Wielokryterialne zadanie optymalizacji schematu agregatów w hurtowniach danych

M. MAZUREK
e-mail:marcin.mazurek@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

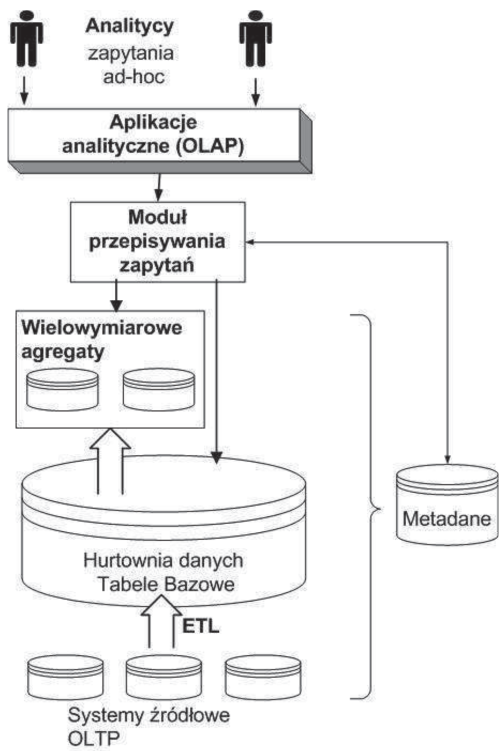
Dominującą technologią wykorzystywaną do zwiększenia wydajności dużych hurtowni danych są, wyliczane w oparciu o dane szczegółowe, agregaty, w tym mające postać wielowymiarowych kostek. Mechanizmy przepisywania zapytań pozwalają na znacznie szybsze wyznaczenie odpowiedzi na zapytanie w oparciu o mniejsze agregaty niż w przypadku odwołań do źródłowych tabel hurtowni. Do kluczowych parametrów opisujących schemat wielowymiarowych agregatów należą czas odpowiedzi na zapytania użytkownika, czas aktualizacji danych w schemacie w oparciu o nowe dane w hurtowni oraz przestrzeń dyskowa wymagana do przechowywania agregatów. Przedstawione zostało dwukryterialne sformułowanie zadania optymalizacji, w którym jako ograniczenie przyjęty został czas aktualizacji danych. Pozostałe dwie zmienne tworzą wektor kryteriów. Wyznaczony został zbiór rozwiązań optymalnych w sensie Pareto oraz zaproponowano metodę znalezienia jednoznacznego rozwiązania w oparciu o punkt idealny.

Słowa kluczowe: hurtownia danych, OLAP, wielowymiarowe agregaty, optymalizacja

1. Wprowadzenie

Wykorzystywane w systemach analizy danych klasy OLAP wielowymiarowe agregaty, zwane kostkami (ang. *cube*), umożliwiają zmniejszenie czasu odpowiedzi na zapytania użytkowników. Zarówno logiczny model danych, jak też fizyczna organizacja tych zbiorów podporządkowane są minimalizacji czasu oczekiwania na wyniki zapytań klasy SPJG (zapytania Select-Project-Join z pojedynczą instrukcją agregacji group-by na końcu [1]), formułowanych w trybie ad-hoc przez użytkowników.

Umiejscowienie schematu agregatów w architekturze hurtowni danych przedstawia rys. 1. W głównej składnicy hurtowni przechowywane są dane szczegółowe. Ze względu na duży wolumen danych, wielokrotnie przekraczający rozmiar pamięci operacyjnej, każdorazowe wyliczanie odpowiedzi na zapytania użytkownika o zagregowane dane wiązałoby się ze stosunkowo długim czasem oczekiwania na odpowiedź, ze względu na dużą liczbę operacji wejścia-wyjścia. Rozwiązaniem jest wyliczenie i przechowywanie części zagregowanych danych tak, by w chwili zapytania użytkownika nie było konieczności przeprowadzania obliczeń. Jeżeli schemat agregatów nie będzie zawierał danych, o które pyta użytkownik, wówczas zapytanie zostanie przekierowane przez moduł przepisywania zapytań do tabel bazowych w głównym repozytorium hurtowni danych. Uzupełnieniem systemu jest baza metadanych, zawierająca istotne informacje sterujące działaniem podsystemu agregatów: model danych przechowywanych w hurtowni, model agregatów oraz informacje wolumetryczne.



Rys. 1. Architektura hurtowni danych

Zagadnienia badawcze w obszarze wielowymiarowych struktur danych dla aplikacji OLAP dotyczyć mogą następujących obszarów:

- projekt schematu agregatów: wybór zakresu danych przechowywanych w postaci wielowymiarowych agregatów
- utrzymanie schematu: określenie wydajnego sposobu odświeżania danych w kostkach, po aktualizacji danych w tabelach bazowych
- wykorzystanie schematu: mechanizmy przepisywania zapytań użytkownika, umożliwiające wyznaczenie odpowiedzi na zapytanie użytkownika w oparciu o schemat agregatów, bądź, jeżeli jest to niemożliwe, w oparciu o tabele bazowe.

Przedmiotem rozważań w artykule jest pierwsze z wymienionych zagadnień, przy czym pozostałe dwa muszą zostać uwzględnione w postaci kształtu odpowiednich funkcji występujących jako dane w zadaniu wyznaczenia schematu agregatów.

W literaturze problemu zaproponowano kilka sformułowań zadania optymalizacji różniących się funkcją kryterium i ograniczeniami. W praktyce, wyznaczenie schematu agregatów w oparciu o jedno, wybrane kryterium optymalizacji nie jest satysfakcjonujące dla użytkownika.

W artykule przedstawiony został model schematu agregatów (meta-model M2 zgodnie z klasyfikacją MOF [2][3]) oraz sformułowanie wielokryterialnego zadania optymalizacji. Dla wskazanego zadania wyznaczony został zbiór Pareto-optymalny oraz sposób znalezienia rozwiązania w tym zbiorze.

2. Model wymiarowy danych i schemat agregatów

Wprowadźmy oznaczenie zbioru wymiarów analizy:

$$\mathbf{D} = \{dim_1, dim_2, ..., dim_n, ..., dim_N\} \qquad (1)$$

gdzie dim_n oznacza n-ty wymiar analizy.

W każdym z wymiarów można wyróżnić jeden lub więcej poziomów szczegółowości (atrybutów). Niech L_n oznacza liczbę poziomów agregacji wyodrębnionych dla n-tego wymiaru.

Zbiór atrybutów odpowiadających poziomom agregacji wymiarów wyznaczonym przez atrybuty określa macierz:

$$\mathbf{L} = \begin{Bmatrix} attr_1^1, attr_2^1, \ldots, attr_{L_1}^1, \\ attr_1^2, attr_2^2, \ldots, attr_{L_2}^2, \\ \ldots\ldots, \\ attr_1^N, attr_2^N, \ldots, attr_{L_N}^N \end{Bmatrix}. \qquad (2)$$

Każdy podzbiór poziomów agregacji $\mathbf{L}_n$

$$\mathbf{L}_n = \left\{attr_1^n, attr_2^n, \ldots, attr_{L_n}^n\right\} \qquad (3)$$

tworzy kratę $\langle \mathrm{L}_n, \preceq \rangle$ w oparciu o relację porządku $\preceq$, określającą zależność funkcyjną pomiędzy wartościami atrybutów (wartości atrybutu $attr_i^n$ wymiaru dim_n zależą funkcyjnie od wartości atrybutu $attr_j^n$), co zostało opisane w postaci definicji (4).

$$\begin{aligned} & attr_i^n \preceq attr_j^n \overset{df}{\Leftrightarrow} \\ & \exists\, f : Dom\left(attr_j^n\right) \xrightarrow{na} Dom\left(attr_i^n\right) \end{aligned} \qquad (4)$$

gdzie:

$Dom\left(attr_j^n\right)$ – oznacza zbiór wartości, jakie przyjmuje atrybut $attr_j^n$.

W oparciu o listę wymiarów i ich poziomy szczegółowości można utworzyć M typów wielowymiarowych agregatów, określonych przez kombinacje poziomów agregacji poszczególnych wymiarów.

$$M = \prod_{n=1}^{N} L_n \qquad (5)$$

Niech

$$\mathbf{C} = \{c_1, c_2, \ldots c_M\} \qquad (6)$$

oznacza zbiór typów wielowymiarowych agregatów (kostek).

Każda kostka c_i jest jednoznacznie definiowana przez n-elementowy wektor opisujący poziomy agregacji poszczególnych wymiarów.

Kostka c_i może zostać wyliczona w oparciu o kostkę c_j, jeżeli dla każdego wymiaru poziom agregacji kostki jest nie mniejszy niż poziom agregacji kostki źródłowej, czyli zachodzi:

$$\underset{n=\overline{1,N}}{\forall} \left(attr_{a_n^i}^i \preceq attr_{a_n^j}^j \right) \qquad (7)$$

Możliwość wyliczenia zawartości agregatu w oparciu o dane przechowywane w innych agregatach można przedstawić w postaci grafu G, w którym zbiór wierzchołków **C** odpowiadających kostkom został rozszerzony o dodatkowy wierzchołek reprezentujący tabelę bazową ($\mathbf{C}^0$):

$$\mathbf{G} = \left\langle \mathbf{C}^0, \prec \right\rangle, \tag{8}$$

Relacja $c_i \prec c_j$ oznacza, że kostka $c_i \in \mathbf{C}$ może zostać wyliczona w oparciu o inną kostkę $c_j \in \mathbf{C}$, czyli zachodzi warunek (7).

Wyznaczenie schematu agregatów polega na znalezieniu podzbioru kostek **C**, który zostanie „zmaterializowany", to jest zostaną wyliczone wartości miar w tych agregatach i odświeżane każdorazowo po załadowaniu hurtowni nowymi danymi.

W dalszej części artykułu przyjmujemy, że zmaterializowany podzbiór agregatów jest opisany binarnymi zmiennymi decyzyjnymi:

$$x_i \in \{0,1\},\ i = \overline{1,M} \tag{9}$$

określającymi, czy kostka c_i zostanie wskazana do wyliczenia.

Graf **G** jest wykorzystywany do znalezienia takiego planu wyliczania danych w agregatach, który przy zadanej wartości zmiennej *x* będzie prowadził do najkrótszego czasu trwania tego procesu.

3. Odświeżanie danych

W chwili, kiedy do tabeli bazowej wstawiane są rekordy, niezbędne jest ponowne wyliczenie zawartości kostek danych ze względu na dezaktualizację podsumowań znajdujących się w komórkach agregatów. Zmiany w tabeli bazowej pociągają za sobą konieczność aktualizacji danych w każdym z wyliczonych agregatów. Jeżeli przyjmiemy, że detaliczna hurtownia danych jest ładowana w trybie dziennym, wówczas w tym samym cyklu należy wyliczać dane w agregatach.

Dane w kostce mogą być aktualizowane w oparciu o dane z tabeli bazowej bądź innej kostki pozostającej w odpowiedniej relacji $\prec$. Jako koszt aktualizacji danych przyjmujemy czas wymagany do wyliczenia i zapisania zawartości zmienionych komórek agregatu. Koszt związany z aktualizacją kostki może być opisany jako funkcja na krawędziach grafu **G** pokazującego zależności obliczeniowe pomiędzy kostkami.

W zależności od użytych algorytmów aktualizacji danych w agregatach i zakresu zrównoleglenia operacji, czas odświeżenia tego samego schematu agregatów może się różnić w odmiennych systemach zarządzania wielowymiarową bazą danych.

4. Zapytania użytkowników

Typy zapytań klasy SPJG kierowane do hurtowni danych należą do skończonego zbioru, wyznaczonego przez wymiary analizy oraz warunki selekcji.

Oznaczmy przez **Q** zbiór numerów typów zapytań:

$$\mathbf{Q} = \{1, 2, \ldots, M\} \tag{10}$$

Klasy równoważności zapytań, odpowiadające typom zapytania, zostały wyróżnione w oparciu o poziom agregacji wymiarów występujący w treści zapytania.

Każdemu typowi zapytania odpowiada dokładnie jeden typ agregatu (jest to ta kostka, w której poziomy agregacji wymiarów są identyczne jak w zapytaniu).

Jeżeli kierowane do hurtowni danych zapytanie zawiera warunki selekcji, określone na dziedzinie atrybutu, który nie występuje w klauzuli grupującej zapytania, wówczas według przyjętego modelu należy dla zadanego wymiaru wskazać poziom szczegółowości tegoż atrybutu. Odpowiedź na tego typu zapytanie może zostać udzielona wyłącznie w oparciu o agregat, w którym wymiar występuje w stopniu szczegółowości nie niższym, niż odpowiadający zadanemu atrybutowi.

Zapytania kierowane przez użytkowników do hurtowni danych mogą być modelowane jako strumień zdarzeń oraz odpowiadający mu dyskretny w stanach proces stochastyczny. Wartością $Z(t_i)$ procesu jest numer typu zapytania skierowanego do hurtowni danych w chwili t_i.

Zmienne losowe $Z(t_i), Z(t_{i+1}), \ldots, Z(t_{i+k})$ są niezależnymi zmiennymi losowymi o tym samym, znanym rozkładzie prawdo--podobieństwa. Możemy przyjąć, że numer typu zapytania kierowanego przez użytkownika do hurtowni w dowolnej chwili jest realizacją tej samej zmiennej losowej *Z*.

Rozkład zmiennej losowej Z wyznaczają wartości prawdopodobieństw wystąpienia zapytania *i*-tego:

$$w_i = P\{Q = i\} \tag{11}$$

Dysponując rozkładem zmiennej losowej *Z* można wyznaczyć zależność oczekiwanego czasu odpowiedzi *T(x)* od podzbioru zmaterializowanych agregatów. Zakładamy przy tym, że znamy czas odpowiedzi $\tau_i(x)$ na *i*-te zapytanie dla schematu agregatów zdefiniowanego przez *x*; przy czym:

$$T(x) = \sum w_i \cdot \tau_i(x) \tag{12}$$

Mając dany zbiór numerów typów zapytań $\mathbf{Q}$ oraz rozszerzony o tabelę bazową zbiór agregatów $\mathbf{C}^0$, możliwość wyznaczenia odpowiedzi na zapytanie można przedstawić w postaci dwudzielnego grafu:

$$\mathbf{H} = \left\langle \mathbf{Q} \cup \mathbf{C}^0, \mathbf{E} \right\rangle \tag{13}$$

Graf ten stanowi podstawę do wyznaczenia takiego planu wykonania zapytania przez moduł przepisywania zapytań, aby czas odpowiedzi był najmniejszy z możliwych, przy określonym zbiorze zmaterializowanych agregatów.

5. Kryteria i ograniczenia zadania optymalizacji

Oczywistym kryterium optymalizacji wydaje się być wartość oczekiwana czasu odpowiedzi na zapytanie użytkownika. Taka funkcja celu pojawia się w najstarszych chronologicznie publikacjach [4][5] i większość proponowanych algorytmów optymalizacji schematu agregatów, przyjmując jako ograniczenie przestrzeń dyskową, minimalizuje właśnie tę wielkość.

W pozycji [6] zaproponowane zostało użycie jako kryterium liczby typów zapytań, na które odpowiedź może zostać wyznaczona przy wykorzystaniu schematu agregatów.

Próbę uwzględnienia więcej niż jednego kryterium można odnaleźć w [7], gdzie minimalizowana jest suma składników: oczekiwanego czasu odpowiedzi na zapytanie oraz oczekiwanego czasu wymaganego do odświeżenia zawartości agregatów.

Głównym ograniczeniem związanym z utrzymywaniem schematu agregatów jest czas niedostępności struktur danych dla użytkowników powodowany odświeżaniem danych. W przypadku, gdy hurtownia jest ładowana wsadowo w cyklu dziennym, czas odświeżenia danych *u(x)* nie może przekroczyć wydzielonego okna ładowania danych do hurtowni.

Przy założeniu, że dane do hurtowni spływają na bieżąco (bez opóźnień pomiędzy chwilą zaistnienia zdarzenia a datą pojawienia się informacji o zdarzeniu w systemie źródłowym), czas aktualizacji struktur danych nie zależy od okresu, w jakim dane przechowywane są w schemacie agregatów, lecz wyłącznie od listy typów zmaterializowanych agregatów.

Ograniczenie związane z przestrzenią dyskową, choć mogłoby się wydawać mało istotne w kontekście spadających cen nośników pamięci, nie może zostań pominięte – zapotrzebowanie na ten rodzaj pamięci rośnie wykładniczo w funkcji wymiarów analizy. Dodatkowo, jest to dobry wskaźnik kosztów związanych z utrzymaniem dodatkowej warstwy danych, obejmujących koszt archiwizacji danych, koszt pracy administratorów związanej z nadzorem nad przebiegiem procesów ładowania danych oraz monitorowaniem parametrów środowiska, czy też wreszcie koszt serwisu sprzętowego.

Przestrzeń dyskowa *D(x,t)*, wymagana do materializacji schematu agregatów, zależy od maksymalnej liczby wymiarów analizy, która w dopuszczalnym uproszczeniu, w danym obszarze tematycznym, można być uznana za niezmienną, oraz długości okresu, za jaki dane są przechowywane w schemacie. Jeżeli przyjąć, że w agregatach przechowywane są kompletne dane historyczne, wymagana przestrzeń dyskowa będzie rosła liniowo w funkcji czasu. W dalszej części artykułu, przez wielkość przestrzeni dyskowej *d(x)* będziemy oznaczali wartość wymaganą do przechowywania danych z pojedynczego interwału czasu:

$$D(x,t) = t \cdot \sigma \cdot d(x) \tag{14}$$

W artykule przedstawione zostało uogólnione podejście, w którym problem wyboru optymalnego schemat agregatów został opisany jako wielokryterialne zadanie optymalizacji.

6.

7. Wielokryterialne zadanie optymalizacji

Jednym z etapów projektowania hurtowni danych jest wskazanie, jakie agregaty zostaną wyliczone – jest to decyzja, którą podejmuje projektant hurtowni danych.

Jako kryteria przyjmujemy dwuwymiarowy wektor:

$$\left\langle T(x), d(x) \right\rangle \in R^+ \times R^+ \tag{15}$$

Ogólną postać zadania optymalizacji schematu agregatów opisano poniżej.

Dane:

1. **H** – graf wykonalności zapytań,
2. **G** – graf zależności obliczeniowych pomiędzy typami agregatów,
3. *d*(*x*) – zależność opisująca wymaganą przestrzeń dyskową od typów materializowanych agregatów,
4. *T*(*x*) – oczekiwany czas odpowiedzi na zapytanie przy założeniu, że zostanie ono wykonane w sposób optymalny,

tzn. gwarantujący najkrótszy możliwy czas odpowiedzi, przy zadanym zbiorze wyliczonych agregatów opisanym wartością zmiennej x. Plan wykonania takiego zapytania wyznaczony jest w oparciu o graf **G**,

5. $u(x)$ – czas aktualizacji danych w kostce przy założeniu, że aktualizacja będzie wykonana w sposób optymalny, to jest gwarantujący najkrótszy możliwy czas odświeżenia danych w zbiorze agregatów opisanych zmienną x. Plan odświeżania zawartości danych w schemacie agregatów wyznaczany jest w oparciu o graf **H**,
6. U – limit czasu dla aktualizacji danych w schemacie agregatów wynikający z dostępnego okna czasowego ładowania hurtowni danych.

Szukamy $x^* = \langle x_1^*, ..., x_M^* \rangle$ minimalizującego następujące wielkości:

$$\begin{aligned} T(x) &\to \min \\ d(x) &\to \min \end{aligned} \tag{16}$$

przy ograniczeniu:

$$u(x) < U \tag{17}$$

8. Rozwiązanie zadania

Na rys. 2. przedstawiono zbiór rozwiązań dopuszczalnych w przestrzeni kryteriów, uzyskany poprzez rozwiązanie ciągu zadań optymalizacji jednokryterialnej [8].

Wartości kryteriów zostały standaryzowane zakresowo, przy czym maksymalna wartość oczekiwanego czasu T_{max} odpowiedzi odpowiada sytuacji, w której odpowiedzi na wszystkie zapytania wyznaczane są w oparciu o tabele bazowe:

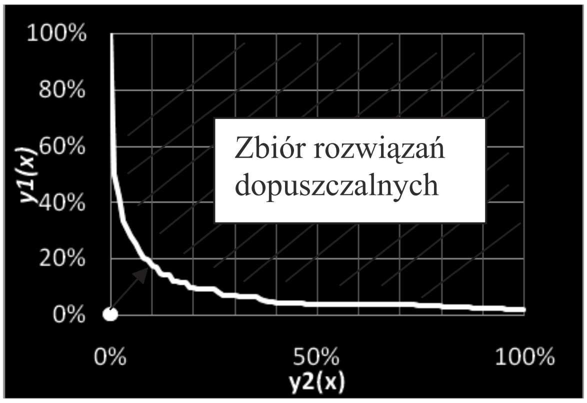
$$y_1 = \frac{T(x)}{T_{\max}} \tag{18}$$

Wartość drugiego kryterium została standaryzowana analogicznie:

$$y_2 = \frac{d(x)}{d_{\max}} \tag{19}$$

gdzie d_{max} oznacza maksymalną przestrzeń dyskową wymaganą do zmaterializowania wszystkich agregatów.

Na tym samym rysunku zaznaczony został dodatkowo zbiór rozwiązań niezdominowanych (optymalnych w sensie Pareto) oraz punkt idealny mający interpretację następującą: zarówno czas odpowiedzi jak i zajmowany obszar dyskowy powinny mieć wartość jak najbliższą zeru.



Rys. 2. Zbiór rozwiązań Pareto-optymalnych

Przedstawione powyżej zadanie nie posiada rozwiązania dominującego. Satysfakcjonujący punkt można odnaleźć wyszukując punkt najbliższy punktowi idealnemu o współrzędnych <0,0>.

Dla wyszukania takiego punktu należy odpowiednio przeskalować wielkości kryteriów, na przykład w sposób zaprezentowany na wykresie. W oryginalnej postaci, wielkości kryteriów odłożone na osiach mają różne wymiary i mierzone są w innych, nieporównywalnych jednostkach.

9. Podsumowanie

Wielokryterialne sformułowanie zadanie optymalizacji pozwala na uwzględnienie preferencji użytkownika w zakresie wagi kryteriów. Uwzględnienie więcej niż jednego kryterium w zadaniu optymalizacji umożliwia modelowanie preferencji użytkownika np. w postaci wag przypisywanych poszczególnym kryteriom.

Należy wspomnieć o innych, niż agregaty, podejściach do problemu zwiększania efektywności systemu hurtowni danych. Prekalkulacja wyników zapytań w postaci wielowymiarowych agregatów nie jest jedynym kierunkiem poszukiwań w zakresie zmniejszanie czasu odpowiedzi systemów. W systemach Business Intelligence „in-memory” [9] zakłada się, że wszystkie dane niezbędne do wyliczenia odpowiedzi na zapytanie można zmieścić w pamięci operacyjnej dzięki efektywnym algorytmom kompresji i dzięki temu ograniczyć liczbę operacji IO. W systemach ROLAP, w których nie jest budowana warstwa pośrednia, kładzie się nacisk na przygotowanie właściwych indeksów dla relacyjnych zbiorów danych.

10. Bibliografia

[1] M. Jarke, M. Lenzerini., Y. Vassiliou, P. Vassiliadis, *Fundamentals of Data Warehouses*, Springer-Verlag, Berlin, 2003.

[2] *MetaObjectFacility(MOF) Specification,* http://www.omg.org/technology/documents/formal/mof.htm/.

[3] *Common Warehouse Metamodel Specification*, http://www.omg.org/spec/CWM/1.1/PDF/

[4] V. Harinarayan, A. Rajaraman, J.D. Ullman, *Implementing Data Cubes Efficiently*, ACM SIGMOD, 205-216, 1996.

[5] H. Gupta, S. Mumick, *Selection of views to materialize under a maintenance cost constraint*, International. Conference on Database Theory (ICDT), 453-470, 1999.

[6] E. Hung, D.W. Cheung, B. Kao, Y.L. Liang, *An Optimization Problem Design in Data Cube System Design* Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Artificial Intelligence 1805, 74-85, Springer-Verlag, 2000.

[7] H. Gupta, *Selection of views to materialize in a data warehouse*. International. Conference. on Database Theory (ICDT), 98-112, 1997.

[8] M. Mazurek, *Optymalizacja schematu wielowymiarowych agregatów w hurtowni danych*, WAT, Warszawa, 2008, rozprawa doktorska.

[9] http://www.qlikview.com.

[10] J. Kusiak, A. Danielewska-Tułecka, P. Oprocha, *Optymalizacja*, Wydawnictwo Naukowe PWN, Warszawa, 2009.