Wojciech Mueller, Piotr Boniecki, Sebastian Kujawa
Instytut Inżynierii Rolniczej
Akademia Rolnicza w Poznaniu

TECHNOLOGIE OLEDB, ADO I ADO.NET W SYSTEMACH INFORMATYCZNYCH WSPOMAGAJĄCYCH PROCES WERYFIKACJI WIEDZY STUDENTÓW

Streszczenie

Proces weryfikacji wiedzy klasycznymi metodami, przy wzrastającej liczbie studentów oraz generalnie niskiej umiejętności przelewania swoich myśli na papier, jest wyjątkowo czasochłonny i w coraz większym stopniu subiektywny. W tej sytuacji uzasadniona jest budowa systemów informatycznych, pozwalających na sprawdzenie wiedzy studentów. Wśród technologii ułatwiających tworzenie tego typu aplikacji, które powinny mieć charakter internetowy, jest technologia OLEDB wraz z interfejsem ADO i ADO.NET. Upraszczają one w znaczący sposób pobór informacji z różnych źródeł danych, które stanowią podstawę budowy pytań. Na bazie tych technologii wytworzono systemy informatyczne, wspomagające proces weryfikacji wiedzy zdobytej przez studentów.

Słowa kluczowe: weryfikacja wiedzy, OLEDB, ADO, ADO.NET

Wprowadzenie

Wzrastająca dynamika procesu kształcenia, mająca swoje źródło w stosunkowo szybko pojawiających się nowych obszarach wiedzy, które muszą sobie przyswoić wpierw wykładowcy, a w następnej kolejności studenci, w połączeniu ze zbyt daleko idącym liberalizmem w zakresie form wypowiedzi ustnej i pisemnej, powoduje znaczne komplikacje w procesie weryfikacji wiedzy słuchaczy. Szczególnie wnikliwej uwagi, w trakcie sprawdzania, wymagają prace związane z nowymi gałęziami wiedzy, albowiem tutaj napotykamy ciągłą modyfikację już istniejących definicji i pojawianie się całkiem nowych pojęć, które niejednokrotnie są rozszerzeniem znaczeniowym terminów potocznych. Wygodnym rozwiązaniem tej sytuacji problemowej, aczkolwiek nie wolnym od wad, jest wprowadzanie testów udostępnianych w postaci elektronicznej lub tradycyjnej formie papierowej. Pierwsza forma wymaga większych nakładów, bowiem wymusza budowę systemu informatycznego i to z reguły o architekturze rozproszonej. Jednak za takim rozwiązaniem przemawiają dodatkowo szersze możliwości formułowania zapytań i odpowiedzi,

wzbogaconych o prezentacje graficzną, w postaci obrazu lub filmu i dźwiękową. Jest to szczególnie istotne w zagadnieniach, które są zlokalizowane na styku techniki i rolnictwa. Drugoplanową zaletą użycia testów w postaci elektronicznej jest możliwość generowania dla każdego uczestnika indywidualnego zestawu pytań oraz pełna automatyzacja procesu sprawdzania wraz z późniejszym udostępnianiem wyników.

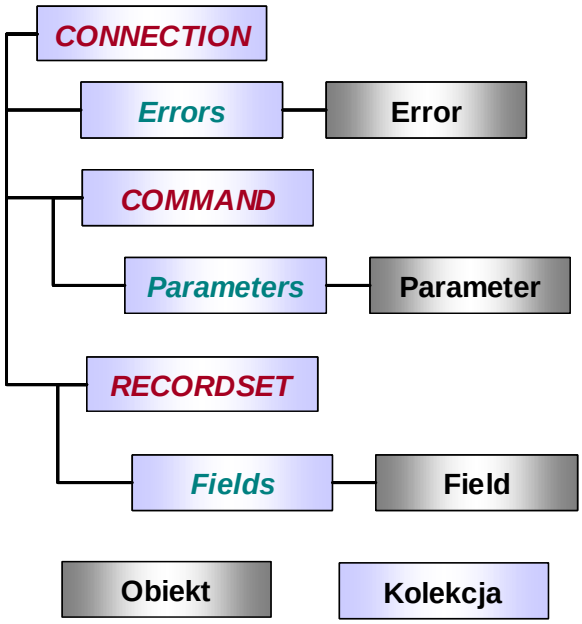
Celem publikacji jest zaprojektowanie, połączone z wieloaspektowym modelowaniem, oraz wytworzenie dwóch systemów informatycznych, w tym jednego o architekturze rozproszonej, wspoma-gających proces weryfikacji wiedzy studentów. Do wytworzenia systemu wykorzystano najnowsze technologie, takie jak: XML, .NET w tym ASP.NET

Technologia OLEDB i ADO.NET

Większość tworzonych systemów informatycznych przetwarza dane, które są przechowywane w SZBD. Zaletą takiego rozwiązania jest zapewnienie bezpieczeństwa danych, szybkość ich wyszukiwania, eliminowanie potencjalnych konfliktów w przypadku jednoczesnego dostępu do tych samych informacji wielu użytkowników. Wgląd do danych zawartych w dowolnym SZBD z poziomu aplikacji uzyskujemy przez zastosowanie technologii ODBC lub nowszej OLE DB.
Aktualnie częściej korzystamy z tego drugiego standardu, jeżeli mamy wybór, za którym kryje się szereg interfejsów COM pozwalających na korzystanie z informacji zawartych w bazie danych [Gunderloy, Chipman 1999]. Z punktu widzenia tej technologii aplikacje, również te, które wchodzą w skład określonego SZBD, dzielą się na dostawców i konsumentów danych. Oznacza to, iż zbiór wspomnianych interfejsów musi być zaimplementowany zarówno przez dostawców, jak i odbiorców danych, wtedy dopiero uzyskujemy możliwość manipulowania przechowywanymi informacjami. Zatem proces tworzenia systemu informatycznego, wykorzystującego standard OLE DB, musi zawierać w sobie implementację sygnalizowanych interfejsów COM.

Realizacja ta w sposób bezpośredni jest wyjątkowo kłopotliwa i wymaga sporej wiedzy ze strony programisty. Łatwiejsza droga prowadzi przez wykorzystanie modelu obiektowego ADO (ActiveX Data Objects), będącego zbiorem obiektów implementujących zdefiniowane interfejsy COM. Ów produkt Microsoft można wykorzystać w środowisku Visual C++, Visual Basic oraz w każdym innym środowisku programistycznym, pozwalającym na realizację odwołań do biblioteki typów, w efekcie czego zyskujemy dostęp do obiektów tworzących te biblioteki.

Ten prosty interfejs programistyczny ma niepełną strukturę hierarchiczną, (rys. 1), i korzystanie z niego nie jest zbyt kłopotliwe. Tworzące go obiekty, wykorzystujące język SQL, możemy traktować jako elementy składowe języka do manipulacji danymi. Obiektem zlokalizowanym najwyżej w hierarchii ADO jest Connection, który poprzez ustawienie odpowiednich właściwości w skojarzeniu z metodą Open lub tylko przez jej wywołanie z określonymi parametrami pozwala połączyć się ze źródłem danych.

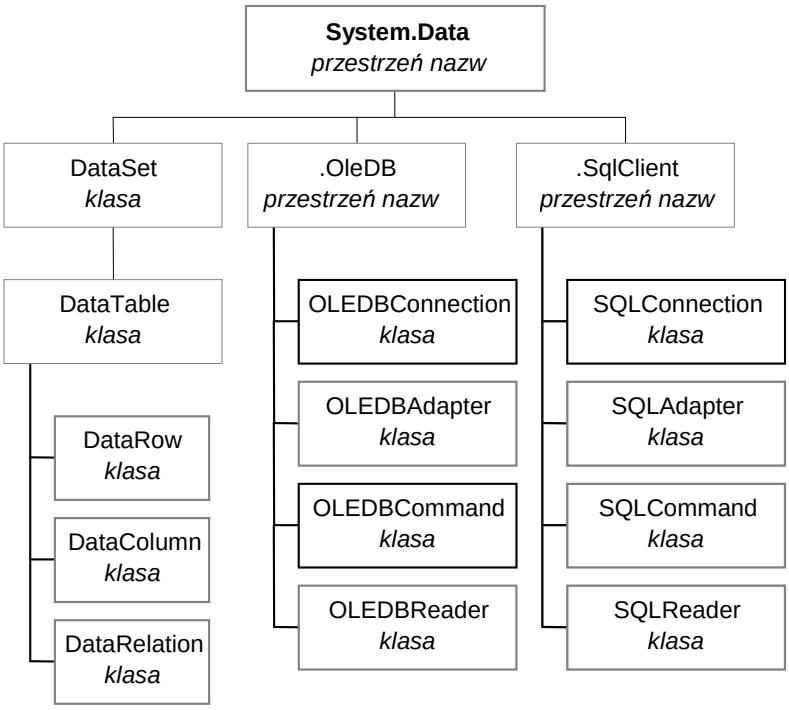


Rys. 1. Struktura obiektów ADO [Gunderloy M., Chipman M.]
Fig. 1. The structure of the ADO objects

Dla prostych, jednorazowych operacji wykonywanych na bazie danych przeważnie korzystamy z tego obiektu w sposób niejawny. Pobór danych do dalszego przetwarzania z wspomnianego źródła może odbywać się na klika sposobów, ale niezależnie od tego jak to zrobimy, informacje zostaną przekazane do obiektu Recordset. Oprócz obiektu Conection jest to kluczowy obiekt w ADO. Wykorzystując jego właściwości oraz metody zyskujemy pełen wachlarz operacji pozwalających na wyszukiwanie danych, przekazywanie informacji do zmiennych języka bazowego Visual C++ lub innego, modyfikację itp. W zależności od typu utworzonego obiektu Recordset, zmiany dokonywane przez innych użytkowników w tabelach są dla nas widoczne bądź nie.

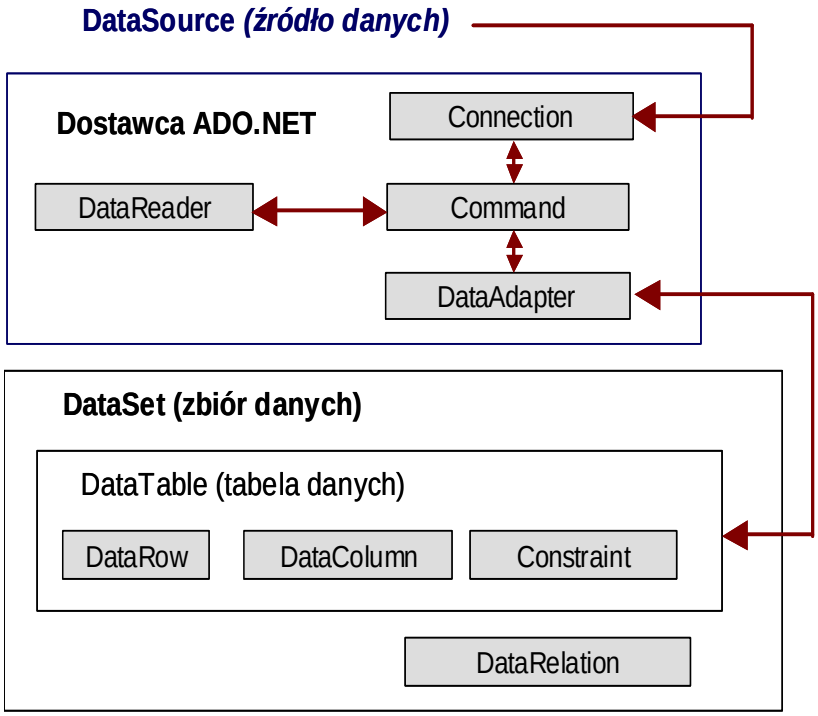
ADO to nie tylko zbiór obiektów pozwalających na manipulowanie danymi, ale to również biblioteki wyposażone w obiekty umożliwiające definiowanie danych i mające wpływ na bezpieczeństwo informacji. Należy podkreślić, że tego typu usługi muszą być również zaimplementowane przez dostawcę, w przeciwnym razie nasze działanie zakończy się niepowodzeniem.

Nowszy model dostępu do danych ADO.NET zaszyty w platformie FrameWork [Payne Ch.] został całkowicie przebudowany, a patrząc na jego strukturę dostrzegamy więcej różnic niż elementów wspólnych. Z uwagi na fakt, iż ADO.NET jest zbiorem klas (rys. 2) technologii .NET, tym samym nie stanowi już fragmentu OLE DB, ale potrafi z nią współpracować. Sposób wykorzystania nowego modelu dostępu do danych w aplikacjach bazodanowych przedstawia rysunek 3.



Rys. 2. Hierarchia przestrzeni nazw i obiektów ADO.NET
Fig. 2. The hierarchy of the ADO.NET namespace and objects

Wgląd w dane z poziomu systemu informatycznego uzyskujemy dwiema drogami, przy czym dłuższy sposób rozszerza operacje, jakie możemy wykonać na zbiorze informacji.



Rys. 3. Sposoby wykorzystania ADO.NET [Payne Ch.]
Fig. 3. Methods of using the ADO.NET

W obu przypadkach tworzymy obiekt Connection na podstawie klas, zlokalizowanych w różnych przestrzeniach nazw. Wybór przestrzeni nazw i związanej z nim klasy jest uzależniony od techniki łączenia się ze źródłem danych, co ilustruje tabela 1.

Tabela 1. Przestrzenie nazw wraz z klasami tworzącymi połączenie
Table 1. Namespaces with the classes creating a link

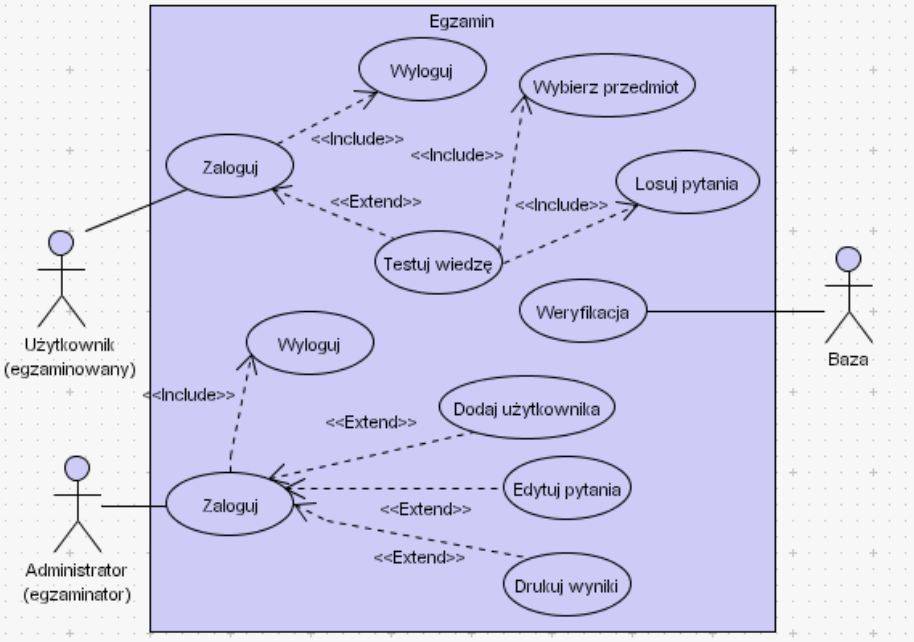
Przestrzeń nazw	Nazwa klasy
System.Data.SqlClient	SQLConnection
System.Data.OleDb	OLEDBConnection
System.Data.ODBC	ODBCConnection

Pierwszy z obiektów, utworzony na bazie klasy SQLConnection, pozwala tylko na łączenie z SQL Serwerem 2000. W pozostałych przypadkach

korzystamy z pośrednictwa ODBC lub OLE DB i zyskujemy możliwość współpracy z dowolnym SZBD, aczkolwiek połączenia te są zdecydowanie wolniejsze. Dalsze postępowanie wymaga użycia obiektu Command oraz DataReader, w którym będą zawarte dane pobrane ze źródła, ale tylko z możliwością odczytu. Tworzenie obiektów odbywa się analogicznie, jak poprzednio, przez wykorzystanie odpowiednich klas zlokalizowanych w trzech przestrzeniach nazw. Drugi sposób pozwalający nie tylko przeglądać pobrane dane, ale również je modyfikować, usuwać i dodawać wymaga zastosowania obiektów DataAdapter i DataSet wraz z jego obiektami podrzędnymi.

Projektowanie

Proces projektowania systemu informatycznego, wspomagający weryfikację wiedzy studentów, przebiegał przy pełnym wykorzystaniu metodyk obiektowych oraz zasad inżynierii oprogramowania. Zidentyfikowane funkcjonalne wymagania systemu zostały odwzorowane w diagramach przypadków użycia przez aktorów i przyporządkowane im scenariusze. Obrazują one możliwe interakcje pomiędzy rozpoznanymi aktorami a aplikacją (rys. 4).



Rys. 4. Diagram przypadków użycia
Fig. 4. The use case diagram

Jako narzędzie do budowy diagramów przypadków użycia i klas zastosowano program Visual Paradigm, implementujący graficzny język modelowania UML [Booch i in. 2000]. Rozpoznane elementy składowe systemu, na etapie projektowania, zostały zaprezentowane w postaci diagramów klas, zawierających klasy nietrwałe i trwałe. Te ostatnie stanowiły punkt wyjścia dalszego postępowania, obejmującego modelowanie logiczne i fizyczne. Ten etap musiał być poprzedzony decyzjami odnośnie zastosowanego modelu danych i w dalszej kolejności o wykorzystanym SZBD. Zdecydowano się na relacyjny model danych, który jest częściej implementowany, w związku z tym modelowanie logiczne oznaczało przekształcenie modelu obiektowego w schemat relacyjny [Muller 2000].

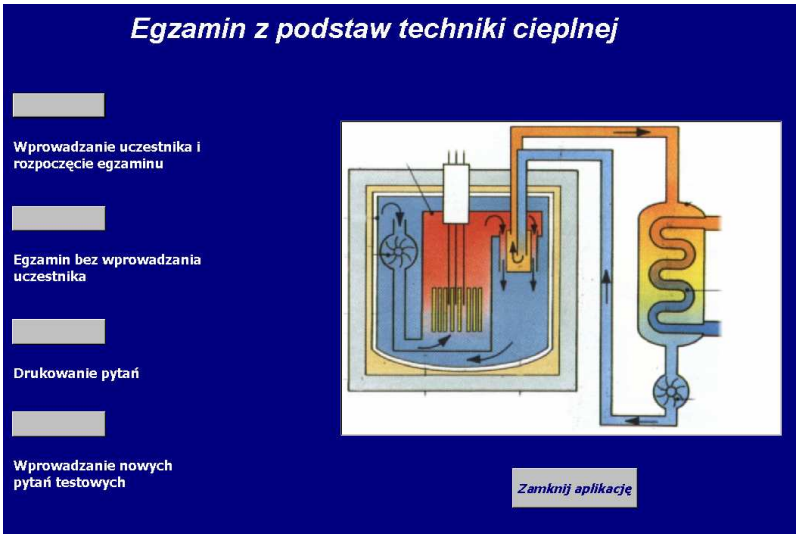
Implementacja

Równie istotną formą przekazu co tekst, na gruncie inżynierii rolniczej i nie tylko, jest szeroko rozumiana grafika, za którą kryje się grafika wektorowa, rastrowa, animacje i filmy. Ten fakt winien znaleźć odzwierciedlenie w pytaniach testowych, stanowiących podstawę weryfikacji wiedzy studentów. Wynika z tego szereg problemów informatycznych; gdzie i w jakich formatach przechowywać dane graficzne, co należy zrobić aby zapewnić spójność danych z jednoczesnym zagwarantowaniem ich bezpieczeństwa, jak pogodzić ze sobą szybkość działania aplikacji z odpowiednią jakością grafiki. Zaprezentowana lista problemów nie wyczerpuje ich.

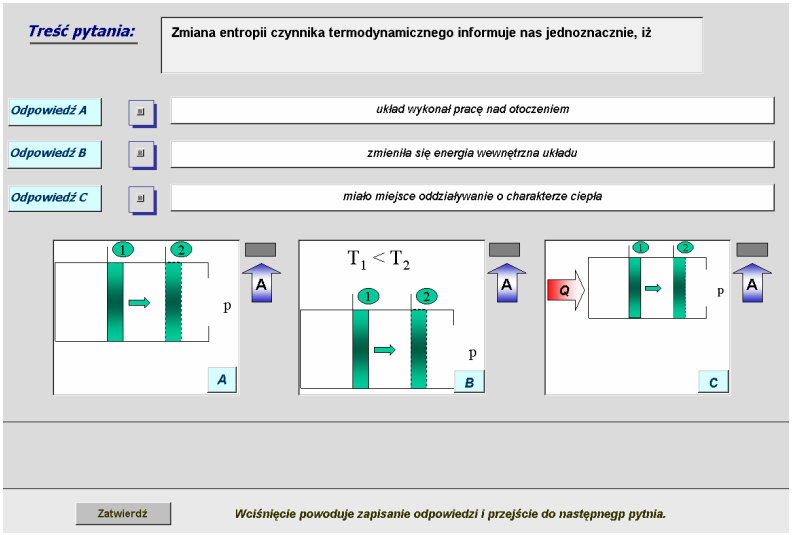
Zaproponowane przez autorów rozwiązanie w postaci dwóch systemów informatycznych nie są próbą odpowiedzi na przedstawione wyżej problemy, a jedynie propozycją szybkiego, niezbyt pracochłonnego, w miarę obiektywnego sposobu sprawdzenia wiedzy słuchaczy.

Pierwsza z aplikacji nie wymaga działającej sieci i powstała przy wykorzystaniu SZBD jakim jest Access oraz obiektów ADO [Forte i in. 2002]. Elementy graficzne wyjaśniające pytania lub będące odpowiedziami zostały zlokalizowane w sposób mieszany. Obrazy statyczne zawarto w bazie danych poprzez wykorzystanie pola typu obiekt OLE, natomiast animacje są przechowywane w plikach zewnętrznych, do których mamy dostęp poprzez informacje zawarte w polu hiperłącze.
Architektura drugiego systemu informatycznego ma charakter rozproszony i aplikacja bazuje na technologii ASP.NET oraz współpracuje ze źródłem danych, którym jest SQL Server 2000 [Waymire, Sawtell 2002]. Grafika wraz z animacjami zlokalizowana jest w plikach zewnętrznych.
Funkcjonowanie obu systemów jest zbliżone i przebiega w wyniku przemieszczania się po hierarchicznie uporządkowanych formularzach,

począwszy od panelu sterowania (rys. 5). Wygenerowany zestaw pytań jest dostępny w postaci elektronicznej (rys. 6) lub w postaci przygotowanego raportu do wydruku.



Rys. 5. Panel sterowania
Fig. 5. The control panel



Rys. 6. Wylosowany zestaw pytań
Fig. 6. A set of questions drawn at random

Na razie wytworzony system informatyczny, z uwagi na zakres wprowadzonych danych do SZBD, może wspomagać proces weryfikacji z zakresu techniki cieplnej. Uzupełnienie go o pytania z pozostałych przedmiotów wyeliminuje to tymczasowe ograniczenie.

Wnioski

Wspomaganie procesu kształcenia przez wykorzystywanie specjalistycznych systemów informatycznych staje się powoli standardem. Jednym z elementów składowych wspomnianego procesu, stwarzającym specyficzne problemy, jest weryfikacja zdobytej wiedzy. Zastosowanie na tym etapie pomocy w postaci aplikacji testującej niewątpliwie przyspieszy oraz zobiektywizuje proces sprawdzania wiadomości. Podjęte przez autorów prace, obejmujące projektowanie, wytwarzanie i testowanie systemów informatycznych wspomagających ów proces pozwoliły na sformułowanie następujących uwag i wniosków:

- Wprowadzenie elementów graficznych w postaci obrazów statycznych i dynamicznych rozszerza możliwości tworzenia pytań jak i odpowiedzi testowych, ale jednocześnie generuje problemy informatyczne, co do sposobu ich przechowywania przy jednoczesnym zachowaniu spójności informacji.
- Spora liczba technologii informatycznych, możliwych do wykorzystania przy tworzeniu tego typu aplikacji, wymaga precyzyjnego sformułowania kryteriów wyboru.
- Wytworzone aplikacje powinny stanowić element składowy większego systemu informatycznego wspomagającego proces kształcenia.

Bibliografia

Booch G., Rumbaugh J., Jacobson I. 2000. UML przewodnik użytkownika. WNT, Warszawa

Forte S., Howe T., Wall K., Kimmel P., Mullen R. 2002. Access 2002 Projektowanie baz danych. Helion, Gliwice

Gunderloy M., Chipman M. 1999. SQL Server 7. Mikom, Warszawa.

Muller R., J. 2000. Bazy danych język UML w modelowaniu danych. Mikom, Warszawa

Payne Ch. 2002. ASP.NET dla każdego. Helion, Gliwice

Waymire R., Sawtell R. 2002. MS SQL Server 2000 dla każdego, Helion, Gliwice

OLEDB AND ADO.NET TECHNOLOGIES IN INFORMATION SYSTEMS SUPPORTING VERIFICATION OF THE STUDENTS' KNOWLEDGE

Summary
Traditional knowledge verification is exceptionally time consuming and subjective as the number of students increases and, generally, their writing skills are relatively low. This situation justifies construction of the information systems allowing verification of students' knowledge. Among the internet technologies facilitating development of such applications there are the OLEDB and its interfaces – ADO and ADO.NET. They significantly simplify acquiring information from various data sources as the basis for constructing questions. With the use of such technologies the information systems were developed to support verification of knowledge gained by students.

Key words: knowledge verification, OLEDB, ADO, ADO.NET

Recenzent – Małgorzata Jaros