H. F. Jr., Sugrue B., Baker E. L., Effects of Motivational Interventions on the National Assessment of Educational Progress Mathematics Performance, Educational Assessment, 3(2), 1996.
- [ORAC01] Oracle iLearning Szkolenie Wersja 4.0, Oracle Corporation, 2001.

- [PAGE94] Page E. B., New Computer Grading of Student Prose, Using Modern Concepts and Software, *Journal of Experimental Education*, 62(2), 1994.
- [PAGE95] Page, E. B., Computer Grading of Essays: A Different Kind of Testing?, Duke University, APA Annual Meeting, 13 sierpnia 1995.
- [PAGE96] Page E. B., Project Essay Grade and Writing Traits: Qualitative and Quantitative Combined?, Duke University, NCME Invited Symposium, Nassau, kwiecień 1996.
- [PAGE96a] Page E. B., Grading Essays by Computer: Why the Controversy?, Duke University, NCME Invited Symposium, Nassau, kwiecień 1996.
- [PANE03] Panek T., Materiały do zajęć publikacje, Instytut Statystyki i Demografii SGH, <http://www2.sgh.waw.pl/sgh/instytuty/isd/isd/publikacje/>, stan strony z dnia 25 kwietnia 2003r.
- [POWE01] Powers D., Burstein J., Chodorow M., Fowles M., Kukich K., Stumping E-Rater: Challenging the Validity of Automated Essay Scoring, GRE Board Professional Report No. 98-08bP, ETS Research Report 01-03, Educational Testing Service, Princeton, marzec 2001.
- [PROM03] Strona domowa firmy Prometric, dane z 11 kwietnia 2003.
<http://www.prometric.com>.
- [PRZE02] Przepiórkowski A., Kupść A., Marciniak M., Mykowska A., Formalny opis języka polskiego. Teoria i implementacja, Exit, Warszawa 2002.
- [QUIR85] Quirk R., Greenbaum S., Leech G., Svartik J., A Comprehensive Grammar of the English Language, Longman, New York 1985.
- [QUOT] Internetowy zbiór cytatów na stronie
<http://www.quotationspage.com>.

- [REHD98] Rehder B., Schreiner M. E., Wolfe M. B., Laham D., Landauer T. K., Kintsch W. Using Latent Semantic Analysis to Assess Knowledge: Some Technical Considerations., *Discourse Processes* 25, 1998.
<http://www.knowledge-technologies.com/>.
- [ROCK85] Rocklin T., Thompson J. M., *Journal of Educational Psychology*, nr 77, American Psychological Association, Washington DC, 1985.
- [ROKI02] Rokicka-Broniatowska A., Nojszeski D., Schab A., Wspomaganie procesu dydaktycznego na studiach zaocznych technologiami multimedialnymi, Materiały z ogólnopolskiej konferencji pt.: „Studia zaoczne – jakość kształcenia i perspektywy rozwoju”, Szkoła Główna Handlowa, Warszawa, 13 grudnia 2002.
- [ROKI02a] Rokicka-Broniatowska A., Nojszeski D., Schab A., Darmetko K., Rubach P., Pura Sz., Modelowe rozwiązania kształcenia na odległość w uczelnianym kampusie SGH, Szkoła Główna Handlowa, Warszawa, listopad 2002.
- [RUSS97] Russel M., Haney W., Testing Writing on Computers: An Experiment Comparing Student Performance on Tests Conducted via Computer and via Paper-and-Pencil, *Education Policy Analysis Archives Volume 5 Number 3*, College of Education, Arizona State University, 1997.
- [RUSS99] Russel M., Testing On Computers: A Follow-up Study Comparing Performance On Computer and On Paper, *Education Policy Analysis Archives Volume 7 Number 20*, College of Education Arizona State University, 1999.
- [SALT71] Salton G., The SMART Retrieval System Experiments in Automatic Document Processing, Prentice Hall, Englewood Cliffs 1971.
- [SALT88] Salton G., Automatic text processing: The transformation, analysis, and retrieval of information by computer, Addison-Wesley, Reading, Mass, 1988.
- [STAL51] Stalnaker, The Essay Type Examination Educational Measurement, ACE, Washington 1951.

- [STEI01] Steinhart D., Summary Street: an Intelligent Tutoring System for Improving Student Writing Through the Use of Latent Semantic Analysis, niepubl. praca doktorska, Institute of Cognitive Science, University of Colorado, Boulder, 2001.
- [SZAF95] Szafran K., Analizator morfologiczny SAM-95 opis użytkowy, Instytut Informatyki Uniwersytetu Warszawskiego, 1995,
<http://www.mimuw.edu.pl/~kszafran/SAM-dists/>.
- [ŚWID92] Świdziński M., Gramatyka formalna języka polskiego, Wydawnictwa Uniwersytetu Warszawskiego, Warszawa, 1992.
- [THOM03] Materiały firmy Thomson Corporation na konferencję pt. „Global Services Growth Conference” z dnia 10 marca 2003,
http://www.thomson.com/corp/pressroom/pr_exec_presentations.jsp.

Spis rysunków

3.1	Panel administracyjny systemu iLearning	33
3.2	Okno zdającego podczas odpowiadania na pytania	37
5.1	Okno logowania do panelu administracyjnego systemu KSE	50
5.2	Architektura systemu KSE	54
5.3	Panel administracyjny – okno wyboru przedmiotu	56
5.4	Okno serwera sesji przedstawiające trwający egzamin	57
5.5	Okno stanowiska egzaminacyjnego w trakcie trwania egzaminu	58
5.6	Panel administracyjny – okno edycji pytań	61
5.7	Okno powitalne kreatora sesji egzaminacyjnej	64
5.8	Wynik końcowy egzaminu, wyświetlany po zakończeniu odpowiadania na pytania	66
5.9	Panel administracyjny wykładowcy – szczegółowe wyniki egzaminu	67
6.1	Przykład działania analizatora morfologicznego SAM, autorstwa Krzysztofa Szafrana	84
6.2	Wektory atrybutów, przykład zaczerpnięty z [JURA00, s.649]	88