Adam Pelikant

Instytut Mechatroniki i Systemów Informatycznych
Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Politechnika Łódzka

ZASTOSOWANIE TYPÓW OBIEKTOWYCH W PRZETWARZANIU ANALITYCZNYM

Wstęp

O wadze prowadzenia przetwarzania analitycznego we współczesnych systemach gromadzenia danych nie trzeba w zasadzie nikogo przekonywać. Należy jednak podkreślić rzadko uzmysławiany fakt, że posiadanie nawet bardzo dużych i dobrze zorganizowanych danych z perspektywy biznesowej nie daje żadnej przewagi. Dane to jeszcze nie jest informacja. Konieczne jest wprowadzenie wartości dodanej. Jej najprostszą formą są systemy raportujące, w których wyznaczono zestawy agregatów. Dopiero w takiej formie dane mogą stanowić użyteczne narzędzie „walki" na konkurencyjnym rynku. Oczywiście przejście od prostego raportowania do hurtowni danych może tę wartość dodaną wzbogacić[1]. Jest prostą – wynikającą ze sposobu odbierania przez nas otaczającego świata – prawidłowością fakt, że najdogodniejszą formą prezentacji wyników będzie ich wizualizacja, złożona pod względem

[1] L. Kiełtyka, *Systemy informatyczne zarządzania informacją*, w: *Informatyka gospodarcza*, red. J. Zawiła-Niedźwiecki, Wydawnictwo C.H. Beck, Warszawa 2010, s. 475–506; C. Olszak, *Systemy informatyczne analityczno-raportujące*, w: *Informatyka gospodarcza*, op.cit., s. 445–474; T. Witkowski, *Systemy informatyczne wspomagania podejmowania decyzji*, w: *Informatyka gospodarcza*, op.cit., s. 547–586; G. Zwoliński, M. Kacperski, *Komputerowa organizacja działań wspierająca proces rekrutacji studentów*, „Metody Informatyki Stosowanej" 2010, nr 3, Polska Akademia Nauk Oddział w Gdańsku, Komisja Informatyki.

programistycznym, ale łatwo postrzegana przez poziom zarządczy firm[2]. Zobaczyć dane ooznacza więcej niż otrzymać dziesiątki podsumowań. Na końcu tej drabiny metod analizy znajdują się metody zgłębiania danych, które przekształcają surowe lub wstępnie przetworzone dane we wskaźniki, klasyfikatory, które mogą być bezpośrednim wsparciem w podejmowaniu decyzji[3]. Niestety, nie wszystkie funkcjonalności niezbędne podczas prowadzenia złożonych analiz są wbudowane w środowiska serwerów baz danych. W takim przypadku pomocne okazują się typy obiektowe, które pozwalają na znaczne rozszerzenie oferowanych metod analizy oraz zaimplementowanie własnych algorytmów[4].

Możemy również stwierdzić, że usługa IT dostarcza wartość biznesowi[5], gdy jednocześnie zapewnia odpowiednią:

- użyteczność – opisującą zakres usługi, jej funkcjonalność (czemu usługa służy);
- gwarancję – definiującą jakość usługi (w jaki sposób i na jakich zasadach usługa będzie dostarczana).

Zastosowanie typów obiektowych ma znaczny wpływ na powiększenie użyteczności i dopasowania funkcjonalnego oferowanych dla rozwiązań biznesowych, w tym również administracji oraz służby zdrowia. Ramy doskonalenia procesów dedykowanych dla organizacji usługowych (CMMI for Services), opracowane przez Software Engineering Institute (SEI) w Carnegie Mellon University w USA 2009 roku, mają za zadanie[6]:

- umożliwić dostawcom oszacowanie ich możliwości w odniesieniu do dostarczania usług IT;
- wskazać dostawcom usług IT niezbędne kroki do dalszego doskonalenia potencjału usługowego.

Możemy stwierdzić, że zastosowanie prezentowanych tutaj rozwiązań wypełnia drugi postulat tych zasad. Niestety, przejmowanie w komercji nowego potencjału

---

[2] S. Brecheisen, H.P. Kriegel, P. Kröger, M. Pfeifle, *Visually Mining Through Cluster Hierarchies*, Proc. SIAM Int. Conf. on Data Mining (SDM '04), Lake Buena Vista 2004, s. 400–412; D. Pérez, M.J. Somodevilla, I.H. Pineda, *Fuzzy Spatial Data Warehouse: A Multidimensional Model*, w: *Advances Decision Support Systems*, red. G. Devlin, INTECH, Croatia 2010, s. 342.

[3] A. Kowalczyk, A. Pelikant, *Implementation of Automatically Generated Membership Functions Based on Grouping Algorithms*, The IEEE Region 8 Eurocon 2007 Conference, Warsaw, 9–12 September 2007; K. Pabjańczyk, A. Pelikant, *Implementacja algorytmów użytkownika w środowisku Business Intelligence SQL Server 2008*, „Studia Informatica" (Gliwice) 2009, vol. 30, no. 2B (84), Silesian University of Technology Press, s. 347–358.

[4] M. Agata, A. Pelikant, *Support methods for weak learning algorithms – Adaboost*, XII International Conference System Modelling and Control, Zakopane, 17–19 October 2007; A. Pelikant, *Hurtownie danych. Od przetwarzania analitycznego do hurtowni danych*, Helion, 2011; A. Pelikant, *Systemy gromadzenia danych*, w: *Informatyka gospodarcza*, op.cit., s. 409–444; D. Pérez, M.J. Somodevilla, I.H. Pineda, op.cit., s. 342.

[5] S. Pushpa, S. Meenakshi, *Implementation of Object Oriented Data Warehousing using a Narrower Compassed Data Model in Oracle 10g*, „International Journal of Computer Applications" 2011, vol. 17, no. 5, March, s. 26–29.

[6] A. Sobczak, *Przegląd wybranych podejść do zarządzania IT w organizacjach*, Komputerowo Zintegrowane Zarządzanie, Zakopane 2010, s. 456–467; A. Sobczak, *Modele i metamodele w architekturze korporacyjnej*, w: *Systemy wspierania organizacji*, Katedra Informatyki, Akademia Ekonomiczna, Katowice 2009.

jest stosunkowo wolne, gdyż wymaga poniesienia dodatkowych nakładów oraz nie pozwala na pełną utylizację już opracowanego oprogramowania.

Zastosowanie typów obiektowych w przetwarzaniu analitycznym na serwerach transakcyjnych OLTP

We współczesnych, komercyjnych bazach danych istnieje zestaw funkcji pozwalających na prowadzenie analiz. Jednakże w większości z nich jest on ograniczony do podstawowych funkcji agregujących. W stanowiącym podstawę niniejszych rozważań środowisku MS SQL są to: suma, średnia, minimum, maksimum, wariancja, odchylenie standardowe oraz wariancja populacji i odchylenie standardowe populacji. Każda z nich może zostać wyznaczona w zapytaniu z określonymi poziomami grupowania, co w przypadku agregacji obliczanych na wielu poziomach prowadzi do budowania złożonych struktur z wielokrotnie łączonymi podzapytaniami. Możliwe jest również zastosowanie opcji ROLLUP, CUBE lub GROUPING SETS, znacznie upraszczających składnie. Najnowszym rozwiązaniem jest wyznaczenie ich nad partycją, oknem logicznym OVER(PARTITION BY...). Rozwiązanie takie pozwala na stosowanie jako bazy zwykłego zapytania wybierającego, bez grupowania – konieczne jest tylko poprawne zrealizowanie złączeń. Niestety, takie rozwiązanie jest stosunkowo nowe (w MS SQL wprowadzono je dopiero w wersji 2008). Zestaw funkcji analitycznych uzupełniają funkcje rangowe wyznaczane w definicji tylko nad oknem logicznym, w którym konieczne jest dodatkowe określenie porządku sortowania względem cechy służącej do sporządzenia rankingu.

Z perspektywy potrzeb praktycznych zestaw funkcji jest bardzo ubogi. Rozwiązaniem problemu jest możliwość utworzenia funkcji agregujących użytkownika. W przypadku omawianego środowiska jest to realizowane przez utworzenie klasy obiektowej (struktury) z zastosowaniem dowolnego języka platformy.NET, która jest kompilowana do postaci biblioteki *.dll. Formalna postać takiej klasy jest ściśle określona, co przedstawia listing 1.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Collections;
using Microsoft.SqlServer.Server;

{Serializable}
{SqlUserDefinedAggregate(Format.UserDefined,
```

```
IsInvariantToDuplicates = false,
IsInvariantToOrder= false,
MaxByteSize =8000 , Name=”Covar”)}
public struct cowariancja : IBinarySerialize
{
public List<double> posr;
public List<double> posr1;
private double temp;
...
public void Init()
{...}
public void Accumulate(double? Value)
{...}
public void Merge(cowariancja Group)
{...}
public SqlDouble Terminate()
{...
return (this.temp);}
public void Write(System.IO.BinaryWriter w)
{...
w.Write(temp);}
public void Read(System.IO.BinaryReader r)
{...
temp = r.Read();}
End Class
```

Listing 1. Schemat tworzenia klasy wykorzystywanej do tworzenia funkcji agregującej

Poza dołączeniem niezbędnych przestrzeni nazw oraz zdefiniowaniem dyrektywy dla kompilatora wskazującej na sposób wykorzystania klasy konieczne jest przeciążenie co najmniej trzech metod[7]:

- Init() – wykonywanej na początku każdej grupy lub partycji logicznej, służącej do zainicjowania początkowych wartości (wyczyszczenia starych wartości pozostałych po obliczeniach dla poprzednich grup).
- Accumulate (double? Value) – wykonywanej cyklicznie dla każdego rekordu grupy lub partycji. Metoda ta ma zdefiniowany co najmniej jeden parametr zgodny co do typu z danymi, dla których agregacja będzie wykonywana. Ponieważ część

[7] A. Pelikant, *Hurtownie danych*, op.cit.

danych po stronie relacyjnej może nie mieć określonej wartości, zastosowano typ pozwalający na przyjęcie takich wartości.
- Terminate() – statycznej metody, w której są wykonywane obliczenia po przejściu wszystkich rekordów grupy. Zwykle służy ona do finalnego sformatowania wyniku.
- Merge(struktura Group) – metody, która może pozostać nieprzeciążona, ponieważ jest wykonywana tylko wtedy, gdy są wykonywane obliczenia w procesach równoległych. W metodzie tej następuje połączenie wyników z rozwidlonych obliczeń. Przekazywana do niej zmienna musi mieć typ taki sam jak definiowana struktura (klasa).

W przypadku stosowania w definicji struktury zmiennych o niestandardowych typach (w pokazywanym przykładzie są nimi listy wartości, również należą do nich zmienne łańcuchowe) użytkownik musi zapewnić sposób serializacji. Realizacja tego odbywa się przez przeciążenie metod Read i Write odpowiednio nad typami IO.BinaryReader oraz IO.BinaryWriter. Metody te pobierają i zwracają zmienną zgodną co do typu z wartością zwracaną przez funkcję agregującą.

Drugim krokiem jest inkapsulacja skompilowanej klasy do obiektu proceduralnego środowiska T-SQL – listing 2.

```
DROP AGGREGATE GeoAvg_v
GO
DROP ASSEMBLY {GeoAvg_a_v}
GO
CREATE ASSEMBLY {GeoAvg_a_v}
AUTHORIZATION {dbo}
FROM 'C:\sciezka\biblioteka.dll'
WITH PERMISSION_SET = SAFE
GO

CREATE AGGREGATE GeoAvg_v(@value float)
RETURNS float
EXTERNAL NAME GeoAvg_a_v.{sred_geom_v.GEOAVE};
GO
```

Listing 2. Mapowanie typu obiektowego do funkcji agregującej TSQL

Proces mapowania jest dwuetapowy i składa się z utworzenia elementu pośredniczącego *assemblies* oraz właściwego elementu proceduralnego. W definicji *assemblies* są podawane – kwalifikowana ścieżka do skompilowanej biblioteki *.dll oraz

sposób ochrony dostępu do tej biblioteki[8]. Należy zaznaczyć, że biblioteka może zawierać wiele klas (struktur) obiektowych definiujących różne co do funkcjonalności byty. Kolejny krok to utworzenie właściwego elementu proceduralnego – w prezentowanym przykładzie funkcji agregującej. Tym razem każdy element jest tworzony oddzielnie, nawet wtedy, gdy mapowanie jest realizowane przez ten sam obiekt pośredniczący.

W toku prowadzonych badań została zrealizowana biblioteka funkcji agregujących dla środowiska MS SQL Server 2008. Poza utworzoną do celów weryfikacji stosowanych metod, analizy czasów wykonania i poprawności wyników funkcją wyznaczającą odchylenie standardowe skonstruowano funkcje wyznaczające: średnią geometryczną, skośność, kurtozę, momenty centralne rzędu n, kowariancje i korelacje Pearsona.

Analizy statystyczne wyznaczane na poziomie transakcyjnym powinny być stosowane w systemach o stosunkowo niewielkim obciążeniu. Dlatego w przypadku wdrożeń w administracji powinny dotyczyć warstwy pośredniej (*Middleware*) pracującej na rzecz urzędu, a nieobsługującej transakcji pochodzących bezpośrednio od klientów.

Prace nad rozbudową biblioteki o kolejne funkcjonalności trwają. Na uwagę zasługuje porównanie dokładności wyników w przypadku wyznaczania współczynnika korelacji, który można obliczyć, używając w wyrażeniu standardowych funkcji agregujących zgodnie z zależnością (1).

$$CORR(x,y) = \frac{E(x^2) + E(y^2) - E(xy)}{\sigma(x)\sigma(y)} \tag{1}$$

W tym przypadku otrzymane rezultaty były obarczone dużym błędem wynikającym z zaokrągleń wyników elementarnych funkcji wyrażenia. Wyniki otrzymane za pomocą funkcji użytkownika były dokładne. Również czas obliczeń przy zastosowaniu zaproponowanego rozwiązania wykorzystującego CLR był znacząco krótszy.

W konkurencyjnym rozwiązaniu komercyjnym, jakim jest serwer ORACLE, liczba wbudowanych funkcji agregujących jest znacznie większa (gwałtowny przyrost odnotowano począwszy od wersji 10g). Jednak i tutaj wprowadzony został mechanizm tworzenia funkcji agregujących definiowanych przez użytkownika[9]. Pomimo iż stosuje się na tej platformie struktury obiektowe tworzone bezpośrednio w języku PL/SQL, to główna idea ich budowy jest analogiczna.

[8] Ibidem.

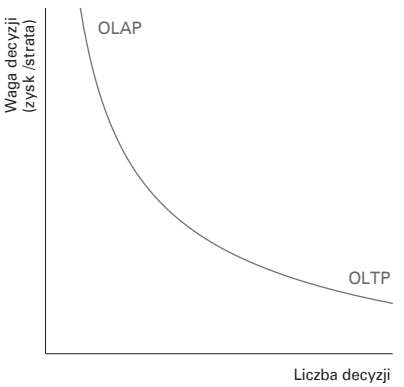
[9] D. Pérez, M.J. Somodevilla, I.H. Pineda, op.cit., s. 342.

Wielowymiarowe struktury analityczne – hurtownie danych

Mimo że możliwości prowadzenia analiz po stronie transakcyjnej OLTP są bardzo duże[10], w rozbudowanych systemach nie korzysta się z nich bezpośrednio[11]. Głównymi przyczynami takiego postępowania są następujące fakty:

1. Zapytania analityczne są bardzo złożone, co powoduje, że wykonują się stosunkowo wolno i wymagają dużej ilości zasobów pamięci, a to znacznie obciąża systemy transakcyjne OLTP.
2. Analizy mogą dotyczyć danych gromadzonych na różnych serwerach i różnych systemach, co wymaga albo zastosowania przetwarzania rozproszonego, albo migracji danych między systemami.
3. Ze względu na formalne różnice sposobów zapisu danych (np. różne kalendarze u różnych dostawców baz danych) oraz różnice w metodzie słownikowania dane podczas migracji powinny zostać zintegrowane do wspólnej postaci.
4. Nie wszystkie dane istotne z punktu widzenia transakcyjnego są istotne dla prowadzenia analiz, co wymaga wstępnego filtrowania danych.

Z tych powodów stosuje się przetwarzanie analityczne OLAP, wykorzystujące z reguły mechanizmy hurtowni danych, dla których wykorzystuje się oddzielne, odseparowane instancje serwerów baz danych, zainstalowane na oddzielnych względem OLTP komputerach[12].



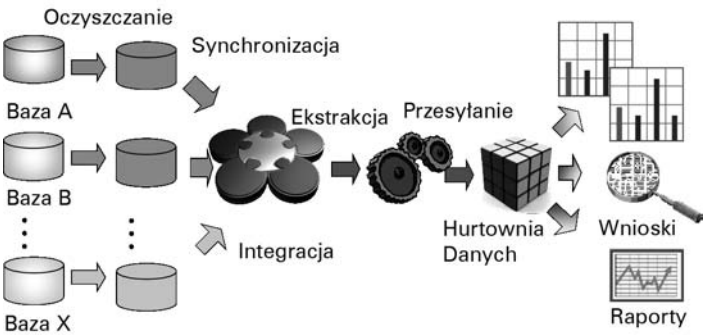
Wykres 1. Miejsce przetwarzania analitycznego OLAP oraz transakcyjnego OLTP w zależności od poziomu szczebla zarządzającego

[10] A. Pelikant, *Systemy gromadzenia danych*, op.cit., s. 409–444.

[11] G. Zwoliński, M. Kacperski, op.cit.

[12] C. Olszak, op.cit., s. 445–474; L. Kiełtyka, op.cit., s. 475–506; T. Witkowski, op.cit., s. 547–586.

Tego typu systemy są przeznaczone dla szczebli zarządczych firm, co ilustruje wykres 1. Należy podkreślić, że dotyczy to również zadań zarządzania na poziomie administracji publicznej i służby zdrowia. Jednakże wydaje się, że instytucje tej sfery na razie są dopiero na etapie budowania i wykorzystywania transakcyjnych systemów gromadzenia danych, a zapotrzebowanie na analizę będzie dopiero kolejnym krokiem. Na wykresie 1 widać również, że ponieważ na wysokim szczeblu zarządzania decyzje są podejmowane niezbyt często, a ich liczba jest stosunkowo niewielka, systemy analityczne nie muszą pracować na danych aktualnych względem zatwierdzonych transakcji w bazowych systemach OLTP. Daje to czas niezbędny do przeprowadzenia wstępnego przygotowania, przetworzenia danych źródłowych.



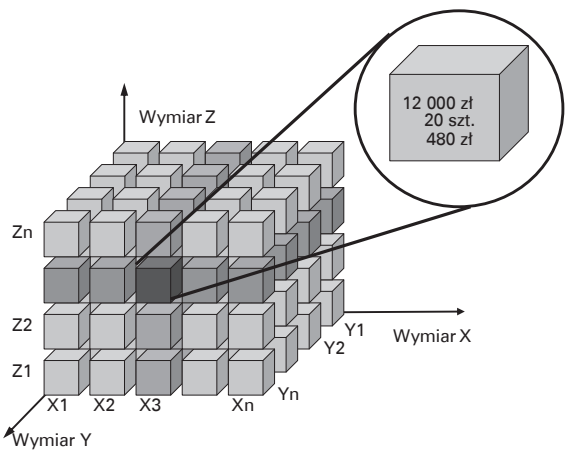
Rysunek 1. Ogólna idea tworzenia i działania systemów przetwarzania analitycznego

Ogólne założenia dla systemów przetwarzania analitycznego mówią, że dane dla nich pochodzą z wielu różnych, pod względem zarówno przeznaczenia, jak i zastosowanych, rozwiązań i platform programistycznych, źródeł transakcyjnych. Zakłada się ponadto, że dane w tych systemach są „brudne", czyli zawierają wpisy, które nie zawierają istotnej informacji, a z których istnieniem godzimy się w systemach transakcyjnych, np.: puste rekordy, nie w pełni wypełnione dane, ślady po częściowo wycofanych transakcjach etc. Tak jak wskazano wcześniej, dane te mogą być zapisane w różnych formatach, przy zastosowani różnych typów danych, co wymaga ich zsynchronizowania, a w kolejnym etapie doprowadzenia do jednolitej postaci pośredniej – integracji. Końcowym etapem takiego działania jest załadowanie przygotowanych danych do systemu (systemów) analitycznego, np. hurtowni danych, co przedstawia rysunek 1. Omawiany proces nazywany jest zwyczajowo ETL – *extract*, *transform*, *load*.

Dane składowane w hurtowni danych podlegają przetworzeniu, a wyniki są prezentowane użytkownikowi w czytelnej dla niego postaci – raportu. Ponieważ ludzka percepcja preferuje informacje wizualne, najczęściej wyniki są przekazywane w postaci graficznej – różne warianty wykresów, wykresów przestawnych, wskaźników

sukcesu, trendu, map. Te wysoce przetworzone i zagregowane dane bardzo często stanowią źródło dla systemów wspomagania decyzji czy systemów zgłębiania danych, które oferują wskazówki pozwalające na podjęcie właściwej na danym szczeblu zarządczym decyzji. Można powiedzieć, że oferują one już nie dane, ale na skutek zastosowania złożonych metod analizy generujących „wartość dodaną" – informację. Niestety, pomimo wysokiej jakości rozwiązań komercyjnych informacje te, a przede wszystkim wypracowana na ich podstawie decyzja, są weryfikowane *ex post*, co wymusza na twórcach dużą dbałość o jakoś zaproponowanych metod analitycznych.

W uproszczeniu można przedstawić przetworzoną strukturę hurtowni danych jako trójwymiarową kostkę w przestrzeni euklidesowej (rysunek 2). Każdy z wymiarów reprezentuje opis pewnego zestawu atrybutów określających zagadnienie biznesowe i może być charakterystyczny dla tego problemu. Najczęściej w tego typu strukturach pojawiają się jednak pewne kategorie wymiarów opisujących: czas (transakcji, dostawy, ekspedycji), położenie geograficzne (miejsce ekspedycji, miejsce sprzedaży, pochodzenie klienta), towar (rozumiany szeroko – jako podmiot transakcji, usługa, funkcjonalność). Jak widać, ten sam rodzaj wymiaru może opisywać różne atrybuty formalne. Dodatkowo można stwierdzić, że wymiary nie muszą być ortogonalne, ponieważ nietrudno zauważyć, iż np. czas transakcji i czas dostawy są ze sobą silnie dodatnio skorelowane (a przynajmniej powinny być). Dodatkowo problem stanowi dyskretyzacja danych ciągłych opisujących wymiar, porządkowanie i grupowanie danych kategorycznych czy też głębokość reprezentacji danych hierarchicznych. Problemy te są oddzielnie poruszane w innych publikacjach[13].



Rysunek 2. Idea przechowywania danych w hurtowni danych – ze względów formalnych ograniczono się do struktury trójwymiarowej

[13] A. Kowalczyk, A. Pelikant, op.cit.; C. Olszak, op.cit., s. 445–474; A. Pelikant, *Systemy gromadzenia danych*, op.cit., s. 409–444.

Załóżmy jednak dla uproszczenia, że na rysunku 2 przedstawiono dane mające reprezentację dyskretną, stąd wybranie jednej wartości na osi generuje przecięcie struktury wielowymiarowej. Wybranie wartości atrybutów na wszystkich wymiarach wskazuje elementarną komórkę, w której są przechowywane zmaterializowane agregaty. Takie przygotowanie struktury powoduje, że wykonanie zapytania wybierającego jest bardzo wydajne, ponieważ wymaga tylko odczytania przetworzonych danych. Praktycznie agregaty są wyznaczane na poziomie każdego poziomu hierarchii definiującej wymiar, również dla węzła najwyższego – dla całej kostki. Inny problem stanowi rodzaj wyznaczanych funkcji agregujących. Ze względu na wydajność wymaga się, aby były to funkcje addytywne (łączne), jak suma czy liczebność, albo semi addytywne, tzn. takie, jakie da się wyznaczyć przy pomocy wyrażeń zawierających tyko funkcje addytywne. W jednym i drugim przypadku dodanie nowych danych do hurtowni umożliwia przyrostowe przeliczenie agregatów, a nie przeliczanie całej hurtowni od początku, chociaż i takie przetworzenie może zostać wymuszone na życzenie twórcy hurtowni.

Również w przypadku narzędzi OLAP istnieje możliwość wykorzystania klas obiektowych[14]. Jednak tutaj możliwe jest użycie tylko zwykłych funkcji zwracających wartość skalarną albo zmienną tabelaryczną (tabelę). Przykład takie prostej klasy zawiera listing 3.

```
using System;
using Microsoft.SqlServer.Server;
using System.Collections;
using System.Data;
using System.Data.Sql;
using System.Data.SqlTypes;
using System.Data.SqlClient;
public class wynik{
{Microsoft.SqlServer.Server.SqlFunction}
public static double? reszta(double? a, double? b)
{double? wyn = (double?)a % b;
return wyn;}
public static double resztaa(double a, double b)
{double wyn = (double)a % b;
return wyn;}
}
```

Listing 3. Sposób budowania klasy wykorzystywanej do tworzenia funkcji dla potrzeb systemów analitycznych i transakcyjnych

[14] A. Pelikant, *Hurtownie danych*, op.cit.

Przykład zawiera dwie proste klasy wyznaczające resztę z dzielenia dwóch liczb będących jej argumentami. Różnią się one zastosowanym typem danych wejściowych i wyniku. W pierwszym przypadku są to typy umożliwiające przyjęcie wartości NULL, co dedykuje klasę dla analiz po stronie transakcyjnej. W drugim – typy niezezwalające na przyjęcie wartości NULL, co umożliwia ich zastosowanie w przetwarzaniu OLAP, ponieważ są w tym przypadku niedopuszczalne. Zastosowanie klasy obiektowej wymaga tylko jej zarejestrowania do *assemblies*.

```
WITH MEMBER Measures.rest as
funkcja.resztaa({Measures}.{Ilosc}, {Measures}.{Wartosc})
SELECT {{Measures}.{Wartosc},rest} ON COLUMNS
FROM {Zysk}
```

Listing 4. Zastosowanie funkcji z klasy obiektowej w zapytaniu MDX odpytującym strukturę wielowymiarową

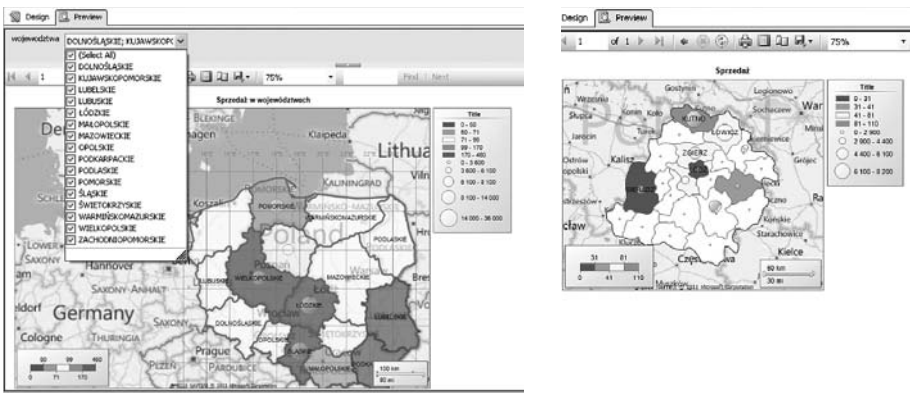
Tak zdefiniowane funkcje mogą być użyte do formułowania zapytań do struktur wielowymiarowych w języku MDX SQL (listing 4). W przypadku funkcji skalarnych dopuszczalne jest to tylko w sekcji WITH, definiującej miary *ad hoc* lub zestawy atrybutów generowanych podczas przetwarzania.

Przeciwnie do przetwarzania analitycznego OLTP, hurtownie danych pozwalają na analizy na podstawie danych pochodzących z systemów o dużym ruchu, ponieważ odciążają tę warstwę. Wdrożenia mogą więc dotyczyć przetwarzania danych pochodzących bezpośrednio od klientów urzędów.

Zastosowanie obiektowych typów definiujących grafikę wektorową – *spatial*

Na takich samych zasadach, na jakich buduje się obiekty proceduralne z zastosowaniem klas CLR, zostały zaimplementowane przez producenta serwera typy pozwalające na przechowywanie obiektów graficznych – *spatial*. Są one podzielone na dwa typy – *geometry* i *geography*, różniące się w zasadzie sposobem odwzorowania. W pierwszym przypadku są to współrzędne kartezjańskie, w drugim współrzędne sferyczne (na globusie) z dodatkowym zestawem informacji określającym parametry narodowe, tzw. SRID.

Oba typy złożone mogą zostać zastosowane do prowadzenia analiz po stronie zarówno transakcyjnej, jak i analitycznej[15]. Wydaje się jednak, że typ reprezentujący geografię ma szersze zastosowanie praktyczne i jest bardziej spektakularny podczas prezentacji wyników. Przykładem jest prowadzona dla firmy handlowej analiza, której jednym z elementów jest określenie poziomu sprzedaży dla klientów pochodzących z różnych miejscowości w kraju. Dane na potrzeby analizy uzyskano na podstawie rzeczywistego podziału administracyjnego kraju. Natomiast zarówno asortyment, jak i poszczególne transakcje sprzedaży zostały wygenerowane za pomocą autorskiego generator zawartości baz, odwołującego się do słowników, różnych algorytmów pseudolosowych pozwalających na uzyskanie dystrybucji zgodnych z wybranymi rozkładami. Zakres generowania danych został znacznie ograniczony, co spowodowało, że dla niektórych alokacji nie uzyskano żadnych wartości. Ma to uzasadnienie w rzeczywistych rozkładach, sprawdza też „odporność" stosowanych metod na pojawienie się takich wartości.



Rysunek 3. Prezentacja wyników analizy w połączeniu z typami obiektowymi reprezentującymi grafikę wektorową (*spatial*) przedstawiającą podział administracyjny Polski

W analizie zastosowano połączenie danych *spatial* przechowywanych w oddzielnej tabeli wygenerowanych na podstawie map w formacie *.shp, których konwersja odbyła się za pomocą darmowego narzędzia Shape2SQL. Połączenie między danymi

[15] M. Ester, S. Gundlach, H.P. Kriegel, J. Sander, *Database Primitives for Spatial Data Mining*, Proc. 8. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'99) (Int. Conf. on Databases in Office, Engineering and Science), Freiburg 1999, s. 137–150; M. Ester, A. Frommelt, H.P. Kriegel, J. Sander, *Algorithms for Characterization and Trend Detection in Spatial Databases*, Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98), New York 1998, s. 44–50; M. Ester, H.P. Kriegel, J. Sander, *Spatial Data Mining: A Database Approach*, Proc. 5th Int. Symposium on Large Spatial Databases (SSD'97), Berlin 1997, s. 47–66; A. Pelikant, *Bazy danych w zastosowaniach praktycznych*, WSInf, Łódź 2007; A. Pelikant, *Systemy gromadzenia danych*, op.cit., s. 409–444.

analitycznymi zostało zrealizowane przez nazwy na poziomie: województw, powiatów oraz miast. Wyniki zostały zaprezentowane w MS SQL Server 2008 R2 Reporting Services, która zawiera wbudowane narzędzia warstwy prezentacji operujące na mapach. Przykłady takiego raportu ilustruje rysunek 3.

Na przykładowych raportach jest widoczna możliwość zastosowania jako tła rzeczywistej mapy dostępnej w serwisie (tutaj mapa konturowa, która może być zastąpiona zdjęciem satelitarnym lub kombinacją tych dwóch map). Można również zastosować mapy z innych, własnych źródeł. Zaprezentowano możliwość filtrowania wielowartościowego na poziomie województw. Zaprezentowano dwie wartości analityczne: za pomocą koloru tła – wartość sprzedaży, a za pomocą wielkości punktów (kół) centralnych – ilość sprzedanych towarów w wybranym okresie. Możliwe jest potencjalne przypisanie trzeciej wielkości analitycznej do koloru punktów centralnych. W zrealizowanym rozwiązaniu zastosowano akcję przypisaną do kształtu województwa, pozwalającą na ograniczenie zasięgu wyświetlanych analiz do wybranego obszaru i przeniesienie na bardziej szczegółowy poziom reprezentujący powiaty. Postępowanie takie odpowiada drążeniu danych – *drill down*. Definicje akcji mogą być związane z innymi funkcjonalnościami, jak choćby przeniesieniem do strony WWW o wskazanym dynamicznie adresie wyszukiwarki z dynamicznym pytaniem (również wyszukiwarki map) czy też prezentacją tabelaryczną wyników niekoniecznie bezpośrednio związanych z wyjściową analizą.

Zastosowanie raportowania wykorzystującego prezentację na mapie jest bardzo użyteczne nie tylko w przypadku aplikacji dla biznesu, lecz także w innych dziedzinach. Każda jednostka administracyjna jest przecież związana z jakimś obszarem, a wizualizacja danych powoduje, że są one łatwiejsze do analizy przez gremia decyzyjne. Najbardziej oczywiste wydaje się odniesienie do geodezji, ale równie użyteczne jest dla utrzymania infrastruktury, takiej jak: drogi, sieci wodociągowe, ciepłownicze, media. Także prezentacja zagrożeń związanych z rozprzestrzenianiem się epidemii, wylewami wód może być w ten sposób szybko i skutecznie opracowywana[16].

Dane geograficzne mogą również stanowić elementy modeli zgłębiania danych[17]. W pracy zaprezentowano to w postaci drzewa decyzyjnego generowanego z zastosowaniem entropii Shanona (2) – rysunek 4.

$$E(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{n} p_i \lg(p_i) \tag{2}$$

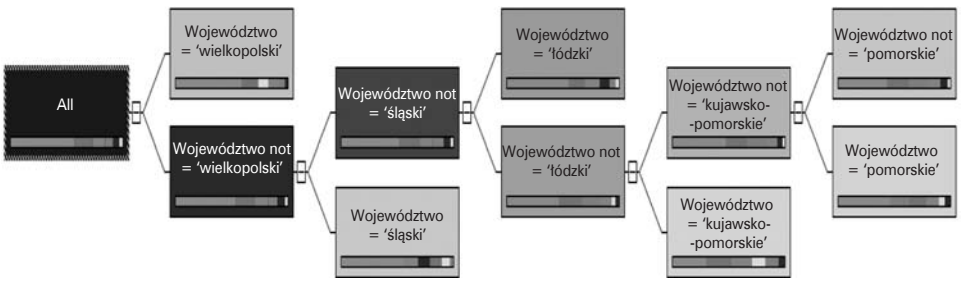
[16] J.M. Czajkowski, *Nowe rozumienie e-usług. Interoperacyjna wielokanałowa platforma kontaktów miasta i mieszkańców*, XIV konferencja „Miasta w Internecie", Zakopane 2010.

[17] M. Agata, A. Pelikant, op.cit.; A. Kowalczyk, A. Pelikant, op.cit.; H.P. Kriegel, M. Renz, M. Schubert, A. Züfle, *Statistical Density Prediction in Traffic Networks*, Proc. 8th SIAM Conf. on Data Mining (SDM 2008), Atlanta 2008, s. 692–703; M. Ester, H.P. Kriegel, J. Sander, *Spatial Data Mining: A Database Approach*, op.cit., s. 47–66; T. Witkowski, op.cit., s. 547–586.

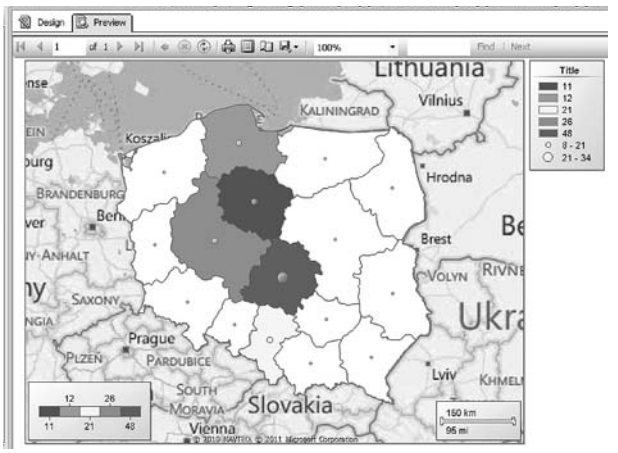
Model opierający się na klasyfikatorze Bayesa generował „płytkie drzewo" ze względu na małą różnorodność danych otrzymanych w procesie generacji.

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)} \tag{3}$$

Dystrybucja wartości parametru analitycznego (ilość sprzedanych towarów) jest pokazana za pomocą mapy barwnej zawartej w każdym węźle drzewa. Zastosowano domyślną, ciągłą reprezentację tego atrybutu, co spowodowało wygenerowanie przedziałów wartości zgodnie z metodą *k-means*.



Rysunek 4. Drzewo decyzyjne wygenerowane na podstawie analizy danych pochodzących z hurtowni danych



Rysunek 5. Prezentacja danych pochodzących z systemu zgłębiania danych, hurtowni danych w połączeniu z typami obiektowymi reprezentującymi grafikę wektorową (*spatial*) przedstawiającą podział administracyjny Polski

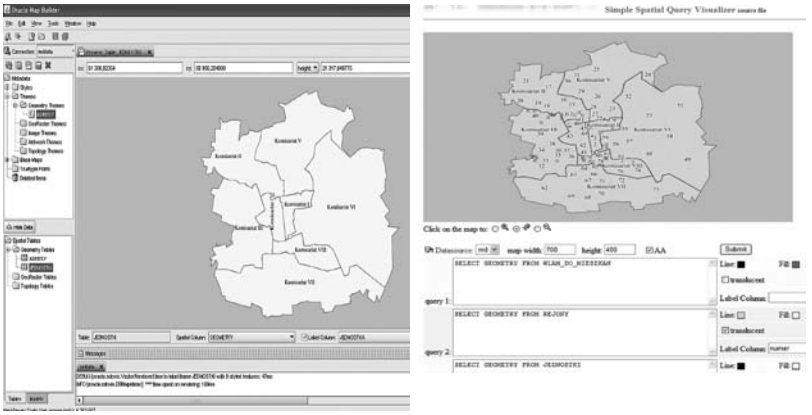
Model zgłębiania może stanowić oś (wymiar) w kostce hurtowni danych, a ta może zostać wizualizowana na zasadach przedstawionych poprzednio. W tym przypadku kolorem tła zilustrowano liczbę atrybutów węzła, a wielkością punktu wagę węzła (rysunek 5). Pomimo formalnego podobieństwa otrzymanych wyników prezentują one zupełnie inne wielkości.

Praktyczna realizacja w administracji

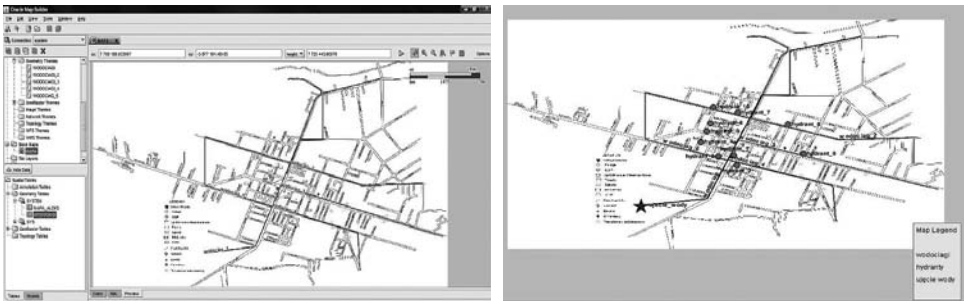
Jednymi z pionierów wprowadzania rozwiązań dla administracji publicznej operujących na danych przestrzennych są: Urząd Miasta Łodzi i działający w nim Wojewódzki Ośrodek Dokumentacji Geodezyjnej i Kartograficznej (http://modgik.lodz.pl/), który wprowadził system informatyczny ZSI Magistrat 2000. Zawiera on wiele podsystemów: finansowo-budżetowy, wymiar i windykacja podatków oraz opłat lokalnych, ewidencja gruntów budynków i lokali, moduł obsługi świadczeń rodzinnych, informowania zarządu. Oferuje on również system informacji przestrzennej, na który składa się wiele warstw tematycznych i który jest dostępny przez Internet. Wykorzystuje on system ewidencji gruntów, a na poziomie serwera bazę danych ORACLE. Niestety, należy zauważyć, że inne ośrodki nie są na tak zaawansowanym poziomie wykorzystywania nowoczesnych rozwiązań. Rezultaty prezentowane za pośrednictwem tej platformy, chociaż są bardzo użyteczne, ograniczają się jednak w większości przypadków do danych transakcyjnych, nieprzetworzonych analitycznie. Wyjątkiem wydaje się mapa akustyczna miasta.

Również w naszym ośrodku były prowadzone prace nad zastosowaniem opisanych metod w praktyce. Wykorzystywane były technologie zarówno MS SQL, jak i ORACLE. Przykład stanowi analiza przestępczości na terenie miasta, w której zostały zastosowane technologie Oracle Map Builder, a po stronie prezentacyjnej Oracle Map Viewer. Ponieważ projekt miał być wykorzystywany jako narzędzie wewnętrzne sekcji informatyki, zastosowano możliwość tworzenia zapytań generujących warstwy mapy, z odpowiednimi podziałami administracyjnymi, oraz zapytania analityczne z wyrażeniami MDX (*multidimensional extension*) dla ostatniej z warstw, która niosła informacje o stanie zagrożenia dla wybranych obszarów (rysunek 6). Z oczywistych przyczyn zamieszczone w pracy dane nie są rzeczywiste, a zostały wygenerowane dla potrzeb testowych i prezentacyjnych. Ostatnim elementem pracy było przeniesienie całości wyników na stronę WWW. Stanowiła ona końcówkę kliencką dostępną dla pracowników szczebla dowodzenia i zarządzania.

Na podobnych zasadach została przygotowana aplikacja pozwalająca na zarządzanie miejską siecią wodociągów (rysunek 7). Zostały wykorzystane te same technologie. W warstwie danych zastosowano API dostarczone przez producenta serwera baz danych.



Rysunek 6. Zastosowanie narzędzi ORACLE do analizy zagrożenia na terenie miasta



Rysunek 7. Zastosowanie narzędzi ORACLE prezentacji i analizy sieci wodociągów

Przedstawione przykłady nie stanowią spektrum opracowanych aplikacji, a jedynie mają wskazywać na potencjalne obszary zastosowania.

Wnioski

W niniejszym artykule przedstawiono możliwości zastosowania do prowadzenia analiz typów obiektowych definiowanych zarówno po stronie transakcyjnego serwera baz danych, jak i po stronie hurtowni danych. Szczególne miejsce przyznano zastosowaniu typów obiektowych reprezentujących grafikę wektorową, ukazując przede wszystkim zastosowanie praktyczne typów *geography*. Główną ideą przyświecającą przedstawieniu tej problematyki jest stosunkowo małe jej rozpowszechnienie w dedykowanych rozwiązaniach komercyjnych, głównie koncentrujących się na części transakcyjnej, oraz mała świadomość możliwości, jakie oferują, wśród przedstawicieli środowisk biznesowych. Pomimo że można powiedzieć, iż jest to wiedza czysto

techniczna, to przeniesienie samej świadomości możliwości prezentowanych w pracy rozwiązań uważam za istotną wartość. Ważne jest nie tylko to, co analizujemy, lecz także, w jaki sposób i przy wykorzystaniu jakich narzędzi. Brak świadomości istnienia metod i narzędzi prowadzi do ograniczenia wymagań względem oferowanych rozwiązań biznesowych.

Zakres możliwości narzędzi przetwarzania analitycznego jest znacznie większy i w wielu miejscach został tylko zasygnalizowany[18]. Dotyczy to przede wszystkim możliwości odpytywania hurtowni danych z zastosowaniem języka SQL MDX oraz możliwości predykcji agregatów na podstawie znanych analiz też z zastosowaniem tego rozszerzenia. Podobna sytuacja istnieje w przypadku podniesionych w pracy problemów dotyczących reprezentacji atrybutów i definiowania elementów hurtowni danych. Również dotyczy to dyskusji nad stosowalnością ich modeli formalnych: gwiazda, płatek śniegu, konstelacja faktów. Wiele z nich zostało omówionych w załączonej do pracy literaturze nieco szerszej, niż to wynika z przedstawionych problemów stanowiących zasadniczą podstawę artykułu.

Prace nad stosowaniem typów obiektowych użytkownika stanowiących analogię typów geometrycznych, ale tworzonych od podstaw przez programistę są rozwijane w naszym ośrodku. Dotyczy to budowy typów reprezentujących atrybuty w n-wymiarowej przestrzeni oraz ich grup (klastrów) wraz z metodami wyznaczającymi odległości względem wielu metryk oraz transformacjami tych obiektów. Ma to bezpośrednie zastosowanie w algorytmach zgłębiania danych.

Ostatnio prace dotyczyły reprezentacji w takiej postaci sieci słabo unormowanych. Reprezentacja taka jest przydatna w analizach sieci, ze szczególnym uwzględnieniem sieci społecznościowych.

Literatura

1. Agata M., Pelikant A., *Support methods for weak learning algorithms – Adaboost*, XII International Conference System Modelling and Control, Zakopane, 17–19 October 2007.
2. Brecheisen S., Kriegel H.P., Kröger P., Pfeifle M., *Visually Mining Through Cluster Hierarchies*, Proc. SIAM Int. Conf. on Data Mining (SDM'04), Lake Buena Vista 2004.

[18] M. Kacperski, G. Zwoliński, *Communication with users in the computerized recruitment system of candidates for higher education studies*, International Conference on System Modelling and Control (SMC), Zakopane, October 12–14 2009; L. Kiełtyka, op.cit., s. 475–506; A. Kowalczyk, A. Pelikant, op.cit.; A. Kowalczyk-Niewiadomy, A. Pelikant, *Zagadnienia grupowania w kontekście budowania zapytań rozmytych,* w: *Bazy danych. Rozwój metod i technologii. Architektura, metody formalne i zaawansowana analiza danych*, Wydawnictwa Komunikacji i Łączności, Warszawa 2008, s. 175–186; A. Pelikant, *Hurtownie danych*, op.cit.; A. Pelikant, *Bazy danych w zastosowaniach praktycznych*, op.cit.; S. Pushpa, S. Meenakshi, op.cit., s. 26–29; R. Stępniak, A. Pelikant, *Wykorzystanie interfejsu JAVA API do budowy narzędzi eksploracyjnych z wykorzystaniem Oracle Data Mining na platformie Oracle*, XIII konferencja PLOUGH, Kościelisko, październik 2007.

3. Czajkowski J.M., *Nowe rozumienie e-usług. Interoperacyjna wielokanałowa platforma kontaktów miasta i mieszkańców*, XIV konferencja „Miasta w Internecie", Zakopane 2010.
4. Ester M., Frommelt A., Kriegel H.P., Sander J., *Algorithms for Characterization and Trend Detection in Spatial Databases*, Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98), New York 1998.
5. Ester M., Gundlach S., Kriegel H.P., Sander J., *Database Primitives for Spatial Data Mining*, Proc. 8. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'99) (Int. Conf. on Databases in Office, Engineering and Science), Freiburg 1999.
6. Ester M., Kriegel H.P., Sander J., *Spatial Data Mining: A Database Approach*, Proc. 5th Int. Symposium on Large Spatial Databases (SSD'97), Berlin 1997.
7. Kacperski M., Zwoliński G., *Communication with users in the computerized recruitment system of candidates for higher education studies*, International Conference on System Modelling and Control (SMC), Zakopane, 12–14 October 2009.
8. Kiełtyka L., *Systemy informatyczne zarządzania informacją*, w: *Informatyka gospodarcza*, red. J. Zawiła-Niedźwiecki, Wydawnictwo C.H. Beck, Warszawa 2010.
9. Kowalczyk A., Pelikant A., *Fuzzy clustering in relational databases*, XII International Conference System Modelling and Control, Zakopane, 17–19 October 2007.
10. Kowalczyk A., Pelikant A., *Implementation of Automatically Generated Membership Functions Based on Grouping Algorithms*, The IEEE Region 8 Eurocon 2007 Conference, Warsaw, 9–12 September 2007.
11. Kowalczyk-Niewiadomy A., Pelikant A., *Zagadnienia grupowania w kontekście budowania zapytań rozmytych,* w: *Bazy danych. Rozwój metod i technologii. Architektura, metody formalne i zaawansowana analiza danych*, Wydawnictwa Komunikacji i Łączności, Warszawa 2008.
12. Kriegel H.P., Renz M., Schubert M., Züfle A., *Statistical Density Prediction in Traffic Networks*, Proc. 8th SIAM Conf. on Data Mining (SDM 2008), Atlanta 2008.
13. Olszak C., *Systemy informatyczne analityczno-raportujące*, w: *Informatyka gospodarcza*, red. J. Zawiła-Niedźwiecki, Wydawnictwo C.H. Beck, 2010.
14. Pabjańczyk K., Pelikant A., *Implementacja algorytmów użytkownika w środowisku Business Intelligence SQL Server 2008*, „Studia Informatica" (Gliwice) 2009, vol. 30, no. 2B (84), Silesian University of Technology Press.
15. Pelikant A., *Hurtownie danych. Od przetwarzania analitycznego do hurtowni danych*, Helion, 2011.
16. Pelikant A., *Bazy danych w zastosowaniach praktycznych*, WSInf, Łódź 2007.
17. Pelikant A., *Systemy gromadzenia danych*, w: *Informatyka gospodarcza*, red. J. Zawiła-Niedźwiecki, Wydawnictwo C.H. Beck, 2010.
18. Pérez D., Somodevilla M.J., Pineda I.H., *Fuzzy Spatial Data Warehouse: A Multidimensional Model*, w: *Advances Decision Support Systems*, red. G. Devlin, INTECH, Croatia 2010.

19. Pushpa S., Meenakshi S., *Implementation of Object Oriented Data Warehousing using a Narrower Compassed Data Model in Oracle 10g*, „International Journal of Computer Applications" 2011, vol. 17, no. 5, March.
20. Sobczak A., *Modele i metamodele w architekturze korporacyjnej,* w: *Systemy wspierania organizacji*, Katedra Informatyki, Akademia Ekonomiczna, Katowice 2009.
21. Sobczak A., *Przegląd wybranych podejść do zarządzania IT w organizacjach*, Komputerowo Zintegrowane Zarządzanie, Zakopane 2010.
22. Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-2009- TR-001, *CMMI for Services, Version 1.2 – Improving processes for better services*, February 2009.
23. Stępniak R., Pelikant A., *Wykorzystanie interfejsu JAVA API do budowy narzędzi eksploracyjnych z wykorzystaniem Oracle Data Mining na platformie Oracle*, XIII konferencja PLOUGH, Kościelisko, październik 2007.
24. Witkowski T., *Systemy informatyczne wspomagania podejmowania decyzji*, w: *Informatyka gospodarcza*, red. J. Zawiła-Niedźwiecki, Wydawnictwo C.H. Beck, 2010.
25. Zwoliński G., Kacperski M., *Komputerowa organizacja działań wspierająca proces rekrutacji studentów*, „Metody Informatyki Stosowanej" 2010, Polska Akademia Nauk Oddział w Gdańsku, Komisja Informatyki.

Summary

Application of object types in analytical processing

The paper is dedicated to the issues of the analytical processing executes both a transactional database (OLTP), as well as on the part of the multidimensional structures of the data warehouse (OLAP). It shows you how to create the object-oriented types on the server-side, their encapsulation to elements of procedure and use of complex processing algorithms. Against this background explains the application of built-in object types describing vector graphics (spatial – geometry, geography). In paper present their application in systems that report using the analysis of systems with transactional and data warehouse on the background of possible deployment in government and healthcare. Discusses use to data mining tasks, together with the propagation results to reporting systems. It provides directions of using object types to describe networks analysis tasks, including social networks. Beyond the conclusions arising from the experience of using complex types, including types of object, underlined small use them in commercial solutions.