

XIV Konferencja PLOUG
Szczyrk
Październik 2008

Od interMedia do Multimedia – obsługa danych multimedialnych w Oracle 10g/11g

Marek Wojciechowski
Politechnika Poznańska

e-mail: Marek.Wojciechowski@cs.put.poznan.pl

Abstrakt. Oracle Multimedia to cecha (ang. feature) serwera bazy danych Oracle 11g dostępna zarówno w edycji serwera Enterprise Edition jak i Standard Edition, w poprzednich wersjach serwera znana pod nazwą Oracle interMedia. Oracle Multimedia umożliwia składowanie, wyszukiwanie i w pewnym stopniu przetwarzanie obrazów, danych audio i danych wideo. Celem artykułu jest przedstawienie nowości i zmian w Oracle interMedia 10.2 i Oracle Multimedia 11.1, ze szczególnym uwzględnieniem ekstrakcji metadanych o obrazach i obsłudze obrazów medycznych.

Informacja o autorze. Dr inż. Marek Wojciechowski jest adiunktem w Instytucie Informatyki Politechniki Poznańskiej. Specjalizuje się w eksploracji danych, technologiach internetowych i obsłudze danych multimedialnych w bazach danych. Jest autorem ok. 70 publikacji o charakterze naukowym i technicznym. Od kilku lat prowadzi szkolenia z technologii internetowych i produktów Oracle w Polsce i za granicą w ramach Oracle University. W trakcie konferencji, szkół i seminariów PLOUG wielokrotnie prowadził warsztaty, tutoriale i wykłady na temat tworzenia aplikacji internetowych i obsługi zaawansowanych typów danych przez system zarządzania bazą danych Oracle.

1. Wprowadzenie

Oracle Multimedia to cecha (ang. feature) serwera bazy danych Oracle 11g. Nie jest to więc samodzielny produkt ani opcja wymagająca dodatkowej licencji, a jedynie nazwany podzbiór funkcjonalności serwera Oracle. Zadaniem Oracle Multimedia jest dostarczenie na poziomie systemu zarządzania bazą danych typowej funkcjonalności wykorzystywanej przez aplikacje przeznaczone do składowania, przetwarzania i przeszukiwania kolekcji multimedialnych. Oracle Multimedia jest cechą serwera dostępną zarówno w edycji serwera Enterprise Edition jak i Standard Edition. W poprzednich wersjach serwera cecha ta występowała pod nazwą Oracle interMedia i była dostępna od wersji Oracle 8i (8.1.5). W wersji 11g nastąpiła zmiana nazwy, ale nie wynikała ona ze zmiany architektury, więc można ją traktować jako zabieg „kosmetyczny”, porządkujący nazewnictwo poszczególnych cech, tak aby ich nazwy możliwie czytelnie odzwierciedlały ich funkcjonalność.

Oracle Multimedia umożliwia składowanie, wyszukiwanie i w pewnym stopniu przetwarzanie obrazów, danych audio i danych wideo. Funkcjonalność ta jest realizowana w oparciu o obiektowe typy SQL, ale dostarczany jest również tzw. interfejs relacyjny. Od wersji 8.1.5 podstawowe znaczenie mają typy ORDIImage, ORDAudio, ORDVideo i ORDDoc. Pierwsze trzy służą do reprezentacji odpowiednio: obrazów, danych audio i danych wideo, a czwarty może być wykorzystany do składowania heterogenicznych kolekcji obiektów multimedialnych. Kluczowym, pomocniczym typem danych Oracle Multimedia/interMedia jest ORDSource, wykorzystywany przez wszystkie cztery wymienione wcześniej typy do reprezentacji zawartości binarnej multimedialnych, z możliwościami składowania ich w bazie danych, w systemie plików serwera bazy danych lub na dedykowanym zewnętrznym serwerze multimedialnym.

Funkcjonalność Oracle Multimedia/interMedia ulega zmianom z wersji na wersję. Najczęściej zmiany te polegają na dodaniu nowej funkcjonalności lub poprawie wydajności i skalowalności, ale zdarza się również, że pewna funkcjonalność zostaje zarzucona. Celem artykułu jest przedstawienie nowości i zmian w Oracle Multimedia/interMedia w dwóch ostatnich wersjach serwera bazy danych Oracle: 10g Release 2 (10.2) i 11g release 1 (11.1), ze szczególnym uwzględnieniem ekstrakcji metadanych o obrazach i obsługi obrazów medycznych.

2. Obsługa metadanych o zawartości obrazów

Metadane, czyli „dane o danych” pełnią istotną rolę w opisywaniu, indeksowaniu i wyszukiwaniu obiektów multimedialnych. Serwer bazy danych Oracle od momentu pojawienia się w nim cechy interMedia umożliwia ekstrakcję metadanych określanych mianem „technicznych”, czyli takich, które opisują techniczne aspekty obiektu multimedialnego związane z jego formatem, rozmiarem, sposobem kodowania, czyli ogólnie - sposobem reprezentacji w postaci cyfrowej. Metadane techniczne wspólne dla wszystkich obsługiwanych przez Oracle typów obiektów multimedialnych, czyli obrazów, danych audio i danych wideo, to: format, typ MIME i rozmiar w bajtach. Poza tym podstawowym zestawem metadanych technicznych poszczególne typy multimedialnych posiadają dodatkowo zestaw specyficznych dla nich metadanych. W przypadku obrazów są to: wysokość i szerokość wyrażona w pikselach oraz zastosowana metoda kompresji.

Drugą ważną kategorią metadanych opisujących zawartość multimedialną są metadane o zawartości. Ich dostępność zależy od wykorzystywanego formatu pliku, gdyż nie wszystkie formaty przewidują możliwość umieszczania metadanych towarzyszących binarnej zawartości multimedialnej, i oczywiście od tego czy aplikacja generująca czy zapisująca plik z tej możliwości skorzystała i rzeczywiście umieściła w pliku metadane o zawartości. Co ciekawe, w wersjach interMedia wcześniejszych niż 10.2 ekstrakcja metadanych o zawartości dostępna była dla danych audio i wi-

deo, a niedostępna dla obrazów. Ekstrahowane metadane określone były mianem adnotacji i zapisywane w formacie XML. Wynikowy XML zawierał zarówno metadane techniczne jak i metadane o zawartości. Przykładem tych drugich mogły być np. wykonawca i tytuł utworu wydobyte z obiektu audio w formacie MP3.

Od wersji 10.2 (z niezmienną funkcjonalnością w wersji 11g) Oracle w ramach funkcjonalności interMedia/Multimedia obsługuje metadane o zawartości również dla obrazów, wspierając trzy popularne standardy:

- EXIF,
- IPTC-IIM,
- XMP.

EXIF (pełna nazwa: Exchangeable Image File Format) to standard wykorzystywany w cyfrowych aparatach fotograficznych, opracowany przez Japan Electronic Industry Development Association (JEIDA). Metadane w standardzie EXIF mogą być zawarte w obrazach w formacie TIFF i JPEG. Oracle umożliwia ekstrakcję metadanych EXIF z obu tych formatów graficznych.

IPTC-IIM (pełna nazwa: International Press Telecommunications Council-Information Interchange Model) to standard rozwijany wspólnie przez International Press Telecommunications Council i Newspaper Association of America. Standard ten jest zorientowany na informacje istotne z punktu widzenia zbierania i publikowania newsów. Podobnie jak w przypadku EXIF, metadane IPTC-IIM mogą być zawarte w obrazach w formacie TIFF lub JPEG i z obu tych formatów Oracle potrafi je wyekstrahować.

XMP (pełna nazwa: Extensible Metadata Platform) jest standardem opracowanym przez firmę Adobe do tworzenia, przetwarzania i wymiany metadanych w różnych zastosowaniach. Oracle obsługuje metadane XMP dla formatów TIFF, JPEG oraz GIF i co wymaga podkreślenia, potrafi nie tylko je odczytywać, ale również zapisywać we wszystkich trzech wspieranych formatach obrazów.

Odczytane przez interMedia/Multimedia metadane o zawartości obrazu są reprezentowane w formie dokumentów XML i mogą być składowane w bazie danych Oracle w kolumnach typu XMLType. Dla każdego z trzech standardów metadanych w bazie danych zarejestrowany jest odpowiedni schemat XML Schema wyznaczający strukturę dokumentu XML zawierającego wyekstrahowane metadane. XML jako format reprezentacji w bazie danych odczytanych metadanych o zawartości obrazów ma tę zaletę, że Oracle oferuje wiele opartych o standardy mechanizmów przeszukiwania danych w tym formacie w ramach funkcjonalności Oracle XML DB.

W celu ilustracji sposobu korzystania z mechanizmów obsługi metadanych o zawartości obrazów rozważymy przykład, w którym z załadowanej do bazy danych fotografii w formacie JPEG odczytane zostaną metadane zagnieżdżone w pliku. Na początek utworzymy tabelę o nazwie HOLIDAY_PHOTOS, która będzie zawierać między innymi kolumnę typu ORDImage do składowania obrazów oraz kolumnę typu XMLType do składowania wyekstrahowanych metadanych:

```
CREATE TABLE holiday_photos
(id NUMBER PRIMARY KEY,
 metaEXIF XMLTYPE,
 photo ORDSYS.ORDIMAGE)
XMLType COLUMN metaEXIF
XMLSCHEMA "http://xmlns.oracle.com/ord/meta/exif"
ELEMENT "exifMetadata";
```

Przygotowując w tabeli kolumnę do składowania metadanych musieliśmy określić w jakim standardzie będą składowane odczytane metadane poprzez wskazanie odpowiedniego schematu XML Schema dla zawartości XMLType. W naszym przypadku jest to standard EXIF.

Następnie załadujemy obrazek z systemu plików do bazy danych. W przypadku większej liczby ładowanych plików należałoby z pewnością rozważyć wykorzystanie do tego celu narzędzia SQL Loader. W naszym przykładzie, dla pojedynczego pliku, do załadowania obrazka użyjemy poniższego bloku kodu w języku PL/SQL:

```
DECLARE
    temp ordsys.ORDImage;
    ctx RAW(64) := NULL;
BEGIN
    INSERT INTO holiday_photos (id, photo)
    VALUES (42,
        ordsys.ORDImage.init('FILE', 'MEDIAFILES', 'szklarka.jpg'))
    RETURNING photo INTO temp;
    temp.import(ctx);
    temp.setProperties();
    UPDATE holiday_photos SET photo=temp WHERE id=42;
    COMMIT;
END;
```

W celu załadowania obiektu multimedialnego z pliku do bazy danych za pomocą PL/SQL należy wykonać kilka kroków, co może wydawać się skomplikowane. Zaletą takiego rozwiązania jest jednak to, że operacje niezbędne do załadowania danych binarnych do bazy można już na tym etapie uzupełnić o operacje opcjonalne, ale w praktyce bardzo potrzebne, tj. ekstrakcję różnego rodzaju metadanych.

Pierwszym krokiem programu PL/SQL ładującego obrazek do bazy danych jest polecenie INSERT, które wstawia obiekt multimedialny do tabeli reprezentując jego zawartość binarną jako BFILE. Jednym z argumentów polecenia INSERT jest inicjalizacja obiektu typu ORDImage jego metodą init(). Pierwszy argument tej metody wskazuje, że obiekt znajduje się w lokalnym systemie plików, drugi argument to nazwa obiektu DIRECTORY reprezentującego w bazie danych katalog systemu plików zawierający ładowany plik, a ostatni argument to nazwa tego pliku. Wykorzystanie BFILE do składowania danych binarnych oznacza oczywiście, że po wykonaniu operacji INSERT zawartość binarna pozostaje fizycznie poza bazą danych, a w bazie danych znajduje się jedynie wskaźnik do niej. Takie rozwiązanie pozwala np. na ekstrakcję metadanych i składowanie ich w bazie, ale nie chroni binarnej zawartości obiektu przed dostępem spoza bazy danych ani nie obejmuje jej przetwarzaniem transakcyjnym. Dlatego też w drugim kroku na rzecz obiektu ORDImage wywołana jest metoda import() realizująca kopiowanie zawartości binarnej obiektu z zewnętrznego źródła (w naszym przypadku lokalnego pliku) do obiektu BLOB. Aby możliwe było wywoływanie metod-procedur na rzecz obiektu, obiekt ten musi być dostępny z poziomu PL/SQL. Stąd klauzula RETURNING w poleceniu INSERT, która umieszcza obiekt w zmiennej lokalnej. W kolejnym kroku metodą setProperties() realizowana jest ekstrakcja podstawowych metadanych technicznych, co powoduje umieszczenie ich w poszczególnych atrybutach obiektu ORDImage. Ostatnim krokiem jest polecenie UPDATE, które sprawia, że zmiany dokonane na tymczasowym, ulotnym obiekcie PL/SQL są przenoszone do bazy danych.

Wyekstrahowane metodą setProperties() metadane techniczne mogą być odczytane odpowiednimi metodami dostępu do składowych. Poniższy fragment kodu realizuje odczyt szerokości i wysokości obrazu w pikselach:

```
SELECT p.photo.getWidth() WIDTH,
       p.photo.getHeight() HEIGHT
FROM holiday_photos p
WHERE id = 42;
```

Ekstrakcja metadanych o zawartości obrazu, która jest głównym celem niniejszego przykładu, realizowana jest metodą `getMetadata()` obiektu `ORDImage`. Argumentem tej metody jest nazwa standardu metadanych, które chcemy odczytać. Jeśli naszą intencją jest odczytanie metadanych w wielu standardach lub też nie wiemy czy i jakie metadane zostały zawarte w pliku obrazu, możemy w miejscu nazwy standardu podać słowo „ALL”, co ilustruje poniższy fragment kodu PL/SQL:

```

DECLARE
    image ORDSYS.ORDImage;
    metad XMLSequenceType;
BEGIN
    SELECT photo INTO image FROM holiday_photos WHERE id=42;
    metad := image.getMetadata('ALL');
    FOR i in 1..metad.count LOOP
        DBMS_OUTPUT.PUT_LINE('namespace: ' ||
                               metad(i).getNamespace() );
    END LOOP;
EXCEPTION
    WHEN ORDSYS.ORDImageExceptions.NULL_LOCAL_DATA THEN
        DBMS_OUTPUT.PUT_LINE('source local data is null');
    WHEN ORDSYS.ORDImageExceptions.NULL_SOURCE THEN
        DBMS_OUTPUT.PUT_LINE('source is null');
END;
/

```

Ponieważ metoda `getMetadata()` może potencjalnie odczytać w wyniku pojedynczego wywołania metadane w więcej niż jednym standardzie, jej wynik nie jest pojedynczym dokumentem typu `XMLType`, ale kolekcją dokumentów XML typu `XMLSequenceType`. W powyższym bloku kodu, po odczycie metadanych, wykonywana jest pętla po wszystkich odczytanych dokumentach z metadanymi i dla każdego z nich wyświetlana jest przestrzeń nazw, którą można utożsamiać z nazwą standardu metadanych. Ponieważ wykorzystany w przykładzie obrazek jest fotografią wykonaną aparatem cyfrowym, zawarte są w nim metadane EXIF. Ponadto w `interMedia/Multimedia` zawsze dostępne są metadane w „standardzie” `ORDIMAGE`, czyli lista metadanych technicznych obsługiwanych przez Oracle `interMedia/Multimedia`, zapisanych w formie dokumentu XML. Wykonanie powyższego bloku kodu PL/SQL dla rozważanego obrazka spowoduje wyświetlenie na konsoli następujących informacji:

```

namespace: http://xmlns.oracle.com/ord/meta/ordimage
namespace: http://xmlns.oracle.com/ord/meta/exif

```

Komentarza wymagają jeszcze klauzule obsługi wyjątków, które mogą pojawić się przy odczycie metadanych. Nie są to jednak wyjątki specyficzne dla operacji dotyczących metadanych o zawartości. Pojawiają się one w sytuacjach gdy podejmowana jest próba operacji na obiekcie `ORDImage`, którego zawartość nie została prawidłowo zainicjalizowana. Aby uniknąć tych wyjątków należy inicjalizować obiekty `interMedia/Multimedia` odpowiednią metodą `init()`.

W przeciwieństwie do metadanych technicznych odczytywanych metodą `setProperties()`, metadane o zawartości obrazu odczytane metodą `getMetadata()` nie są automatycznie umieszczone w atrybutach obrazu, gdyż typ `ORDImage` zawiera jedynie atrybuty dla metadanych technicznych. Aby przechować odczytane z obrazu metadane o zawartości, należy więc jawnie zapisać je w bazie danych poza obrazem. W naszym przykładzie, w tabeli, w której znajduje się kolumna typu `ORDImage` do przechowywania obrazów, zawarta została również kolumna typu `XMLType` do składowania metadanych EXIF. Skojarzenie metadanych o zawartości obrazu z obrazem realizowane jest więc na poziomie schematu bazy danych przez jego projektanta. Poniższy blok kodu

PL/SQL odczytuje ze znajdującego się w tabeli bazy danych obrazu metadane o zawartości, tym razem już tylko w standardzie EXIF, i umieszcza je w innym atrybucie tego samego wiersza:

```
DECLARE
  image ORDSYS.ORDImage;
  metad XMLSequenceType;
  exif XMLType;
BEGIN
  SELECT photo INTO image
  FROM holiday_photos WHERE id = 42;
  metad := image.getMetadata('EXIF');
  exif := metad(1);
  UPDATE holiday_photos
  SET metaexif = exif WHERE id = 42;
END;
/
```

Dokument XML z metadanymi EXIF wyekstrahowanymi z przykładowego obrazka ma następującą postać:

```
<exifMetadata xmlns="http://xmlns.oracle.com/ord/meta/exif"
  xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/exif
  http://xmlns.oracle.com/ord/meta/exif"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <TiffIfd>
    <Make tag="271">Canon</Make>
    <Model tag="272">Canon PowerShot G6</Model>
    <Orientation tag="274">right top</Orientation>
    <XResolution tag="282">180</XResolution>
    <YResolution tag="283">180</YResolution>
    <ResolutionUnit tag="296">inches</ResolutionUnit>
    <DateTime tag="306">2008-04-30T16:28:18.000000</DateTime>
    <YCbCrPositioning tag="531">centered</YCbCrPositioning>
  </TiffIfd>
  <ApertureValue tag="37378">4,96875</ApertureValue>
  <ExposureBiasValue tag="37380">0</ExposureBiasValue>
  <MaxApertureValue tag="37381">2,65625</MaxApertureValue>
  <MeteringMode tag="37383">Pattern</MeteringMode>
  <Flash tag="37385">
    <Fired>No</Fired>
    <Return>No strobe return function</Return>
    <Mode>Compulsory suppression</Mode>
    <Function>Yes</Function>
    <RedEyeReduction>No</RedEyeReduction>
  </Flash>
  <FocalLength tag="37386">17,59375</FocalLength>
  <FlashpixVersion tag="40960">0100</FlashpixVersion>
  <ColorSpace tag="40961">sRGB</ColorSpace>
  ...
</exifMetadata>
```

Dla kompletności przykładu pokażemy jeszcze jaką postać dla przykładowego obrazka miałyby metadane w „standardzie” ORDIMAGE (odczytane wywołaniem: `getMetadata('ORDIMAGE')`):

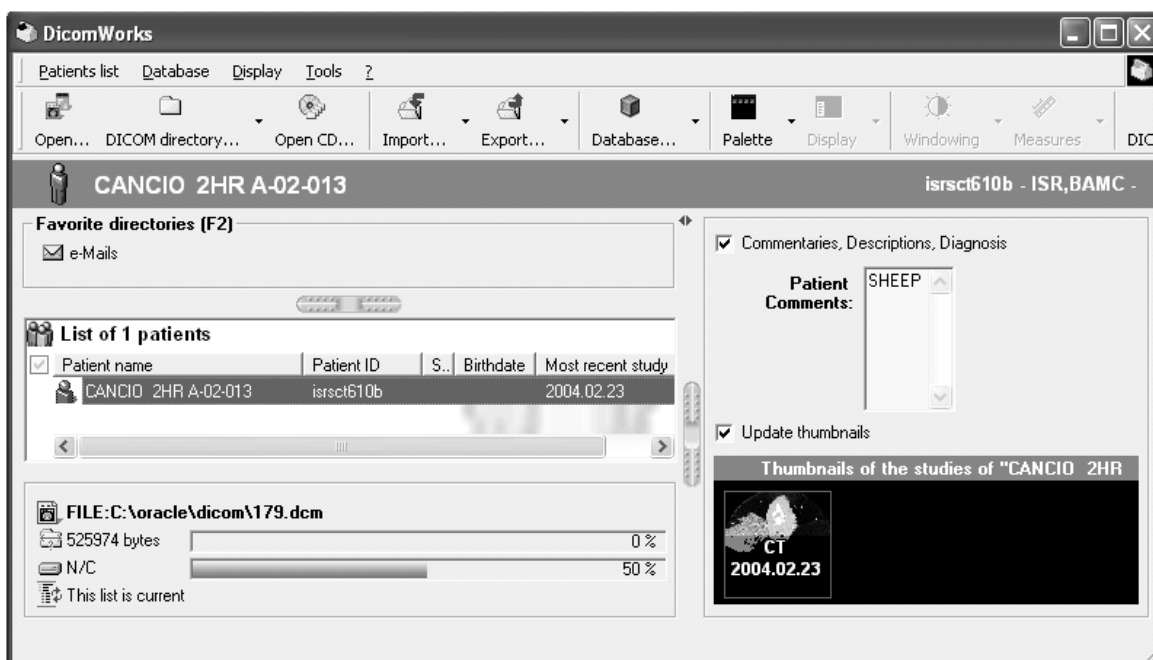
```
<ordImageAttributes
  xmlns="http://xmlns.oracle.com/ord/meta/ordimage">
```

```
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/ordimage
  http://xmlns.oracle.com/ord/meta/ordimage">
  <height>2304</height>
  <width>3072</width>
  <contentLength>1992462</contentLength>
  <fileFormat>JFIF</fileFormat>
  <contentFormat>24BITRGB</contentFormat>
  <compressionFormat>JPEG</compressionFormat>
  <mimeType>image/jpeg</mimeType>
</ordImageAttributes>
```

3. Obsługa zdjęć medycznych w formacie DICOM

Digital Imaging and Communications in Medicine (DICOM) to standard dotyczący obrazów wykorzystywanych w medycynie. Został on opracowany z inicjatywy dwóch organizacji: American College of Radiology (ACR) i National Electrical Manufacturers Association (NEMA) w celu ułatwienia współpracy urządzeń wykorzystywanych w radiologii. Potrzeba opracowania standardu wynikała z tego, że przed jego ustanowieniem każdy z producentów sprzętu radiologicznego używał własnego formatu obrazu i protokołu komunikacyjnego, co w praktyce utrudniało lub wręcz uniemożliwiało łączenie ze sobą urządzeń różnych producentów, czy też tworzenie zewnętrznego oprogramowania do analizy uzyskanych zdjęć. Odpowiedzią na te problemy był standard opublikowany w 1985 roku, początkowo pod nazwą ACR-NEMA. W roku 1993 nazwa standardu została zmieniona na DICOM, z jednoczesnym nadaniem mu numeru wersji 3.0. Ta wersja obowiązuje do dziś mimo pojawiających się w standardzie modyfikacji. DICOM 3.0 jest obecnie dominującym standardem w zakresie zarówno formatu obrazów jak i protokołów komunikacji wykorzystywanych w radiologii. W praktyce, można dziś przyjąć że każde oprogramowanie obsługujące standard DICOM będzie mogło być użyte do analizy zdjęć pochodzących z dowolnego urządzenia.

DICOM jest standardem zorientowanym obiektowo. Obiekty DICOM wiążą zawartość graficzną z atrybutami, które jej dotyczą. Poniższy rysunek przedstawia przykładowy obiekt DICOM otwarty w narzędziu DicomWorks (<http://dicom.online.fr>). Obok miniatury obrazu widoczne są związane z nim atrybuty takie jak np. nazwisko i identyfikator pacjenta.



Rys. 1. Przykładowy obraz DICOM (w narzędziu DicomWorks)

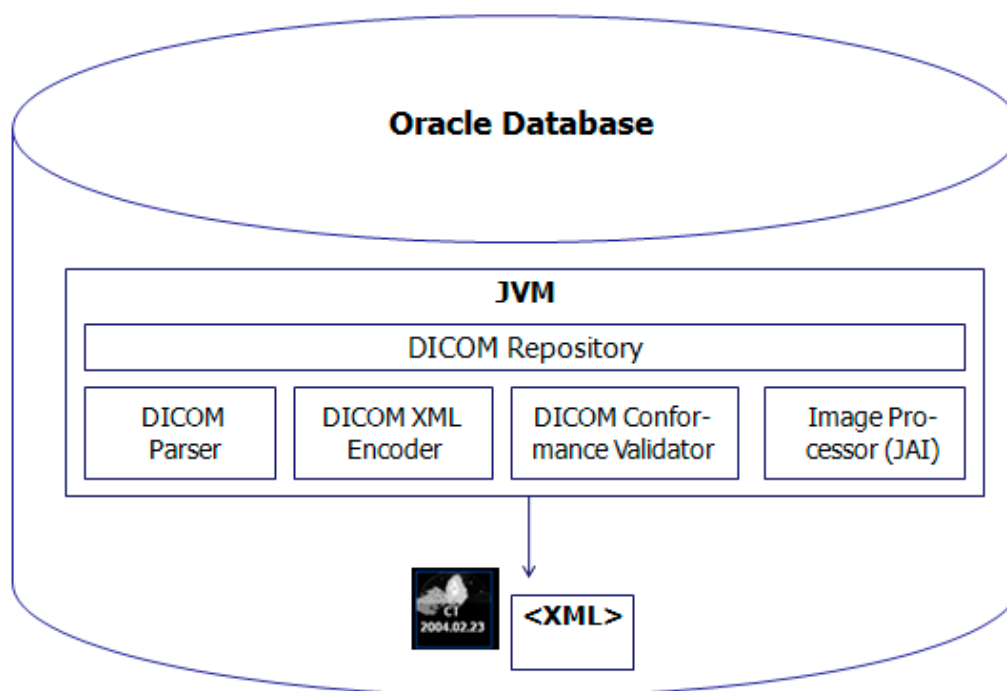
Pierwszą wersją serwera bazy danych Oracle oferującą wsparcie dla standardu DICOM w ramach Oracle interMedia była wersja 10.2. Wsparcie to było ograniczone do rozpoznawania najważniejszych atrybutów formatu DICOM, a obrazy DICOM były składowane w bazie danych, podobnie jak inne obrazy, jako obiekty ORDImage.

Wersja 11g zachowuje wsparcie dla ekstrakcji atrybutów DICOM z obrazów medycznych składowanych jako ORDImage, ale dodatkowo wprowadza nowy, dedykowany do obsługi obrazów DICOM, typ danych ORDDicom oferujący znacznie rozszerzoną funkcjonalność. Ponadto dostępny jest interfejs relacyjny w postaci pakietu ORD_DICOM o funkcjonalności pokrywającej się z obiektowym typem ORDDicom.

Obsługa obrazów medycznych w formacie DICOM w Oracle Multimedia 11g obejmuje:

- pełną obsługę metadanych DICOM,
- testowanie zgodności ze standardem,
- przetwarzanie obrazów (kopiowanie, konwersja formatu),
- anonimizację danych DICOM,
- tworzenie obiektów DICOM poprzez wiązanie zdjęć z metadanymi,
- interfejsy programistyczne w językach Java i PL/SQL.

Rys. 2 przedstawia architekturę Oracle DICOM. Wsparcie dla DICOM w bazie danych Oracle jest realizowane na poziomie wirtualnej maszyny Java (JVM) wbudowanej w serwer bazy danych. Centralnym elementem wykorzystywanym przez pozostałe moduły Oracle DICOM jest repozytorium modelu danych DICOM (DICOM Repository). Za ekstrakcję metadanych z obiektów DICOM odpowiada parser DICOM (DICOM Parser). DICOM XML Encoder odpowiada za generację dokumentu XML z metadanymi odczytanymi z obiektu DICOM. DICOM Conformance Validator służy do sprawdzania składniowej i semantycznej zgodności zawartości DICOM z regułami zapisanymi w repozytorium modelu danych. Image Processor jest oparty o Java Advanced Imaging (JAI) i umożliwia przetwarzanie obrazów DICOM, np. generację miniatur i konwersję obrazu do formatów graficznych ogólnego przeznaczenia.



Rys. 2. Architektura Oracle Multimedia DICOM

W celu ilustracji funkcjonalności Oracle DICOM w ramach Oracle Multimedia 11g przedstawimy proste przykłady operacji na przykładowym obrazie DICOM. Rozpoczniemy od polecenia tworzącego tabelę zawierającą cztery kolumny: identyfikator obrazu, nazwisko pacjenta, zdjęcie DICOM oraz miniaturę zdjęcia w formie zwykłego obrazka:

```
CREATE TABLE medical_images
(id INTEGER PRIMARY KEY,
patient_name VARCHAR2(30),
dicom ordsys.ORDDicom,
thumbnail ordsys.ORDImage);
```

Przed wywołaniem jakiegokolwiek funkcjonalności Oracle DICOM należy raz w sesji załadować repozytorium modelu danych z bazy danych do struktur w pamięci poprzez wywołanie odpowiedniej procedury:

```
EXEC ordsys.ord_dicom.setDataModel()
```

Ładowanie obiektu DICOM z pliku w systemie plików serwera do bazy danych realizowane jest podobnie jak w przypadku obrazów ORDImage. Najpierw obiekt ładowany jest do bazy danych jako BFILE, a w kolejnym kroku wywoływana jest metoda import() typu ORDDicom. Przekazanie wartości 1 jako argumentu tej metody powoduje wyekstrahowanie atrybutów DICOM z importowanego obiektu i zapisanie ich w odpowiednich atrybutach ORDDicom.

```
DECLARE
temp ordsys.ORDDicom;
BEGIN
INSERT INTO medical_images (id, dicom, thumbnail)
VALUES (179,
ordsys.ORDDicom('FILE',
'DICOMFILES', '179.dcm', 0),
ordsys.ORDImage.init())
RETURNING dicom INTO temp;
temp.import(1);
```

```

UPDATE medical_images SET dicom=temp WHERE id=179;
COMMIT;
END;
/

```

Wyekstrahowane w trakcie importu atrybuty obiektu DICOM są dostępne z poziomu SQL. Poniższe zapytanie odczytuje dla wybranego obiektu DICOM jeden z podstawowych atrybutów: SOP_INSTANCE_UID:

```

SELECT id, t.dicom.getSOPInstanceUID() AS SOP_Instance_UID
FROM medical_images t;

```

Wynik powyższego zapytania przy założeniu, że w tabeli znajduje się tylko jeden przykładowy obiekt DICOM będzie następujący:

```

ID          SOP_INSTANCE_UID
-----
179 1.2.392.200036.9116.2.2.2.1762676206.1077529882.102147

```

Komplet metadanych o obiekcie DICOM jest dostępny w formie dokumentu XML poprzez atrybut METADATA:

```

SELECT t.dicom.metadata FROM medical_images t WHERE id=179;

```

Dokument XML z metadanymi DICOM dla naszego przykładowego obiektu został przedstawiony poniżej. Ze względu na duży rozmiar dokumentu niektóre jego fragmenty zostały pominięte. Widoczna jest jednak ogólna struktura dokumentu oraz wspomniane już atrybuty DICOM: unikalny identyfikator (SOP Instance UID) i nazwisko pacjenta:

```

<?xml version="1.0"?>
<DICOM_OBJECT
  xmlns="http://xmlns.oracle.com/ord/dicom/metadata_1_0" ... >
...
  <CODE_STRING tag="00080008" definer="DICOM" name="Image Type"
    offset="352" length="22">
    ORIGINAL
  </CODE_STRING>
...
  <UNIQUE_ID tag="00080018" definer="DICOM"
    name="SOP Instance UID" offset="416" length="54">
    1.2.392.200036.9116.2.2.2.1762676206.1077529882.102147
  </UNIQUE_ID>
...
  <PERSON_NAME tag="00100010" definer="DICOM"
    name="Patient's Name" offset="710" length="22">
    <NAME type="unibyte">
      <FAMILY>CANCIO 2HR A-02-013</FAMILY>
    </NAME>
    <VALUE>CANCIO 2HR A-02-013</VALUE>
  </PERSON_NAME>
...
</DICOM_OBJECT>

```

W przypadku gdy zależy nam na odczytaniu konkretnych, wybranych atrybutów DICOM, które nie są bezpośrednio dostępne jako atrybuty obiektu ORDDicom, możemy je wyekstrahować z powyższego dokumentu zawierającego komplet metadanych. Do tego celu należy oczywiście wykorzystać funkcjonalność XML DB dostępną z poziomu języka SQL. Poniższe zapytanie od-

czytuje nazwisko i identyfikator pacjenta wykorzystując funkcję `extractValue()`, której argumentem jest wyrażenie XPath wyszukujące wartości wybranych węzłów dokumentu XML:

```
SELECT id,
       extractValue(t.dicom.metadata,
                   '/DICOM_OBJECT/*[@name="Patient''s Name"]/VALUE',
                   'xmlns=http://xmlns.oracle.com/ord/dicom/metadata_1_0')
       as PATIENT_NAME,
       extractValue(t.dicom.metadata,
                   '/DICOM_OBJECT/*[@name="Patient ID"]',
                   'xmlns=http://xmlns.oracle.com/ord/dicom/metadata_1_0')
       as PATIENT_ID
FROM medical_images t;

ID          PATIENT_NAME          PATIENT_ID
-----
179 CANCIO    2HR A-02-013  ISRSCT610b
```

Jeśli któreś z metadanych niedostępnych bezpośrednio poprzez atrybuty typu `ODRDicom` są często wykorzystywane w konkretnej aplikacji, to można je wyekstrahować i składować w przygotowanej wcześniej kolumnie tabeli. W naszym przypadku przygotowaliśmy wcześniej kolumnę dla nazwiska pacjenta. Aby wypełnić ją danymi wyekstrahowanymi z metadanych XML należy wykonać poniższe polecenie `UPDATE`:

```
UPDATE medical_images t
SET patient_name =
  extractValue(t.dicom.metadata,
              '/DICOM_OBJECT/*[@name="Patient''s Name"]/VALUE',
              'xmlns=http://xmlns.oracle.com/ord/dicom/metadata_1_0');
```

Typową operacją w zakresie przetwarzania obrazów jest generacji miniatur (ang. thumbnail). Jest ona dostępna w Oracle Multimedia również dla obrazów DICOM, a sposób realizacji tej operacji w zasadzie nie różni się od generacji miniatur dla obrazów `ORDImage`. Wynikiem generacji miniatury jest obraz `ORDImage`, a nie nowy obiekt `ORDDicom` z kompletem metadanych DICOM. Jeśli chcemy składować miniatury zdjęć DICOM obok oryginalnej zawartości, musimy w tym celu przygotować kolumnę tabeli typu `ORDImage`. Poniższy blok kodu PL/SQL generuje miniatury dla obrazów DICOM z tabeli `MEDICAL_IMAGES` i umieszcza je w kolumnie `THUMBNAIL` tej tabeli:

```
DECLARE
  dcmSrc    ordsys.ORDDicom;
  imgDst    ordsys.ORDImage;
BEGIN
  SELECT dicom, thumbnail INTO dcmSrc, imgDst
  FROM medical_images
  WHERE id = 179 FOR UPDATE;
  dcmSrc.processCopy('fileformat=jfif fixedscale=75 100',
                    imgDst);
  UPDATE medical_images
  SET thumbnail = imgDst WHERE id = 179;
  COMMIT;
END;
/
```

Generacja miniatury jest realizowana metodą `processCopy()` z odpowiednim pierwszym argumentem zawierającym w formie łańcucha znaków parametry, zgodnie z którymi ma być zrealizowane przetwarzanie obrazu. Wynikiem przetwarzania jest nowy obraz, który jest zapisywany

w obiekcie przekazany jako drugi argument processCopy(). Aby umożliwić wywołanie metody będącej procedurą (a nie funkcją), oba obiekty (źródłowy i docelowy), których operacja dotyczy, zostały odczytane z bazy danych do zmiennych PL/SQL. Po zakończeniu działania metody processCopy() wynikowy obraz jest zapisywany w bazie danych poleceniem UPDATE.

Ostatnią wartą omówienia funkcjonalnością Oracle DICOM jest tzw. anonimizacja, czyli usuwanie z obiektów DICOM informacji o pacjentach. Jest to bardzo ważne jeśli kolekcja zdjęć pacjentów ma być udostępniona do celów naukowo-badawczych lub dydaktycznych. Anonimizacja realizowana jest metodą makeAnonymous() typu ORDDicom. Jej pierwszym argumentem jest nowy identyfikator SOP Instance UID, który powinien być globalnie unikalny. Ten identyfikator będzie nadany tworzonemu anonimowemu obiektowi DICOM. Drugi argument metody makeAnonymous() wskazuje zmienną typu ORDDicom, w której ma być zapisany tworzony obiekt. Poniższy blok kodu PL/SQL umieszcza w tabeli ANON_IMAGES anonimową wersję obiektu DICOM odczytanego z tabeli MEDICAL_IMAGES:

```
DECLARE
    dcmSrc    ordsys.ORDDicom;
    anonDst   ordsys.ORDDicom;
BEGIN
    INSERT INTO anon_images(id, dicom)
    VALUES (179, ordsys.ORDDicom())
    RETURNING dicom INTO anonDst;
    SELECT dicom INTO dcmSrc FROM medical_images
    WHERE id = 179;
    dcmSrc.makeAnonymous('1.2.3', anonDst);
    UPDATE anon_images SET dicom = anonDst WHERE id = 179;
    COMMIT;
END;
/
```

Anonimizacja powoduje usunięcie danych pozwalających na identyfikację pacjenta zarówno z atrybutów obiektu ORDDicom reprezentujących podstawowe atrybuty DICOM jak i z metadanych w formacie XML zawartych w atrybucie METADATA. Poniższe polecenie SELECT weryfikuje efekty anonimizacji:

```
SELECT id,
    extractValue(t.dicom.metadata,
        '/DICOM_OBJECT/*[@name="Patient's Name"]/VALUE',
        'xmlns=http://xmlns.oracle.com/ord/dicom/metadata_1_0')
    AS PATIENT_NAME,
    extractValue(t.dicom.metadata,
        '/DICOM_OBJECT/*[@name="Patient ID"]',
        'xmlns=http://xmlns.oracle.com/ord/dicom/metadata_1_0')
    AS PATIENT_ID
FROM anon_images t;
```

ID	PATIENT_NAME	PATIENT_ID
179	anonymous	anonymous

4. Inne rozszerzenia i zmiany w Oracle Multimedia 11g

4.1. Składowanie dużych obiektów binarnych

W wersji Oracle 11g zwiększony został maksymalny rozmiar dużych obiektów binarnych (BLOB) i wynosi on obecnie od 8TB do 128TB zależnie od rozmiaru bloku danych. Limit ten jest jednocześnie ograniczeniem na rozmiar pojedynczego obiektu Oracle Multimedia. Jest to znaczące zwiększenie limitu w porównaniu z Oracle 10.2, gdzie ograniczeniem była wartość 4GB.

Rewolucyjną zmianą w Oracle 11g rzutującą na funkcjonalność Oracle Multimedia jest wprowadzenie nowej generacji dużych obiektów o nazwie Oracle SecureFiles. Oracle SecureFiles oferuje funkcjonalność charakterystyczną dla współczesnych zaawansowanych systemów plików, w tym: wysoką wydajność, deduplikację, kompresję, szyfrowanie oraz zaawansowane mechanizmy utrzymywania logu. Oracle zaleca stosowanie SecureFile zamiast tradycyjnego BLOB w obiektach Oracle Multimedia choćby ze względu na zwiększoną wydajność. Należy podkreślić, że wprowadzenie SecureFiles było stosunkowo mało inwazyjne, z dużym naciskiem na zachowanie kompatybilności wstecz, co ilustruje poniższy przykład:

```
CREATE TABLE holiday_photos
(id NUMBER PRIMARY KEY,
 metaEXIF XMLTYPE,
 photo ORDSYS.ORDIMAGE)
LOB(photo.source.localdata) STORE AS SECUREFILE
XMLType COLUMN metaEXIF
XMLSCHEMA "http://xmlns.oracle.com/ord/meta/exif"
ELEMENT "exifMetadata";
```

W przedstawionym poleceniu CREATE TABLE zawarta została klauzula zlecająca składowanie zawartości binarnej obiektu Oracle Multimedia (w tym wypadku obrazu ORDImage) jako obiektu SECUREFILE. Klauzula ta jest jednak opcjonalna, a domyślnym typem danych w przypadku jej pominięcia jest BLOB, dzięki czemu skrypty SQL przygotowane dla wcześniejszych wersji serwera Oracle będą nadal uruchamiane poprawnie w Oracle 11g.

4.2. Wydajność

Oracle Multimedia 11g oferuje zwiększoną wydajność w stosunku do Oracle interMedia 10g w bardzo ograniczonym, ale istotnym podzbiórce funkcjonalności. W Oracle 11g znacząco poprawiono wydajność generacji miniatur obrazów, co jest najbardziej popularną operacją jeśli chodzi o przetwarzanie obrazów. Wzrost wydajności dotyczy obrazów w formatach TIFF i JPEG, a także obiektów DICOM, których treść obrazkowa jest w formacie JPEG.

4.3. Wyszukiwanie obrazów w oparciu o zawartość

Jako ważny obszar funkcjonalny Oracle interMedia aż do wersji 10.2 włącznie wskazywane było wyszukiwanie obrazów w oparciu o zawartość (ang. content-based image retrieval). Polega ono na realizowaniu testów podobieństwa obrazów na podstawie automatycznie ekstrahowanych cech wizualnych. Dla obrazów ORDImage uwzględniane były następujące cechy: kolor, kształt i tekstura z opcjonalnym uwzględnieniem lokalizacji obszarów zawierających te 3 cechy. Cechy wizualne obrazu były reprezentowane przez obiekty typu ORDImageSignature i testy podobieństwa obrazów były realizowane właśnie poprzez ten typ danych. W wersji 11g typ ORDImageSignature został zdeprecjonowany, a w związku z tym i cała funkcjonalność wyszukiwania w oparciu o zawartość dla obrazów przechowywanych jako ORDImage.

Nie został zdeprecjonowany interfejs zgodny ze standardem SQL/MM Still Image. Niemniej, funkcjonalność wyszukiwania w oparciu o zawartość przewidziana przez standard SQL/MM nie

pokrywa w całości funkcjonalności oferowanej przez ORDImageSignature, gdyż SQL/MM nie uwzględnia w testach podobieństwa kształtów występujących na obrazie. Ponadto, zdeprecjonowanie ORDImageSignature można traktować jako wyznacznik strategii Oracle odnośnie funkcjonalności serwera Oracle w obszarze multimediiów, gdyż praktyczne znaczenie specyfikacji SQL/MM Still Image nie jest duże.

Wycofanie się Oracle z wyszukiwania obrazów w oparciu o zawartość na poziomie funkcjonalności serwera bazy danych spowodowane jest niedoskonałością uniwersalnych mechanizmów w tym zakresie. Obecnie uważa się, iż użyteczne mechanizmy wyszukiwania obrazów w oparciu o zawartość muszą być specjalizowane dla konkretnych typów obrazów np. zdjęć twarzy czy zdjęć medycznych i dlatego przyjmuje się, że mechanizmy te powinny być implementowane na poziomie specjalistycznych aplikacji, wykorzystujących serwer bazy danych jedynie do składowania obrazów i obsługi metadanych.

5. Podsumowanie

Oracle Multimedia (dawniej interMedia) to cecha serwera bazy danych Oracle dostępna już od wersji 8.1.5. Jej funkcjonalność ulega ciągłym zmianom w odpowiedzi na potrzeby użytkowników i ogólne trendy w dziedzinie multimedialnych baz danych. Aż do wersji Oracle 10g Release 1 włącznie wydawało się, że zadaniem systemu zarządzania bazą danych w zakresie obsługi multimediiów powinno być dostarczenie mechanizmów możliwie ogólnych, abstrahujących od konkretnych zastosowań, a także położenie nacisku na wyszukiwanie w oparciu o zawartość na podstawie cech automatycznie ekstrahowanych z zapisanej cyfrowo binarnej zawartości multimedialnej. Te dwa postulaty z pewnością w dalszym ciągu są słuszne, niestety nie udało się odpowiadającej im funkcjonalności wdrożyć na poziomie pozwalającym na praktyczne zastosowania.

Zmiany w dwóch ostatnich wersjach serwera bazy danych Oracle (10.2 i 11.1) z jednej strony sankcjonują odwrót od wyszukiwania w oparciu o zawartość (zdeprecjonowanego w wersji 11.1), a z drugiej wskazują na zwiększoną rolę metadanych towarzyszących zawartości multimedialnej (zagnieżdżone metadane o zawartości obrazów od wersji 10.2). Ponadto, ważnym obszarem funkcjonalnym staje się obsługa obrazów medycznych (załączki w wersji 10.2, znacząco rozbudowane możliwości w wersji 11.1), co stanowi pierwszy przypadek funkcjonalności przeznaczonej dla jednej konkretnej, ale z pewnością ważnej dziedziny zastosowań systemów baz danych i aplikacji multimedialnych.

Bibliografia

- [1] Digital Imaging and Communications in Medicine (DICOM), <http://medical.nema.org/>
- [2] ISO/IEC 13249-5:2001, Information Technology – Database Languages – SQL Multimedia and Application Packages – Part 5: Still Image. ISO, 2001
- [3] Oracle interMedia Reference 10g Release 2 (10.2), June 2005
- [4] Oracle interMedia User's Guide 10g Release 2 (10.2), June 2005
- [5] Oracle Multimedia DICOM Developer's Guide 11g Release 1 (11.1), September 2007
- [6] Oracle Multimedia: Feature Overview, July 2007
- [7] Oracle Multimedia Reference 11g Release 1 (11.1), September 2007
- [8] Oracle Multimedia User's Guide 11g Release 1 (11.1), September 2007

