

# Raportowanie w XML dla zwykłych ludzi

## Oracle XML Publisher

Tomasz Traczyk



Wydział Elektroniki i Technik Informatycznych  
Politechnika Warszawska

## Raportowanie w XML dla zwykłych ludzi...

- ❖ Wprowadzenie
- ❖ Czym jest XML Publisher?
- ❖ Model danych
- ❖ Formatowanie – szablony
- ❖ Podsumowanie



T. Traczyk

Raportowanie w XML dla zwykłych ludzi... PLOUG'06

2

## Wprowadzenie

### XML odpowiedni do tworzenia raportów

- Elastyczny
  - można w nim reprezentować źródła danych o różnorodnym charakterze
  - pozwala łączyć wiele danych w większe całości
  - pozwala łączyć dane strukturalne z „wolnym” tekstem i treściami semistukturalnymi
- Obudowany odpowiednimi technologiami
  - XSLT + XSL-FO wydają się idealne do tworzenia raportów
    - » wielkie możliwości transformacji i formatowania
    - » liczne formaty wyjściowe
- Właściwy do przetwarzania w sieci
  - idealny do udostępniania zintegrowanych danych w sieci
  - łatwy do transformacji na HTML

### Użycie technologii XML

- Technologie
  - transformacja danych na XML
    - » narzędzia: np. XSQL, XSU, SQLX
  - XSLT
    - » dowolne transformacje XML
      - na HTML
      - na XSL-FO
    - » duży wybór procesorów
  - XSL-FO
    - » zaawansowany opis wyniku – sformatowanego dokumentu
    - » narzędzia: np. FOP
- Potrzebne umiejętności
  - znajomość w/w technologii i narzędzi
  - umiejętność ich integracji na serwerze aplikacyjnym

➤ Niezbędne wysokie kwalifikacje projektanta

- ❖ Wprowadzenie
- ❖ Czym jest XML Publisher?
- ❖ Model danych
- ❖ Formatowanie – szablony
- ❖ Podsumowanie



T. Traczyk

Raportowanie w XML dla zwykłych ludzi... PLOUG'06

3



T. Traczyk

Raportowanie w XML dla zwykłych ludzi... PLOUG'06

4

## Czym jest XML Publisher?

### Narzędzie do tworzenia raportów

- Dostępne
  - jako część Oracle E-Business Suite
  - jako samodzielne narzędzie XML Publisher Enterprise
- Łatwe w użyciu
  - tworzenie szablonów w MS Word
- Silne
  - wielkie możliwości kształtowania wyglądu raportu
    - » formatowanie niemal jak w RTF
    - » XSLT i większość XSL-FO
- Elastyczne
  - Źródła danych
    - » bazy danych – SQL
    - » pliki XML (przez http)
    - » Web Services
  - formaty wyjściowe raportów
    - » HTML
    - » PDF
    - » RTF
    - » Excel
    - » XML (same dane)

### „Portal” raportowy XMLP Enterprise

- Definiowanie raportów
- Zarządzanie
  - raportami i folderami
  - harmonogramem i dystrybucją wyników
  - użytkownikami i prawami dostępu
- Autoryzacja dostępu
  - własna
  - LDAP
  - SSO (*Single Sign On*)
- Dostęp do zdefiniowanych raportów
  - odnajdowanie
  - wywoływanie
  - określanie wartości parametrów
  - harmonogramowanie
  - wysyłka do miejsca przeznaczenia

## Jak to działa?

### „Architektura” narzędzia

- Serwer: XMLP Enterprise
  - aplikacja J2EE, np. na OC4J
- Klient:
  - przeglądarka WWW
  - Acrobat Reader
  - edytor tekstu zgodny z RTF
  - MS Excel
- Narzędzia pomocnicze
  - MS Word lub inny edytor RTF
  - XMLP Desktop
    - » dodatki do MS Word i MS Excel
  - Adobe Acrobat

### Składniki raportu

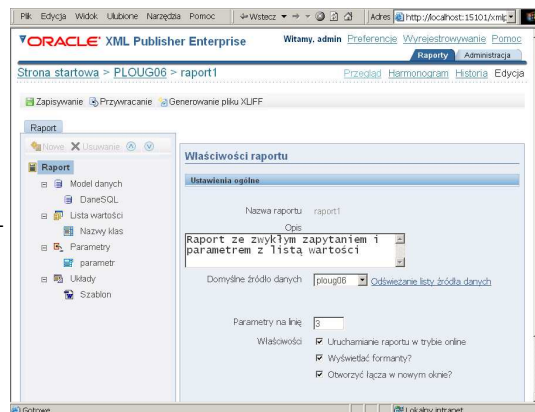
- Model danych
  - źródła danych (*data sources*)
  - zestawy danych (*data sets*)
  - definicje parametrów
- Szablony układu (*layout templates*)
  - może być kilka do jednego raportu
- Specyfikacje wersji językowych
  - pliki tłumaczeń
  - specjalne szablony specyficzne dla języka

## Jak się buduje raport

### Kroki budowania typowego raportu

- Za pomocą interfejsu WWW XMLP
  - utworzyć nowy raport i umieścić go w jednym z folderów
  - zdefiniować źródła danych
    - » parametry połączenia z bazą danych
  - zdefiniować parametry raportu
    - » ew. listy wartości
  - zdefiniować zestaw danych
    - » stworzyć zapytania SQL
      - ręcznie
      - za pomocą kreatora (typu QBE)
  - wyeksportować dane do pliku XML
- Za pomocą MS Word (ew. z dodatkiem Template Builder for Word)
  - zbudować szablony RTF
- Za pomocą interfejsu WWW XMLP
  - załadować szablon
  - określić prawa dostępu

- Ewentualnie zbudować wersje językowe
  - eksportować definicję i szablony do plików XLIFF
  - dokonać tłumaczenia
  - załadować pliki XLIFF



❖ Wprowadzenie

❖ Czym jest XML Publisher?

❖ Model danych

❖ Formatowanie – szablony

❖ Podsumowanie

## Model danych

### Definiuje

- Pochodzenie danych
- Sposób ich przekształcenia na wejściowy XML

### Budowa

- Złożony z 1..n deklaracji zestawów danych
- Możliwe jest połączenie danych ze wszystkich zestawów

### Zestaw danych (data set)

- Określają sposób pobrania i przekształcenia danych z jednego źródła

### Źródło danych (data source)

- Dla danych z baz danych definiuje się tzw. źródła danych
- Deklaracja źródła danych określa
  - parametry połączenia JDBC
  - lub nazwę JNDI (*Java Naming and Directory Interface*)

### Możliwe pochodzenie danych

- Bazy danych – przez zapytania SQL
  - dane są przekształcane na „postać kanoniczną” XSU
    - » ROWSET – ROW – *kołumny*
- Dokumenty XML
  - statyczne lub dynamiczne
  - dostęp przez protokół HTTP
  - można przekazywać parametry
- Usługi sieciowe *Web Services*
  - definiowane przez podanie
    - » URL do specyfikacji usługi w WSDL
    - » nazwy metody
    - » parametrów metody
  - muszą zwracać poprawny XML
- *Data Templates*
  - zapisane w XML deklaracje pobrania i połączenia danych z wielu źródeł SQL

## Model danych, c.d.

### Data Template

- Określa sposób połączenia i przetwarzania danych z 1-n źródeł SQL
- Definiuje
  - zapytania SQL
  - sposób połączenia danych z zapytań
    - » zmienne związane
    - » znacznik <link>
  - strukturę wynikowego XML
    - » zagnieżdżone grupy danych
- Można używać odwołań do PL/SQL w bazie
  - filtry dla grup danych
  - wyzwalacze dla raportu (przed / po)

### Buforowanie

- Pobrane dane mogą być buforowane
- Wykonanie raportu z różnymi szablonami/językami
  - nie wymaga ponownego pobrania danych
  - co odciąża bazę danych

### Parametry raportu

- Deklarowane w definicji raportu
  - nazwa, typ danych, długość
  - mogą mieć listę dopuszczalnych wartości
    - » statyczną
    - » dynamiczną – wynik zapytania SQL
      - zapytanie dla kolejnego parametru może zależeć od wartości poprzednich
- Wykorzystywane do
  - parametryzacji zapytań SQL (*bind variables*)
  - parametryzacji szablonu układu
  - przekazywania jako parametry żądań HTTP oraz *Web Services*

## Model danych – przykład

### Dane przykładowe

- Dwie tabele relacyjne w związku 1-n
- Plik XML z opisami klas tematycznych

KLASY_TEMATYCZNE	
ID_KLASA	A
NAZWA	A

PRZEDMIOTY				
ID_PRZEDMIOTU	A			
NAZWA	A			
ID_WYKLADOWCY	799			
LICZBA_GODZIN	799			
ID_KLASA	A			

### Źródło SQL

- Zwykle zapytanie SQL ze złączeniem
- Wynik zapytania jest „płaski”
  - jeden poziom w XML, bez hierarchii
- Pokazano użycie parametru jako zmiennej związanej
  - znak % nie działa dobrze (URL!)
  - znak \* jest zamieniany na NULL

```
select
  id_klasy, k.nazwa as klasa,
  id_przedmiotu, p.nazwa as przedmiot, liczba_godziny
from klasy_tematyczne k
join przedmioty p using (id_klasy)
where k.nazwa like '%||:parametr||%'
```

```
<ROWSET>
<ROW>
  <ID_KLASA>TEOR</ID_KLASA>
  <KLASA>Przedmioty teoretyczne</KLASA>
  <ID_PRZEDMIOTU>OTW</ID_PRZEDMIOTU>
  <PRZEDMIOT>Ogólna teoria wszystkiego</PRZEDMIOT>
  <LICZBA_GODZIN>10</LICZBA_GODZIN>
</ROW>
...
</ROWSET>
```

## Model danych – przykład, c.d.

### Źródło SQL + XML

- Zestawy danych
  - przedmioty: zapytanie SQL z operatorem CURSOR – wynik zachowuje hierarchię
  - konspekty: plik XML zawierający konspekty klas
- Pokazano połączenie danych
  - dane zostały połączone w jeden wejściowy XML

```
select id_klasy, k.nazwa as klasa,
  cursor (
    select id_przedmiotu, nazwa as przedmiot, liczba_godziny
    from przedmioty p
    where p.id_klasy=k.id_klasy
  ) as przedmioty
from klasy_tematyczne k
```

```
<?xml version = '1.0'
  encoding = 'windows-1250'?>
<KONSPEKT_KLAS>
  <KONSPEKT ID_KLASA="TEOR">
    <TEKST>Klasa obejmuje...</TEKST>
  </KONSPEKT>
  <KONSPEKT ID_KLASA="PRAK">
    <TEKST>Klasa obejmuje...</TEKST>
  </KONSPEKT>
</KONSPEKT_KLAS>
```

```
<DATA>
  <przedmioty>
    <ROW>
      <ID_KLASA>TEOR</ID_KLASA>
      <KLASA>Przedmioty teoretyczne</KLASA>
      <PRZEDMIOTY>
        <PRZEDMIOTY_ROW>
          <ID_PRZEDMIOTU>OTW</ID_PRZEDMIOTU>
          <PRZEDMIOT>Ogólna teoria wszystkiego</PRZEDMIOT>
          <LICZBA_GODZIN>10</LICZBA_GODZIN>
        </PRZEDMIOTY_ROW>
        ...
      </PRZEDMIOTY>
    </ROW>
    ...
  </przedmioty>
  <konspekty>
    <KONSPEKT_KLAS>
      <KONSPEKT ID_KLASA="TEOR">
        <TEKST>
          Klasa obejmuje ogólna teorię wszystkiego
          i wnioski z niej płynące.
        </TEKST>
      </KONSPEKT>
      ...
    </KONSPEKT_KLAS>
  </konspekty>
</DATA>
```

## Użycie Data Template

```
<dataTemplate name="przedmioty" version="1.0">
  <description>klasa i przedmioty</description>
  <parameters>
    <parameter name="parametr" datatype="character" />
  </parameters>
  <dataQuery>
    <sqlStatement name="klasy">
      <![CDATA[
        SELECT id_klasy as klasa, nazwa
        FROM klasy_tematyczne
        WHERE id_klasy = nvl(:parametr, id_klasy)
      ]]>
    </sqlStatement>
    <sqlStatement name="przedmioty">
      <![CDATA[
        SELECT id_przedmiotu, nazwa, liczba_godzin
        FROM przedmioty
        WHERE id_klasy = :klasa
      ]]>
    </sqlStatement>
  </dataQuery>
  <dataStructure>
    <group name="klasa" source="klasy">
      <element name="id" value="KLASA" />
      <element name="nazwa" value="NAZWA" />
      <group name="przedmiot" source="przedmioty">
        <element name="id" value="ID_PRZEDMIOTU" />
        <element name="nazwa" value="NAZWA" />
        <element name="godziny" value="LICZBA_GODZIN" />
      </group>
      <element name="suma_godzin" value="przedmiot.godziny"
        function="SUM()" />
    </group>
  </dataStructure>
</dataTemplate>
```

- Definicja Data Template w XML
- Dwa zapytania SQL
  - wyniki połączone przez zmienną związaną
- Zdefiniowana struktura wynikowa
  - zagnieżdżone grupy zachowujące hierarchię
  - element wyliczający sumę w grupie
- Pokazano użycie parametru jako *bind variable*

```
<przedmioty>
  <parameter>TEOR</parameter>
  <LIST_KLASA>
    <KLASA>
      <ID>TEOR</ID>
      <NAZWA>Przedmioty teoretyczne</NAZWA>
      <LIST_PRZEDMIOT>
        <PRZEDMIOT>
          <ID>OTW</ID>
          <NAZWA>Ogólna teoria wszystkiego</NAZWA>
          <GODZINY>10</GODZINY>
        </PRZEDMIOT>
        ...
      </LIST_PRZEDMIOT>
      <SUMA_GODZIN>15</SUMA_GODZIN>
    </KLASA>
  </LIST_KLASA>
</przedmioty>
```

- ❖ Wprowadzenie
- ❖ Czym jest XML Publisher?
- ❖ Model danych
- ❖ Formatowanie – szablony
- ❖ Podsumowanie

## Szablony RTF

### Zawartość szablonów

- Pliki RTF tworzone np. za pomocą MS Word
  - wspomaganie: *Template Builder for Word*
- Zawierają
  - niezależne od danych elementy dokumentu
    - » sekcje, szpalty, teksty itp.
    - » nagłówki, stopki
    - » tabele
    - » elementy formatowania
      - czcionki, kolory, ramki itp.
      - format liczb i dat
  - grupy powtarzane (*repeating groups*)
    - » sterujące iteracjami przez dane
    - » zagnieżdżane, zgodnie z
      - hierarchią danych w XML
      - „wymuszonym” grupowaniem
  - pola danych (*placeholders*)
  - inne znaczniki sterujące, np. sortowaniem

### Znaczniki XMLP

- Elementy sterujące są umieszczane w dokumencie w postaci znaczników
  - zwykły tekst
    - <?nazwa: parametr; parametr... ?>
  - pola formularzy MS Word (*text form fields*)
    - » tekst j/w wpisany we własność „Tekst pomocy”
      - dla paska stanu
      - lub dla klawisza F1
    - » pomocniczy opis wpisany we własność „Domyślny tekst”
- Elementy definiujące grupy
  - mają znacznik początkowy i końcowy
  - znaczniki te okalają zawartość grupy
  - zagnieżdżenie znaczników odzwierciedla hierarchię grup

## Szablony układu – przykład

### Szablony RTF

- Grupy powtarzane
  - umieszczone
    - » w wolnym tekście
    - » w tabelach
  - zagnieżdżane
- Pola wyświetlające dane
  - nazwa zgodna z elementem XML
- Formatowanie pól i opisów – zachowany format z szablonu

### Szablon z danymi SQL

- Do zapytania SQL zełączeniem
  - brak hierarchii – konieczne grupowanie
- Pokazano
  - dwie zagnieżdżone grupy
    - » odtwarzają hierarchię danych
  - sortowanie
  - sumowanie w grupie

```
<ROWSET>
  <ROW>
    <ID_KLASA>TEOR</ID_KLASA>
    <KLASA>Przedmioty teoretyczne</KLASA>
    <ID_PRZEDMIOTU>OTW</ID_PRZEDMIOTU>
    <PRZEDMIOT>Ogólna teoria wszystkiego</PRZEDMIOT>
    <LICZBA_GODZIN>10</LICZBA_GODZIN>
  </ROW>
  ...
</ROWSET>
```

```
<?for-each-group:ROW:/ID_KLASA?><?sort:ID_KLASA?>
Klasa: <?ID_KLASA?>
Nazwa klasy: <?KLASA?>


| Id przedmiotu                               | Nazwa przedmiotu | Liczba godzin                         |
|---------------------------------------------|------------------|---------------------------------------|
| <?for-each-current-group?><?ID_PRZEDMIOTU?> | <?PRZEDMIOT?>    | <?LICZBA_GODZIN?>                     |
| <?sort:ID_PRZEDMIOTU?>                      | <?end-for-each?> | <?end-for-each?>                      |
| RAZEM                                       |                  | <?sum (current-group)/LICZBA_GODZIN?> |


<?end-for-each-group?>
```

↓

Klasa: PRAK		
Nazwa klasy: Przedmioty praktyczne		
Id przedmiotu	Nazwa przedmiotu	Liczba godzin
APOT	Aspekty praktyczne ogólnych teorii	3
RAZEM		3

Klasa: TEOR		
Nazwa klasy: Przedmioty teoretyczne		
Id przedmiotu	Nazwa przedmiotu	Liczba godzin
OTW	Ogólna teoria wszystkiego	10
STNR	Szczególna teoria niektórych rzeczy	5
RAZEM		15

### Szablon z danymi XML

- Do danych z hierarchią
  - bez dodatkowego grupowania
- Pokazano
  - pobieranie danych XML w kontekście przez XPath
  - proste sumowanie

```
<?for-each:ROW?><?sort:ID_KLASY?>
Klasa: <?ID_KLASY?>
Nazwa klasy: <?KLASA?>
<?for-each:/KONSPEKT[@ID_KLASY=current()/ID_KLASY]?>
<?TEKST?>
<?end for-each?>
```

Id przedmiotu	Nazwa przedmiotu	Liczba godzin
<?for-each:PRZEDMIOTY_ROW?><?PRZEDMIOT?>		<?LICZBA_GODZIN?>
<?ID_PRZEDMIOTU?>		<?end for-each?>
RAZEM		<?sum (LICZBA_GODZIN)?>

```
<?end for-each?>
```

```
<DATA>
<przedmioty>
<ROW>
<ID_KLASY>TEOR</ID_KLASY>
<KLASA>Przedmioty teoretyczne
<PRZEDMIOTY>
<PRZEDMIOTY_ROW>
<ID_PRZEDMIOTU>OTW</ID_PRZEDMIOTU>
<PRZEDMIOT>Ogólna teoria
<LICZBA_GODZIN>10</LICZBA_GODZIN>
</PRZEDMIOTY_ROW>
</PRZEDMIOTY_ROW>
</PRZEDMIOTY>
</ROW>
...
</przedmioty>
<konspekty>
<KONSPEKT_KLAS>
<KONSPEKT_ID_KLASY="TEOR">
<TEKST>
Klasa obejmuje ogólną teorię i wnioski z niej płynące.
</TEKST>
</KONSPEKT>
...
</KONSPEKT_KLAS>
</konspekty>
</DATA>
```

**Klasa: PRAK**  
Nazwa klasy: Przedmioty praktyczne  
Klasa obejmuje ogólną teorię wszystkiego i wnioski z niej płynące.

Id przedmiotu	Nazwa przedmiotu	Liczba godzin
APOT	Aspekty praktyczne ogólnych teorii	3
RAZEM		3

**Klasa: TEOR**  
Nazwa klasy: Przedmioty teoretyczne  
Klasa obejmuje zastosowania ogólnej teorii wszystkiego i teorii pochodnych w praktyce gospodarczej i administracyjnej, ze szczególnym uwzględnieniem problemów integracji europejskiej.

Id przedmiotu	Nazwa przedmiotu	Liczba godzin
OTW	Ogólna teoria wszystkiego	10
STNR	Szczególna teoria niektórych rzeczy	5
RAZEM		15

### Wspierane formatowanie MS Word

- Stronicowanie i numeracja stron
- Justowanie tekstu
- Tabele (także zagnieżdżone)
- Sekcje i szpalty
- Strony parzyste i nieparzyste
- Generowanie spisów treści

### Paginacja

- Wymuszanie zmiany stron za pomocą znaczników XMLP
- Zmiany numeracji stron
- Żywa pagina (nagłówki, stopki)
  - odwołania do pól danych w żywej paginie

### Formatowanie warunkowe

- Warunkowa prezentacja fragmentów raportu
- Sterowanie formatowaniem (kolory itp.)

### Formatowanie liczb i dat

- Formaty MS Word
- Deskryptory formatu Oracle

### Obliczenia

- Podsumowania grup
- Podsumowania stron
- Sumy biegące
- Wyrażenia SQL
 

```
<?xdoxf:wyrazenieSQL?>
```
- Wyrażenia XSLT
 

```
<?xdoxslt:wyrazenieSQL?>
```

### Dynamiczne tabele

- Dynamicznie tworzone kolumny
- Tabele typu *cross-tab* („raport macierzowy”)

### XSLT i XSL-FO w szablonach

- Podzbiór instrukcji XSLT ma odpowiedniki w postaci znaczników XMPL
- W polach *Form Fields* można umieścić
  - skrypty XSLT – pełny zakres
  - fragmenty XSL-FO – znaczny podzbiór standardu

## Grafika i wykresy

### Elementy graficzne

- Stałe elementy umieszcza się w szablonie
- Grafika sterowana danymi:
  - położenie obrazu
  - wielkość
  - liczba powtórzeń
  - nachylenie
  - tekst umieszczony na obrazie

### Wykresy

- Tworzone za pomocą Oracle BI Beans
  - rysują wykresy na podstawie opisu w XML
- Sposób definiowania
  - tworzy się obiekt graficzny-zaślepkę
    - » w odpowiednim miejscu
    - » o odpowiednim rozmiarze
  - we właściwości „Tekst alternatywny” umieszcza się kod XSLT tworzący opis wykresu
- Proste wykresy można tworzyć kreatorem z dodatku Template Builder for Word
- Elementy graficzne mogą być generowane w formacie SVG

## Szablony PDF

### Sposób tworzenia

- Za pomocą Adobe Acrobat na dokumencie PDF umieszcza się pola (*form fields*)
  - odpowiadające danym
  - definiujące grupy i pola powtarzane
- Można tak „spreparować” każdy dokument PDF
  - wytworzony za pomocą Distillera lub podobnego narzędzia
  - uzyskany z sieci itp.
  - zeskanowany

### Przeznaczenie

- Wypełnianie formularzy urzędowych
  - wierne odtworzenie wyglądu formularza
  - automatyczne wstawianie danych

### Repozytorium XMLP

- Zawiera
  - pliki definiujące raporty (.xdo)
  - szablony układów (.rtf oraz .pdf)
  - tłumaczenia XLIFF (.xlf)
- Postać
  - foldery w systemie plików – gałąź xmlpserver/Reports
  - XML Repository w XML DB

### Pliki konfiguracyjne XMLP

- Pliki XML
- Definiują m.in.
  - źródła danych
  - użytkowników, role, uprawnienia
  - serwery druku i inne miejsca przeznaczenia raportów
  - własności generowanych PDF-ów
  - dodatkowe czcionki
  - parametry serwera
    - » wielkość buforów
    - » sposób dostępu do LDAP

### Interfejsy programistyczne XMLP

- Funkcjonalność składników XMLP jest dostępna przez API
- Dostępne API (wybór)
  - *PDF Form Processing Engine* – tworzy raporty PDF
  - *RTF Processor* – przekształca szablony w XSL
  - *FO Engine* – tworzy wynikowe dokumenty na podstawie XSL i danych XML
  - *Document Processor Engine* – steruje wsadowym przetwarzaniem i wysyłką raportów
  - *Data Engine* – tworzy
    - » XML-owe reprezentacje danych na podstawie modelu danych
    - » schematy XML Schema
    - » domyślne szablony RTF



❖ Wprowadzenie

❖ Czym jest XML Publisher?

❖ Model danych

❖ Formatowanie – szablony

❖ Podsumowanie



## XML Publisher a inne technologie raportowania

### XMLP a Oracle Reports

- Model danych, parametry – podobne
  - Źródła danych – szerszy wybór w XMLP
  - Zarządzanie raportami, harmonogram, dostarczanie – lekka przewaga Reports Server
  - Wiele układów do jednego raportu – tylko XMLP
  - Tłumaczenia – tylko XMLP
  - Tworzenie szablonu układu – dużo łatwiejsze w XMLP
  - Tworzenie rozbudowanych opracowań – tylko XMLP
  - Dostosowywanie gotowych wzorców RTF i PDF – tylko XMLP
  - Integracja z Oracle Forms, generatorami Designera itp. – tylko Reports
- W większości zastosowań XMLP może zastąpić Oracle Reports
- Do niektórych zastosowań nadaje się znacznie lepiej

### XMLP a Oracle Discoverer

- Budowanie zapytań i raportów *ad hoc*
  - w XMLP mniejsze możliwości, ale możliwe
- Analiza wielowymiarowa
  - w XMLP możliwa, choć bez wyrafinowanych możliwości czy specjalnego wsparcia dla VLDB
- Wizualizacja wyników
  - w XMLP duże możliwości, choć niezbyt łatwa w użyciu

### XMLP a „ręczne” użycie technologii XML

- Możliwości funkcjonalne
  - w XMLP nieco ograniczone, ale mało istotnie
- Wydajność
  - zapewne przewaga XMLP
- Niezawodność i wsparcie techniczne
  - wyraźna przewaga XMLP
- Łatwość użycia
  - miazdząca przewaga XMLP



## Podsumowanie

### Zalety XML Publisher

- Pomysłowa metoda tworzenia szablonów
  - łatwa: wystarczy niewiele więcej niż opanowanie użycia MS Word
  - pozwalająca świetnie wykorzystać istniejące wzory dokumentów
- Wiele szablonów i wersji językowych do jednego modelu danych
  - dobrze odpowiada potrzebom występującym w praktyce
- Ogromne możliwości kształtowania wyglądu raportu
  - niemal pełne możliwości edytora tekstu
  - rozbudowane możliwości graficzne
- Możliwość tworzenia rozbudowanych opracowań
  - łączenie danych z wielu źródeł o różnym charakterze
  - wiele różnych zestawień w jednej całości
  - z elementami porządnego składu, np. żywą paginą, spisami treści itp.

- API umożliwiające programową kontrolę procesu raportowania
  - możliwość włączenia funkcji XMLP do systemów o niebanalnej logice
- Wbudowane narzędzie do analizy wielowymiarowej

### Wady XML Publisher

- Mało wyczerpująca dokumentacja
- Brak wsparcia dla standardu OpenDocument
- Ograniczenia w wykorzystaniu *Web Services*
- „Jednokierunkowe” kreatory
- Brak integracji ze starszymi narzędziami
- Obecne warunki licencjonowania produktu



**Oracle XML Publisher – raportowanie w XML dla „zwykłych ludzi”**

- Właściwe wykorzystanie XML
  - w pełni zgodne z intencjami twórców XML, XSLT i XSL-FO
  - w znacznym stopniu zgodne ze standardami
- Narzędzie przyjazne dla „zwykłego człowieka”
  - łatwe tworzenie raportów
    - » „użytkownik końcowy” może samodzielnie tworzyć lub dostosowywać układy raportów
    - » średnio zaawansowany użytkownik może tworzyć całe raporty
  - łatwe korzystanie z raportów
    - » interfejs WWW
  - łatwe zarządzanie raportami
- Narzędzie przyjazne dla informatyka
  - zgodne ze standardami
  - udostępniające wyrafinowane konstrukcje w szablonach
  - pozwalające korzystać z możliwości XSLT, XSL-FO i SQL
  - dostępne przez liczne API



---

Raportowanie w XML dla zwykłych ludzi  
Oracle XML Publisher

