

## Oracle8i a Internet

Maciej Zakrzewicz

mzakrz@cs.put.poznan.pl

Politechnika Poznańska, Instytut Informatyki

Poznań, ul. Piotrowo 3a

### **Abstrakt**

Najnowsza wersja systemu zarządzania bazą danych Oracle - Oracle8i - stanowi kolejny krok w kierunku integracji baz danych z internetowymi usługami sieciowymi. W artykule scharakteryzowano własności serwera Oracle8i, bezpośrednio związane z obsługą sieci Internet: WebDB, Internet File System, XML, itp. Ponadto, przedstawiono propozycje zastosowań nowych technologii w globalizacji baz danych przedsiębiorstwa.

## 1 Wstęp

Ostatnie lata, a nawet miesiące rozwoju systemów baz danych to stopniowa ewolucja w kierunku internetowej globalizacji. Po wyczerpaniu się rynku systemów informatycznych przedsiębiorstw nadszedł czas na integrację heterogenicznych baz danych, mającą na celu umożliwienie ich otwartego współdziałania w ogólnosiwiatowej sieci Internet. Oracle8i to kolejny fakt potwierdzający słuszność takiej linii rozwojowej.

Oracle 8i, czyli wersja 8.1 systemu zarządzania bazą danych, został rozbudowany o szereg nowoczesnych technologii internetowych. Podstawowe kierunki „usieciowienia” serwera Oracle8i to jego integracja z usługami WWW oraz wsparcie dla języka XML - następcy HTML. W artykule dokonano krótkiej charakterystyki technologii wbudowanych oraz dobudowanych do najnowszego serwera Oracle.

## 2 Wprowadzenie do technologii WWW

World Wide Web (WWW) rozpoczął podbój Internetu wraz z nadejściem lat dziewięćdziesiątych. Początkowo zaprojektowany jako platforma upowszechniania statycznych dokumentów hipertekstowych, dość szybko stał się środowiskiem aplikacyjnym. WWW pracuje w architekturze klient-serwer, klient-przeglądarka WWW (Netscape Navigator, Microsoft Internet Explorer) wysyła żądania zapisane w postaci adresów URL (Uniform Resource Locator, np. `http://www.oracle.com/index.html`), a serwer WWW odpowiada na te żądania przesyłając treść dokumentu w formacie HTML (Hypertext Markup Language). Format HTML pozwala w prosty sposób definiować układ graficzny przesyłanych dokumentów. Protokołem komunikacyjnym jest HTTP (Hypertext Transfer Protocol). Dokumenty mogą pochodzić ze statycznych plików, bądź być dynamicznie generowane przez zewnętrzne programy wykonywane przez serwer WWW.

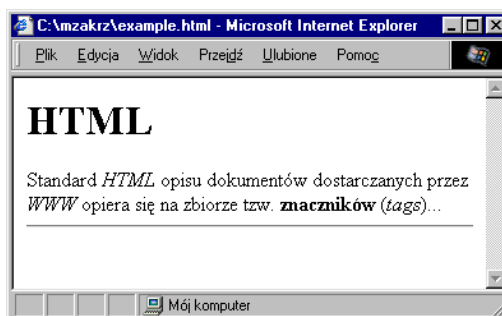
Standard HTML opisu dokumentów dostarczanych przez WWW opiera się na zbiorze tzw. znaczników (tags), opisujących sposób graficznej prezentacji informacji. Znaczniki są zapisywane w postaci kodów sterujących, ujętych w ostre nawiasy. Poniżej przedstawiono przykładowy zapis w formacie HTML oraz jego graficzną reprezentację.

```
<H1>HTML</H1>
```

Standard *HTML* opisu dokumentów dostarczanych przez *WWW* opiera się na zbiorze tzw. **znaczników** (*tags*)...

```
<BR>
```

```
<HR>
```



Pomimo, iż HTML pozwalał budować bardzo skomplikowane układy graficzne dokumentów, to jednak często był krytykowany za brak semantyki – opisu znaczenia danych, które formatował. W wyniku tego na szczyty popularności wspinają się dziś jego bezpośredni następcy: XML (Extensible Markup Language) i XSL (Extensible Stylesheet Language). Standard XML pozwala definiować strukturę danych przy użyciu rozszerzalnego zbioru znaczników, zapisywanych identycznie, jak w HTML. Standard XSL służy do niezależnego opisanie formy graficznej prezentacji danych, zdefiniowanych przez XML. Poniżej przedstawiono przykładowy zapis w formacie HTML.

```
<?xml version="1.0"?>
<Kurs>
  <Tytuł>Oracle8i a Internet</Tytuł>
  <Organizacja>Politechnika Poznańska</Organizacja>
  <Instruktor>
    <Osoba>Maciej Zakrzewicz</Osoba>
  <Słuchacz>
    <Osoba>Jan Kowalski</Osoba>
  </Słuchacz>
  <Słuchacz>
    <Osoba>Adam Nowak</Osoba>
  </Słuchacz>
</Kurs>
```

Błyskawicznie rosnąca popularność formatu XML wynika z łatwości jego zautomatyzowanej analizy. To właśnie XML staje się sieciowym sposobem komunikacji niekompatybilnych systemów komputerowych.

### 3 Własności internetowe Oracle8i

#### 3.1 Pakiet UTL\_HTTP

Serwer Oracle8i jest standardowo wyposażony w nowy pakiet systemowy PL/SQL o nazwie UTL\_HTTP. Pakiet ten oferuje funkcje, pozwalające użyć serwera bazy danych jako klienta WWW innego serwera WWW. Podstawowa funkcja pakietu,

UTL\_HTTP.REQUEST(url VARCHAR2) RETURN VARCHAR2 samodzielnie nawiązuje połączenie z serwerem WWW, pobiera dokument wskazywany przez URL i zwraca jego treść. Poniżej przedstawiono prosty przykład użycia tej funkcji dla pobrania i wyświetlenia źródła dokumentu HTML z serwera www.oracle.com.

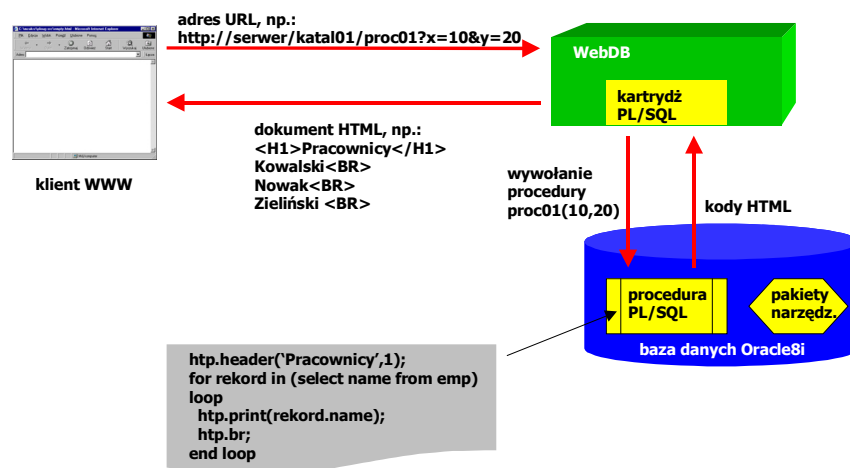
```
select utl_http.request('http://www.oracle.com') from dual;
```

Pakiet UTL\_HTTP umożliwia budowę aplikacji, które samodzielnie pobierają dane ze zdalnych serwisów informacyjnych.

### **3.2 WebDB**

Oracle WebDB jest modułem, przeznaczonym do tworzenia i uruchamiania aplikacji udostępniających zawartość bazy danych Oracle w sieci Internet. Funkcjonalnie kojarzy się on z uproszczeniem filozofii znanych dotychczas narzędzi, takich jak Designer/2000, Developer/2000, Enterprise Manager i Application Server. Twórcy oprogramowania mogą wykorzystywać WebDB do zarówno automatycznego (wizards), jak i programowego (w języku PL/SQL) konstruowania internetowych formularzy, raportów, wykresów, itp. Ponadto, WebDB jest w pełni funkcjonalnym serwerem WWW, udostępniającym, poza aplikacjami, statyczne obiekty przechowywane bądź w systemie plików, bądź wewnątrz bazy danych Oracle.

Ogólną architekturę WebDB przedstawiono na rysunku 1. Programista przygotowuje składowane w bazie danych procedury w języku PL/SQL, które generują kody dokumentów HTML w oparciu o zawartość bazy danych. Kiedy przeglądarka użytkownika wysyła żądanie URL do WebDB, wtedy WebDB uruchamia wymienioną w URL procedurę przy pośrednictwie znanego z poprzednich produktów Oracle kartrydża PL/SQL. Wyniki generowane przez wywołaną procedurę są przesyłane do przeglądarki WWW użytkownika i prezentowane w taki sam sposób, jak każdy inny dokument HTML.



**Rysunek 1.** Ogólna architektura funkcjonowania WebDB

Dla ułatwienia pracy programisty, wraz z WebDB dostarczane są pakiety bibliotecznych procedur PL/SQL, nazywane Web Developer's Toolkit. Programista może wykorzystywać procedury i funkcje tych pakietów, aby uniezależnić się od zmian standardów języka HTML, jak też po to, by realizować zaawansowane operacje związane z technologią WWW. W celu zilustrowania własności pakietów bibliotecznych rozważmy poniższy przykład. Załóżmy, że zadanie programisty polega na wygenerowaniu kodu HTML, prezentującego stronę domową firmy. Kod ten mógłby wyglądać następująco:

```
<H1>Oracle Corporation</H1>
<BR>
<B> Witamy w firmie Oracle <B>
<A HREF=http://www.oracle.com.pl> Oracle Polska </A>
```

Kod PL/SQL generujący taki dokument posiada następującą treść. Pakiet HTP jest jednym z omawianych pakietów bibliotecznych.

```
begin
  http.header('Oracle Corporation',1);
  http.br;
  http.bold('Witamy w firmie Oracle');
  http.anchor('http://www.oracle.com.pl','Oracle Polska');
end;
```

Zatem w pewnym sensie, od programisty nie wymaga się znajomości języka HTML – wystarczająca będzie umiejętność programowania w języku PL/SQL oraz znajomość funkcji i procedur bibliotek Web Developer's Toolkit.

WebDB oferuje również inny sposób tworzenia aplikacji internetowych. Korzystając ze zbioru dostarczanych z WebDB aplikacji-wizardów, użytkownik nie znający języka PL/SQL może w łatwy sposób budować procedury PL/SQL generujące internetowe formularze, raporty i wykresy. Użytkownik wprowadza w kolejnych oknach dialogowych potrzebne informacje dotyczące układu graficznego, struktury i sposobu wykorzystania danych przez aplikację, a następnie WebDB automatycznie tworzy kod PL/SQL w pełni funkcjonalnej aplikacji. Ponadto, WebDB zawiera zbiór graficznych aplikacji ułatwiających administrowanie zawartością bazy danych – możliwe jest przeglądanie i modyfikacja struktury tabel, definicji użytkowników, itd. Należy zaznaczyć, że wszystkie aplikacje, jakie są dostarczane wraz z WebDB są aplikacjami WWW, w związku z czym ich obsługa realizowana jest poprzez dowolną przeglądarkę WWW.

### 3.3 Parser XML

Oracle XML Parser jest modulem bibliotecznym, służącym do programowego dostępu do struktury i zawartości dokumentów XML. API parsera jest aktualnie dostępne dla języków PL/SQL, Java i C/C++. Wykorzystując Oracle XML Parser, programista może odczytać dokument XML dostępny w formie pliku, adresu URL, zmiennej lub zapisany w bazie danych, a następnie wygenerować dla niego drzewo analizy, dostępne poprzez interfejs DOM (Document Object Model) lub SAX (Simple API for XML). Funkcje parsera XML w wersji PL/SQL zostały zamieszczone w tabeli 1.

<b>funkcja/procedura</b>	<b>opis</b>
<code>parse(url VARCHAR2)</code> RETURN DOMDocument	Wykonuje analizę dokumentu XML znajdującego się w pliku o podanej nazwie lub pod podanym adresem URL i zwraca dokument DOM
<code>newParser</code> RETURN Parser	Zwraca nową instancję parsera (potrzebna do zmiany zachowania parsera lub dostępu do innych funkcji analizy)
<code>parse(p Parser, url VARCHAR2)</code>	Wykonuje analizę dokumentu XML dla wskazanej instancji parsera; dokument znajduje się w pliku o podanej nazwie lub pod podanym adresem URL
<code>parseBuffer(p Parser, doc VARCHAR2)</code>	Wykonuje analizę dokumentu XML dla wskazanej instancji parsera; dokument znajduje się w zmiennej/kolumnie VARCHAR2
<code>parseClob(p Parser, doc CLOB)</code>	Wykonuje analizę dokumentu XML dla wskazanej instancji parsera; dokument znajduje się w zmiennej/kolumnie CLOB
<code>setBaseDir(p Parser, dir VARCHAR2)</code>	Ustawia katalog bazowy, wykorzystywany do rozwiązywania ścieżek względnych URL

setErrorLog( p Parser, fileName VARCHAR2)	Ustawia plik, do którego kierowane będą komunikaty o błędach analizy
getDocument( p Parser) RETURN DOMDocument	Pobiera dokument DOM wygenerowany przez wskazaną instancję parsera

**Tabela 1.** Wybrane funkcje parsera XML

### 3.4 Obsługa XML w InterMedia ConText

Dokumenty XML lub ich fragmenty mogą być zapisywane w bazie danych w kolumnach typu CLOB. Interesujące operacje przeszukiwania takich kolumn oferuje moduł ConText, wchodzący aktualnie w skład InterMedia. Programista może wykorzystać strukturę dokumentu XML do zgrabnego zawężenia przeszukiwania SQL wyłącznie do wskazanych sekcji tekstu. W celu zilustrowania tej własności, założmy, że w tabeli INFORMACJE, w kolumnie DATA zapisano datę otrzymania, a w kolumnie TEXT fragmenty wiadomości prasowych w formacie XML, jak poniżej:

```
<informacja>
Jak oświadczył komendant <nazwisko> Kowalski </nazwisko>,
<zdarzenie> pożar </zdarzenie> w miejscowości <miejsce>
Kotowo </miejsce> pod Warszawą był spowodowany przez
<przyczyna> usterkę instalacji elektrycznej </przyczyna>.
</informacja>
```

```
<informacja>
W miejscowości <miejsce> Strzyżewo </miejsce> miał miejsce
<zdarzenie> napad rabunkowy </zdarzenie> na sklep spożywczy.
Sprawców ujęła policja.
</informacja>
```

W odniesieniu do powyższej struktury, moduł ConText pozwala na łatwe formułowanie zapytań typu: „kiedy pojawiła się ostatnia wiadomość o pożarze w miejscowości Kotowo?”. Poniżej zamieszczono przykład zapisu takiego zapytania w języku SQL, rozszerzonym o klauzulę contains, pochodzącą z ConText.

```
SELECT MAX(data)
FROM informacje
WHERE CONTAINS(text, 'pożar WITHIN zdarzenie')
AND CONTAINS(text, 'Kotowo WITHIN miejsce')
```

### 3.5 Internet File System

Oracle iFS jest nowatorską technologią, dzięki której baza danych Oracle8i może być traktowana jak dysk sieciowy o strukturze systemu plików. Zbudowany na platformie Java, iFS umożliwia tworzenie drzew katalogów i składowanie w nich tradycyjnych plików. Dostęp do plików odbywa się za pomocą dowolnego z szerokiej gamy protokołów internetowych, takich jak HTTP, FTP, SMTP, POP, IMAP, SMB.

Pewne dodatkowe właściwości iFS są związane z przetwarzaniem plików zawierających dokumenty XML. Dokumenty takie mogą być automatycznie analizowane w chwili ich umieszczania w iFS i zapisywane w bazie danych w postaci relacyjnej. Przyszłe odczyty wymagają wówczas automatycznej syntezy dokumentu oryginalnego przez specjalizowany moduł, XML Renderer. Dzięki temu, baza danych Oracle8i staje się „przezroczysta” dla standardu XML. Oracle iFS występuje w serwerach Oracle8i począwszy od wersji 8.1.6.

### 3.6 Servlet XSQL

Oracle XSQL Servlet jest servletem Java, pozwalającym na automatyczne generowanie dokumentów XML, opartych na wyniku jednego lub wielu zapytań SQL. Technologicznie, jest to aplikacja wykorzystująca Oracle XML SQL Utilities for Java, rozpowszechniana wraz z wersją 8.1.6 serwera Oracle8i oraz wersją 4.0.8 Oracle Application Servera, obsługiwana praktycznie przez każdy serwer WWW obsługujący servlety Java. Osoby wykorzystujące wcześniej narzędzia Microsoft Internet Database Connector dostrzegą tu liczne analogie zastosowanego przez Oracle podejścia.

W celu użycia servletu XSQL, programista umieszcza znacznik `<query>` wewnątrz pliku XML, w miejscu, w którym powinien zostać włączony wynik zapytania. Poniżej przedstawiono przykład prostego dokumentu XML, korzystającego z servletu XSQL. Atrybut `connection` znacznika `<query>` służy tu do wskazania połączenia z konkretną bazą danych - połączenia takie muszą zostać uprzednio wyspecyfikowane w specjalnym pliku konfiguracyjnym o nazwie `XSQLConnection.xml`. Pomiędzy znacznikiem `<query>` a symetrycznym znacznikiem zamykającym `</query>` zapisywana jest treść zapytania SQL.

```
<?xml version="1.0"?>
<query connection="kadry">
  SELECT nazwisko, stanowisko FROM pracownicy
</query>
```



Zakładając, że tabela PRACOWNICY zawiera trzy rekordy opisujące nazwiska i stanowiska pracowników banku, po przetworzeniu powyższego dokumentu przez servlet XSQL, wywołany przez serwer WWW, do użytkownika końcowego dotrze następujący dokument XML:

```
<?xml version="1.0"?>
<ROWSET>
  <ROW id="1">
    <NAZWISKO> Kowalski </NAZWISKO>
    <STANOWISKO> KASJER </STANOWISKO>
  </ROW>
  <ROW id="2">
    <NAZWISKO> Nowak </NAZWISKO>
    <STANOWISKO> STAŻYSTA </STANOWISKO>
  </ROW>
  <ROW id="3">
    <NAZWISKO> Zieliński </NAZWISKO>
    <STANOWISKO> REFERENT </STANOWISKO>
  </ROW>
</ROWSET>
```

Dokument XML może zawierać dowolną liczbę znaczników `<query>`, które mogą też być zagnieżdżone wewnątrz innych znaczników języka XML. Ponadto, znacznik `<query>` może być wyposażony w szereg atrybutów, wpływających na format wyniku pracy servletu XSQL. Atrybuty te zostały opisane w tabeli 2.

atrybut	opis
doc-element	nazwa znacznika otaczającego cały wynik zapytania (domyślnie <code>&lt;ROWSET&gt;</code> )
row-element	nazwa znacznika otaczającego pojedynczy rekord zwrócony przez zapytanie (domyślnie <code>&lt;ROW&gt;</code> )
max-rows	maksymalna liczba rekordów, jakie zostaną pobrane z wyniku zapytania (domyślnie wszystkie rekordy)
skip-rows	liczba początkowych rekordów, które zostaną pominięte z wyniku zapytania (domyślnie 0)
id-attribute	nazwa atrybutu identyfikującego każdy rekord z wyniku zapytania (domyślnie id)
id-attribute-column	nazwa kolumny z wyniku zapytania, używanej jako wartość atrybutu identyfikującego każdy rekord z wyniku zapytania (domyślnie numer kolejny rekordu)

**Tabela 2.** Wybrane atrybuty znacznika `<query>`

Zapytanie SQL zapisane przy użyciu znacznika `<query>` może odwoływać się do zmiennych substytucyjnych specyfikowanych jako `{@nazwa_zmiennej}`. Zmienne takie zostaną podstawione przed wykonaniem zapytania wartościami domyślnymi, wartościami pobranymi z zapytania nadrzędnego, bądź parametrami przekazanymi przez formularz HTML. W poniższym przykładzie dokumentu XML, po przesłaniu przez klienta żądania `http://serwer/ścieżka/dokument.xsql?stan=REFERENT`, wygenerowany zostanie dokument XML opisujący wszystkich zatrudnionych referentów. Jeżeli klient nie przekaze wartości parametru `stan`, wtedy przyjmie on wartość domyślną, zamieszczoną jako atrybut znacznika `<query>`, czego efektem będzie zwrócenie danych wszystkich pracowników.

```
<?xml version="1.0"?>
<query connection="kadry" stan="%">
  SELECT nazwisko, stanowisko FROM pracownicy
  WHERE stanowisko LIKE {@stan}
</query>
```

Wewnątrz znacznika `<query>` może zostać zagnieżdżony znacznik `<no-rows-query>`, który opisuje zapytanie awaryjne – takie, które wykona się zamiast zapytania `<query>`, jeżeli nie znalazło ono żadnych rekordów. Poniżej przedstawiono przykład, wykorzystujący znacznik `<no-rows-query>` dla umożliwienia elastycznego wyszukiwania pracowników według stanowiska, bądź według nazwiska.

```
<?xml version="1.0"?>
<query connection="kadry">
  SELECT nazwisko, stanowisko FROM pracownicy
  WHERE stanowisko LIKE {@stan}
  <no-rows-query>
    SELECT nazwisko, stanowisko FROM pracownicy
    WHERE nazwisko LIKE {@stan}
  </no-rows-query>
</query>
```

Powyższy dokument będzie umożliwiał wyszukiwanie pracowników zarówno przy użyciu wywołań typu `http://serwer/ścieżka/dokument.xsql?stan=REFERENT`, jak i `http://serwer/ścieżka/dokument.xsql?stan=Kowalski`.

### 3.7 Transformacja dokumentów XML przy użyciu XSLT

Moduł XSLT jest stosowany dla zapewnienia współpracy ze starszymi przeglądarkami WWW, które nie potrafią interpretować języka XML. Jeżeli na początku pliku XSQL zostanie zapisany znacznik `<?xml-stylesheet?>`, wtedy generowany dokument XML będzie mógł być automatycznie przetłumaczony do formatu HTML i w takiej postaci przesłany do przeglądarki użytkownika. Poniżej przedstawiono przykładowy zapis znacznika `<?xml-stylesheet?>`.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="uklad.xsl"?>
...
```

Atrybut HREF znacznika `<?xml-stylesheet?>` specyfikuje nazwę pliku XSL, w którym zapisano wzorzec układu graficznego, jaki zostanie zastosowany do dokumentu przesyłanego do przeglądarki WWW użytkownika. W efekcie, użytkownik otrzyma zsyntetyzowany dokument HTML.

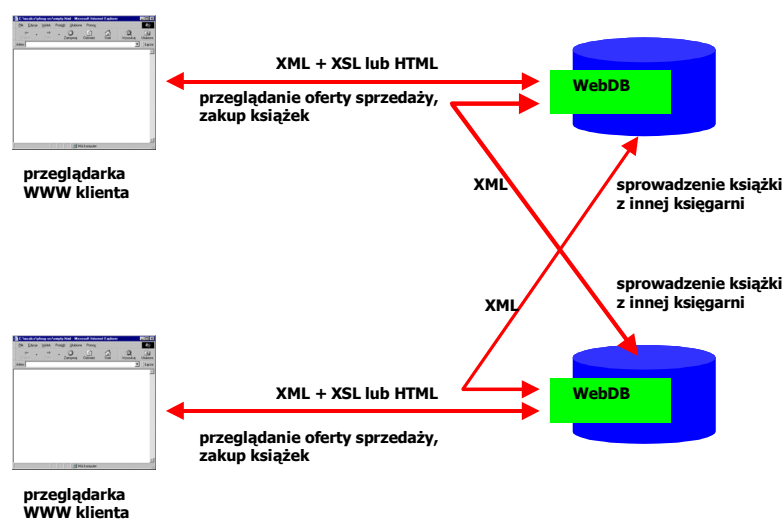
Możliwe jest zapisanie kilku znaczników `<?xml-stylesheet?>` w jednym dokumencie XML i wyposażenie ich w atrybut `media`. Na podstawie jego wartości, wybór sposobu formatowania będzie mógł być uzależniony od typu przeglądarki WWW, jaką posługuje się użytkownik, np.:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" media="lynx"
href="u_1.xsl"?>
<?xml-stylesheet type="text/xsl" media="msie"
href="u_m.xsl"?>
...
```

## 4 Podsumowanie

Nowe własności obsługi Internetu, jakie zostały zaimplementowane w środowisku Oracle8i, swą prostotą stanowią zachętę do konstrukcji rozproszonych systemów bazodanowych, w których publikowanie i wymiana informacji jest w pełni zautomatyzowana. W celu zilustrowania hipotetycznej implementacji systemu sieciowego na platformie Oracle8i, na rysunku 2 przedstawiono ogólną strukturę sieci księgarń internetowych. Każda księgarnia sprzedaje wysyłkowo książki, które posiada we własnym

magazynie, a także książki, jakie dla klienta może sprowadzić z innej księgarni. Klient uzyskuje dostęp do katalogu sprzedawanych książek przy pomocy swojej przeglądarki WWW, która kieruje żądania do WebDB, a w rezultacie otrzymuje dokumenty HTML bądź XML+XSL (np. wykorzystujące pliki multimedialne z iFS). Dokumenty te są generowane przez procedury PL/SQL lub servlet XSQL na podstawie danych składowanych w bazie danych. Jeżeli klient jest zainteresowany pozycją książkową, która nie występuje aktualnie w magazynie, wtedy oprogramowanie PL/SQL, pracujące na rzecz księgarni poszukuje tej pozycji, w imieniu klienta, w innych księgarniach. Komunikacja z innymi księgarniami jest realizowana poprzez pakiet UTL\_HTTP, pobierający oferty sprzedaży w postaci XML (tak, jak zwykły klient) oraz parser XML, który dokonuje logicznej analizy oferty i pozwala stwierdzić dostępność lub niedostępność książki. Ostateczne żądanie zakupu może być wówczas przekierowane przez księgarnię w sposób niezauważalny dla użytkownika. Na zakończenie należy zwrócić uwagę, że powyższa architektura może być budowana również przy udziale obcych, innych niż Oracle środowisk (np. dBase + oprogramowanie w języku C) – język XML staje się wtedy uniwersalnym integratorem systemów.



Rysunek 2. Architektura sieci księgarni internetowych opartych na Oracle8i

## 5 Literatura

- [1] Data Management for XML J. Widom, <http://www-db.stanford.edu/~widom/xml-whitepaper.html>
- [2] Hypertext Transfer Protocol -- HTTP/1.0, T. Berners-Lee, R. Fielding, H. Frystyk, Internet Draft

- [3] Internet Standards and Protocols, D.C. Naik, Microsoft Press, 1998
- [4] Introduction to XML, <http://msdn.microsoft.com/xml/general/intro.asp>
- [5] Oracle Application Server, dokumentacja techniczna
- [6] Oracle XSQL Servlet, release notes for 0.9 technology preview,  
[http://technet.oracle.com/tech/xml/xsql\\_servlet/doc/relnotes.htm](http://technet.oracle.com/tech/xml/xsql_servlet/doc/relnotes.htm)
- [7] Oracle8i, dokumentacja techniczna
- [8] W3C DOM APIs, [http://technet.oracle.com/tech/parser\\_plsql/doc/package-dom.html](http://technet.oracle.com/tech/parser_plsql/doc/package-dom.html)
- [9] XML Parser API, [http://technet.oracle.com/tech/parser\\_plsql/doc/package-parser.html](http://technet.oracle.com/tech/parser_plsql/doc/package-parser.html)
- [10] XML Support in Oracle8i and Beyond,  
[http://www.oracle.com/xml/documents/xml\\_twp/](http://www.oracle.com/xml/documents/xml_twp/)