

Selektywne rozpowszechnianie informacji z relacyjnych baz danych w postaci dokumentów XML-owych

Tadeusz Pankowski

Katedra Automatyki, Robotyki i Informatyki, Politechnika Poznańska

pankowski@put.poznan.pl

Streszczenie. W pracy omówiono problem rozpowszechniania danych relacyjnych w postaci dokumentów XML-owych. Potrzeba taka wynika z faktu, że wciąż jeszcze większość danych w systemach informatycznych przechowywana jest w relacyjnych bazach danych. Wynika to z faktu, że systemy te stanowią niezawodne środowisko pamiętania i przetwarzania danych, przede wszystkim z punktu widzenia systemów transakcyjnych (OLTP), a ostatnio także analitycznych (OLAP). Okazuje się jednak, że z punktu widzenia powszechnego rozpowszechniania informacji (głównie z wykorzystaniem Internetu) bezpośrednie korzystanie z relacyjnej bazy danych jest niewygodne (dane pamiętane są w postaci trudnej do interpretacji), a często niemożliwe ze względu na nieprzystawalność platform sprzętowo-programowych funkcjonujących w sieci. Istnieje więc konieczność korzystania ze standardów umożliwiających wymianę informacji w formacie niezależnym od stosowanego sprzętu i oprogramowania. Takim uznanym standardem stał się język XML. Jednocześnie prezentacja (wizualizacja) tej informacji powinna odbywać się na komputerze użytkownika końcowego w tym środowisku sprzętowo-programowym, którym dysponuje użytkownik, a więc najczęściej za pomocą przeglądarki internetowej. W pracy omawiana jest metoda przekształcania danych relacyjnych do danych XML-owych z wykorzystaniem rozszerzonych możliwości języka SQL (zaimplementowanych w systemie MS SQL Server 2000). Wskazano także na potrzeby dalszego rozszerzenia tych możliwości i osadzenia ich w szerszym kontekście systemów integracji i rozpowszechniania informacji.

1. Wprowadzenie

Język XML [1, 8] stał się powszechnie akceptowanym standardem wymiany wszelkiego rodzaju informacji (naukowej, technicznej, ekonomicznej i innej) poprzez Internet. Jednak obecnie, jak i w najbliższej przyszłości, większość informacji przechowywana jest i będzie w systemach relacyjnych baz danych. Ważnym problemem staje się więc opracowywanie metod i narzędzi transformacji wybranych fragmentów relacyjnych baz danych do postaci dokumentów XML-owych. Tak przygotowane dokumenty mogą być publikowane w Internecie i mogą mieć do nich powszechny dostęp wszyscy użytkownicy niezależnie od wykorzystywanej przez nich platformy sprzętowej i programowej. Pozwala to nie tylko na proste udostępnianie danych, ale może stanowić podstawę budowy systemów organizacji współpracy w sieci WWW.

Istotnym problemem przy transformacji danych relacyjnych do danych XML-owych jest to, że metody strukturalizowania w obu tych podejściach różnią się zasadniczo. W relacyjnej bazie danych mamy do czynienia z płaskimi, regularnymi tabelami wypełnionymi prostymi wartościami. Uzyskanie pełnej informacji o semantycznie spójnych, ale złożonych jednostkach świata rzeczywistego, wymaga łączenia wielu tabel bazy danych. W przypadku danych XML-owych mamy do czynienia ze strukturami hierarchicznymi, w których występują zagnieżdżenia. Znaczenie poszczególnych elementów struktury oddawane jest za pomocą znaczników. Jeden dokument XML-owy może mieć bardzo złożoną strukturę, ale jednocześnie może opisywać w naturalny sposób złożone jednostki świata rzeczywistego. Dokument taki staje się jednak zrozumiały dla użytkownika najczęściej dopiero wtedy, gdy zostanie odpowiednio wyświetlony przez przeglądarkę.

Celem pracy jest przedstawienie metody konwersji danych relacyjnych do postaci dokumentów XML-owych. Szczególna uwaga zwrócona jest na przypadek, gdy konwersja ma objąć zbiór tabel pozostających ze sobą w pewnych związkach semantycznych wynikających z faktu, że łącznie odnoszą się do określonego, spójnego fragmentu rzeczywistości. Dyskutowane są:

- możliwości relacyjnych serwerów baz danych w zakresie wspomaganie procesu konwersji: dane relacyjne - dokumenty XML;
- proponowane w literaturze rozszerzenia funkcjonalności serwerów baz danych (języka SQL) w zakresie wspomaganie procesu konwersji;

- własne rozwiązania i propozycje dotyczące metod i narzędzi związanych z selektywnym rozpowszechnianiem zawartości relacyjnych baz danych w postaci dokumentów XML-owych.

Współczesne serwery relacyjnych baz danych włączają w zestaw swoich standardowych operacji funkcje związane z operowaniem na danych XML-owych. W systemie MS SQL Server 2000 są to na przykład: SELECT ... FOR XML ... – określa, że wynik zapytania udostępniany jest jako dokument XML-owy; OPENXML umożliwia otwieranie dokumentu XML-owego i przekształcenie go do pewnej reprezentacji tabelarycznej.

Wspomniane powyżej funkcje są jednak niewystarczające i wprowadzane są doraźnie przez poszczególnych producentów serwerów baz danych. Stąd też w literaturze przedmiotu intensywnie dyskutowane są propozycje rozwiązania tego problemu [2, 3, 6, 7]. Zwraca się przy tym uwagę na dwa zasadnicze problemy związane z publikowaniem danych relacyjnych w postaci dokumentów XML. Pierwszym z nich jest potrzeba zdefiniowania *języka* do specyfikacji konwersji danych relacyjnych do dokumentów XML, a drugi dotyczy efektywnej *implementacji* procesu konwersji. Wspomniany język powinien w szczególności umożliwić określanie sposobu strukturalizowania i przypisywania znaczników danym relacyjnym pobieranym z jednej lub z wielu tabel i przekształcanym w hierarchiczny dokument XML-owy.

Prowadzone w naszym zespole prace nad integracją heterogenicznych źródeł danych [4, 5] obejmują również problemy będące przedmiotem niniejszego referatu. W szczególności modele i algorytmy opracowane w projekcie bazy danych obiektów częściowo etykietowanych (stanowiącej repozytorium integrujące dane heterogeniczne) mają znaczenie dla efektywnej implementacji procesów konwersji między danymi relacyjnymi, a dokumentami XML-owymi. W tej pracy ograniczamy się jedynie do przedstawienia metody transformacji danych relacyjnych do dokumentów XML-owych, przy czym metoda ta oparta jest na możliwościach systemu SQL Server 2000.

2. Zastosowanie relacyjnych baz danych do zarządzania dokumentami XML-owymi

Mówiąc o zastosowaniu systemów relacyjnych do zarządzania danymi XML-owymi mamy na myśli co najmniej dwa zagadnienia:

1. przechowywanie i przetwarzanie dokumentów XML-owych w relacyjnej bazie danych,
2. rozpowszechnianie (selektywne) danych relacyjnych w postaci dokumentów XML-owych.

W każdym z tych przypadków istnieje szansa na wykorzystanie wielkich możliwości i niezawodności systemów zarządzania relacyjnymi bazami danych.

Wykorzystanie relacyjnego systemu zarządzania bazą danych do przechowywania, przetwarzania danych XML-owych ma wiele zalet. Systemy relacyjne są techniką dojrzałą, sprawdzoną i rozwijaną od wielu lat; dostarczają szereg przydatnych rozwiązań dotyczących między innymi zarządzania pamięcią, zarządzania współbieżnością, odtwarzania, wersjonowania. Jednak ze względu na specyfikę danych semistrukturalnych ich składowanie i przetwarzanie w relacyjnej, a więc strukturalnej, bazie wymaga wprowadzenia do tej bazy szeregu rozszerzeń funkcjonalnych. Konieczne jest między innymi:

1. opracowanie schematu relacji odpowiedniego do składowania danych semistrukturalnych,,
2. translacja zapytań z języka semistrukturalnego do relacyjnego,
3. optymalizacja zapytań,
4. utrzymywanie integralności i spójności.

Podobnie w przypadku rozpowszechniania danych relacyjnych w postaci dokumentów XML-owych. Podstawowym problemem, jaki pojawia się na tym etapie jest uwzględnienie takich cech danych XML-owych jak:

- dowolna głębokość grafu reprezentującego ich strukturę,
- atrybuty wielowartościowe i rekursje,
- brak typów,
- konieczność uwzględnienia kolejności elementów,
- referencje.

Możliwość rozpowszechniania danych relacyjnych w postaci dokumentów XML-owych ma jednak bardzo istotne zalety:

- rozpowszechnianie informacji staje się naprawdę powszechne, uwalniamy się od ograniczeń sprzętowo-programowych,

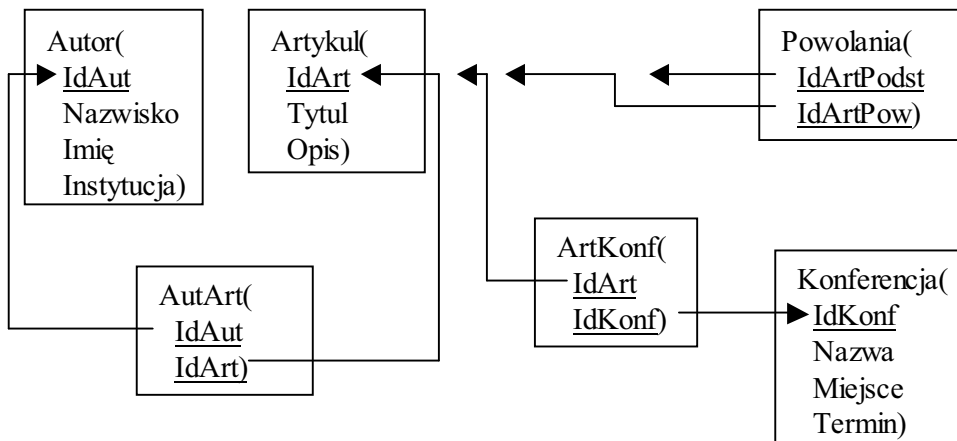
- dane mogą być strukturyzowane w postaci najbardziej wygodnej dla użytkownika – zarówno sposób strukturyzowanie danych, jak i sposób ich prezentacji pozostaje w zakresie decyzji użytkownika,
- przekazywanie danych, ich ochrona oraz wizualizacja odbywają się za pomocą standardowych środków.

Realizacja omówionych powyżej celów wymaga rozszerzenia funkcjonalności komercyjnych systemów relacyjnych baz danych o cechy wspomagające udostępnianie danych w postaci dokumentów XML-owych.

3. Metod transformacja danych relacyjnych do postaci dokumentów XML

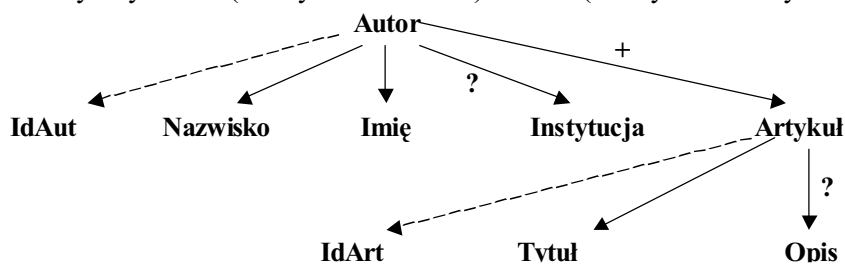
Przedstawiona w tym rozdziale metoda transformacji danych relacyjnych do dokumentów XML-owych zorientowana jest na możliwości systemu MS SQL Server 2000, a w szczególności wykorzystuje rozszerzenie wyrażenia SELECT o frazę FOR XML.

Informacje w relacyjnej bazie danych przedstawione są w postaci znormalizowanych relacji (tabel), między którymi mogą być określone zależności w postaci tzw. *więzów referencyjnych* oddają-



Rys. 1. Przykład schematu relacyjnej bazy danych

cych powiązania między *kluczami obcymi* tabeli podrzędnej i *kluczami głównymi* tabel nadrzędnych. Na rysunku 1 atrybuty IdAut (identyfikator autora) i IdArt (identyfikator artykułu) są kluczami obcy-



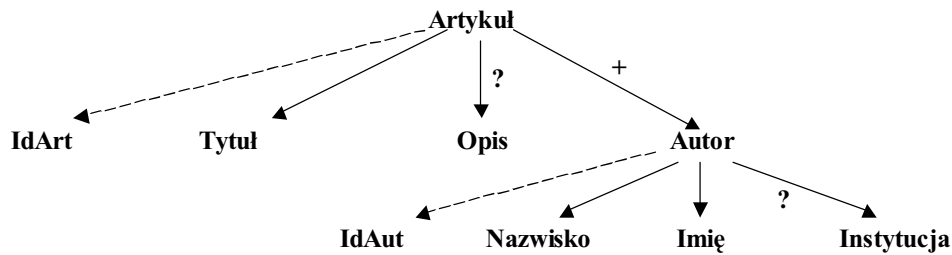
```

<!ELEMENT Autor(Nazwisko, Imię, Instytucja?, Artykuł+)>
<!ELEMENT Nazwisko (#PCDATA)>
<!ELEMENT Imię (#PCDATA)>
<!ELEMENT Instytucja (#PCDATA)>
  <!ATTLIST Autor IdAut ID #REQUIRED>
<!ELEMENT Artykuł (Tytuł, Opis?)>
<!ELEMENT Tytuł (#PCDATA)>
<!ELEMENT Opis ANY>
  <!ATTLIST Artykuł IdArt ID #REQUIRED>

```

Rys. 2. Graf schematy dokumentu XML i jego definicja DTD opisujące hierarchiczną strukturę danych o autorach i ich artykułach

mi w tabeli AutArt. Więzy referencyjne wiążą je z kluczami głównymi odpowiednio IdAut tabeli Autor i IdArt tabeli Artykuł. Uwzględnienie tego typu zależności ma istotne znaczenie w procesie transformacji do dokumentów XML-owych.



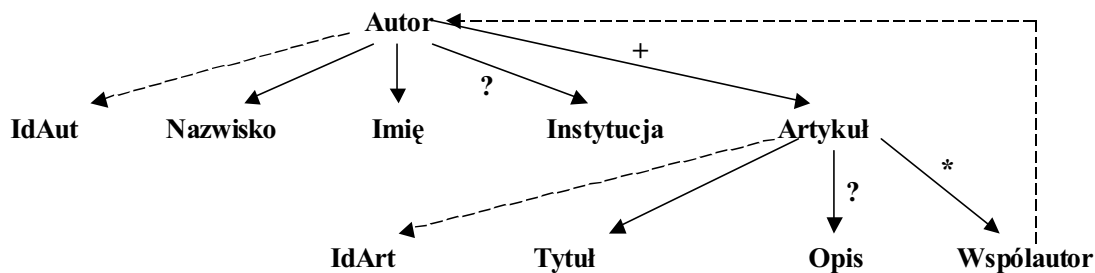
```

<!ELEMENT Artykuł (Tytuł, Opis?, Autor+)>
<!ELEMENT Tytuł (#PCDATA)>
<!ELEMENT Opis ANY>
  <!ATTLIST Artykuł IdArt ID #REQUIRED>
<!ELEMENT Autor(Nazwisko, Imie, Instytucja?)>
<!ELEMENT Nazwisko (#PCDATA)>
<!ELEMENT Imię (#PCDATA)>
<!ELEMENT Instytucja (#PCDATA)>
  <!ATTLIST Autor IdAut ID #REQUIRED>

```

Rys. 3. Graf schematy dokumentu XML i jego definicja DTD opisujące hierarchiczną strukturę danych o artykułach i ich autorach

Dla uproszczenia ograniczymy nasze rozważania do tabel Autor, Artykuł, AutArt. Użytkownik może żądać prezentacji zawartej w nich informacji w rozmaitej postaci, na przykład: dane o wybranych autorach i ich publikacjach, dane o wybranych artykułach i ich autorach, dane o autorach, artykułach i współautorach artykułów. Można żądać prezentacji informacji o autorach i artykułach z punktu widzenia instytucji zatrudniającej autorów lub z punktu widzenia zawartości (opisu) artykułów. Organi-



```

<!ELEMENT Autor(Nazwisko, Imie, Instytucja?, Artykuł+)>
<!ELEMENT Nazwisko (#PCDATA)>
<!ELEMENT Imię (#PCDATA)>
<!ELEMENT Instytucja (#PCDATA)>
  <!ATTLIST Autor IdAut ID #REQUIRED>
<!ELEMENT Artykuł (Tytuł, Opis?, Współautor*)>
<!ELEMENT Tytuł (#PCDATA)>
<!ELEMENT Opis ANY>
<!ELEMENT Współautor EMPTY>
  <!ATTLIST Artykuł IdArt ID #REQUIRED>
  <!ATTLIST Współautor IdAut IDREF IMPLIED>

```

Rys. 4. Graf schematy dokumentu XML i jego definicja DTD opisujące hierarchiczną strukturę danych o autorach, artykułach i współautorach artykułów

zacja i sposób prezentacji w tych przypadkach różnią się istotnie. Na rysunkach 2, 3 i 4 przedstawiamy postaci graficzne struktur dokumentów XML-owych oraz ich definicję w postaci dokumentów DTD.

Przedstawione definicje DTD podają strukturę elementów (frazę !ELEMENT), oraz ich atrybuty (frazę !ATTLIST). Składowe elementów mogą być opcjonalne (?), mogą występować zero lub wiele razy (*) lub co najmniej jeden raz (+). Niektóre nie mają składowych (EMPTY). Atrybuty mogą być jednoznaczными identyfikatorami (ID) lub odwołaniami do identyfikatorów (IDREF), dzięki czemu tworzone są struktury rekurencyjne. Słowo kluczowe ANY oznacza, że zawartością pola może być dalej nie opisywany dokument XML-owy.

Na podstawie definicji relacyjnej bazy danych oraz na podstawie wymagań określonych w definicji DTD, system powinien być w stanie wygenerować oczekiwany dokument XML. Generowanie to poprzedzone jest zainicjowaniem odpowiednich operacji w systemie relacyjnym – im bardziej potrafimy wykorzystać standardowe możliwości serwera, tym większej możemy oczekiwać efektywności przetwarzania.

Zilustrujemy procedurę takiego przetwarzania opierając się na przykładzie przedstawionym na rysunku 5.

Autor				Artykuł			AutArt	
IdAut	Nazwisko	Imię	Instytucja	IdArt	Tytuł	Opis	IdAut	IdArt
au1	Nowak	Jan	PP	ar1	tyt1	opis1	au1	ar1
au2	Wiech	Adam	PW	ar2	tyt2	opis2	au1	ar2
au3	Lata	Piotr	pp	ar3	tyt3	opis3	au1	ar3
							au2	ar1
							au2	ar2
							au3	ar1

Rys. 5. Przykładowy stan fragmentu relacyjnej bazy danych o schemacie z rysunku 1

Przykład wyrażenia SQL-owego generującego oczekiwany dokument XML-owy przedstawione jest na rys. 6. Właściwe zrozumienie sposobu działania tego wyrażenia wymaga odwołania się do tabeli 1. Podaje ona strukturę pewnej pośredniej tabeli w procesie transformacji – tabeli uzyskanej bez uwzględnienia frazy FOR XML. Nazwy kolumn tej tabeli podają sposób strukturyzowania danych w procesie tworzenia dokumentu XML. Kolumna Tag podaje poziom zagnieżdżenia elementu, a Parent rodzica tego elementu (NULL mówi, że element jest korzeniem). W przypadku nazw pozostałych kolumn, np. Artykuł!2!Tytuł!Element, występują trzy lub cztery elementy rozdzielone wykrzyknikiem: Artykuł oznacza tabelę bazową, 2 podaje poziom zagnieżdżenia (odpowiada wartości z kolumny Tag), trzeci element, Tytuł, użyty jest jako znacznik XML-owy, a czwarty element podaje sposób traktowania pozycji w kolumnie – słowo Element poleca traktowanie go jako elementu, a gdy brak czwartej składowej, to pozycja traktowana jest jako atrybut.

Tab. 1. Postać tabeli transformacji uzyskanej w trakcie realizacji wyrażenia z rys. 6

Tag	Parent	Autor!1! IdAut	Autor!1! Nazwisko !Element	Autor!1! Imię !Element	Autor!1! Instytucja !Element	Artykuł!2! IdArt	Artykuł!2! Tytuł !Element	Artykuł!2! Opis !Element
1	NULL	au1	Nowak	Jan	PP	NULL	NULL	NULL
2	1	au1	NULL	NULL	NULL	ar1	tyt1	opis1
2	1	au1	NULL	NULL	NULL	ar3	tyt3	opis3
1	NULL	au2	Wiech	Adam	PW	NULL	NULL	NULL
2	1	au2	NULL	NULL	NULL	ar1	tyt1	opis1
2	1	au2	NULL	NULL	NULL	ar2	tyt2	opis2
1	NULL	au3	Lato	Piotr	PP	NULL	NULL	NULL
2	1	au3	NULL	NULL	NULL	ar1	tyt1	opis1
2	1	au3	NULL	NULL	NULL	ar2	tyt2	opis2

Na rysunku 6 podana jest wyrażenie SQL-owe tworzące dokument XML-owy według wymagań sprecyzowanych na rysunku 3.

```
SELECT 1 as Tag, NULL as Parent,
       Autor.IdAut as [Autor!1!IdAut],
       Autor.Nazwisko as [Autor!1!Nazwisko!Element],
       Autor.Imie as [Autor!1!Imie!Element],
       Autor.Instytucja as [Autor!1!Instytucja!Element],
       null as [Artykuł!2!IdArt],
       null as [Artykuł!2!Tytuł!Element],
       null as [Artykuł!2!Opis!Element]
FROM   Autor
UNION ALL
SELECT 2, 1,
       Autor.IdAut, null, null, null, Artykuł.IdArt, Tytuł, Opis
FROM   Autor, Artykuł, AutArt
WHERE  AutArt.IdAut=Autor.IdAut and AutArt.IdArt=Artykuł.IdArt
ORDER BY [Autor!1!IdAut]
FOR XML EXPLICIT
```

Rys. 6. Wyrażenie w języku SQL (z frazą FOR XML) tworzące dokument XML-owy zgodnie z wymaganiami z rysunku 3

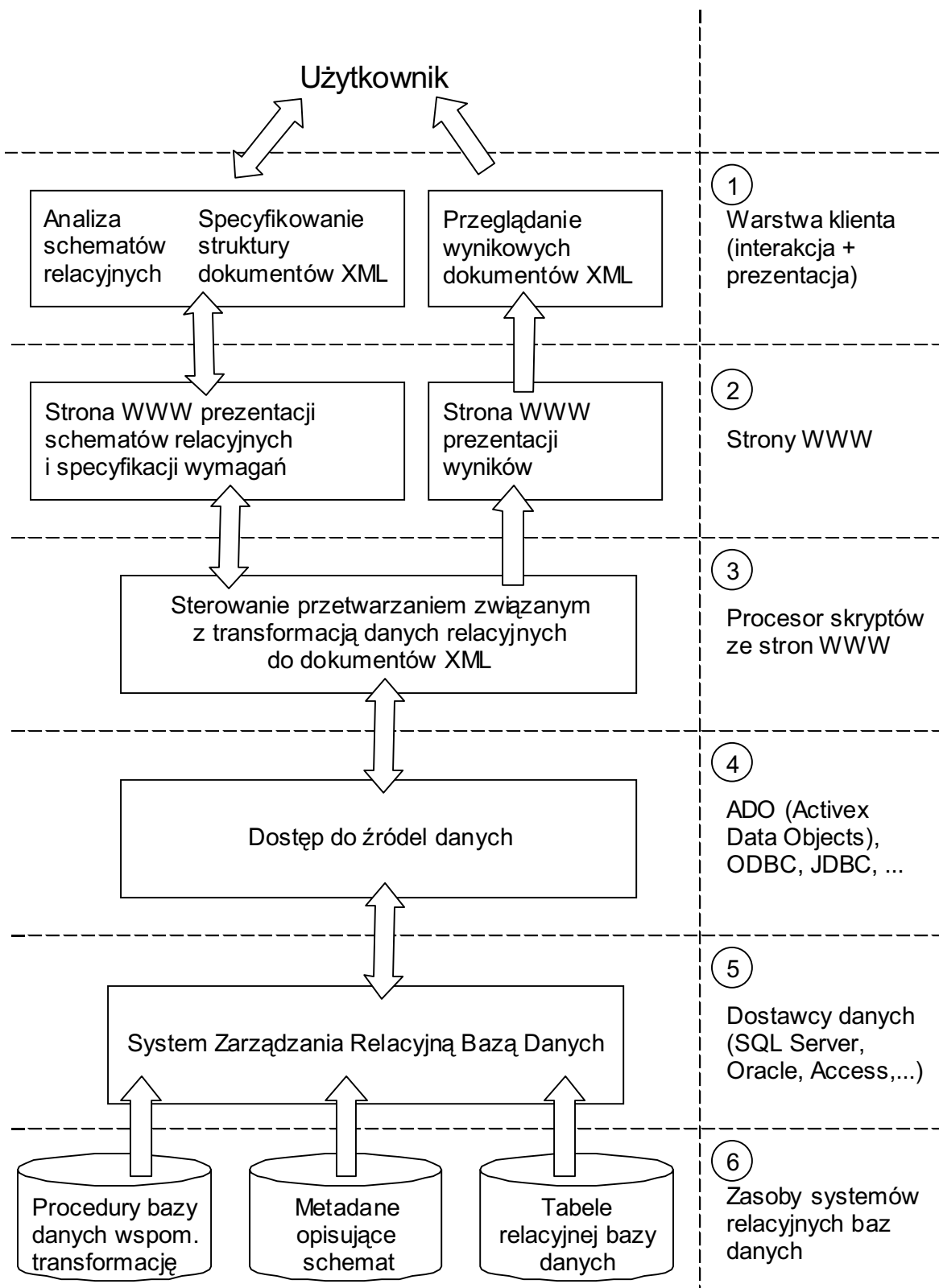
Wynik działania wyrażenia z rysunku 6 przedstawia rysunek 7. Jest to dokument XML-owy zgodny z wymaganiami określonymi na rysunku 3.

```
<Autor IdAut = "au1">
  <Nazwisko> Nowak </Nazwisko>
  <Imię> Jan </Imię>
  <Instytucja> PP </Instytucja>
  <Artykuł IdArt = "ar1">
    <Tytuł> tyt1 </Tytuł>
    <Opis> opis1 </Opis>
  </Artykuł>
  <Artykuł IdArt = "ar3">
    <Tytuł> tyt3 </Tytuł>
    <Opis> opis3 </Opis>
  </Artykuł>
</Autor>
<Autor IdAut = "au2">
  ...
```

Rys. 7. Dokument XML otrzymany w wyniku wykonania wyrażenia z rys. 6

4. Architektura systemu rozpowszechniania informacji

Na rysunku 8 przedstawiono wielowarstwową architekturę systemu rozpowszechniania informacji z relacyjnych baz danych w postaci dokumentów XML-owych. Przyjęto założenia, że środowiskiem rozpowszechniania jest globalna sieć Internet, a więc system oparty jest na technologiach internetowych i na serwerach konwencjonalnych baz danych. Zakładamy, że w bazie relacyjnej gromadzone są dane podlegające powszechnemu upowszechnianiu, na przykład dane bibliograficzne, dane turystyczne, o wydarzeniach kulturalnych itp. Innym obszarem zastosowań tego typu systemów są wyspecjalizowane serwisy informacyjne, z których korzystają specjaliści z różnych dziedzin mający zdalny dostęp do usług tego serwisu, na przykład informacje agencji prasowych, ośrodków badania opinii publicznej, wyniki zawodów sportowych, czy wyspecjalizowanych danych naukowo technicznych. Specjalnym rodzajem zastosowań jest budowa tego typu systemów dla potrzeb organizacji gospodarczych, w tym rozproszonych korporacji i organizacji wirtualnych.



Rys. 8. Wielowarstwowa architektura systemu udostępniania danych relacyjnych w postaci dokumentów XML

W tego typu systemach selektywny dostęp do różnego rodzaju informacji gromadzonych w relacyjnych bazach danych jest szczególnie istotny. Znaczenie poszczególnych warstw systemu jest następujące:

1. Warstwa pierwsza związana jest z procesem interakcji z użytkownikiem i z prezentacją mu wyników realizacji jego poleceń. Użytkownikowi udostępniane są informacje o strukturze relacyjnej bazy danych i na tej podstawie użytkownik specyfikuje swoje wymagania. Jest to jak się wydaje kluczowy etap w całym procesie – od naturalności i prostoty formułowania wymagań zależy akceptacja lub odrzucenie całego systemu przez użytkownika. W rozdziale 3 pokazaliśmy możliwy sposób prezentacji struktury bazy danych (rys. 1) oraz sposób specyfikowania wymagań za pomocą wyboru (pod)grafu schematu i/lub jego specyfikowania za pomocą DTD (rys. 2, 3, 4). Wydaje się, że najwłaściwsze byłoby tutaj stosowanie narzędzi graficznych. Wymaganie aby użytkownik posługiwał się językiem definicji DTD wydaje się zbyt wygórowane. Wybór właściwej strony realizującej interakcję z systemem odbywa się z wykorzystaniem standardowej przeglądarki.
2. Druga warstwa dotyczy obsługi stron WWW po stronie klienta. Strony są wyświetlane i przesyłane z wykorzystaniem standardowych przeglądarek i protokołów komunikacyjnych – w specjalnych przypadkach wymagane byłyby protokoły zapewniające bezpieczeństwo, na przykład wykorzystujące SSL.
3. Kluczowe znaczenie z punktu widzenia realizacji transformacji ma warstwa trzecia, w której realizuje się sterowanie całym procesem. Sterowanie przetwarzaniem zakodowane jest za pomocą języka skryptowego osadzonego na stronie WWW. Należy tak je zorganizować, aby możliwie największą jego część zawrzeć w regułach przetwarzania zakodowanych za pomocą procedur pamiętanych przechowywanych w bazie danych.
4. Odwołanie do docelowej bazy danych nie mogą odbywać się bezpośrednio, ale wykorzystywane są w tym celu mechanizmy realizujące ten dostęp. Najbardziej zaawansowana i najbardziej uniwersalna technika tego rodzaju to technika ADO oparta na ActiveX i pośrednicząca w dostępie do różnorodnych zewnętrznych źródeł danych. Dzięki temu w jednolity sposób widzimy wszystkie systemy zarządzania bazami danych, w jednolity sposób możemy na nich operować i inicjować procedury pamiętane.
5. Warstwa piąta obejmuje komercyjny system zarządzania bazami danych, ewentualnie inne źródło danych dostępne poprzez ADO. ODBC, JDBC itp. Ważne jest aby system ten był w stanie wykonywać wyspecjalizowane funkcje na danych w bazie danych.
6. Warstwa szósta to zasoby systemu relacyjnego. Obok tabel pamiętających dane, w bazie danych gromadzone są metadane opisujące zarówno strukturę tabel, jak i powiązania między tabelami. Specjalne znaczenie mają procedury pamiętane napisane w wyspecjalizowanym języku systemu bazy danych (np. Transact-SQL), skompilowane i zoptymalizowane z punktu widzenia efektywności przetwarzania dla potrzeb transformacji danych relacyjnych w dokumenty XML-owe. Wykonywanie tych procedur na serwerze gwarantuje dużą efektywność systemu.

Efekt przetwarzania uzyskany w wyniku działania procedur pamiętanych i/lub języka SQL w serwerze bazy danych, jest następnie przejmowany ponownie przez oprogramowanie wykonywane na warstwie trzeciej. Tworzony jest wynik tego przetwarzania w postaci strony WWW i przekazywany jest użytkownikowi.

5. Podsumowanie

Selektywny dostęp do informacji jest jednym z ważniejszych czynników warunkujących skuteczne działanie w świecie współczesnym. Podejmowanie właściwych decyzji w działalności gospodarczej, naukowej czy społecznej wymaga szybkiego i wygodnego dostępu do wiarygodnych informacji. Okazuje się jednak, że często mimo, że wiarygodne informacje zgromadzone są w licznych bazach danych i nie ma przeszkód aby je udostępniać, to dla niewprawnego użytkownika istnieje nieprzekraczalna granica w dostępie do tych danych. Istniejące technologie i metody dostępu ciągle nie są wystarczająco przyjazne dla użytkowników, którzy nie są specjalistami w przetwarzaniu danych.

Nowe możliwości w tym zakresie daje standard XML. Aby jednak dane mogły być prezentowane w postaci dokumentów XML-owych należy opracować metody konwersji danych relacyjnych do postaci tych dokumentów. W niniejszej pracy opisujemy jedną z takich metod. Kolejnym problemem jest opracowanie wygodnych dla użytkownika mechanizmów formułowania wymagań. Omawiamy także wielowarstwową architekturę systemu realizującego omawiane funkcje. System taki należy widzieć w kontekście zaawansowanych systemów integrowania i rozpowszechniania danych.

Literatura

1. Abiteboul S., Buneman P., Suci D., *Data on the Web. From Relational to Semistructured Data and XML*, Morgan Kaufmann, San Francisco, 2000.
2. Altinel M., Franklin M. J., *Efficient filtering of XML documents for selective dissemination of information*, Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000, ss. 53-64.
3. Bonifati A., Ceri S., Comparative Analysis of Five XML Query Languages, *ACM SIGMOD Record* **29**(1), 2000.
4. Pankowski T., *Model bazy danych obiektów częściowo etykietowanych w procesie integrowania danych*, W: Z. Mazur (red.) Bazy danych, Nr 1, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2000, s. 49-84.
5. Pankowski T., *Relational completeness of a query language for databases of labeled objects*, Proceedings of the 4th IEEE International Conference "Databases & Information Systems", Vilnius, 59-74 (2000).
6. Shanmugasundaram J., i in., *Efficiently publishing relational data as XML documents*, Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000, ss. 65-76.
7. Shanmugasundaram J., i in., *Relational databases for querying XML documents: limitations and opportunities*, Proceedings of the 25th International Conference on Very Large Databases, Edinburgh, Scotland, 1999, ss. 302-314.
8. World Wide Web Consortium, *Extensible markup language (xml) 1.0*, 1998. <http://www.w3.org/TR/REC-xml>