# Rozdział 4

# Integracja danych w teorii i praktyce – przegląd problemów i rozwiązań

**Streszczenie.** W rozdziale omawiane są problemy integracji danych z punktu widzenia ogólnych potrzeb identyfikowanych w praktyce, charakteryzowane są niektóre badania teoretyczne inspirowane potrzebami integracji danych oraz przedstawione są wybrane rozwiązania komercyjne przeznaczone do budowy systemów integracji danych. Integracja danych rozpatrywana jest jako kolejny etap rozwoju technologii baz danych włączający nowe dyscypliny związane z usługami sieciowymi, semantyką danych i inżynierią wiedzy. W zakresie badań teoretycznych rozważamy zagadnienia odwzorowywania schematów XML jako podstawę wymiany danych i reformułowania zapytań. W konkluzji formułujemy postulat podejścia do procesów integracji *na poziomie danych* (tzw. *integracja danocentryczna*) w przeciwieństwie do dominującej obecnie integracji *na poziomie aplikacji*.

## 1 Wstęp

Z badań przeprowadzonych przez IBM wśród kadry zarządzającej i opublikowanych w Raporcie CEO [11] (ang. *Chief Executive Officer*) wynika, że tylko 13% z nich uważa, że są dobrze przygotowani do szybkiego reagowania na zmiany pojawiające się na rynku. Podkreślają potrzebę uwzględniania i rozumienia wszystkich rodzajów dostępnej informacji, aby w sposób właściwy podejmować decyzje biznesowe. 68% badanych wymienia integrację niedopasowanych do siebie aplikacji i infrastruktur jako kluczowy problem zwalniający i blokujący przepływ informacji. Jednocześnie twierdzą, że 30% swego czasu poświęcają na wyszukiwanie informacji, których potrzebują w swojej pracy.

Problemy z dotarciem do informacji związane są ze wzrastającym wolumenem danych dostępnych *on line*. Jednak innym, poważniejszym problemem jest fragmentacja danych i zwiększające się rozproszenie ich źródeł. W firmach pojawia się coraz więcej baz danych, często ukrytych w obrębie różnych aplikacji, a powszechne jest istnienie różnorodnych repozytoriów dokumentów oraz innych źródeł nieustrukturalizowanych danych (np. poczta elektroniczna, strony internetowe). Z badań [7] wynika, że 79% firm ma więcej niż dwie bazy dokumentów, a 25% więcej niż piętnaście.

Tadeusz Pankowski
Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej,
pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Polska
Uniwersytet im. Adama Mickiewicza, Wydział Matematyki i Informatyki,
ul. Umultowska 87, 61-614 Poznań,
email:tadeusz.pankowski@put.poznan.pl

Informację jest nie tylko trudno odnaleźć, ale nie do uniknięcia są również inne problemy takie jak nakładanie się danych (informacja w różnych źródłach może się powtarzać), niezgodność między sobą w różnych źródłach czy niepełność. Okazuje się, że prawie wszędzie, gdzie występuje kilka źródeł danych (o klientach, pacjentach, studentach, pracownikach, itp.) dane zawarte w tych źródłach są w mniejszym lub większym stopniu niezgodne ze sobą.

Celem tego rozdziału jest omówienie ważniejszych praktycznych i teoretycznych problemów integracji danych. Integracja danych rozpatrywana jest przy tym jako naturalne rozwinięcie technologii baz danych włączające dorobek programowania rozproszonego, zarządzania wiedzą, usług sieciowych i semantycznej sieci Web.

W podrozdziale 2 dyskutujemy podstawowe problemy integracji danych, w tym jej etapy według metodyki opracowanej przez IBM [12]. W podrozdziale 3 przedstawiono wybrane problemy teoretyczne, a w podrozdziale 4 scharakteryzowano niektóre rozwiązania komercyjne przeznaczone do integrowania danych. W podrozdziale 5 zawarto podsumowanie i uwagi na temat prac nad integracją danych prowadzonych na Politechnice Poznańskiej.

## 2 Problemy integracji danych

### 2.1 Cel i znaczenie integracji danych

Celem integracji jest umożliwienie sprawnego tworzenia nowych aplikacji wymagających danych z wielu źródeł [7]. Realizacja tego celu wymaga rozwiązania wielu problemów, takich jak [7], [8], [10]:

- określenie najlepszego źródła informacji dla danych potrzeb,
- utworzenie odpowiedniego interfejsu do integrowanych danych lub ich schematów,
- sposób formułowania zapytań do różnorodnych źródeł danych i opracowywanie planów ich efektywnego wykonania,
- czyszczenie i rozwiązywanie konfliktów w celu uzyskania spójnego zbioru danych,
- postępowanie z danymi niepewnymi i śledzenie ich pochodzenia,
- identyfikowanie tych samych obiektów w różnych źródłach,
- przekształcanie danych w celu uzyskania struktury spełniającej wymagania określonego modelu (schematu),
- uwzględnienie wymagań bezpieczeństwa, poufności i wiarygodności danych.

W procesie integracji należy uwzględniać dwa następujące czynniki jakościowe:

- jakość danych (ang. *data quality*) – na przykład: aktualność, spójność, kompletność;
- jakość usługi (ang. *quality of service*) – na przykład: czas odpowiedzi, dostępność, koszt udostępnienia, bezpieczeństwo.

Integracja danych może być rozpatrywana z różnych punktów widzenia i w różnych obszarach, gdzie waga i znaczenie omawianych powyżej problemów może być bardzo różna:

- integracja danych bibliograficznych gromadzonych w różnych bibliotekach,
- integracja danych naukowych pozyskiwanych i udostępnianych przez instytuty badawcze,
- integracja danych w procesach biznesowych (analiza rynku, negocjowanie i realizacja umów biznesowych),
- integracja danych medycznych (związanych np. z leczeniem konkretnego pacjenta),
- integracja danych na potrzeby instytucji publicznych (opieka społeczna, kontrola zagrożeń ekologicznych, kontrola zagrożeń kryminalnych),

– integracja danych dotyczących zjawisk społecznych, politycznych, kulturalnych, itp.

### 2.2 Etapy procesu integracji danych

Integracja danych powinna być traktowana jako proces składający się z następujących czterech etapów [7], [10]: zrozumienie, standaryzacja, specyfikacja i przetwarzanie.

***Zrozumienie*** (ang. *understanding*)**.** Pierwszym zadaniem w procesie integracji jest zrozumienie danych podlegających integracji. Kluczowe znaczenie ma tutaj analiza metadanych opisujących poszczególne źródła danych, a więc analiza schematów, kluczy głównych, zależności referencyjnych (tzn. kluczy obcych) oraz analiza innych warunków spójności. Analiza może dotyczyć charakterystyki statystycznej danych (na przykład, rozkład statystyczny wartości danych, selektywność, częstość wystąpień). Analizie podlegają wzajemne zależności między poszczególnymi źródłami danych. Dotyczy to określenia jakie zakresy informacji występują w kilku źródłach i czy dane je reprezentujące nie są sprzeczne ze sobą. Czy w przypadku określenia globalnych warunków spójności dane w poszczególnych źródłach są z nim zgodne, czy nie. Istotne jest przy tym to, czy zbiór integrowanych źródeł danych jest z góry określony, czy też dopuszczamy dynamiczną zmianę tego zbiory, tzn. pewne źródła danych przestają być brane pod uwagę, podczas gdy inne mogą być na bieżąco włączane do systemu (np. w systemach P2P) [3], [5], [13].

***Standaryzacja*** (ang. *standardization*). Na tym etapie następuje określenie najlepszego sposobu reprezentacji oraz metody czyszczenia integrowanych danych. Obejmuje to przede wszystkim określenie *schematu docelowego*, tj. schematu, według którego zintegrowane dane udostępniane są użytkownikom. Podejmowane są tutaj decyzje dotyczące ujednolicenia używanych nazw i sposobów reprezentacji wartości danych. Na przykład, czy nazwa ulicy w adresie ma być poprzedzona słowem „Ulica", skrótem „ul.", z małej czy z dużej litery, z kropką czy bez? Na tym etapie podejmowane są również decyzje dotyczące sposobów postępowania z danymi niespójnymi (ang. *inconsistent*) i niepełnymi (ang. *Incomplete*). Gdy na przykład jedna osoba będzie miała w różnych źródłach przypisane różne adresy, wówczas mamy do czynienia z niespójnością (naruszona zależność funkcyjna między osobą i jej adresem). Co wtedy zrobić? – zachować obydwa adresy czy tylko jeden z nich (najnowszy?). Określane jest to jako *reperowanie danych* (ang. *data repair*). Wreszcie kluczowym problemem jest właściwa *identyfikacja obiektów*, to znaczy określenie czy dane dotyczą tego samego obiektu rzeczywistego. Czy na przykład dane związane z nazwiskiem „Jan Nowak" (np. w relacyjnej bazie danych) dotyczą tej samej osoby co dane związane z nazwiskiem „Jan R. Nowak" (np. w poczcie elektronicznej)?

***Specyfikacja*** (ang. *specification*). Specyfikacja określa, w jaki sposób ma być wykonane przetwarzanie danych na następnym etapie. Metody i techniki specyfikacji związane są ściśle z wybranym systemem przetwarzania. Stosowane są tutaj narzędzia *mapowania* (określania *odwzorowań*) danych ustalających powiązania między danymi źródłowymi i danymi docelowymi. Wynikiem mapowania jest wygenerowane zapytanie (na przykład w SQL, XSLT lub XQuery [23]) transformujące dane źródłowe do wymaganej postaci docelowej.

***Przetwarzanie*** (ang. *execution*). Na etapie tym wykonywana jest faktyczna integracja. Integracja może być zrealizowana poprzez: *materializację*, *federację* lub *indeksowanie*.

1) *Materializacja* polega na utworzeniu bazy zintegrowanych danych (hurtowni danych, magazynu danych). Podstawową techniką materializacji jest proces ETL (*Ekstrakcja-/Transformacja/Ładowanie*). W procesie tym dane pobierane są z jednego lub z wielu źródeł danych, poddawane są transformacji zgodnie z opracowaną specyfikacją, a następnie są zapamiętywane w docelowej bazie danych. Odmianami materializacji są *replikacja* i *caching*:

- *replikacja* to tworzenie kopii danych oraz ich synchronizacji z danymi źródłowymi według określonych strategii;
- *caching* polega na gromadzeniu wyników wykonywanych zapytań w celu ich późniejszego wykorzystania do udzielania odpowiedzi na te same lub inne zapytania.

2) *Federacja* polega na tworzeniu wirtualnej reprezentacji integrowanych danych. Dane są prezentowane poprzez *schemat docelowy* (lub *schemat mediujący*), który nie jest bezpośrednio związany z danymi, istnieją natomiast definicje określające powiązania tego schematu ze schematami źródłowymi. W zależności od sposobu wykonywania zapytań formułowanych względem schematu docelowego mówimy o *wymianie danych* (ang. *data exchange*) i *reformułowaniu zapytań* (ang. *query reformulation*).
   - w przypadku *wymiany danych* odpowiednie zbiory danych źródłowych materializowane są według schematu docelowego i na tak utworzonej bazie danych wykonywane jest zapytanie docelowe;
   - przy *reformułowaniu zapytań* zapytanie docelowe jest przekształcane (przeformułowywane) w zbiór zapytań, które mogą być wykonane bezpośrednio na danych źródłowych. Uzyskane w ten sposób częściowe odpowiedzi łączone są następnie tworząc oczekiwaną odpowiedź na zapytanie.

   Bardziej zaawansowaną metodą federacji są systemy *integracji P2P* (*peer-to-peer*). Wówczas nie ma jednego wyróżnionego schematu docelowego (mediującego), a rolę takiego schematu może pełnić każdy ze schematów związanych z poszczególnymi hostami sieci (niezależnie czy są w nim zapamiętane również dane, czy nie). Wówczas zarówno wymiana danych, jak i reformułowanie zapytań propagowane są między hostami sieci zgodnie ze ścieżkami semantycznych powiązań określających odwzorowania między schematami.
3) *Indeksowanie* jest techniką integracji polegającą na utrzymywaniu informacji umożliwiającej dotarcie do pełnych danych. Tworzone są indeksy słów kluczowych zawierające adresy URL dokumentów, w których te słowa są zawarte. Dokumenty te pobierane są dynamicznie dopiero wtedy, gdy wykonywane jest zapytanie.

Rozpoczęta integracja jest procesem, który nigdy się nie kończy, a omówione etapy i problemy przeplatają się przy poszukiwaniu zadawalającego rozwiązania. W obszarze zainteresowania integracji mogą pojawiać się coraz to nowe źródła danych i nowi użytkownicy (aplikacja) z coraz to bardziej zaawansowanymi wymaganiami.

## 3 Teoretyczne problemy integracji danych

Tematyka integracji danych znajduje bardzo obszerne odzwierciedlenie w literaturze poświęconej teoretycznym problemom przetwarzania danych. Mieści się bowiem w nurcie badań związanych zarówno z teorią modeli, jak i z teorią języków i translacji, a także dotyczy szeregu zagadnień związanych z przetwarzaniem transakcji w środowisku rozproszonym. Problemy z tych obszarów rozważane są w odniesieniu zarówno do relacyjnych, jak i XML-owych baz danych [2], [6], [18], [24]. W kolejnych podpunktach przedstawimy niektóre z tych zagadnień nawiązujące również do naszych wcześniejszych prac [3], [19], [20].

### 3.1 Odwzorowywanie schematów

Podstawowym zagadnieniem rozważanym niezależnie od obranej strategii integracji jest problem *odwzorowywania schematów* (ang. *schema mapping*). Specyfikacja odwzorowania jest wykorzystywana do tego, aby dla zadanej instancji (wystąpienia) schematu źródłowego utworzyć odpowiednią instancję schematu docelowego.

W odniesieniu do relacyjnych baz danych, problem odwzorowywania schematów sformułowany został przez Fagina i in. [6]. Dla schematu relacyjnego R=($R_1$,…,$R_k$), gdzie każdy symbol relacyjny $R_i$ ma przypisaną dodatnią liczbę całkowitą $m_i$ zwaną *arnością* symbolu, *instancją I* jest funkcja (interpretacja) przypisująca każdemu symbolowi relacyjnemu $R_i$ $m_i$-członową relację $I(R_i)$. Pojęcia te można rozszerzyć na dane XML, gdzie schemat dany jest za pomocą DTD lub XML Schema, a instancją schematu jest dokument XML reprezentowany zgodnie z modelem DOM [22], [2], [19] i spełniający schemat.

Niech S i T będą odpowiednio *schematem źródłowym* i *schematem docelowym*. Kluczowym problemem jest określenie *odwzorowania* $M_{ST}$ między S i T definiującego *korespondencję* między S i T w taki sposób, aby odwzorowanie to można było wykorzystać w procesach:

- wymiany danych (ang. *data exchange*) – dla zadanej instancji I schematu S (tj. I |= S) wyznaczyć instancję J schematu T (tj. J |= T) taką, że para (I, J) spełnia formułę $M_{ST}$, tj. (I, J) |= $M_{ST}$; wówczas J nazywamy *rozwiązaniem* dla I względem odwzorowania $M_{ST}$ (symbol |= oznacza relację spełniania);
- reformułowania zapytań (ang. *query reformulation*) – zapytanie Q względem T przekształcić w takie zapytanie Q' względem S, że dla każdej instancji I schematu S, Q'(I) = Q($M_{ST}$(I)).

W rozważaniach teoretycznych odwzorowania między schematami wyrażane są za pomocą formuł logicznych zwanych formułami STD (ang. *source-to-target dependencies*). Formuły STD były wykorzystywane dla potrzeb badania teorii zależności (zależności funkcyjnych i zależności zawierania) w relacyjnych bazach danych [1]. Ogólna postać formuły STD w logice pierwszego rzędu, FO STD, (ang. *first order STD*) jest następująca:

$$\forall \mathbf{x}\ (\varphi(\mathbf{x}) \Rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})),$$

gdzie: **x** jest wektorem zmiennych źródłowych; $\varphi(\mathbf{x})$ jest koniunkcją formuł definiujących wiązania zmiennych w strukturach źródłowych (w podejściu relacyjnym są to formuły relacyjne o postaci $R(x_1,...,x_m)$, a w przypadku XML są to wzorce drzew zwane formułami TPF omawiane w następnym podrozdziale) i równości o postaci $x = x'$; a $\psi(\mathbf{x}, \mathbf{y})$ jest koniunkcją formuł definiujących wiązania zmiennych w strukturach docelowych.

W wyniku skolemizacji, formuły STD można przekształcić w formuły drugiego rzędu, SO STD (ang. *second order STD*), tj. w formuły o postaci:

$$\exists \mathbf{f} \forall \mathbf{x}, \mathbf{y}\ (\varphi(\mathbf{x}) \wedge \chi(\mathbf{x}, \mathbf{y}) \Rightarrow \psi(\mathbf{x}, \mathbf{y})),$$

gdzie **f** jest wektorem symboli funkcyjnych, a $\chi(\mathbf{x}, \mathbf{y})$ jest koniunkcją równości o postaci $x = f(x_1, ..., x_k)$ lub $y = f(x_1, ..., x_k)$, gdzie $x, x_1, ..., x_k \in \mathbf{x}$, a $y \in \mathbf{y}$.

### 3.2 Odwzorowania schematów XML

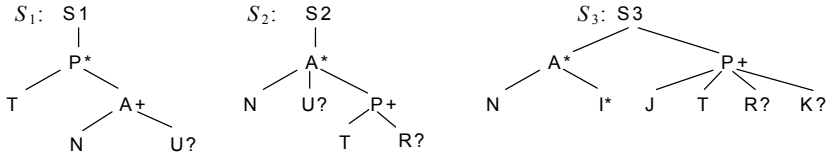
W przypadku specyfikowania odwzorowań dla danych XML formuły definiujące wiązanie zmiennych przyjmują postać formuł TPF (ang. *tree-pattern formula*) [2]. Formułę SO STD, w której występują formuły TPF będziemy nazywać SO XSTD (ang. *second order XML source-to-target dependences*).

**Definicja 1.** Formułą TPF $\pi$ nad zbiorem etykiet $L$ jest wyrażenie o składni:

$$\pi ::= /\text{top}[E],$$
$$E ::= l = x \mid l[E] \mid E \wedge E,$$

gdzie *top* jest najbardziej zewnętrzną (szczytową) etykietą dokumentu XML; $l$ jest dowolną etykietą w $L$; $x$ jest zmienną tekstową wiązaną z wartością tekstową elementu o etykiecie $l$.

Niech $S_1$, $S_2$ i $S_3$ będą drzewami schematów XML przedstawionymi na rys. 1, gdzie znaczenie etykiet jest następujące: P – publikacja, T – tytuł, A – autor, N – nazwisko, U – uniwersytet, z którego pochodzi autor; R – rok publikacji, K – konferencja, na której publikacja była przedstawiana; I – referencja do wartości J łącząca autora z jego publikacją. Rozważane schematy na różne sposoby przedstawiają dane bibliograficzne i mogą reprezentować różne źródła danych, w których dane są zapamiętane. Dane w poszczególnych źródłach mogą się wzajemnie uzupełniać, mogą się także powtarzać.




**Rys. 1.** Drzewa przykładowych schematów XML

Specyfikacje odwzorowań (korespondencji) $M_{ij}$ między schematami $S_i$ i $S_j$ mają następującą postać (numery linii dodajemy dla ułatwienia objaśnień):

(1) $M_{12} = /S1[P[T = x_T \wedge A[N = x_N \wedge U = x_U]]] \wedge$
(2) $x_R = f_R(x_T) \Rightarrow$
(3) $/S2[A[N = x_N \wedge U = x_U \wedge P[T = x_T \wedge R = x_R]]]$

(4) $M_{32} = /S3[A[N = x_N \wedge I = x_I \wedge P[J = x_J \wedge T = x_T \wedge R = x_R \wedge K = x_K]] \wedge$
(5) $x_I = x_J \wedge$
(6) $x_U = f_U(x_N) \Rightarrow$
(7) $/S2[A[N = x_N \wedge U = x_U \wedge P[T = x_T \wedge R = x_R]]]$

Formuły TPF w specyfikacji odwzorowań są faktycznie wyrażeniami logicznymi języka XPath – przyjmują wartość prawda, gdy przy danym wartościowaniu zmiennych w instancji I wyznaczają niepustą sekwencję wierzchołków i wartość fałsz w przeciwnym przypadku. Dla zadanej instancji źródłowej I, poprzednik implikacji wyznacza zbiór wartościowań zmiennych, dla których jest prawdziwy.

Linie (1) i (4) określają sposób wiązania zmiennych źródłowych – każda zmienna przyjmuje wartości tekstowe odpowiedniego wierzchołka tekstowego lub atrybutu.

Linie (2) i (6) podają w jaki sposób wartości zmiennych docelowych obliczane są z wartości zmiennych źródłowych. Podana nazwa funkcji może pełnić tylko rolę symboliczną, mówi na przykład tylko tyle, że rok publikacji $x_R$ jest funkcyjnie zależny od tytułu tej publikacji, choć faktyczna postać funkcji $f_R()$ nie jest znana. W pełniejszej specyfikacji postać funkcji może być zdefiniowana: na przykład w rozwiązaniach komercyjnych [15], [21] ma postać funkcji w XSLT lub XQuery; możemy również zakładać, że obliczenie wartości tej funkcji wymaga interakcji z użytkownikiem.

Linia (5) odzwierciedla zależność referencyjną między I i J w schemaccie $S_3$.

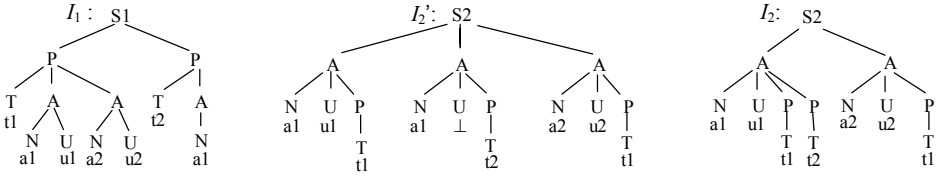
Formuły TDF będące następnikami implikacji (docelowa TPF) wyznaczają fragmenty instancji docelowej odpowiadającej instancji źródłowej zdefiniowanej w poprzedniku implikacji. Docelowa TPF pozwala wygenerować *instancję kanoniczną* schematu docelowego

za pomocą techniki zwanej *pogonią* (ang. *chase*) [1]. W przypadku docelowej TPF z linii (3), postępujemy następująco:

- tworzymy korzeń instancji docelowej i jej najbardziej zewnętrzny element S2;
- pod wierzchołek S2 „doczepiamy" poddrzewa o korzeniu A – liczba tych poddrzew jest równa liczbie różnych wartościowań zmiennych ($x_N$, $x_U$, $x_T$, $x_R$), jeśli jakieś wartościowanie jest niepełne, tzn. zmienna nie ma przypisanej wartości, to jako jej wartość przyjmujemy $\perp$ (*null*) lub tekst o postaci np. „$f_R$(Podstawy SQL)", gdzie tekst „Podstawy SQL" jest wartością zmiennej $x_T$ (tytułem publikacji).

Otrzymana instancja kanoniczna nie musi być jedyną instancją docelową spełniającą odwzorowanie. Na rysunku 2 podano instancje $I_2$ i $I_2$' schematu $S_2$, otrzymane z instancji $I_1$ o schemacie $S_1$ ($I_2$' jest instancją kanoniczną). Obydwie są rozwiązaniami dla $I_1$ względem odwzorowania $M_{12}$. Mamy bowiem wówczas następujące spełniania:

$$I_1 \models S_1,\ I_2 \models S_2,\ I_2' \models S_2,\ (I_1, I_2) \models M_{12},\ (I_1, I_2') \models M_{12}.$$




**Rys. 2.** Instancja źródłowa $I_1$ o schemacie $S_1$ i dwie instancje schematu $S_2$ będące jej rozwiązaniami względem odwzorowania $M_{12}$ ($I_2$' jest instancją kanoniczną)

Jak widać na powyższym przykładzie, rozwiązanie dla zadanej instancji źródłowej i przy zadanym odwzorowaniu nie musi być jednoznaczne. W [19] pokazujemy, w jaki sposób można uzyskać jednoznaczne rozwiązanie poprzez uwzględnienie informacji o kluczach. Klucze określają oczekiwany sposób identyfikacji poddrzew w drzewach XML, a także sposób grupowania i zagnieżdżania elementów. Uzyskujemy to poprzez wprowadzenie nowej klasy formuł zwanych formułami KPF (ang. *key-pattern* formula). Formuły KPF pozwalają ujmować klucze zgodnie z teorią kluczy dla XML zaproponowaną w [4].

# 4 Integracja danych w praktyce

W ostatnich latach producenci komercyjnych systemów baz danych i rozwiązań dla biznesu oferują na rynku liczne produkty realizujące lub wspomagające integrację [5], [12], [14], [15], [21].

## 4.1 IBM Information Server

IBM Information Server (IIS) [12] – jest platformą integracji informacji i obejmuje szereg produktów wspomagających różne funkcje na poszczególnych etapach integracji danych. Na poszczególnych etapach występują następujące narzędzia:

1) Na etapie wspomagającym zrozumienie (analizę integrowanych danych) występują narzędzia przeznaczone do odkrywania, modelowania i zarządzania strukturami danych i ich zawartością.

- *WebSphere Information Analyzer* – analizuje dane źródłowe, odkrywa klucze główne i obce, sprawdza spełnianie reguł integracji i reguł jakości definiowanych przez użyt-

kownika; wspomaga zrozumienie danych fizycznych; odkrywa szczegółowe charakterystyki danych w poszczególnych kolumnach (liczebność, wartości *null*, zakresy wartości, długość, dokładność, selektywność, itp.);
- *WebSphere Business Glossary* – wspomaga definiowanie i zarządzanie słownictwem (ontologią) dziedziny i wiąże pojęcia z danymi (zapewnia to właściwe rozumienie danych). Przeznaczony jest dla użytkowników biznesowych i ekspertów dziedzinowych i wykorzystywany jest zarówno na etapie rozpoznawania danych, jak i na etapie standaryzacji;
- *Rational Data Architect* (RDA) – jest w pełni funkcjonalnym narzędziem modelowania danych, które może być używane z dowolnym systemem zarządzania bazą danych. RDA wspomaga zrozumienie danych na poziomie logicznym. Pozwala na projektowanie i eksplorację schematu logicznego (np. odnajdywanie definicji kluczy obcych) oraz na tworzenie schematów fizycznych. RDA zawiera także możliwości definiowania odwzorowań opracowane w systemie Clio [8] między schematami danych w procesie integracji (zarówno danych SQL, jak i XML). Na podstawie odwzorowań RDA generuje programy transformacji danych. Narzędzie to posiada również funkcje przeznaczone dla etapów standaryzacji i specyfikacji.

2) Standaryzacja – narzędzie służące do standaryzacji, scalania i korygowania danych.
- WebSphere Metadata Server – udostępnia ujednolicone repozytorium metadanych. Zapewnia import/eksport między dwudziestoma różnymi narzędziami modelowania danych i inteligentnych systemów biznesowych;
- *WebSphere QualityStage* – jest podstawowym narzędziem związany z czyszczeniem danych. Umożliwia definiowanie formatów danych, sprawdzania, poprawiania i wzbogacania pól danych, dopasowywania i łączenia rekordów mogących reprezentować te same obiekty. Interfejs graficzny pozwala na określanie reguł czyszczenia i organizowania procesów czyszczenia w postaci przepływów czynności (ang. *workflows*), a także na obserwowanie wpływu reguł na zbiory danych.

3) Specyfikacja – narzędzie służące do opisu, jak łączyć i strukturalizować dane dla potrzeb nowych zastosowań.
4) Wykonywanie – narzędzie służące do synchronizacji, wirtualizacji i przekazywania danych w trybie on-line.
- *WebSphere DataStage* – używany do materializacji danych poprzez ich ekstrakcję, transformacje i ładowanie (realizacja procesu ETL);
- *WebSphere Federation Server* – umożliwia dostęp do heterogenicznych źródeł danych tak jakby dane te były w pojedynczej (wirtualnej) bazie danych;
- *WebSphere Replication Server* – zarządza replikami danych i synchronizuje ich zawartość.
- *WebSphere Data Event Publisher* – używany jako sposób integracji aplikacji poprzez wysyłanie komunikatów między nimi, a także jako sposób inicjowania zasilania danymi w procesach ETL.

### 4.2 Rozwiązania firmy Microsoft

Microsoft SQL Server 2005 Integration Services (SSIS) [17] są zestawem usług do tworzenia materializującej integracji danych (hurtowni danych) w wyniku procesów ETL (ekstrakcji, transformacji i ładowania). SSIS realizuje następujące zadania integracji danych.

1) Scalanie danych z heterogenicznych źródeł danych

Scalanie różnorodnych źródeł danych jest szczególnie istotne, gdy dane pamiętane są w systemach danych budowanych według przestarzałych technologii, ale także wtedy, gdy źródła danych budowane są niezależnie przy zastosowaniu różnych modeli i różnych rozwiązań inżynierskich. SSIS pozwala na utworzenie połączeń do wielu źródeł danych w ramach pojedynczego pakietu. Do relacyjnych i XML-owych baz danych można odwoływać się za pośrednictwem sterowników ADO i ADO.NET lub ODBC. Można także łączyć się z plikami tekstowymi, Excel i projektami OLAP Analysis Services. Wynikiem scalania jest utworzenie pojedynczego, spójnego magazynu danych.

2) Wypełnianie i aktualizacja hurtowni danych

Dane w hurtowni danych są zwykle aktualizowane dość często, a rozmiar ładowanych danych jest bardzo duży. Usługi SSIS posiadają mechanizmy masowego ładowania danych bezpośrednio z plików do bazy danych SQL Server, a pakiety SSIS mogą być restartowane automatycznie. Zapewnia to dużą elastyczność i efektywność oraz umożliwia automatyzację procesów.

3) Czyszczenie i standaryzacja danych

Przed załadowaniem do bazy OLTP lub OLAP, dane muszą być oczyszczone i sprowadzone do standardowej postaci. Usługi SSIS mają wbudowane transformacje ułatwiające ten proces – umożliwiają konwertowanie danych lub tworzenie nowych danych na podstawie wyrażeń definiujących te dane. Możliwe jest także grupowanie podobnych wartości danych, co jest potrzebne do identyfikacji rekordów, które mogą się duplikować i nie można ich włączyć do bazy danych bez dodatkowej analizy.

4) Inteligencja biznesowa w procesach transformacji

Proces transformacji danych wymaga dynamicznego reagowania na zawartość przetwarzanych danych. Dane mogą wymagać agregacji, konwersji lub dystrybucji na podstawie ich wartości. Reakcją może też być ich odrzucenie. Realizacja tych funkcji wymaga zaprogramowania odpowiedniej logiki przetwarzania (inteligencji biznesowej). Pakiety i kontenery SSIS umożliwiają dynamiczne tworzenie przepływów (ang. *workflows*) w zależności od wartości przetwarzanych danych.

5) Automatyzacja funkcji administracyjnych i ładowania danych

Takie funkcje administracyjne jak archiwowanie i odtwarzanie zarchiwowanej bazy danych, kopiowanie baz danych lub ich wybranych obiektów, a także ładowanie danych realizowane są regularnie i wymagają automatyzacji. Usługi SSIS zawierają mechanizmy realizujące tę automatyzację.

Microsoft BizTalk Server [15] przeznaczony jest do integracji funkcji systemów klasy EAI (ang. *Enterprise Application Integration*) i B2B (ang. *Business To Business*) ze szczególnym uwzględnieniem środowiska Internetu. Pozwala na współdziałanie luźno połączonych i długotrwałych procesów biznesowych zarówno wewnątrz, jak i na zewnątrz organizacji, gdzie transakcje mogą trwać bardzo długo (tygodnie lub miesiące). BizTalk udostępnia narzędzia do specyfikacji dokumentów, odwzorowań między dokumentami oraz ich transformacji. Monitoruje przebieg procesów w systemie. W szczególności oferuje wygodny interfejs graficzny do definiowania odwzorowań między schematami XML [16], co z teoretycznego punktu widzenia dyskutowaliśmy w podrozdziale 3.

## 5 Podsumowanie

W rozdziale przedstawiono przegląd problemów związanych z integracją danych. Dziedzina ta rozwija się bardzo dynamicznie, co jest inspirowane z jednej strony realnymi potrzebami formułowanymi ze strony praktyki przetwarzania danych, a z drugiej – oferowanymi rozwiązaniami ze strony producentów oprogramowania w odpowiedzi na te potrzeby. Otwiera się przy tym bardzo obszerny i interesujący obszar badań teoretycznych. Można zaobserwować, że na najważniejszych konferencjach z baz danych (SIGMOD, VLDB, EDBT, ICDT) prace poświęcone podstawowym problemom teoretycznym integracji danych przygotowywane są przez osoby związane z wielkimi producentami oprogramowania, takimi jak IBM, Microsoft, Oracle czy Google.

Główne propozycje rozwiązań pochodzą przy tym od specjalistów z baz danych. Do tej pory w zakresie integracji dominują systemy EAI, a więc dotyczące integracji *na poziomie aplikacji*. Polega ona na tym, że wykorzystując różnorodne narzędzia (również te omówione w podrozdziale 4 tworzona jest aplikacja, która w sposób proceduralny zbiera dane z różnorodnych źródeł i odpowiednio je transformuje. Integrację na poziomie aplikacji cechują następujące wady [7], [10]: (1) wcale nie jest oczywiste, że pisanie aplikacji dla potrzeb integracji jest proste nawet na początku procesu integracji – jaką integrację wybrać: materializację, federację czy indeksowanie? ile zmian będzie koniecznych przy rozszerzaniu zestawu źródeł danych?; (2) jeśli kod programu integracji zawarty jest w aplikacji, to może być optymalizowany tylko przez programistę, a więc wydajność może być tylko tak dobra, jak dobry jest programista [9]; (3) trudne jest ponowne wykorzystanie pracy wykonanej na potrzeby jednej aplikacji, gdy powstaje potrzeba pisanie kolejnej aplikacji nawet korzystającej z tych samych źródeł danych.

Najnowsze propozycje dotyczą integracji *na poziomie danych* lub *integracji dano-centrycznej*. Systemy tej klasy określane są jako EII (ang. *Enterprise Information Integration*) [7], [10], [9]. W [7] podejście to określane jest jako „Big I" i zgodnie z nim powinno się dążyć do utworzenia pojedynczego systemu dla wszystkich potrzeb integrowania (silnik integracji), który akceptuje nieproceduralną specyfikację tych potrzeb i automatycznie wybiera właściwą metodę (rozwiązanie) lub kombinację metod.

Badania prowadzone w Instytucie Automatyki i Inżynierii Informatycznej Politechnik Poznańskiej dotyczą wspomnianej powyżej integracji zorientowanej na dane. Jednym z głównych problemów jest wtedy opracowanie nieproceduralnych metod specyfikowania wymagań w zakresie integracji danych. Niezbędne jest wówczas odwoływanie się do problemów znaczenia (semantyki) danych oraz definiowanie procesów semantycznej integracji danych. Prace w tym zakresie realizujemy w ramach projektu SIX-P2P (*Semantic Integration of Xml data in Peer-To-Peer environment*) (grant KBN 1553).

## Literatura

1. Abiteboul, S., Hull, R., Vianu, V., Foundations of Databases, Addison-Wesley, Reading, Massachusetts, 1995.
2. Arenas, M., Libkin, L., XML Data Exchange: Consistency and Query Answering, PODS Conference, 2005, 13–24.
3. Brzykcy G., Bartoszek J., Pankowski T.: Semantic data integration in P2P environment using schema mappings and agent technology, Poznań, 2006 (w druku).
4. Buneman, P., Davidson, S. B., Fan, W., Hara, C. S., Tan, W. C., Reasoning about keys for XML, Information Systems 28 (8), 2003, 1037–1063.
5. Enterprise Data Integration, http://www.informatica.com/.

6. Fagin, R., Kolaitis, P. G., Popa, L., Data exchange: getting to the core, ACM Trans. Database Syst. 30 (1), 2005, 174–210.
7. Haas L., Beauty and the Neast: The Theory and Practice of Information Integration, Database Theory – ICDT 2007, Lecture Notes in Computer Science 4353, Springer 2007, 28–43.
8. Haas L., i in. Clio grows up: from research prototype to industrial tool. SIGMOD Conference, 2005, 805–810.
9. Halevy A., Franklin M., Maier D., Principles of dataspace systems, PODS Conf, 2006, 1–9.
10. Halevy, A. Y., Rajaraman, A., Ordille, J. J., Data Integration: The Teenage Years, VLDB, 2006, 9–16.
11. IBM Business Consulting Services, Your Turn: The Global CEO Study 2004, http://www.bitpipe.com/detail /RES/1129048329_469.html
12. IBM Information Integration, http://www-306.ibm.com/software/data/integration/
13. Koloniari, G., Pitoura, E., Peer-to-peer management of XML data: issues and research challenges, SIGMOD Record 34(2), 2005, 6–17.
14. Krishnaprasad M., i in., Query Rewrite for XML in Oracle XML DB, VLDB 2004, 1122–1133
15. Microsoft BizTalk Server, http://msdn2.microsoft.com/en-us/library/aa286554.aspx.
16. Microsoft MSDN, Extending the Microsoft BizTalk Accelerator for Suppliers to Support the BMEcat Catalog Standard, http://msdn2.microsoft.com/en-us/library/ms978361.aspx.
17. Microsoft SQL Server Integration Services (SSIS), SQL Server 2005 Books Online, http://msdn2.microsoft.com/en-us/library/ms141026.aspx.
18. Namyoun Choi, I., Han, H.: A Survey on Ontology Mapping, SIGMOD Record 35 (3), 2006, 34–41.
19. Pankowski T., Cybulka J., Meissner A., XML Schema Mappings in the Presence of Key Constraints and Value Dependencies, Emerging Research Opportunities for Web Data Management, Database Theory ICDT 2007 Workshop EROW 2007, CEUR Workshop Proceedings Vol. 229, http://CEUR-WS.org/Vol-229/, 1–15.
20. Pankowski T., Management of executable schema mappings for XML data exchange, In: Database Technologies for Handling XML Information on the Web, EDBT 2006 Workshop DataX, Lecture Notes in Computer Science 4254, Springer 2006, 264–277.
21. Stylus Studio 2007, http://www.stylusstudio.com
22. XML Schema Part 1: Structures: 2004. www.w3.org/TR/xmlschema-1.
23. XQuery 1.0: An XML Query Language. W3C XQuery 1.0: An XML Query Language W3C Proposed Recommendation 2006, http://www.w3.org/TR/xquery/.
24. Yuan, J., Bahrami, A., Wang, C., Murray, M. O., Hunt, A.: A Semantic Information Integration Tool Suite, VLDB, 2006, 1171–1174.