

Analiza porównawcza środowisk rozwoju aplikacji internetowych

Zbyszko Królikowski

Zbyszko.Krolikowski@cs.put.poznan.pl

Maciej Zakrzewicz

mzakrz@cs.put.poznan.pl

Politechnika Poznańska, Instytut Informatyki

Poznań, ul. Piotrowo 3a

Streszczenie

W ostatnich latach, World Wide Web stał się dominującą platformą dla budowy rozproszonych systemów aplikacyjnych, udostępniających zawartość baz danych w sieci Internet. Niekwestionowanymi zaletami WWW są: skalowalność i przenaszalność systemów, łatwość obsługi aplikacji przez użytkowników, minimalne nakłady administracyjne.

W artykule porównano cztery najpopularniejsze architektury budowy bazodanowych aplikacji internetowych: Oracle Application Server + PL/SQL Cartridge, Apache Web Server + CGI + Oracle Call Interface, Apache Web Server + Java + JDBC i Microsoft IDC + SQL Server. Przedstawiono własności konstrukcyjne każdego z rozwiązań oraz oceniono jakość, złożoność i koszt implementacji.

1 Wprowadzenie

Technologia World Wide Web, pierwotnie projektowana jako platforma sieciowego udostępniania statycznych dokumentów hipertekstowych, szybko awansowała do poziomu środowiska aplikacyjnego. Dzięki jej przenaszalności, skalowalności i uniwersalnemu dostępowi, użytkownicy mogą wykorzystywać przeglądarki WWW (Netscape Navigator, Microsoft Internet Explorer) do przesyłania żądań wykonania aplikacji przechowywanych po stronie serwerów WWW (Apache Web Server, Netscape, Microsoft Internet Information Server, Oracle Web Listener). Wyniki działania interaktywnych aplikacji internetowych są prezentowane przez graficzny interfejs przeglądarki WWW.

Już w początkach rozwoju „programowania internetowego” okazało się, że znaczna liczba zastosowań aplikacji udostępnianych przez serwery WWW to aplikacje o charakterze bazodanowym – pozwalające użytkownikom przeszukiwać i prezentować zawartości baz danych. Właśnie ten fakt spowodował narastanie ilości informacji dostępnej dziś w Internecie – oferty sprzedaży towarów, wiadomości finansowe, informacje o połączeniach kolejowych i lotniczych, katalogi biblioteczne, przeszukiwarki sieciowe – to przykłady bazodanowych aplikacji tworzonych dla Internetu.

Wzrost zainteresowania WWW jako platformą aplikacyjną spowodował oczywiste poszukiwania efektywnych środowisk programowania. Dziś, dzięki istniejącej ofercie technologicznej, programista i projektant mogą wybierać spośród wielu różnorodnych metod budowy oprogramowania, wykorzystującego zawartość bazy danych do dynamicznego generowania dokumentów HTML prezentowanych przez przeglądarkę użytkownika. Od dokonanego wyboru będzie jednak znacząco zależeć czas budowy, elastyczność i wydajność tworzonych systemów.

W artykule dokonano analizy porównawczej czterech, wybranych jako najpopularniejsze, środowisk rozwoju bazodanowych aplikacji internetowych (tabela 1). Trzy z nich wykorzystują system zarządzania bazą danych Oracle, a jeden – w celu odniesienia do pozostałej części rynku produktów WWW – Microsoft SQL Server. Wszystkie środowiska zostały krótko scharakteryzowane, a następnie porównane pod kątem najczęściej formułowanych kryteriów. W celu uzyskania prezentowanych tutaj wniosków, dokonano eksperymentalnej implementacji identycznego systemu internetowego (księgarnia internetowa) na każdej z platform.

nazwa środowiska	serwer WWW	interfejs z bazą danych	system operacyjny serwera WWW/bazy danych	punkt wykonania aplikacji
OAS + PL/SQL Cartridge	Oracle Web Listener	PL/SQL Cartridge	dowolny	serwer bazy danych
Apache + CGI + OCI	Apache	OCI	dowolny	serwer WWW
Apache + Java + JDBC	Apache	JDBC	dowolny	klient
IDC + SQL Server	Microsoft Internet Information Server	IDC	Windows NT	serwer WWW

Tabela 1. Ogólne własności porównywanych środowisk aplikacyjnych

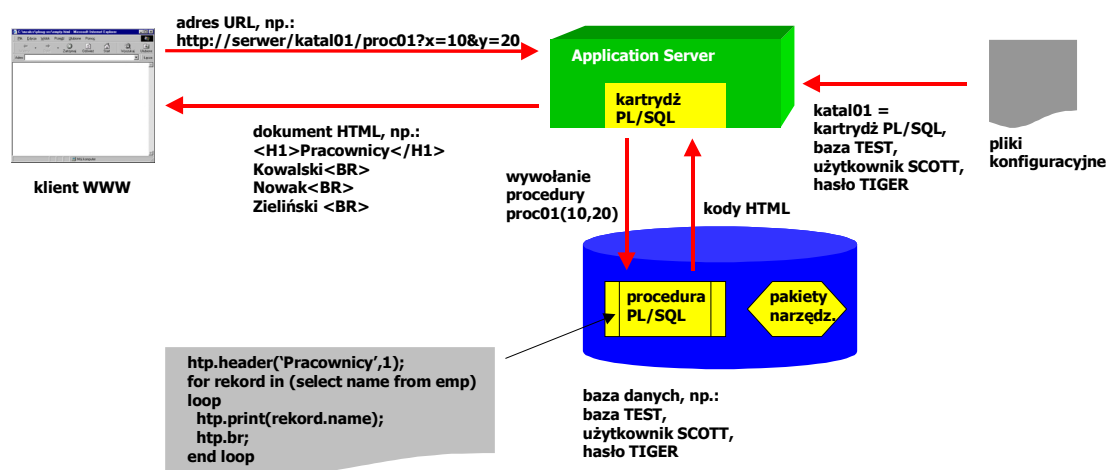
2 Charakterystyka środowisk rozwoju aplikacji internetowych

1.1 Oracle Application Server i PL/SQL Cartridge

Oracle Application Server (OAS), poprzednio nazywany Oracle Web Server, jest obecnie od kilku lat na rynku silną platformą udostępniania trójwarstwowych aplikacji internetowych, współpracujących z bazą danych Oracle. OAS realizuje funkcjonalność nowoczesnego serwera HTTP, a dodatkowo wyposażony jest w środowiska uruchomieniowe (kartrydże), pozwalające na wykonywanie niewielkich aplikacji, dynamicznie generujących dokumenty HTML, przesyłane do przeglądarki WWW użytkownika. Kartrydże OAS obsługują aplikacje tworzone w języku PL/SQL, Java, C/C++, Perl, standard LiveHTML (odpowiednik Server Side Includes), standard Oracle Worlds (automatyczne generowanie światów wirtualnych VRML). Programiście związanemu dotychczas z narzędziami Oracle, najbardziej atrakcyjny wyda się kartrydż PL/SQL.

Architekturę funkcjonowania oprogramowania internetowego opartego na kartrydżu PL/SQL i OAS przedstawiono na rysunku 1. Programista przygotowuje aplikację generującą dokument HTML jako procedurę składowaną, zapisaną w bazie danych. Procedura taka korzysta z pakietów narzędziowych OAS, które są instalowane w tej samej bazie danych. Pakiety te pozwalają m.in. na obsługę wejścia/wyjścia, wysyłanie kodów standardu HTML, formatowanie tekstów, budowę formularzy itd. W celu wykonania utworzonej procedury, użytkownik, korzystając z przeglądarki WWW, wysyła do OAS żądanie w postaci adresu URL, zawierającego adres serwera, nazwę katalogu wirtualnego, nazwę procedury i jej parametry wejściowe. Adres URL jest następnie

analizowany przez OAS i na podstawie skojarzenia nazwy katalogu wirtualnego z zapisami w plikach konfiguracyjnych nawiązywane jest połączenie z serwerem bazy danych. W bazie danych uruchamiana jest wówczas procedura o nazwie podanej przez użytkownika jako przedostatni człon adresu URL. Z adresu tego mogą zostać odczytane również wejściowe parametry jej wywołania. W oparciu o zawartość bazy danych, uruchomiona procedura generuje na standardowym wyjściu kody HTML, które są odbierane przez OAS a następnie przesyłane do przeglądarki WWW, która zainicjowała cały proces.

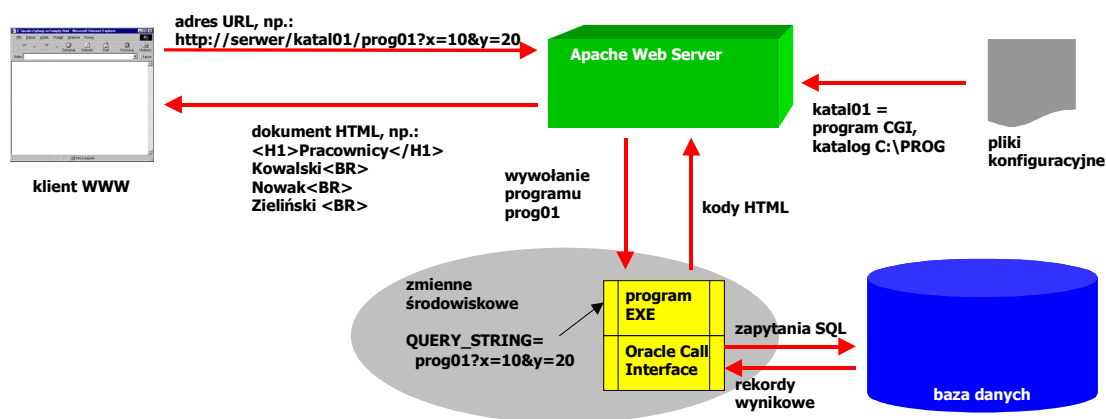


Rysunek 1. Architektura funkcjonowania Oracle Application Server i PL/SQL Cartridge

1.2 Apache Web Server i Oracle Call Interface

Praktycznie każdy popularny serwer WWW (jak np. Apache Web Server) dostarcza mechanizmów wołania zewnętrznych programów wykonywalnych, które mogą generować kod HTML przesyłany do przeglądarki WWW użytkownika. Najpowszechniejszym standardem komunikacji pomiędzy serwerem WWW a innym programem zewnętrznym jest Call Gateway Interface (CGI). Programy wołane poprzez CGI (często nazywane „programami CGI”) mogą być tworzone w dowolnych językach programowania: C/C++, Pascal, Perl, UNIX Shell, DOS Batch, itd. Jeżeli wymagany jest dostęp programu CGI do bazy danych Oracle, wtedy programista zmuszony jest do samodzielnego oprogramowania komunikacji z serwerem Oracle. Najwydajniejszym, chociaż najbardziej skomplikowanym interfejsem komunikacyjnym w środowisku serwera Oracle jest Oracle Call Interface (OCI).

Architekturę funkcjonowania oprogramowania opartego na CGI i OCI przedstawiono na rysunku 2. Programista przygotowuje program wykonywalny, który przy pomocy interfejsu OCI nawiązuje połączenie z bazą danych, wysyła do niej treść zapytania SQL, odbiera wyniki zapytania i na ich podstawie generuje kody HTML. W celu wykonania programu, użytkownik, korzystając z przeglądarki WWW, wysyła do serwera Apache żądanie w postaci adresu URL, zawierającego adres serwera, nazwę katalogu wirtualnego, nazwę programu i jego parametry wejściowe. Adres URL jest następnie analizowany przez Apache i na podstawie skojarzenia nazwy katalogu wirtualnego z zapisami w plikach konfiguracyjnych lokalizowany jest fizyczny katalog dyskowy, w którym znajduje się program o podanej nazwie. Przed uruchomieniem programu, w środowisku systemowym definiowany jest zbiór zmiennych CGI, pod które podstawiane są takie wartości, jak adres IP klienta, nazwa przeglądarki WWW klienta, wersja protokołu HTTP, itd. Jedną ze zmiennych jest QUERY_STRING, pod którą podstawiany jest tekst zawierający wszystkie parametry wywołania programu CGI, wyłonię z adresu URL. Program CGI odczytuje potrzebne zmienne środowiskowe, komunikuje się z bazą danych i na swoim standardowym wyjściu (stdout) generuje kody HTML. Kody te są odbierane przez serwer Apache i przesyłane do przeglądarki WWW użytkownika, który zainicjował proces.



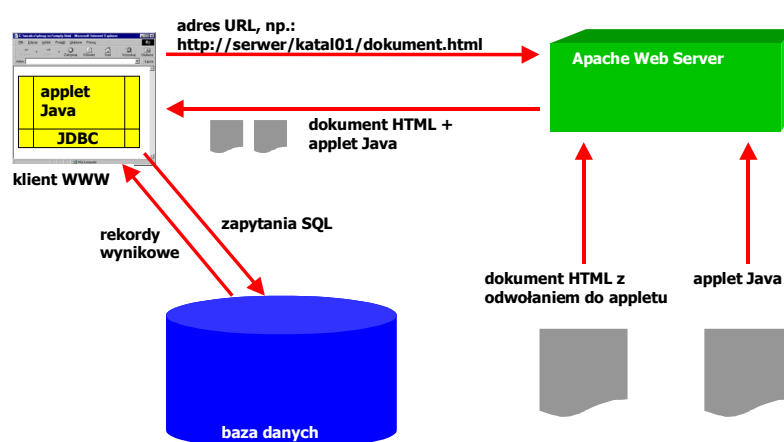
Rysunek 2. Architektura funkcjonowania Apache Web Server, CGI i OCI

1.3 Apache Web Server i JDBC

Od kilku lat systematycznie rośnie popularność nowego obiektowego języka programowania Java. Java umożliwia tworzenie interesującego typu aplikacji, nazywanych

appletami. Applet Java to niewielki program, znajdujący się w systemie plików serwera WWW, który na żądanie użytkownika jest przesyłany przez sieć i wykonywany wewnątrz przeglądarki WWW. Ważną zaletą appletów Java jest ich przenaszalność i niezależność od platformy – ten sam applet może być wykonywany zarówno przez przeglądarkę Netscape Navigator w systemie Solaris, jak i przez przeglądarkę Microsoft Internet Explorer w systemie Windows NT. Applety mogą się komunikować z serwerami baz danych – służy do tego celu uniwersalny standard i zarazem biblioteka Java Database Connectivity (JDBC).

Architekturę funkcjonowania oprogramowania opartego na appletach Java i JDBC przedstawiono na rysunku 3. Programista przygotowuje i kompiluje kod appletu, który będzie nawiązywać połączenie z bazą danych poprzez JDBC oraz obsługiwać interfejs użytkownika. Dodatkowo, tworzony jest dokument HTML pełniący rolę pojemnika, w którym osadzony jest applet. Po otrzymaniu żądania przesłanego przez przeglądarkę WWW użytkownika, serwer WWW przesyła bazowy dokument HTML, a następnie applet Java. Applet uruchamiany jest przez przeglądarkę WWW i komunikuje się z bazą danych w celu pobrania rekordów i ich zaprezentowania na ekranie.



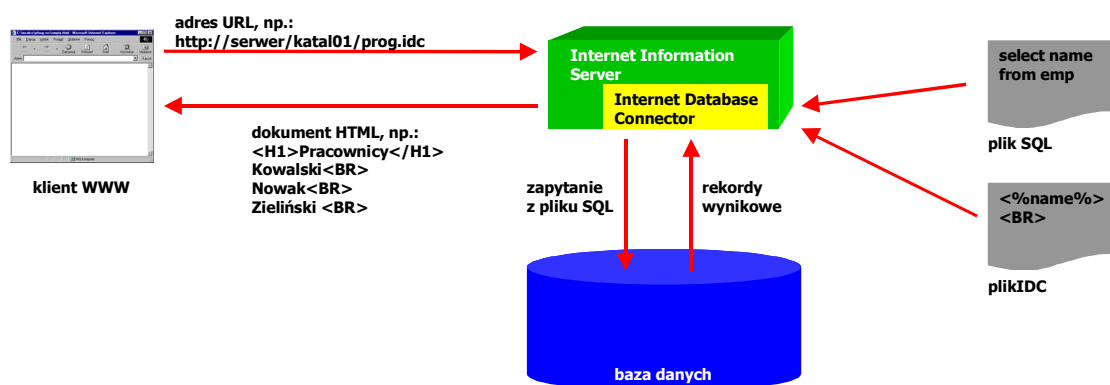
Rysunek 3. Architektura funkcjonowania appletów Java i JDBC

1.4 Microsoft Internet Database Connector i SQL Server

SQL Server jest flagowym bazodanowym produktem Microsoftu. Do publikowania jego zawartości w sieci Internet wykorzystywany jest Internet Database Connector (IDC), związany z Internet Information Serverem, wchodzącym z kolei w skład Windows NT Servera. Programowanie prostych aplikacji internetowych nie wymaga od programisty znacznych umiejętności – sprowadzają się one do poznania przez niego składni poleceń

języka SQL oraz składni poleceń IDC, służących do formatowania dokumentu HTML zawierającego wyniki zapytań.

Architekturę funkcjonowania oprogramowania opartego na IDC i SQL Server przedstawiono na rysunku 4. Programista przygotowuje dwa pliki tekstowe: plik o rozszerzeniu SQL, zawierający treść polecenia SQL pobierającego rekordy z bazy danych i plik o rozszerzeniu IDC, zawierający szkieletowy dokument HTML, do którego włączone zostaną rekordy wynikowe. Żądanie przesłane przez przeglądarkę WWW użytkownika odwołuje się do pliku IDC. Serwer WWW – Internet Information Server – odczytuje pliki IDC i SQL (o takiej samej nazwie, jak IDC), wykonuje zapytanie z pliku SQL, rekordy wynikowe włącza do szablonu HTML i ostateczny dokument przesyła do przeglądarki WWW użytkownika, który zainicjował cały proces.



Rysunek 4. Architektura funkcjonowania Internet Database Connector i SQL Server

3 Porównanie środowisk

3.1 Łatwość i szybkość budowy aplikacji

Pod hasłem łatwości budowy aplikacji rozumiano m.in. liczbę linii kodu, jakie musi przygotować programista dla uzyskania podobnego efektu wynikowego. Rozmiar kodu wpływa pośrednio na szybkość jego tworzenia i łatwość utrzymywania.

Zdecydowanie najmniejszy rozmiar posiada kod źródłowy aplikacji dla IDC/SQL Server – z praktycznego punktu widzenia, poza zapytaniem SQL nie istnieje tam nawet kod źródłowy. Podobnie niewielki rozmiar posiadają aplikacje OAS/PL/SQL Cartridge. Przeciętny dokument HTML może zostać wygenerowany przy użyciu kilku-kilkunastu wierszy kodu źródłowego w języku PL/SQL.

Dużo pracy musi włożyć programista w budowę programów wykorzystujących OCI bądź JDBC do samodzielnej komunikacji z bazą danych. Najmniejsze aplikacje wykonane w takich technologiach to dziesiątki wierszy kodu. Ponadto, OCI i JDBC nie są popularnie wykorzystywane w innych zastosowaniach, w związku z czym konieczne będzie podniesienie kwalifikacji programisty.

1.2 Elastyczność aplikacji

Ta kategoria porównawcza reprezentuje swobodę programisty w zakresie wykorzystywania różnorodnych technik i rozwiązań programistycznych. Im większa jest elastyczność oferowana przez środowisko, tym szerszy jest zakres zastosowań przygotowanych w nim aplikacji.

Pełną swobodę posiada programista w środowiskach CGI/Java – wynika ona z własności obiektowych języków programowania 3GL. Możliwe jest wykorzystywanie dostępnych na rynku bibliotek programowych, złożone i wydajne przetwarzanie obliczeniowe oraz dostęp do wszelkich zasobów systemowych.

Elastyczność PL/SQL Cartridge wyznaczona jest przez własności języka PL/SQL. Składnia PL/SQL przypomina Pascal/Ada, podobne są zatem możliwości konstrukcji programowych. Programowanie jest nieobiektywne, programista czuje się hermetycznie zamknięty wewnątrz bazy danych.

Natomiast IDC nie oferuje praktycznie żadnej elastyczności. Programista formułuje zapytanie SQL i otacza jego wyniki kodem HTML. Nie istnieją możliwości algorytmicznego przetwarzania danych i wyników zapytań. Jest to cena, jaką należy zapłacić za łatwość stosowania.

1.3 Koszt środowiska

W niewielkich zastosowaniach WWW istotna jest cena, jaką należy zapłacić za platformę, na której pracują aplikacje internetowe. Najdroższym z rozważanych rozwiązań jest niewątpliwie OAS/PL/SQL Cartridge – cena minimalnej instalacji sięga 20 tysięcy złotych. Prawie dwukrotnie tańszy jest wariant Microsoft IDC. Ceny minimalne – od 8 tysięcy złotych (wyłącznie koszt serwera Oracle – C i Java są zwykle dostępne bezpłatnie) oferowane są przez środowiska wykorzystujące JDBC i OCI.

1.4 Wymagania systemowe i sprzętowe

Wymagania definiowane przez producentów wykorzystywanych rozwiązań dotyczą głównie rozmiaru dostępnej pamięci operacyjnej serwera WWW. Dla uzyskania zadowalających wyników prowadzonych testów należało zapewnić:

- 96MB RAM dla OAS (64MB) i serwera bazy danych Oracle (32MB)
- 32MB RAM dla Microsoft IDC/SQL Server
- 32MB RAM dla Apache (pomijalne) i serwera bazy danych Oracle (32MB) (z CGI lub JDBC)

1.5 Wydajność

Wydajność pracy aplikacji internetowych wiąże się głównie ze sprawną obsługą dużej liczby równoczesnych, krótkich żądań. Wewnętrzna złożoność obliczeniowa aplikacji wydaje się być pomijalna, ze względu na prostotę przetwarzania (głównie obsługa we/wy).

Z tej kategorii porównawczej należy wyodrębnić aplikacje budowane w formie appletów Java, ponieważ, jako wykonywane po stronie klienta, nie generują one rozważanego obciążenia obliczeniowego serwera. Skalowalność takiego rozwiązania jest absolutna – teoretycznie każda liczba appletów Java może pracować równocześnie na nieograniczonej liczbie komputerów-klientów.

Natomiast pozostałe rozwiązania szeregują się według malejącej wydajności w następujący sposób:

1. OAS – wyposażony w zaawansowane mechanizmy buforowania, równoważenia obciążenia, przekierowywania połączeń, itd. jest środowiskiem najlepiej dostosowanym do obsługi bardzo dużych ilości połączeń.
2. Microsoft IDC – brak szczególnych rozwiązań ukierunkowanych na optymalizację obsługi żądań.
3. CGI/OCI – najmniej wydajna platforma, głównie ze względu na technologię CGI. Dla obsługi każdego żądania uruchomiony musi być oddzielny proces w systemie operacyjnym. Narzuty systemowe związane z każdorazowym powoływaniem do życia nowego procesu są ogromne w porównaniu z prostotą samego przetwarzania realizowanego przez programy wykonywane w ramach tych procesów.

4 Podsumowanie

Przeprowadzone porównanie nie dostarcza prostej odpowiedzi na pytanie: „Jaka jest najlepsza platforma rozwoju aplikacji internetowych?”. Podstawą do jej udzielenia powinna być dokładna specyfikacja wymagań, dotyczących złożoności i skalowalności budowanych aplikacji, a także ograniczeń o charakterze sprzętowo-finansowym. Niewątpliwie najbardziej zaawansowanym technicznie jest Oracle Application Server – bardzo dobra jest jego wydajność, na wyróżnienie zasługuje elastyczność i łatwość budowy aplikacji. Najtańszymi i postulującymi najmniejsze wymagania zasobowe (finansowe i sprzętowe) będą te, które korzystają z serwera Apache (bezpłatny) i mechanizmów CGI/Java. Z kolei jeżeli aspiracje przedsiębiorstwa nie są równoległe do kierunku rozwoju Internetu, a kilka prostych aplikacji powinno powstać minimalnym nakładem sił – wtedy zdecydowanie godne polecenia jest „user-friendly” Microsoft IDC.

5 Literatura

- [1] Analiza porównawcza środowisk rozwoju aplikacji internetowych, raport techniczny, Instytut Informatyki, Politechnika Poznańska, 1999
- [2] Exploring Java, P. Niemeyer, J. Peck, O'Reilly & Associates, Inc., 1996
- [3] Hypertext Transfer Protocol -- HTTP/1.0, T. Berners-Lee, R. Fielding, H. Frystyk, Internet Draft
- [4] Internet Standards and Protocols, D.C. Naik, Microsoft Press, 1998
- [5] Java, A. van Hoff, S. Shaio, O. Starbuck, Helion, 1996
- [6] Java™ 2 Platform: Technology for the Enterprise, L. Perlstein, <http://java.sun.com/>
- [7] Microsoft Internet Database Connector, dokumentacja techniczna
- [8] Oracle Application Server, dokumentacja techniczna
- [9] Oracle Call Interface, dokumentacja techniczna
- [10] The WWW Common Gateway Interface Version 1.1, D.R.T. Robinson, Internet Draft