PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 133
Pozyskiwanie wiedzy i zarządzanie wiedzą 2010

Małgorzata Nycz
Uniwersytet Ekonomiczny we Wrocławiu

POZYSKIWANIE WIEDZY Z HURTOWNI DANYCH

Streszczenie: referat jest poświęcony pozyskiwaniu wiedzy z hurtowni danych będącej głównym elementem systemu Business Intelligence, z wykorzystaniem przetwarzania analitycznego oraz *data mining*. Odkryta wiedza podlega weryfikacji i ocenie, a następnie prezentacji. Całość rozważań kończy krótkie podsumowanie.

Słowa kluczowe: pozyskiwanie wiedzy, hurtownia danych, Business Intelligence.

1. Wstęp

Od mniej więcej połowy lat osiemdziesiątych ubiegłego wieku obserwujemy przyspieszenie rozwoju cywilizacyjnego, co jest związane z ogromnym postępem w rozwoju technologii informacyjno-komunikacyjnych, które wywierają ogromny wpływ na praktycznie wszystkie sektory naszego życia. Obecnie żyjemy w czasach określanych jako era informacji i wiedzy. Oznacza to, że nie posiadanie ziemi czy uzyskiwanie np. ogromnych ilości wydobywanego węgla bądź wytopu stali, lecz posiadanie informacji i wiedzy oraz właściwe jej wykorzystanie na potrzeby zarządcze decyduje o pozycji przedsiębiorstwa na rynku. Jednym ze źródeł informacji i wiedzy są posiadane bazy danych, innymi są zasoby dostępne z zewnątrz np. z Internetu. Technologią, która jest obecnie coraz częściej wykorzystywana w celu dostarczania wiedzy menedżerskiej, będącej wynikiem przetwarzania zasobów danych dostępnych przedsiębiorstwu, jest Business Intelligence, dla której kluczowym elementem jest hurtownia danych.

2. Techniki przetwarzania struktur danych w HD

Każde przedsiębiorstwo wykorzystuje różne systemy informatyczne (działające na różnym sprzęcie) w swojej bieżącej działalności, umożliwiające tworzenie różnego rodzaju raportów. Jednak nie wszystkie informacje da się z nich pozyskać, ponieważ potrzebne są dane historyczne, często przechowywane na różnych nośnikach i nie udostępniane na bieżąco. Systemy BI wspomagają proces podejmowania decy-

zji w przedsiębiorstwie. Jeśli mają spełniać swoje zadanie, muszą mieć dostęp do danych o odpowiedniej jakości i postaci, aby te dane mogły być wykorzystane do różnego rodzaju analiz. Rozwiązaniem tego typu problemów jest technologia hurtowni danych (HD) oparta na wielowymiarowej strukturze danych, której celem jest dostarczenie użytkownikom właściwej informacji, w odpowiednim czasie i po niskiej cenie. Hurtownia danych (zwana też magazynem danych) coraz częściej staje się źródłem wiedzy wykorzystywanej w procesach decyzyjnych. HD można traktować jako technologię służącą gromadzeniu oraz przechowywaniu danych i operowaniu na danych zarówno znajdujących się w organizacji, jak i pochodzących z jej otoczenia. Technologia ta nie zależy od platformy sprzętowej, systemu operacyjnego czy bazy danych. Integruje informacje potrzebne do analiz z różnych heterogenicznych źródeł i z organizacji, i spoza niej. Jest ona wykorzystywana w HD[1], która jest elektronicznym magazynem danych (ang. *storehouse*) oczyszczającym i transformującym dane z wielu źródeł i wielu form. Według Inmona przez pojęcie „hurtownia danych" rozumie się taki zbiór danych (ang. *data set*) wspomagających podejmowanie decyzji, który jest uporządkowany tematycznie, zintegrowany, zawierający wymiar czasowy oraz nieulotny [5].

Dane w hurtowni danych powinny być statyczne i przeznaczone głównie do odczytu. Nie zakłada się zwykle możliwości ich zmiany. Dane w hurtowni reprezentują pewien przedział czasowy, dlatego też nie mogą być uaktualniane. Dane są aktualizowane wyłącznie w swoich systemach źródłowych, a następnie „wstrzykiwane" całą porcją do hurtowni z odpowiednią adnotacją czasową (stempel czasowy). Jedyne operacje, które powinny być wykonywane, to wprowadzanie i agregacja nowych danych podczas procesu ładowania i następnie ich selekcja w zapytaniach.

Istnieją cztery główne kategorie danych w hurtowni. Są to fakty, wymiary, dane zagregowane i metadane. Fakty stanowią najistotniejszy obszar danych w hurtowni, gdyż na ich podstawie dokonuje się bezpośrednio wszelkich analiz. Mogą one mieć bardzo dużą objętość, nawet rzędu kilku terabajtów. Zależy to od tego, jak wiele danych historycznych jest potrzebnych do analizy. Dane faktów mogą być fizycznie podzielone (partycjonowane) na mniejsze tablice, logicznie zaś reprezentowane jako jedna.

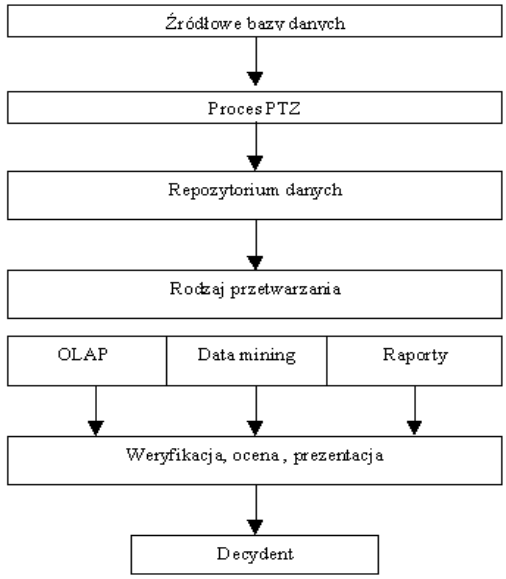
Z hurtownią danych związane są następujące pojęcia:

- **schemat gwiazdy** (ang. *star schema*) – specjalny rodzaj organizacji danych, projektowany pod kątem szybkości dostępu do danych; jest schematem upraszczającym nawigację po danych,
- **tabela faktów** (ang. *fact table*) – centralna tabela w schemacie gwiazdy; zwykle tabela faktów jest zbiorem kluczy obcych do tabel wymiarów w schemacie gwiazdy,

[1] Twórcą idei hurtowni danych jest William H. Inmon [5].

– **tabela wymiaru** (ang. *dimension table*) – tabela przechowująca atrybuty związane z pewnym aspektem danych gromadzonych w schemacie gwiazdy; są to zwykle tabele danych o klientach, magazynach, produktach itp.,
– **OLAP** (ang. *Online Analytical Processing*) – dziedzina hurtowni danych związana z analizą i prezentacją danych w procesie wspomagania decyzji.

Dane składowane w hurtowniach danych mają zazwyczaj charakter wielowymiarowy, który zapewnia im odpowiednia struktura logiczna. Podstawową strukturą logiczną w HD jest struktura gwiaździsta. Inne struktury danych to płatek śniegu oraz konstelacja faktów. Można też spotkać strukturę danych pośrednią pomiędzy gwiazdą a płatkiem śniegu określaną jako gwiazda – płatek śniegu. Dane w hurtowni danych mogą być, generalnie rzecz ujmując, wykorzystywane w ramach trzech rodzajów ich przetwarzania. Są nimi wielowymiarowa analiza danych, *data mining* oraz raportowanie. Wyniki są weryfikowane, oceniane i prezentowane jako raport w postaci wygodnej dla użytkownika. Schematycznie operacje wykonywane na danych zgromadzonych w HD przedstawia rys. 1.



Rys. 1. Rodzaje przetwarzania danych w hurtowni danych

Źródło: opracowanie własne.

Podstawowym (ale nie jedynym: innym, często spotykanym jest klasyczne, relacyjne przetwarzanie danych) rodzajem przetwarzania danych w hurtowni danych jest przetwarzanie analityczne OLAP, a wyniki mogą być prezentowane w postaci raportów o np. trendach, osiągnięciach lub niepowodzeniach konkretnych strategii

marketingowych. W ramach operacji na danych zgromadzonych w HD mogą być wykonywane operacje związane z odkrywaniem wiedzy (*data mining*), o czym w dalszej części.

OLAP składa się z trzech podstawowych elementów, którymi są:

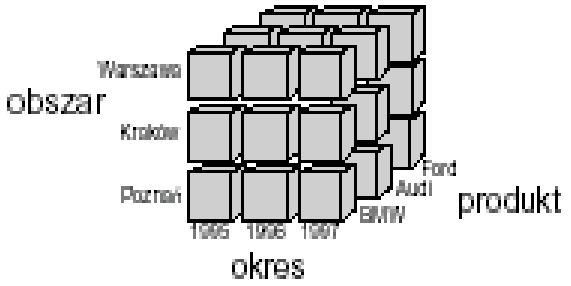
1) struktura danych opisująca logiczną organizację danych oraz sposób postrzegania danych przez użytkowników,

2) zbiór operatorów umożliwiających wyszukiwanie i modyfikowanie danych,

3) ograniczenia integralnościowe, specyfikujące poprawność danych.

W OLAP dane postrzegane są przez użytkownika w postaci wielowymiarowej perspektywy, przedstawianej poglądowo jako kostki OLAP. Obiektem analizy jest zbiór miar numerycznych, które nazywane są faktami. Fakt opisuje pojedyncze zdarzenie i jest daną ilościową (numeryczną) reprezentującą aktywność biznesową, np. sprzedaż produktów, średni zysk, wartość produktu krajowego. Wartość miary zależy od zbioru wymiarów, który z kolei określa kontekst miary. I tak np. sprzedaż produktów można rozpatrywać w kontekście miasta, dostawców, klientów, okresów sprzedaży czy konkretnego produktu. Miara jest przedstawiana jako punkt w wielowymiarowej przestrzeni wymiarów. Z każdym wymiarem związany jest zbiór atrybutów. Na przykład wymiar klient może mieć takie atrybuty, jak identyfikator klienta, jego nazwa, adres, NIP, telefon, faks itd.; wymiar produkt może być opisany przez takie atrybuty, jak obszar, okres, produkt. Dane wymiarów są zdenormalizowane, co pozwala użytkownikowi na wygodne eksplorowanie „w dół" (ang. *drill-down*), „w górę" – agregowanie (ang. *drill-up*) oraz „w poprzek" (ang. *drill across*). Dane referencyjne mogą zawierać tysiące wierszy, ale w porównaniu z nimi dane faktów – miliony. Dane wymiarów nie zmieniają się tak często, jak dane faktów czy dane zagregowane. Atrybuty wymiaru mogą tworzyć hierarchię atrybutu [7; 11].

Na rysunku 2 przedstawiono przykładową kostkę trójwymiarową z wymiarami obszar, produkt i okres dla faktu wielkość sprzedaży samochodów w Polsce.

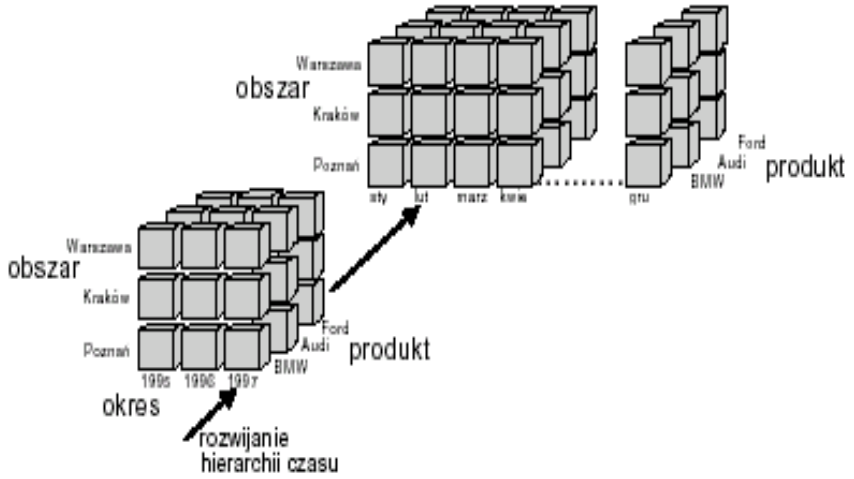


Rys. 2. Kostka trójwymiarowa OLAP

Źródło: [6].

Operacje wykonywane na strukturach danych hurtowni określane są za pomocą operatorów. Zatem zbiór operacji (operatorów) określa operacje przetwarzania struktur danych. Do podstawowych zalicza się takie, jak:

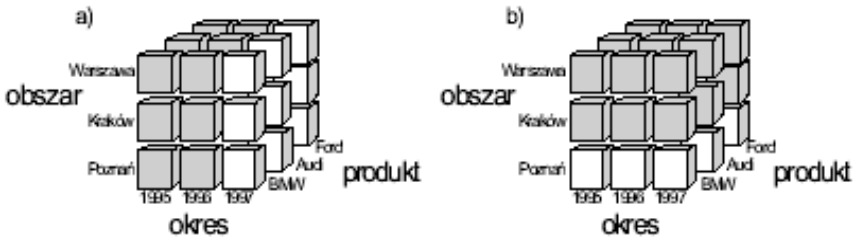
– wyznaczanie punktu centralnego (ang. *pivoting*) mające za zadanie wskazanie miary, która jest interesująca dla użytkownika, i wybranie dwóch wymiarów, w których ma być ona reprezentowana. Na przykład w wymiarze produktu reprezentującego samochody Fabia i w wymiarze dealerzy może być prezentowana liczba sprzedaży samochodów;
– rozwijanie wymiaru (ang. *drill-down*) oznaczające rozwijanie agregatu na części składowe, np. sprzedaż w poszczególnych branżach, sprzedaż w poszczególnych kategoriach samochodów, poszczególnych okresach. Jako przykład niech będą dane informacje o sprzedaży samochodów różnych marek, w latach 1995, 1996, 1997, w poszczególnych miastach. W celu sporządzenia analizy sprzedaży w poszczególnych miesiącach w 1997 r. rozwija się hierarchię okres reprezentującą rok 1997 (rys. 3). Analiza sprzedaży w poszczególnych dniach danego miesiąca jest możliwa po rozwinięciu hierarchii reprezentującej dany miesiąc;



Rys. 3. Operacja rozwijania hierarchii wymiaru

Źródło: [6].

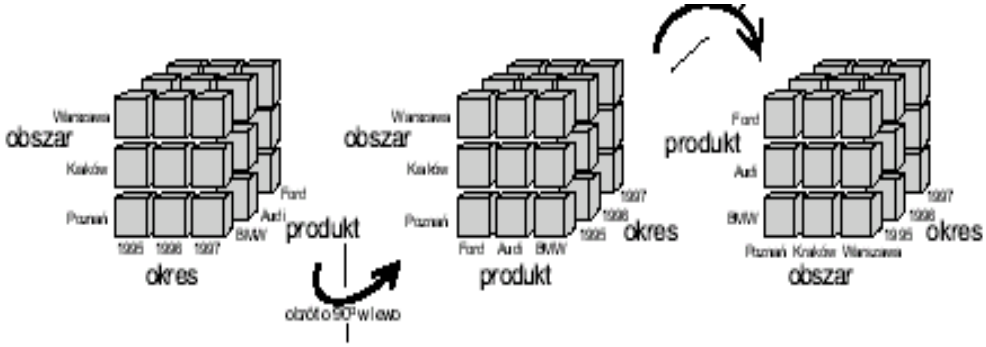
– zwijanie (ang. *roll-up*) – operacja oznaczająca dla danego wymiaru zwijanie w górę hierarchii wymiaru w celu prezentacji większych agregatów,
– wycinanie (ang. *slice and dice*) – odpowiada operacji redukcji liczby wymiarów, a więc zachodzi projekcja danych na wybranym podzbiorze wymiarów dla wybranych wartości innych wymiarów (rys. 4),



Rys. 4. Operacje wycinania danych w różnych wymiarach

Źródło: [6].

– obracanie – umożliwiające prezentowanie danych w różnych układach (rys. 5);



Rys. 5. Operacje obracania danych

Źródło: [6].

– rankowanie (ang. *ranking*) umożliwiające wybór pierwszych n elementów, np. trzy najlepiej sprzedające się produkty w miesiącu styczniu;
– inne operacje, takie jak selekcja, procedury składowane itp.

Zbiór ograniczeń integralnościowych określa poprawność przechowywanych danych. Ograniczenia integralnościowe dla kostki wielowymiarowej można podzielić na dwie grupy, jakimi są:

– ograniczenia integralnościowe pojedynczej kostki danych (ang. *intra cube constraints*), związane z definicjami zależności między atrybutami wymiarów, wymiarami a miarami, oraz hierarchiami wymiarów,
– ograniczenia integralnościowe pomiędzy kostkami danych (ang. *inter cube constraints*) określające związki pomiędzy dwoma bądź więcej kostkami danych, takie jak zależności między miarami kostek, wymiarami kostek, miarą jednej kostki a wymiarami innych kostek.

Wynikiem wykonanych na danych operacji wielowymiarowych analiz są informacje, które przedstawione w postaci raportu być może okażą się wystarczające dla menedżera w procesie decyzyjnym. Ale menedżer może zażyczyć sobie wykonania dalszych czynności, związanych z poszukiwaniem wśród tych informacji wiedzy. Wówczas uruchamiany jest *data mining*, czyli „uzupełnieniem" technologii OLAP są w takiej sytuacji techniki *data mining*, których zadaniem jest m.in. odkrywanie wzorców i trendów w danych, trudnych do odkrycia innymi metodami.

Data mining może zostać uruchomiony w dwóch sytuacjach: albo jako samodzielne zadanie zlecone przez użytkownika, zupełnie abstrahując od hurtowni danych, realizowane na przygotowanym uprzednio zbiorze danych, albo jako kolejny krok po zakończeniu realizacji wielowymiarowych analiz w hurtowni danych, na zestawie wygenerowanych informacji. W obu przypadkach uzyskujemy „na wyjściu" użyteczną wiedzę. Z punktu widzenia *data mining* nie ma znaczenia, czy wystąpił pierwszy przypadek, czy drugi, albowiem sposób realizacji technik *data mining* jest taki sam.

Istnieje wiele różnych technik *data mining*. Można je ująć np. następująco (por. [2; 3; 8]):

1) odkrywanie zależności (ang. *mining association rules*),
2) wielopoziomowe uogólnianie danych (ang. *multi-level data generalization*),
3) klasyfikacja (ang. *data classification*),
4) grupowanie (ang. *clustering analysis*),
5) odkrywanie podobieństw w oparciu o wzorce (ang. *pattern similarity search*),
6) odkrywanie schematów ścieżek (ang. *mining path traversal patterns*).

Odkrywanie zależności w przypadku bazy danych o transakcjach przeprowadzonych w sklepie będzie polegać na identyfikacji artykułów nabywanych razem (np. mleko i chleb). Jeśli $A = \{a_1, a_2, ..., a_n\}$ będzie zbiorem elementów reprezentujących artykuły w sklepie, a $T = \{T_1, T_2, ..., T_n\}$ będzie zbiorem transakcji reprezentującym fakt zakupienia dwóch artykułów, to oznacza, że $T_i \subset A$. Jeśli założymy, że $X \subset A$, wówczas o transakcji T_i możemy powiedzieć, że zawiera zbiór X wtedy i tylko wtedy, gdy $X \in T_i$. Zależność tę możemy przedstawić w formie implikacji $X \Rightarrow Y$, gdzie $X \subset A$, $Y \subset A$, a $X \cap Y = \varnothing$. O zależności $X \Rightarrow Y$ można powiedzieć, że posiada wiarygodność c (gdy c% transakcji ze zbioru T zawierających podzbiór X zawiera również podzbiór Y) określającą siłę zależności. Jeśli s% transakcji ze zbioru T zawiera podzbiór X lub Y, wówczas o zależności $X \Rightarrow Y$ można powiedzieć, że ma wsparcie o wartości s, informujące nas o częstości pojawiania się zależności w bazie danych. Główne zadanie algorytmu *data mining* to znalezienie silnych zależności, które charakteryzują duża wiarygodność i silne wsparcie, a więc zidentyfikowanie największych zbiorów elementów w bazie o wsparciu powyżej wyznaczonej granicy i wykorzystanie ich do wygenerowania poszukiwanych zależności. Ilustracją może być tutaj algorytm Apriori [1]. Algorytm ten konstruuje zestaw różnych, równolicznych podzbiorów, złożonych z ele-

mentów transakcji będących kandydatami na podzbiory o wystarczająco dużym wsparciu. W kolejnych iteracjach tworzone są podzbiory (jedno-, dwu-, trójelementowe itd.), aż do utworzenia odpowiednio licznych podzbiorów z wystarczająco dużym wsparciem lub do momentu, gdy nie można utworzyć podzbiorów w kolejnej iteracji. Dla każdego zbioru kandydującego oblicza się wsparcie, a następnie wybiera te, których wsparcie jest większe od założonego przez użytkownika.

Przyjmijmy, że D_k będzie zestawem podzbiorów kandydujących (po iteracji k), a L_k zestawem podzbiorów z D_k posiadających wystarczające wsparcie. Zestaw D_{k+1} w kolejnej iteracji algorytmu Apiori jest następujący:

$$\{X \cup Y; X, Y \in L_k; |X \cap Y| = k - 1\}.$$

Każdy podzbiór z zestawu D_{k+1} musi spełniać warunek, że dowolny jego podzbiór (od 1 do k-elementowego) ma wystarczające wsparcie, czyli znajduje się w dowolnym zestawie od L_1 do L_k z poprzednich iteracji. Jako przykład możemy przyjąć bazę danych do analizy za pomocą algorytmu Apriori. Algorytm Apriori znajduje reguły asocjacyjne w relacyjnej bazie danych. Asocjacją pomiędzy danymi nazywamy zależność implikacyjną reprezentowaną za pomocą reguł logicznych połączonych zależnością – jeśli X, to Y. Przykładem może być: jeśli garnitur, to koszula, jeśli buty, to torebka, jeśli płaszcz, to apaszka. Przez wsparcie reguły w danym zbiorze rozumiemy stosunek liczby zbiorów zawierających daną regułę do liczby wszystkich zbiorów. Duża wartość wsparcia może zawierać informacje dotyczące strategii działania firmy. Można na tej podstawie ustalić rozmieszczenie towarów w sklepie. Jeśli towary po obniżonej cenie rozmieścimy pomiędzy produktami X oraz Y zachodzi duże prawdopodobieństwo, że zostaną zakupione pomimo podwyższenia ceny towaru Y.

Przykładem może być transakcja opisana za pomocą identyfikatora transakcji T_i (identyfikator klienta sklepu, data i godzina robienia zakupów) oraz produkty zakupione w ramach tej transakcji. W przypadku eksploracji danych w bazie danych supermarketu będzie to znalezienie prawidłowości w kolejności kupowania różnych produktów.

T_i – identyfikator transakcji	Produkty
1	P1 P3 P4
2	P2 P3 P5
3	P1 P2 P3 P5
4	P2 P5

D_1 jest zestawem następujących podzbiorów: {P1}, {P2}, {P3}, {P4}, {P5}. Dla każdego podzbioru obliczamy wsparcie i eliminujemy te, które mają najmniejsze wsparcie:

D_1

Podzbiory	Wsparcie
{P1}	2
{P2}	3
{P3}	3
{P4}	1
{P5}	3

L_1

Podzbiory	Wsparcie
{P1}	2
{P2}	3
{P3}	3
{P4}	3

W następnej iteracji bierzemy pod uwagę wszystkie dwuelementowe i postępujemy jak poprzednio. W L_2 otrzymujemy:

Podzbiory	Wsparcie
{P1 P3}	2
{P2 P3}	2
{P2 P5}	3
{P3 P5}	2

W trzeciej iteracji C_3 zawiera podzbiory {P2 P3 P5}, dla których wsparcie wynosi 2. W tej sytuacji algorytm Apriori kończy swoje działanie, gdyż nie jest już możliwe utworzenie dalszych iteracji.

Innym jest algorytm GSP (ang. *Generalized Sequential Patterns*) służący do odkrywania reguł sekwencyjnych w bazie danych. Podczas pracy algorytmu wykonuje się wiele odczytów baz danych, z których pierwszym zadaniem jest obliczenie wsparcia poszczególnych pozycji, będącego liczbą sekwencji zawierających te pozycje. Po pierwszym odczycie znane są jednoelementowe zbiory częste, a więc pozycje mające minimum wparcia. Następnie algorytm startuje ze zbiorami wzorców częstych znalezionych w poprzedniej fazie. Każdy następny wzorzec ma o jedną pozycję więcej niż ten, którym posłużyliśmy się do jego wygenerowania. Wsparcie dla tych wzorców oblicza się podczas jednego pełnego odczytu bazy danych, co pozwala stwierdzić, które wzorce można uznać potencjalnie za częste i wykorzystać do generacji w następnym kroku. Algorytm kończymy, gdy po kolejnym odczycie brak jest wzorców częstych lub nie udało się wygenerować nowych wzorców potencjalnie częstych. Ograniczenia czasowe związane z sekwencjami (dotyczące bazy danych z przykładu Apriori) to m.in.:

– minimalna odległość (*min-gap*) – minimalna różnica czasów wystąpień dwóch kolejnych sekwencji, pozwalająca na sprowadzenie pewnego produktu, którego wzrost jest bardzo prawdopodobny, gdyż z obserwacji wynika wzrost innego produktu,
– maksymalna odległość (*max-gap*) – maksymalna różnica czasów wystąpień dwóch kolejnych pozycji wzorca sekwencji, która pozwala na odrzucenie klientów, którzy rzadko robią zakupy w naszym sklepie.

Algorytmy służące do wielopoziomowego uogólniania danych oprócz cech odkrywania zależności mogą zawierać takie elementy ułatwiające przeprowadzanie

analiz, jak np. generowanie zależności zbudowanych na różnych poziomach abstrakcji, definiowanie różnych minimalnych wartości wsparcia dla różnych poziomów hierarchii, warunkowe badanie zależności na niższym poziomie, wówczas gdy ta zależność ma na wyższym poziomie odpowiednie wsparcie.

Celem technik klasyfikacji jest znalezienie wspólnych cech charakterystycznych wśród obiektów bazy danych i przyporządkowanie ich do odpowiednich klas (grup), które pozwolą odróżnić je od pozostałych klas obiektów. Do konstruowania klasyfikacji mogą być wykorzystane różne metody klasyfikacyjne, np. drzewa decyzyjne, których tworzenie odbywa się przez rekurencyjny podział zbioru na podzbiory aż do uzyskania ich jednorodności ze względu na przynależność obiektów do klas. Aby zbudowane drzewo było jak najmniejsze, dokonuje się jego porządkowania, usuwając te fragmenty, które mają niewielkie znaczenie dla jakości wyników klasyfikacji. Każdy algorytm tworzący drzewo decyzyjne musi rozwiązać trzy problemy: (1) jak wybrać jedną lub kilka cech, na podstawie których nastąpi podział zbioru obiektów, (2) kiedy zakończyć podział powstałego podzbioru obiektów oraz (3) w jaki sposób przydzielić obiekty znajdujące się w liściu drzewa do pewnej klasy.

Efektywność algorytmu zależy od sposobu podziału zbiorów obiektów w węzłach drzewa, a więc pojedynczych cech lub ich kombinacji liniowych. Wyboru dokonujemy w oparciu o pewną miarę jakości podziału (miary jednorodności lub zróżnicowania). W przypadku miary jednorodności wybieramy podział, który maksymalizuje wartość stosowanej miary, a wybierając miary zróżnicowania – podział, który minimalizuje jej wartość. Jeśli przyjmiemy, że $O = \{o_1, o_2, ..., o_n\}$ będzie zbiorem obiektów należących do jednej z klas $K_1, K_2, ..., K_k$, przy czym liczność klasy K_i oznaczymy jako l_i, to dla każdego zbioru obiektów możemy zbudować wektor prawdopodobieństwa przynależności do klas w postaci:

$$p = (p_1, p_2, ..., p_k) = (\frac{l_1}{n}, \frac{l_2}{n}, ..., \frac{l_k}{n}), \qquad \text{gdzie } \sum_{i=1}^{k} p_i = 1.$$

Możemy powiedzieć, że pewien zbiór obiektów jest jednorodny, gdy $\exists i = 1, ..., k\, p_i = 1$. Natomiast jego maksymalne zróżnicowanie występuje wówczas, gdy $\forall i = 1, ..., k\, p_i = 1/n$.

Grupowanie pozwala identyfikować grupy zdarzeń lub podobnych do siebie obiektów ze względu na kryteria. Wykorzystując określony sposób pomiaru odległości (podobieństwa) obiektów w wielowymiarowej przestrzeni cech, można zbiór podzielić na podzbiory tak, aby zawierały obiekty najbardziej do siebie podobne. Można tu wykorzystać jedną z poniższych technik:

1) optymalno-iteracyjne (dokonuje się podziału zbioru na k rozłącznych podzbiorów, gdzie k jest podane przez badacza),

2) hierarchiczne (w ramach których skupienia tworzą binarne drzewa, liście reprezentują obiekty, a węzły ich grupy; skupienia wyższych poziomów zawierają w sobie skupienia niższych poziomów),

3) tworzące skupienia nierozłączne (niektóre obiekty ze zbioru mogą należeć do więcej niż jednej grupy) [4].

Technika odkrywania podobieństw na podstawie wzorców najczęściej wykorzystywana jest do analizy szeregów czasowych, a więc zbiorów danych, w których jednym z atrybutów jest czas bądź inny atrybut zależny od czasu. Możemy mieć tutaj do czynienia z dwoma przypadkami: zapytaniami związanymi z określonym wzorcowym obiektem, których celem jest znalezienie obiektów spełniających wcześniej zdefiniowane warunki dotyczące podobieństwa do określonego wzorcowego obiektu, lub zapytaniami porównującymi wszystkie pary elementów ze sobą, których celem jest znalezienie par obiektów spełniających określony przez użytkownika warunek podobieństwa [3, s. 866-883].

W rozproszonym środowisku pomiędzy dokumentami i obiektami utrzymywane są połączenia, które ułatwiają interaktywny dostęp do nich. Zrozumienie wzorców dostępu użytkowników w takim środowisku nie tylko ułatwia projektowania systemu, lecz także prowadzi do podejmowania lepszych decyzji marketingowych. Uchwycenie wzorców dostępu użytkownika w środowiskach rozproszonych określane jest mianem odkrywania wzorców ścieżek powiązań w sposób krzyżowy.

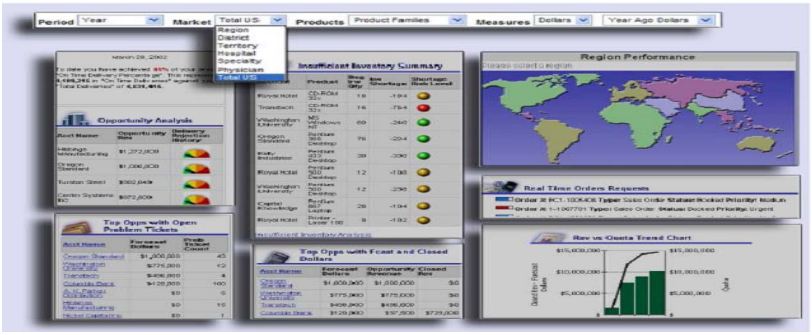
3. Weryfikacja, ocena i prezentacja pozyskanej wiedzy

Zanim odkryta wiedza zostanie przeznaczona od użytku, podlega ocenie (interpretacji), jaką jest jej weryfikacja. Weryfikacja ma na celu identyfikację oraz skorygowanie takich niepożądanych właściwości wiedzy, jak niekompletność i niespójność. Obydwa rodzaje anomalii wiedzy występują w różnym stopniu, zależnie od sposobu reprezentowania wiedzy czy dziedziny, której dotyczy. Ocena odkrytej wiedzy może być realizowana na dwa sposoby. Oceny może dokonać ekspert dziedzinowy i/lub można sporządzać oceny w sposób zautomatyzowany [10, s. 111]. Proces oceniania odkrytej wiedzy sprowadza się do wyznaczenia jej zgodności z założeniami i celami *data mining*. Przyjmuje się zazwyczaj dwa podstawowe kryteria weryfikacji wiedzy, jakimi są kompletność oraz spójność [9]. *Kompletność* oznacza, że wiedza – będąca do dyspozycji określonego podmiotu – jest wystarczająca do generowania odpowiednich wniosków wynikających z wyznaczonego celu systemu. Ujmując ten problem w rozumieniu bardziej intuicyjnym: wiedza kompletna oznacza „pokrycie" wszystkich możliwych przypadków, w jakich będzie ona wykorzystywana (por. np. [9; 10]). W praktyce – w szczególności w odniesieniu do generowanych baz wiedzy – mamy do czynienia z pewnym podzbiorem przekształconej wiedzy dziedzinowej lub bazy wiedzy utworzonej na podstawie rozwiązywanych problemów, którą możemy uznać za kompletną wobec wykonywanych zadań.

Przyjmuje się z kolei, że baza wiedzy jest *spójna*, jeżeli w bazie faktów nie ma takich, które dla określonych więzów spójności (ang. *consistency constraints*) nie pozwoliłyby na osiąganie celów systemu. Więzy spójności dotyczą istotnych dla bazy właściwości strukturalnych, takich jak przykładowo wykluczanie wiedzy konfliktowej czy redundantnej. Są one dość często ignorowane w przypadku wygenerowanych baz wiedzy.

Rys. 6. Przykład kokpitu Oracle BI

Źródło: www.oracle.com.



Rys. 7. Dynamiczna natura kokpitu

Źródło: www.oracle.com.

Odkryta wiedza prezentowana jest menedżerowi w żądanej przez niego postaci. Stosowane są różne techniki wizualizacji, takie jak zestawienia tabelaryczne, wykresy czy opisy. W ostatnich latach coraz bardziej upowszechniają się tzw. kokpity (ang. *dashboards*). Kokpit jest taką organizacją ekranu, która umożliwia użytkownikowi nie tylko „ogląd" sytuacji decyzyjnej, lecz także dynamiczne dostosowanie dostępnych w systemie a wyświetlanych na monitorze informacji w zależności od zapotrzebowania

użytkownika. Zależnie od tego, dla jakiego użytkownika kokpit jest przeznaczony, nosi nazwy: kokpit informacyjny, kokpit menedżerski itp. Przykładowy kokpit oferowany w ramach Oracle BI przez firmę Oracle przedstawiono na rys. 6.

a)

b)

Rys. 8. Przykłady kokpitu informacyjnego (a) i menedżerskiego (b)

Źródło: www.oracle.com.

Kokpit ma naturę dynamiczną, co oznacza, że użytkownik zależnie od swych potrzeb może – dokonując wyboru bezpośrednio z ekranu – obejrzeć np. żądane zestawienia finansowe czy wyniki analiz, co pokazuje rys. 7.

Przykłady kokpitu informacyjnego i menedżerskiego pokazano na rys. 8.

4. Podsumowanie

Artykuł przedstawia pozyskiwanie wiedzy z hurtowni danych będącej elementem centralnym systemu Business Intelligence, za pomocą przetwarzania analitycznego oraz *data mining*, a następnie jej weryfikację i prezentację użytkownikowi za pomocą tzw. kokpitu.

Literatura

[1] Agrawal R., Srikant R., *Fast algorithms for mining association rules in large databases*, [w:] *Proceedings of the 20th International Conference on Very Large Data Bases, September 1994*, s. 478-499.
[2] Berry M.J.A., Linoff G., *Data Mining Techniques for Marketing, Sales and Customer Support*, Wiley Computer Publishing, New York 1997.
[3] Chen M.S., Han J., Yu P.S., *Data mining: An overview from a database perspective*, „IEEE Transactions on Knowledge and Data Engineering" 1996, Vol. 8, No. 6, s. 866-883.
[4] Gatner E., *Symboliczne metody klasyfikacji danych*, Wydawnictwo Naukowe PWN, Warszawa 1998.
[5] Inmon W.H., *Building the Data Warehouse*, Wiley Computer Publishing, New York 2002.
[6] Morzy T., *Przetwarzanie danych w magazynach danych*, [w:] *Projektowanie i implementowanie magazynów* (*hurtowni*) *danych*, V seminarium PLOUG, Warszawa 29.05.2002.
[7] Nycz M., *Problemy związane z pozyskiwaniem wiedzy z baz danych*, Prace Naukowe Akademii Ekonomicznej nr 850, AE, Wrocław 2000.
[8] Nycz M. (red.), *Pozyskiwanie wiedzy menedżerskiej*. *Podejście technologiczne*, AE, Wrocław 2007.
[9] Owoc M., *Wartościowanie wiedzy w inteligentnych systemach wspierających zarządzanie*, AE, Wrocław 2004.
[10] Owoc M. (red.), *Elementy systemów ekspertowych*, *Część 1*. *Sztuczna inteligencja i systemy ekspertowe*, AE, Wrocław 2006.
[11] Smok B. (red.), *Środowisko ORACLE w odkrywaniu wiedzy z baz danych*, UE, Wrocław 2008.

KNOWLEDGE ACQUISITION FROM DATA WAREHOUSE

Summary: The paper presents knowledge acquisition from the data warehouse that is the main element of any BI system. It concentrates on analytical processing as well as data mining within the data warehouse. Discovered knowledge has to be verified, assessed and then presented in form of tables, graphs or with the use of dashboards.