ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol. 1381

Paweł KASPROWSKI
Politechnika Śląska, Instytut Informatyki

ZASTOSOWANIE MECHANIZMU JAVA SERVLETS DO UDOSTĘPNIANIA ZASOBÓW W SIECI INTERNET

Streszczenie. Artykuł omawia sposoby udostępniania zasobów w sieci, szczególny nacisk kładąc na zastosowanie mechanizmu servletów. Opisane zostały właściwości i budowa servletów oraz sposób ich współpracy z serwerem.

USING JAVA SERVLETS TO PRESENT RESOURCES IN INTERNET

Summary. This article decribes new possibilities of presenting information via Internet. A new Sun technology of servlets is described in details. There is information about using and creating servlets and accessing database servers from Internet in this paper.

1. Wstęp

Od momentu upowszechnienia sieci Internet podstawowym problemem było proste udostępnienie zasobów serwera pozostałym użytkownikom sieci. Proste usługi w rodzaju FTP lub TELNET szybko przestały wystarczać - ich używanie było niewygodne i nieefektywne. Tak więc zaistniała potrzeba stworzenia nowego rodzaju usługi umożliwiającej łatwą wymianę w pełni multimedialnej informacji pomiędzy użytkownikami Sieci. Tą usługą stała się World Wide Web, czyli WWW. Jej idea opiera się na prezentacji zasobów serwera na specjalnie przygotowanych i przesyłanych do użytkownika za pomocą protokołu HTTP tak zwanych stronach WWW.

Do tworzenia stron WWW służy język HTML. Za jego pomocą można dowolnie sformatować informacje, które chcemy przedstawić na stronie. Można tworzyć akapity, nagłówki, wstawiać obrazki, tabelki itp. Jednak strona WWW stworzona w języku HTML jest zupełnie

statyczna - jest to po prostu odpowiednio sformatowany multimedialny dokument. Za pomocą odsyłaczy można poruszać się po stronach znajdujących się na serwerze, jednak zawsze będą to gotowe i wcześniej zaprojektowane strony. W miarę rozwoju Internetu i zwiększania ilości dostępnej w nim informacji zaistniała potrzeba „ożywienia" stron WWW. W języku HTML stworzono do tego celu tak zwane formularze. Pozwalają one na umieszczenie na stronie typowych elementów formularza jak pole tekstowe, checkbox, radio button. Wprowadzone przez użytkownika dane są następnie wysyłane do serwera, a ten poprzez mechanizm CGI (Common Gateway Interface) wywołuje odpowiedni program, który na podstawie tych danych generuje nową stronę WWW. Program ten może być napisany w dowolnym języku programowania. Na standardowe wejście otrzymuje on listę danych wprowadzonych przez użytkownika, a na standardowe wyjście (przechwytywane przez serwer WWW) wysyła wygenerowaną przez siebie stronę WWW. Jak widać, dzięki tej metodzie strona WWW może być dostosowana dokładnie do potrzeb użytkownika. Jednak mimo że strony są teraz generowane dynamicznie, są one wciąż statyczne. Użytkownik nie może bezpośrednio wpłynąć na wygląd strony, z którą aktualnie pracuje, może jedynie zażądać nowej strony.

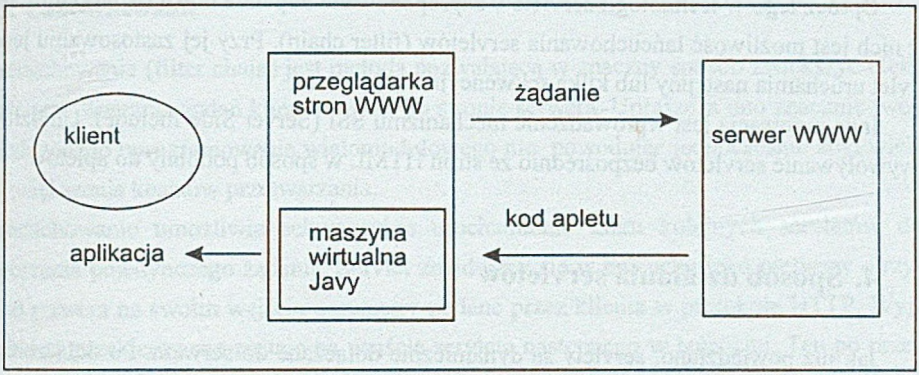
2. Język Java

Usługę WWW zrewolucjonizował język Java. Powstał on w firmie Sun Microsystems jako nowy zorientowany obiektowo i niezależny sprzętowo język programowania. Zaprojektowano go w grupie zajmującej się oprogramowaniem dla domowego sprzętu elektronicznego. W czasie tego rodzaju pracy czynnikiem kluczowym jest niezależność sprzętowa - nigdy przecież nie wiadomo jaki procesor zostanie użyty w urządzeniu. Właśnie konieczność zapewnienia niezależności sprzętowej spowodowała decyzję o zaprojektowaniu języka Java. Osiągnięto to przez stworzenie pojęcia tzw. byte-code (kodu pośredniego lub wirtualnego). Otóż kompilator Javy tworzy program w kodzie pośrednim. Kod ten może być wykonywany na tak zwanej „maszynie wirtualnej". Jest to aplikacja stanowiąca pewnego rodzaju interpreter, tłumacząca kod na język maszynowy danego procesora. Tak więc można powiedzieć, że Java stanowi coś pośredniego między językiem kompilowanym a interpretowanym.

Pełna niezależność sprzętowa i duża zwartość kodu zadecydowała o popularności Javy w świecie Internetu. Niezależność sprzętowa zagwarantowała, że program w Javie może być uruchomiony bez żadnych przeszkód na dowolnym komputerze połączonym z siecią. Z kolei zwartość kodu umożliwiła przesyłanie programów napisanych w Javie przez sieć w odpowiednio szybkim czasie.

Korzystając z Javy firma Sun stworzyła i wprowadziła na rynek technologię apletów - programów przesyłanych z serwera przez sieć i uruchamianych na komputerze klienta.

Umożliwiła ona tworzenie prawdziwie dynamicznych, reagujących na działania użytkownika stron WWW. Technologia ta spotkała się z bardzo przychylną reakcją - w chwili obecnej wszystkie liczące się przeglądarki potrafią już odczytywać i uruchamiać aplety na wbudowanych maszynach wirtualnych.



Rys. 1. Sposób wywołania apleta
Fig. 1. The way of aplet invoking

3. Technologia servletów

Po opanowaniu strony klienckiej WWW język Java rozpoczął podbój strony serwera. W czerwcu 1997 firma Sun wprowadziła na rynek Java Web Server v.1.0 - serwer stron WWW w całości stworzony w języku Java. Jedną z ważniejszych nowości w nim wprowadzonych była technologia servletów. Najprościej można powiedzieć, że servlety są dla serwera tym, czym aplety dla przeglądarki. Servlety są to obiekty w kodzie wirtualnym Javy, które mogą być dynamicznie dołączane do serwera za pomocą metody zbliżonej do plug-in Netscape'a. Mogą one - podobnie jak aplety - być dołączane bezpośrednio z sieci. Istotną różnicą w stosunku do apletów jest nieposiadanie przez servlety interfejsu graficznego. Jak każdy program w języku Java servlety są oczywiście w pełni niezależne sprzętowo. Są one ładowane dynamicznie i mogą dynamicznie rozszerzać możliwości serwera.

Najczęstszym zastosowaniem servletów jest generacja stron w HTML. Na tym polu wypierają one tradycyjne skrypty CGI. Dzieje się tak z co najmniej kilku powodów:

- servlety są szybsze i prostsze od skryptów CGI,
- zajmują mniej zasobów,
- mogą być ładowane i uruchamiane z dysku lokalnego jak i dostępne z sieci,
- używają standardowego API (servlet API).

Poza tym dają wszystkie zalety Javy:

- uruchamianie na różnych serwerach i w różnych systemach bez konieczności rekompilacji,
- dostęp do baz danych za pomocą wbudowanego w Javę mechanizmu JDBC,
- prostota i łatwość tworzenia zwartego kodu.

Oprócz tego w technologii servletów zaproponowano zupełnie nowe rozwiązania. Jednym z nich jest możliwość łańcuchowania servletów (filter chain). Przy jej zastosowaniu jeden servlet uruchamia następny lub kilka sekwencyjnie.

Inną innowacją jest wprowadzenie mechanizmu SSI (Server Side Include). Umożliwia on wywoływanie servletów bezpośrednio ze stron HTML w sposób podobny do apletów.

4. Sposób działania servletów

Jak już powiedziano, servlety są dynamicznie dołączane do serwera. Po dołączeniu traktowane są jak integralna jego część, co wydatnie wpływa na szybkość ich działania. Jeśli servlet nie będzie już używany, można go dynamicznie wyładować - odłączyć jego kod od kodu serwera.

Servlety mogą być ładowane na trzy sposoby:

- z katalogu, który jest opisany w zmiennej środowiskowej CLASSPATH,
- z katalogu /servlets/ - specjalnego katalogu zawierającego servlety,
- z sieci - przez podanie adresu bezwzględnego kodu servleta.

Servlety mogą być identyfikowane przez nazwę swej klasy, jeśli znajdują się w katalogu /servlets/ lub przez specjalną nazwę (rodzaj aliasu) nadawaną im przy ładowaniu do serwera.

Program uruchamiający servlety (który sam jest servletem predefiniowanym) wywołuje metodę service żądanego servleta. Jeśli servlet ten nie jest jeszcze załadowany - uruchamiacz najpierw go ładuje. Jeśli podano bezwzględny adres servleta, może on być załadowany skądkolwiek. Jeśli odwołanie nie jest bezwzględne, program szuka go na lokalnym dysku.

Servlety mogą być wywoływane na kilka sposobów. Najczęstsza sytuacja występuje, gdy klient prosi o dokument, który jest tworzony przez servlet. Serwer otrzymuje żądanie wysłania dokumentu, sprawdza parametry konfiguracyjne i stwierdza, że dokument nie jest statycznym plikiem znajdującym się na dysku, ale jest dynamicznie generowany. Wtedy wysyła żądanie do servleta który na podstawie danych wejściowych generuje odpowiednią stronę WWW. Jak łatwo zauważyć, metoda ta jest analogiczna z procedurą obsługi skryptów CGI.

Klient ma także możliwość wywołania servleta bezpośrednio przez podanie odpowiedniego adresu URL zawierającego ścieżkę do servleta lub podanie nazwy zdefiniowanego na serwerze aliasu.

Inną metodą uruchomienia servleta jest umieszczenie go w łańcuchu wywołań (filter cha-

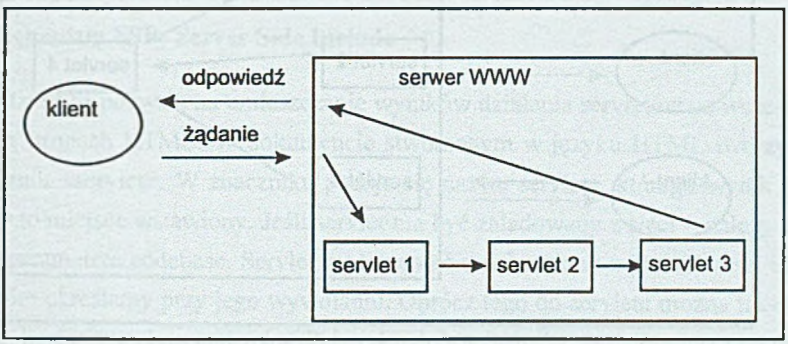
in). Jest on wtedy uruchamiany bezpośrednio po poprzednim w łańcuchu servlecie.

Servlet może zostać wywołany przez serwer, jeśli znajdzie on na którejś z wysyłanych klientowi stron WWW tak zwany SSI tag - czyli znacznik <servlet>. Należy pamiętać, że w ten sposób obsługiwane mogą być jedynie strony HTML z rozszerzeniem .shtml.

4.1. Łańcuchowanie

Łańcuchowanie (filter chain) jest metodą pozwalającą w znaczny sposób zwiększyć efektywność przetwarzania żądań klientów sieci po stronie serwera. Upraszcza ono znacznie tworzenie złożonego oprogramowania wielomodułowego nie powodując jednocześnie zbyt wielkiego zwiększenia kosztów przetwarzania.

Łańcuchowanie umożliwia sekwencyjne uruchamianie kilku kolejnych servletów do przetworzenia pojedynczego żądania. Servlet znajdujący się w łańcuchu jako pierwszy otrzymuje od serwera na swoim wejściu parametry podane przez klienta w protokole HTTP. Wyjście servleta przekierowane zostaje na wejście servleta następnego w łańcuchu. Ten po przetworzeniu danych generuje parametry dla kolejnego ogniwa. Ostatni w łańcuchu servlet na wyjściu tworzy stronę WWW zwracaną do serwera i następnie po dodaniu nagłówka wysyłaną jako odpowiedź do klienta.



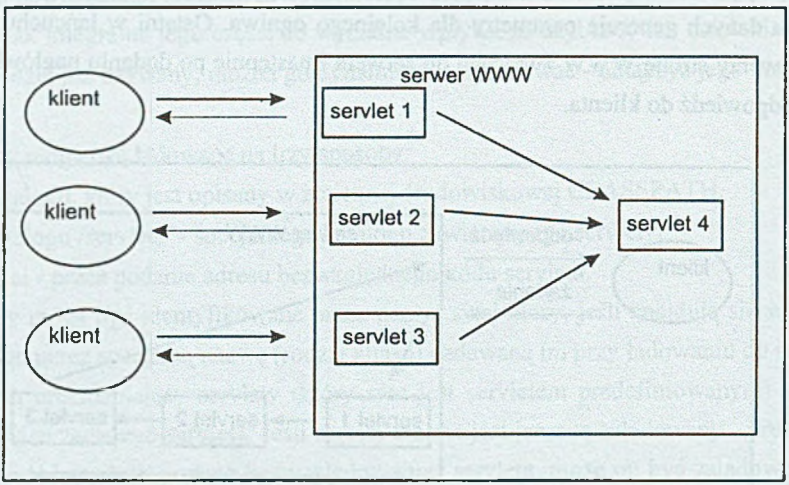
Rys. 2. Przykład łańcuchowania serwletów
Fig. 2. An example of servlets chaining

Jak łatwo zauważyć, metoda ta pozwala na dużą elastyczność przy obsłudze żądań klientów. Ułatwia ona także wymianę części oprogramowania na inne. Dla przykładu wyobraźmy sobie serwer udostępniający zawartość bazy danych. Do serwera tego ładujemy servlet tłumaczący zestaw parametrów wprowadzonych przez użytkownika na tekst zapytania w języku SQL. Drugi servlet na wejście otrzymuje gotowe zapytanie, kontaktuje się z bazą danych i zwraca wynik tego zapytania. Może w tym przypadku zaistnieć sytuacja, że kontaktu z bazą żądać mogą użytkownicy pracujący na różnych stronach WWW. Zestaw parametrów przesyłanych do serwera będzie oczywiście inny. W tym przypadku projektant musi tylko stworzyć

drugą wersję programu tłumaczącego. Dostęp do bazy danych będzie w obu przypadkach identyczny i będzie mógł być realizowany przez drugi servlet.

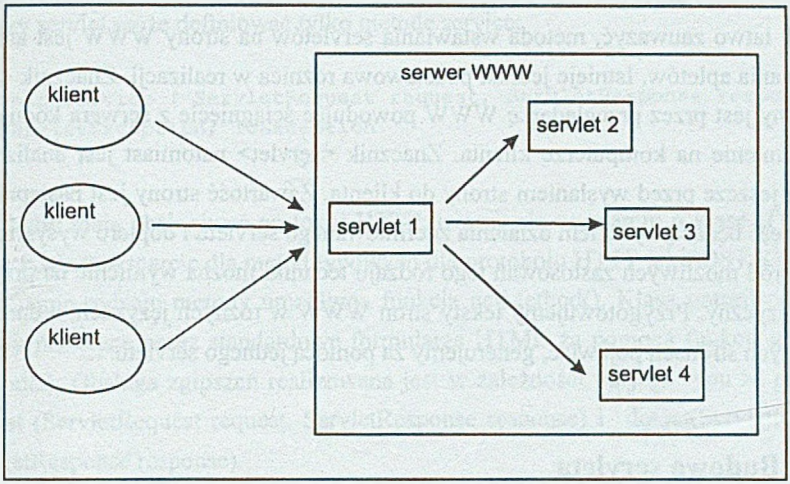
Łańcuchy servletów można zdefiniować odpowiednio konfigurując serwer WWW. Po podaniu odpowiednich łańcuchów serwer sam dba o sekwencyjne wywoływanie kolejnych servletów.

Innym bardzo użytecznym sposobem użycia łańcuchowania jest zdefiniowanie typu MIME, na który servlet ma reagować. Servlet tak załadowany czeka na zgłoszenie z pewnym określonym typem MIME i jeśli się ono pojawi, przejmuje jego obsługę. W podanym powyżej przykładzie można sobie wyobrazić ustawienie parametrów servleta obsługującego dostęp do bazy danych na obsługę wszystkich żądań typu SQL. Nie ma wtedy konieczności definiowania łańcuchów dla każdego rodzaju odwołania do bazy. Wystarczy, że servlety przetwarzające dane od klientów wygenerują na wyjściu żądanie typu SQL. Zostanie ono automatycznie przekazane odpowiedniemu servletowi.



Rys. 3. Konfiguracja serwera typu pierwszego
Fig. 3. First type of server configuration

Ciekawe mogłoby także być odwrócenie problemu. Otóż można tak skonfigurować serwer, że wszystkie odwołania klienta kierowane będą do jednego servleta. Ten po ich wstępnym rozpoznaniu zdecyduje, który servlet powinien je przetworzyć. Następnie wyśle je dalej zaopatrując w odpowiedni nagłówek MIME.



Rys. 4. Konfiguracja serwera typu drugiego
Fig. 4. Second type of server configuration

Metoda ta pozwala na ukrycie przed klientem rzeczywistej struktury serwera, a ponadto umożliwia prostą, bo wykonywaną tylko w jednym miejscu, kontrolę dostępu do zasobów serwera.

4.2. Mechanizm SSI - Server Side Include

Mechanizm SSI pozwala na umieszczanie wyników działania servletów na wcześniej zdefiniowanych stronach HTML. W dokumencie stworzonym w języku HTML tworzy się specjalny znacznik <servlet>. W znaczniku podaje się nazwę servleta, którego wynik działania może być w to miejsce wstawiony. Jeśli servlet ma być załadowany z sieci – należy podać jego adres w parametrze codebase. Servlet może zostać uruchomiony z parametrami inicjalizacyjnymi, które określamy przy jego wywołaniu. Oprócz tego do servleta można także przekazać dowolną liczbę innych parametrów. Pełna składnia znacznika <servlet> jest następująca:

```
<servlet code=ServletName.class codebase=http://server/
  initParam1=val1 initParam2=val2>
<param name=serviceParam1 value=val3>
<param name=serviceParam2 value=val4>
Tekst pokazujący się gdy serwer nie obsługuje servletów
</servlet>
```

Aby korzystać ze znacznika, konieczne jest nadanie dokumentowi w HTML'u go zawierającemu rozszerzenia .shtml. Tylko takie dokumenty są przez serwer przeszukiwane pod kątem uruchamiania na nich servletów.

Jak łatwo zauważyć, metoda wstawiania servletów na strony WWW jest analogiczna do wstawiania apletów. Istnieje jednak podstawowa różnica w realizacji. Znacznik <applet> analizowany jest przez przeglądarkę WWW powodując ściągnięcie z serwera kodu apletu i jego uruchomienie na komputerze klienta. Znacznik <servlet> natomiast jest analizowany przez serwer jeszcze przed wysłaniem strony do klienta. Zawartość strony jest następnie uzupełniana o część będącą wynikiem działania zdefiniowanego servletu i dopiero wysyłana do klienta.

Wśród możliwych zastosowań tego rodzaju techniki można wymienić na przykład serwer wielojęzyczny. Przygotowujemy teksty stron WWW w różnych językach, a dane, które mają się na tych stronach pojawiać, generujemy za pomocą jednego servletu.

5. Budowa servleta

Servlet jest programem napisanym w języku Java. Klasa bazowa tego programu musi implementować interfejs Servlet zwykle poprzez dziedziczenie klas GenericServlet lub HTTPServlet . Interfejs Servlet zawiera definicje funkcji wspólnych dla wszystkich servletów.

Pierwszą wywoływaną funkcją jest funkcja init(). Jest ona wywoływana przy ładowaniu servleta. Warto tu umieścić na przykład ustawianie parametrów servleta pobieranych z dysku – zajmuje to zwykle sporo czasu i dzięki umieszczeniu w metodzie init() nie będzie musiało być wykonywane przy każdym żądaniu uruchomienia. W momencie gdy servlet jest usuwany z serwera, uruchamiana jest metoda destroy().

Po inicjalizacji każdy klient zgłaszający się do servleta powoduje wystartowanie metody service(). Kolejne zgłoszenia są obsługiwane równoległe – może więc istnieć jednocześnie wiele instancji danego servleta. Każda instancja może pracować z innymi parametrami inicjalizacyjnymi i przechowywać swoje dane w wydzielonym miejscu pamięci. Można także za pomocą zmiennych statycznych współdzielić dane pomiędzy kolejnymi zgłoszeniami. Użytkownicy korzystający z jednego serwera mogą w ten sposób współdzielić dane i współpracować ze sobą. Umożliwia to na przykład prowadzenie telekonferencji.

Metoda service posiada dwa parametry – odnośnik wejścia i wyjścia:

```
public void service ( ServletRequest request, ServletResponse response )
```

Za pomocą metod obiektu typu ServletRequest można pobierać dane z serwera. Wynik przetwarzania wysyłany jest z powrotem do serwera poprzez obiekt typu ServletResponse.

Servlety zwykle pobierają parametry z wejścia w postaci obiektu typu ServletInputStream za pomocą metody getInputSteam():

```
ServletInputStream  in = request.getInputStream ();
```

i wysyłają na wyjście za pomocą metody getOutputStream():

```
ServletOutputStream out = response.getOutputStream ();
```

Najprostszy servlet może definiować tylko metodę service:

```
public void service ( ServletRequest request, ServletResponse response )
throws ServletException, IOException
{...}
```

Servlety, które mają obsługiwać protokół HTTP, tworzy się w oparciu o klasę HTTPServlet. Klasa ta posiada wsparcie dla metod typowych dla protokołu HTTP jak POST, GET czy HEAD. Odczytanie rodzaju metody umożliwia funkcja getMethod(). Klasa potrafi odczytywać parametry przesłane przez standardowe formularze HTML za pomocą funkcji getParameter(nazwapola). Obsługa zgłoszeń realizowana jest w zależności od jego typu za pomoca funkcji doPost (ServletRequest request, ServletResponse response) i doGet(ServletRequest request, ServletResponse response).

6. Współpraca z bazą danych

Język Java udostępnia możliwości komunikacji z dowolnym serwerem bazy danych. Dokonuje się tego za pomocą mechanizmu JDBC. JDBC jest interfejsem programowym stworzonym w języku Java służącym do wykonywania komend w języku SQL. Program napisany z użyciem mechanizmów JDBC może działać z dowolnym serwerem bazy danych przyjmującym zapytania w języku SQL (a więc z praktycznie każdym serwerem). JDBC udostępnia drivery, za pomocą których można połączyć się z bazą, wykonać na niej zapytanie i otrzymać wyniki.

Z uwagi na ogromną popularność standardu ODBC firmy Microsoft w bibliotece driverów JDBC znajduje się również driver JDBC-ODBC Bridge umożliwiający połączenie z bazą za pomocą narzędzi ODBC.

Wszystko to odbywa się za pomocą odpowiednich predefiniowanych klas. Połączenie realizowane jest za pomocą metody getConnection(url,user,password) obiektu typu DriverManager. Połączenie przechowywane jest w obiekcie typu Connection. Obiekt ten zawiera metodę createStatement(), która tworzy obiekt typu Statement, na którym można wykonywać zapytania metodą executeQuery(tekst zapytania). Rezultat zapytania zwracany jest w postaci obiektu typu RecordSet. Z obiektu tego oprócz czystych danych możemy odczytać informacje o ilości kolumn, nazwach kolumn itp.

Poniższy przykład prezentuje najprostszy możliwy seans z połączenia i wykonania zapytania na bazie danych:

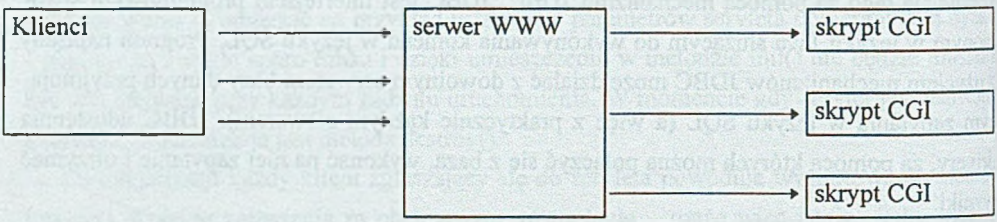
```
Connection con =
DriverManager.getConnection("jdbc:odbc:baza", "użytkownik", "hasło");
Statement stmt = con.createStatement ("SELECT * FROM tablica");
```

```
ResultSet rs = stmt.executeQuery ();
rs.next();
// pobiera zawartość pola nazwa z 1-go rekordu wyniku zapytania
String s = rs.getString("nazwa");
System.out.println(s);
rs.close();
stmt.close();
con.close();
```

Z powyższego przykładu widać, że połączenie z bazą jest bardzo proste i pozwala na bezproblemową zmianę serwerów bazy danych bez potrzeby rekompilacji programu.

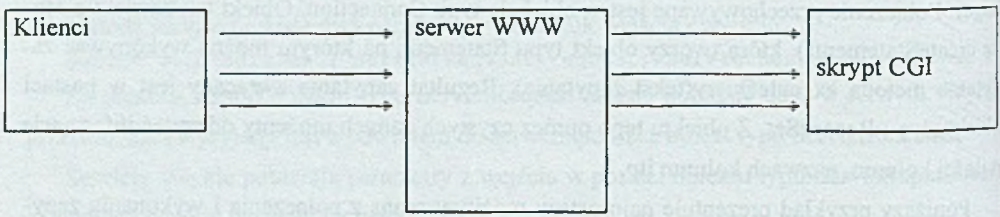
7. Porównanie servletów i skryptów CGI

Jak już stwierdzono, servlety stały się alternatywą dla skryptów CGI. Ich przewaga polega na lepszej integracji z serwerem, a co za tym idzie mniejszym zapotrzebowaniu na zasoby. Dla standardowego połączenia CGI serwer uruchamia aplikację obsługującą dane połączenie tyle razy, ile jest jednocześnie zgłoszeń.



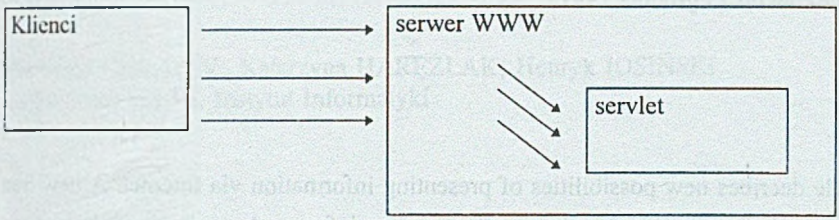
Rys. 5. Sposób działania CGI
Fig. 5. The way of CGI functioning

Nawet przy metodzie Fast CGI, mimo że dany skrypt jest uruchamiany tylko raz – czas tracony jest na jego załadowanie do pamięci.



Rys. 6. Sposób działania Fast CGI
Fig. 6. The way of Fast CGI functioning

Servlety po załadowaniu stają się jakby częścią serwera i dzięki temu nie ma dodatkowych nakładów czasowych na ich uruchomienie.



Rys. 7. Sposób działania servletów
Fig. 7. The way of servlets functioning

Ważną przewagą servletów nad skryptami CGI jest także ich niezależność sprzętowa. Servlet stworzony na komputerze Sun może być w każdej chwili przeniesiony na serwer Windows NT i będzie na nim działał w ten sam sposób.

8. Podsumowanie

Servlety zaprezentowane zostały jako element Java Web Server – serwera WWW napisanego w całości w języku Java. Jednak używanie ich nie jest uzależnione od posiadania tego serwera. Sun udostępnił rozszerzenia dla najpopularniejszych obecnie serwerów WWW umożliwiające używanie na nich servletów w sposób analogiczny do podanego powyżej. Rozszerzenia do serwerów Netscape Enterprise, Apache i Microsoft Internet Information Server są dostępne na serwerze Suna. Jest na nim także dostępna biblioteka do tworzenia servletów, tak zwana JSDK. W skład pakietu JSDK wchodzi także program srun.exe, który umożliwia testowanie servletów bez udziału serwera WWW.

LITERATURA

[1] http://www.java.sun.com.
[2] Jamsa K., Lalani S.: Programowanie WWW. MIKOM, Warszawa 1997.
[3] Hoff A., Shaio S., Starbuck O.: Java. Helion, Gliwice 1996.
[4] Jamsa K.: Java. MIKOM, Warszawa 1996.
[5] Walter S.J., Weiss A.: Język JavaScript. Intersoftland, Warszawa 1996.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 23 grudnia 1997 r.

Abstract

This article decribes new possibilities of presenting information via Internet. A new Sun technology of servlets is described in details. First section informs about previous ways of publishing information including WWW pages and CGI scripts. Next remarks Java language and explains its role in Internet. Fig. 1 shows the way of invoking aplets. In following sections servlets are discussed. There are methods of accessing servlets from Internet and a structure of a servlet decribed including servlet chaining (Fig. 2) and different ways of using it (Fig. 3 and Fig.4). In next section there is a brief example of using JDBC – Java access to databases. The last is a comparison between Java Servlets and CGI Scripts technologies. Fig. 5 shows the way of CGI is working. In Fig. 6 the way of Fast CGI technology is showed and the last Fig. 7 shows full advantage of Java Servlets.