

Tworzenie aplikacji z wykorzystaniem języka Java i systemu Oracle na przykładzie “Naukowej Bazy Danych”

Agnieszka Węgrzyn, Bogdan Kromski, Adam Wawrowicz,
Politechnika Zielonogórska
ul. Podgórna 50, 65-246 Zielona Góra
E-mail: A.Wegrzyn@pz.zgora.pl

Streszczenie

Powstanie „Naukowej Bazy Danych” jest odpowiedzią na rosnące zapotrzebowanie na informacje o działalności naukowej ośrodków akademickich i naukowych w regionie lubuskim. W czasach gdzie wygodny dostęp do informacji związanych z naukowym dorobkiem instytucji i organizacji akademickich, staje się kluczem do rozwiązywania nowych problemów i wyzwań nauki, pomysł stworzenia takiej aplikacji i udostępnienia jej w sieci Internet staje się idealnym rozwiązaniem. Choć istnieje dość duża liczba podobnych aplikacji bazodanowych, są to zwykle aplikacje pracujące ze ściśle określonymi strukturami baz danych i prezentujące dane w ściśle określony sposób (zdefiniowany przez twórców oprogramowania). Nie wydaje się więc możliwe przeniesienie takiej aplikacji i struktury bazy do innego ośrodka naukowego bez konieczności ingerowania w kod programu.

1. Wprowadzenie

Charakter twórczości naukowej jest odmienny dla różnych organizacji naukowych, nie wydaje się więc możliwe stworzenie uniwersalnej aplikacji udostępniającej dane o takiej twórczości i dorobku naukowym. Korzystne byłoby stworzenie takiej aplikacji, która umożliwiałaby prezentowanie danych w sposób, który odpowiada lokalnej strukturze bazy danych oraz wizualnym upodobaniom użytkowników.

Korzystając z szerokiej możliwości i udogodnień jakie daje programiście środowisko Javy w systemach rozproszonych stworzyć można ciekawą aplikację. Chociaż użyteczność i atrakcyjność aplikacji zależy wyłącznie od programisty to nie bez znaczenia jest w tym przypadku system bazy danych, z którego program ma korzystać. Java jest językiem, który cały czas dynamicznie się rozwija. Mimo, że nie istnieje jeszcze w pełni ustalony standard, nie zraża to nawet renomowanych producentów systemów baz danych, wprost przeciwnie, starają się oni nadążać za zmianami w Javie dostarczając nowe wersje sterowników umożliwiających korzystanie z baz za pomocą tego języka. Dobrym przykładem może tu być Oracle, który opracowuje nowy standard strukturalnego języka zapytań SQL w pełni zgodnego z Javą (specyfikacja JDBC - Java DataBase Connectivity). Grono producentów dostarczających sterowniki do swoich baz danych wciąż się powiększa co pozwala aplikacji napisanej w Javie już w tej chwili korzystać z większości popularnych systemów baz danych.

Szkielet interakcji aplikacji z systemem bazy danych opracowywany jest w oparciu o strukturę bazy danych. Z tego wynika, że etap projektowania struktury bazy jest równie ważny jak proces oprogramowania. Uważnie opracowana baza danych w połączeniu z wolnym od błędów, w pełni wykorzystującym jej możliwości programem, daje użyteczną i atrakcyjną aplikację.

Doświadczenie w pisaniu aplikacji z wykorzystaniem baz danych po pewnym czasie sprawia, że programista zdaje sobie sprawę z tego, że nawet najlepszy jego program wciąż zależy od struktury wykorzystywanej bazy. Zmiana w tabeli, dodanie lub usunięcie kolumn, zmiana relacji w najlepszym przypadku sprawia, że poświęcić trzeba wiele czasu na zmiany w kodzie. Czasami aplikacja musi być nawet napisana od początku.

2. Opis aplikacji

2.1. Tworzenie i definiowanie interfejsu

Prezentowana aplikacja jest w pełni „odporna” na zmiany wykorzystywanej przez nią struktury bazy danych. Informacje o niej przechowywane są w pliku, w formacie, który jest „jasny” dla aplikacji. Startujący program odczytuje i odpowiednio interpretuje dane o strukturze zawarte we wcześniej stworzonym pliku. Oprócz samej struktury w pliku tym przechowywane są również informacje w jaki sposób dane z bazy mają być przedstawiane. Innymi słowy opisany jest w nim interfejs użytkownika. W momencie kiedy w bazie nastąpiły jakieś zmiany wystarczy poprzez odpowiednie narzędzie wygenerować ponownie dla danej bazy interfejs użytkownika.

Narzędziem służącym do generowania interfejsu użytkownika jest *zarządca aplikacji*. Pozwala on wybrać tabele, na podstawie których stworzona ma zostać struktura, opisać właściwości tabel i kolumn oraz zapisać plik, z którego korzystać będzie aplikacja. Właściwości tabel i kolumn zawierają między innymi informacje o tym, w jaki sposób dane w nich przechowywane mają być udostępnione użytkownikowi.

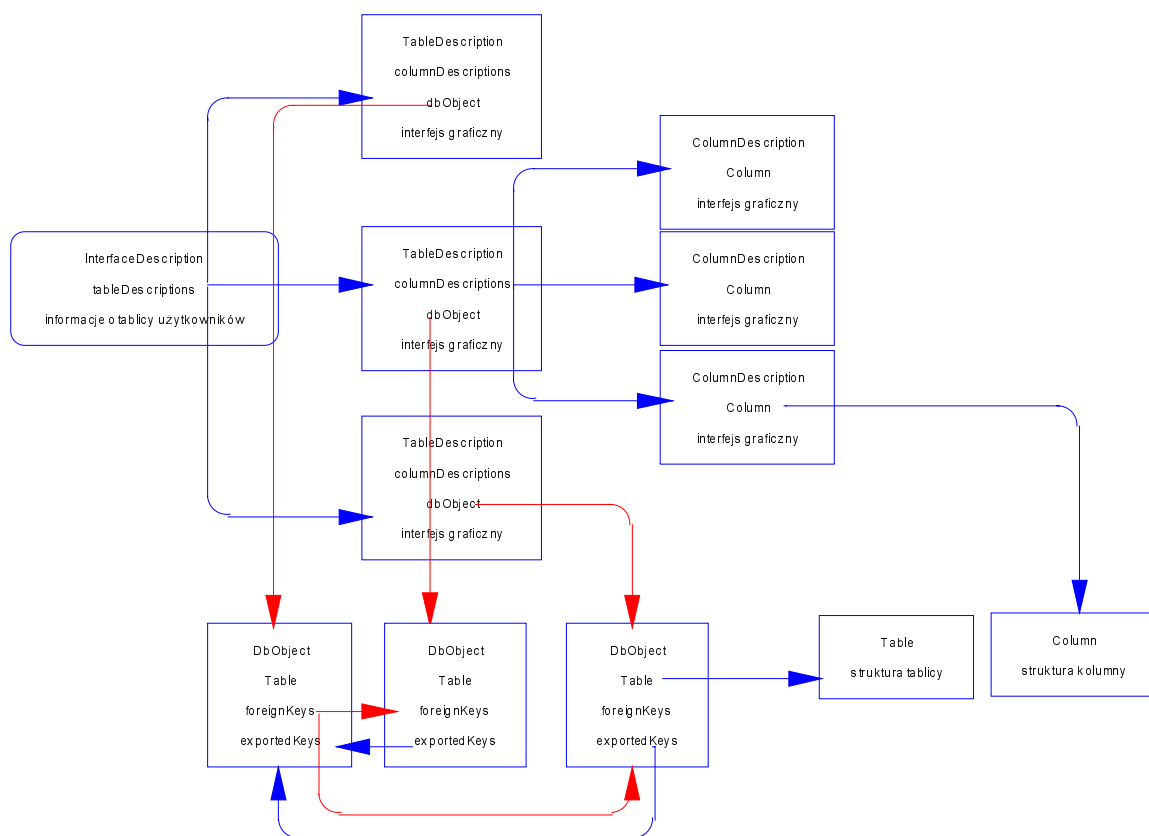
Na obecnym etapie aplikacja przedstawia dane za pomocą drzewa, powiązanej z nim tabeli, info panela wyświetlającego dane wybranego w tabeli rekordu oraz odpowiednich dialogów służących do dodawania i edycji rekordów. Idea elastyczności aplikacji może być równie dobrze przeniesiona na komponowanie przez użytkownika własnych formularzy, wstawianie różnych komponentów (np. zakładek, tabel, przycisków, różnego rodzaju pól edycyjnych) o zdefiniowanym wyglądzie i pozycji na formularzu oraz powiązaniu ich z akcjami wykonywanymi na bazie danych. W odróżnieniu od istniejących już systemów umożliwiających użytkownikowi definiowanie własnego interfejsu graficznego (np. MSAccess firmy Microsoft), pomijając korzyści jakie niesie sama Java, narzędzie takie pozwoli oddzielić właściwy program od zapisanego wcześniej w formie pliku, jego interfejsu.

Za pomocą *zarządcy aplikacji* istnieje możliwość wybrania trybu autoryzacji dostępu. Informacja ta także zapisywana jest w pliku zawierającym strukturę, i powoduje, że w aplikacji odczytującej z niego swój interfejs konieczne będzie wprowadzenie nazwy i hasła użytkownika. W *zarządcy aplikacji* wybierana jest tablica przechowująca informacje o użytkownikach (konieczne są takie kolumny jak: nazwa użytkownika, jego hasło i uprawnienia). W chwili kiedy wybrany zostaje tryb autoryzacji dostępu i podana zostaje tablica użytkowników, jest ona traktowana inaczej niż inne tablice zawierające „zwykłe dane”. Przede wszystkim dane z tej tablicy nie będą dostępne dla „zwykłego użytkownika”, a jedynie dla osób do tego uprawnionych. W chwili obecnej aplikacja rozpoznaje dwa typy uprawnień: administrator oraz zwykły użytkownik. W momencie wprowadzenia nazwy użytkownika i odpowiadającego mu hasła program rozpoznaje również jego uprawnienia. Tylko użytkownik o uprawnieniach administratora ma możliwość ingerowania w dane zawarte w bazie (edycja, dodawanie, usuwanie), pozostali mogą jedynie przeglądać zawartość bazy danych. System uprawnień można w przyszłości również poszerzyć o inne prawa (do zapisu, odczytu, usuwania, prawa na konkretne zasoby) oraz grupy użytkowników.

Jeśli w zarządcy aplikacji nie zostanie wybrany tryb autoryzacji dostępu, program jest dostępny dla każdego w pełnym zakresie.

Autoryzacja dostępu może być szczególnie przydatna w systemach rozproszonych (wykorzystując program np. jako aplet w sieci WWW), gdzie celowe jest ograniczanie lub rozszerzanie praw poszczególnych użytkowników, gdzie istotne jest kto ma możliwość ingerencji, przeglądu, w jaką część danych, o jakiej porze itp.

Obiektem wiążącym aplikację z bazą danych jest zserializowana (zapisana w formie pliku) struktura przechowująca informacje o tabelach, kolumnach, relacjach oraz o formie wyświetlania danych zawartych w bazie. Poniżej przedstawiony jest schemat obrazujący tę strukturę.



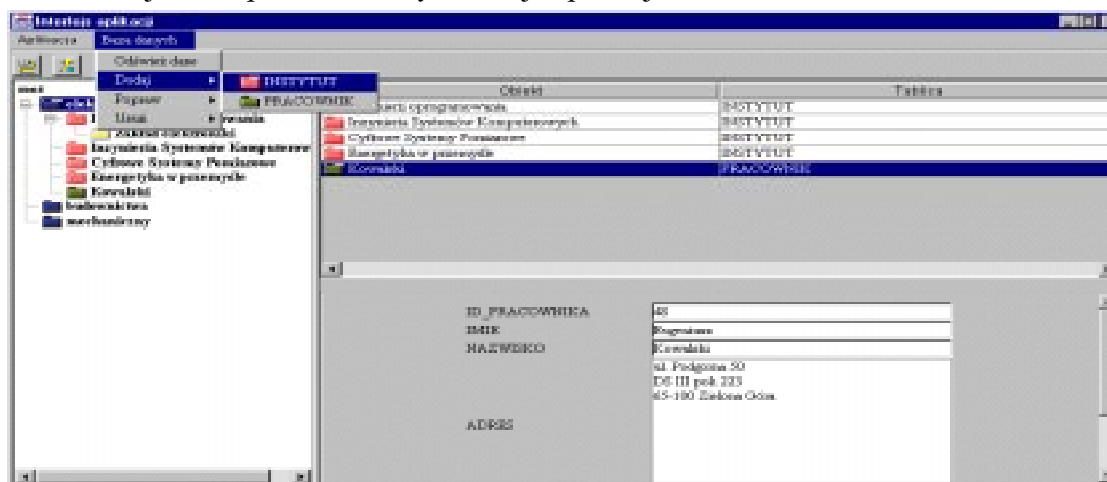
Rys. 1 Schemat obrazujący strukturę przechowującą informacje o tabelach, kolumnach i relacjach

Struktura generowana jest na podstawie informacji uzyskanych z bazy za pomocą odpowiednich sterowników odpowiadających specyfikacji JDBC. Wszystkie modele zaimplementowane zostały w systemie ORACLE a komunikacja z bazą danych odbywa się przy wykorzystaniu sterowników tej firmy.

Za pomocą interfejsu DatabaseMetaData (pakiet java.sql) z bazy danych wyciągana jest informacja o jej strukturze. W pierwszym kroku pobierane są dane o tablicach należących do określonego schematu (odpowiadającego użytkownikowi). Po wybraniu tabel, z których ma zostać stworzona struktura, są one analizowane pod względem kolumn i relacji jakie zawierają.

Łączenie się z serwerem bazy danych odbywa się poprzez dialog logujący, którego zawartość jest pobierana z pliku konfiguracyjnego. Istotne dla prawidłowego funkcjonowania *zarządcy* jest zgodność użytkowników (schematów).

Poniżej został przedstawiony interfejs aplikacji.



Rys. 2 Interfejs aplikacji

2.2. Sposoby uruchamiania aplikacji na serwerze WWW

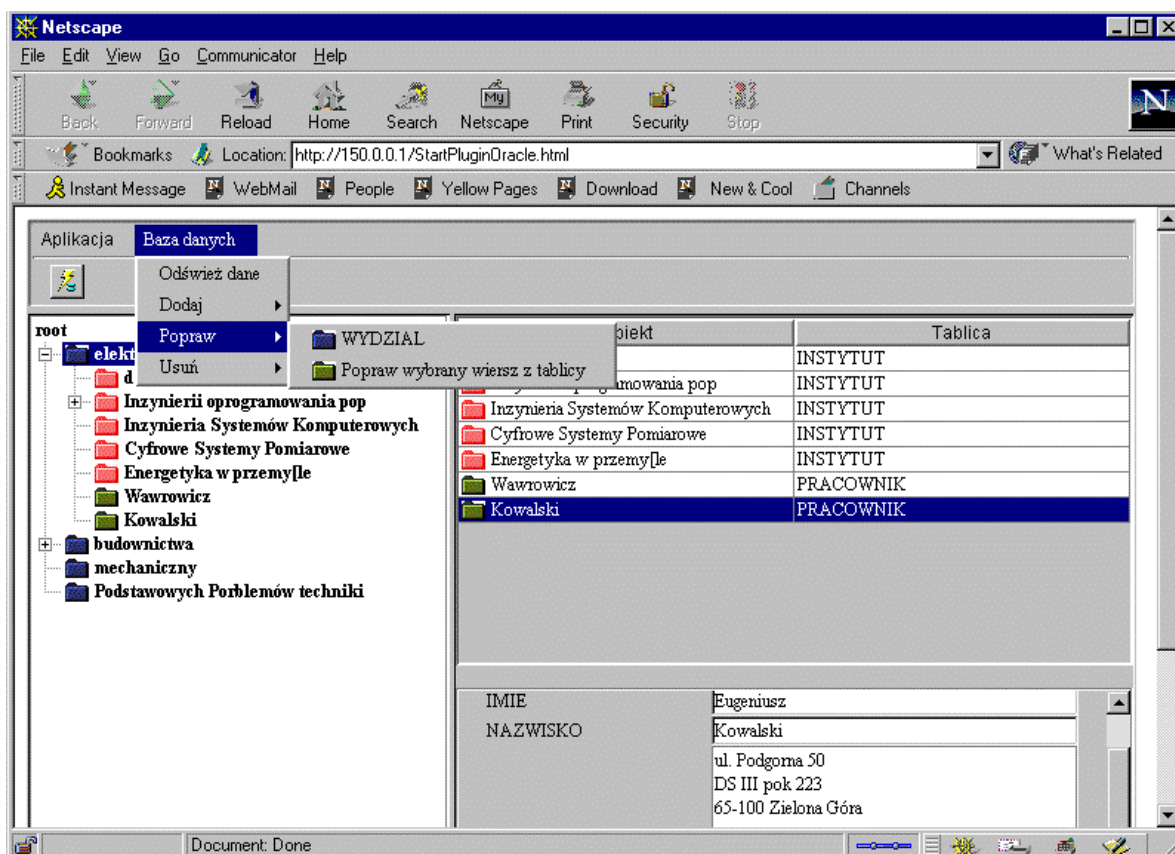
Aplikacja, uruchomiona jako aplet na serwerze WWW, nie potrzebuje żadnych zmian (nie trzeba jej będzie przekompilowywać, co mogłoby się okazać trudnym zadaniem dla zwykłego użytkownika). Wystarczy zmienić dane w pliku konfiguracyjnym (wpisując ścieżkę dostępu i nazwę do pliku z zapisanym w nim interfejsem użytkownika) lub po prostu podmienić plik z interfejsem użytkownika na nowy, wygenerowany przez *zarządcę aplikacji*.

Komunikacja między apletem a bazą danych odbywa się na tej samej zasadzie co podczas pracy w formie aplikacji. Jediną różnicą jest zastosowanie innego sterownika. Aplikacja wykorzystuje bazę Oracle, której producent dostarcza dwa rodzaje sterowników: jeden wykorzystywany przez aplikacje oraz drugi, zalecany dla apletów.

W przeglądarkach implementujących JDK 1.0.2 (np. Netscape 3.0) zabezpieczenia nie pozwalają nawiązać połączenie z innym komputerem. Oznacza to, że aplet może się połączyć tylko z bazą pracującą na tym samym komputerze. W przeglądarkach z maszyną wirtualną w wersji 1.1.1 (np. Netscape 4.0) możliwe jest poproszenie użytkownika o przydzielenie uprawnienia „socket connection”, i jeśli użytkownik przyzna je, aplet będzie mógł połączyć się z jakimkolwiek innym komputerem.

Działanie apletu nie różni się niczym od aplikacji omówionej w punkcie 2.1. Istnieje możliwość pracy w trybie autoryzacji dostępu (wyświetlany jest wówczas dialog logujący). Plik konfiguracyjny znajduje się na serwerze WWW w tym samym miejscu co klasy Javy i inne zasoby, z których aplet korzysta (np. ikony). Nie jest on wyświetlany jako niezależne okno, lecz cały zawiera się na stronie WWW.

Rysunek zamieszczony poniżej przedstawia wygląd okna przeglądarki z apletem.



Rys. 3 Wygląd okna przeglądarki Netscape z apletem

3. Podsumowanie

Główną ideą było napisanie aplikacji na tyle elastycznej by mogła wykorzystać każdą strukturę relacyjnej bazy danych.

Prezentowana struktura Naukowej Bazy Danych zawiera tablice gromadzące informację: o strukturze organizacji, zespołach badawczych, pracownikach naukowych, raportach, wydawnictwach, współpracy zagranicznej, stopniach naukowych oraz raportach.

W przyszłości można rozbudować interfejs użytkownika o nowe komponenty dostępne w *zarządcy aplikacji* np. formularze drukowania, zakładki, raporty itp.

Zaletą tej aplikacji jest jej możliwość prezentacji w ogólnodostępnej sieci Internet, co daje ośrodkom naukowym możliwość wymiany doświadczeń i informacji z innymi uczelniami i organizacjami na całym świecie.

Bibliografia

1. Beynon-Devis P., *Systemy baz danych*, Wydawnictwo Naukowo-Techniczne, Warszawa, 1998
2. Bielecki J., *Java po C++*, Interfotland Warszawa 1997,
3. Vanhelsuwe L., Philips I., Goang-Tay Hsu, Sankar K., Ries E., Rohaly T., Zukowski J., *Programujemy w Java*, SYBEX Inc., 1997,
4. Wrembel R., Wiczerzycki W., *Projektowanie aplikacji bazy danych ORACLE*, Wydawnictwo Nakom, Poznań, 1997