

Obiektowe bazy danych – wybrane kierunki rozwoju

K. TOMASZEWSKI

karol.tomaszewski@wat.edu.pl

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

Systemy bazodanowe stanowią jedną z kluczowych gałęzi rozwoju współczesnych systemów informatycznych. Są one jednym z podstawowych elementów architektury i mają krytyczny wpływ na funkcjonowanie całości rozwiązań informatycznych. Nieustannie trwają prace rozwojowe w obszarach systemów bazodanowych wszystkich typów. W tym artykule opisana została skrócona charakterystyka obiektowych baz danych. Następnie przedstawione zostały wybrane propozycje nowych rozwiązań, dotyczące kategorii obiektowych baz danych. Wybrane zagadnienia zostały podzielone na innowacje w zakresie funkcjonalności oraz nowe realizacje istniejących mechanizmów, wpływające na poprawę efektywności ich działania. Dla każdej z tych grup wyszczególniono szereg zagadnień będących przedmiotem najnowszych opracowań w dziedzinie wraz ze skróconym opisem działania.

Słowa kluczowe: obiektowe bazy danych, OODB, kierunki rozwoju.

1. Wprowadzenie

Na przestrzeni ostatnich lat można zaobserwować znaczący rozwój obszaru wiedzy, zajmującej się zagadnieniami utrwalania danych przetwarzanych we wszelkiego rodzaju systemach informatycznych. Od czasu pierwszych realizacji komputerowych systemów bazodanowych, w latach sześćdziesiątych ubiegłego stulecia, koncepcje implementacji rozwinęły się w kilka indywidualnych ścieżek rozwoju. W wyniku prac poszczególnych gałęzi rozwoju powstawały różne rozwiązania, takie jak płaskie (kartotekowe) bazy danych, hierarchiczne bazy danych, sieciowe bazy danych, relacyjne bazy danych, relacyjno-obiektowe bazy danych, czy czysto obiektowe bazy danych.

Obecnie, w poszczególnych gałęziach, stale trwa rozwój technologiczny, owocujący nowymi produktami na rynku komputerowych systemów baz danych.

Artykuł ten ma na celu przybliżenie wybranych zagadnień dotyczących rozwoju w dziedzinie obiektowych baz danych.

Na początku przytoczone zostaną podstawowe cechy charakterystyczne obiektowych baz danych na tle pozostałych rozwiązań.

Następnie zostanie omówiony szereg rozwiązań wskazujących kierunki badań i rozwoju na przestrzeni ostatnich lat. Zagadnienia zostaną pogrupowane na dwa rodzaje, mianowicie na zagadnienia rozwoju nowych funkcjonalności,

wykorzystywanych w projektowaniu i pracy z systemami obiektowych baz danych, oraz zagadnienia rozwoju nowych sposobów realizacji istniejących funkcjonalności, wpływających na poprawę efektywności ich działania, w szczególności optymalizacji, opisanej różnymi kryteriami. Wśród kryteriów optymalizacji mechanizmów działania obiektowych baz danych możemy wyszczególnić między innymi takie cechy jak minimalizowana złożoność obliczeniowa realizacji żądań, minimalizowana złożoność dostępu do pamięci podczas realizacji zapytań, minimalizowany czas przygotowania odpowiedzi na żądania czy poprawa charakterystyk niezawodności systemów obiektowych baz danych.

2. Manifesty obiektowych baz danych

Prace rozwojowe w dziedzinie obiektowych baz danych przybliżają te rozwiązania do stabilnej pozycji wśród powszechnych rozwiązań bazodanowych.

Brak wytyczonych standardów stoi jednak na przeszkodzie ujednoczenia prac i masowego zastosowania obiektowych baz danych. Dlatego już w latach 90. trwała dyskusja i podejmowane były próby stworzenia wspólnego standardu dla rozwiązań tego typu. Działania na rzecz standaryzacji cech obiektowych baz danych objawiały się m.in. kolejnymi manifestami dla rozwiązań OODBMS. Pierwszy taki manifest – *The OODBMS Manifesto* – został zaprezentowany w roku 1989 [1].

W opracowaniu tym przedstawiono 13 cech nawiązujących do obiektowego modelu danych. Zostały one podzielone na trzy grupy, tzn. cechy, które obiektowa baza danych powinna posiadać, cechy opcjonalne, które mogą być dodatkowo dołączone w celu rozwinięcia funkcjonalności, oraz cechy otwarte, pozostawione do wyboru projektantów konkretnych rozwiązań. Opisane cechy stanowiły oderwanie od właściwości systemów relacyjnych baz danych, stawiając jako punkt wyjścia obiektowy model danych.

Drugi manifest – *Third generation database system manifesto* [2] – został zaprezentowany w 1990 roku. W opracowaniu tym autorzy zaproponowali trzy doktryny oraz 13 postulatów składających się na definicję systemów, tzw. „systemów baz danych trzeciej generacji”. Definicja oparta była na założeniu, że systemy baz danych trzeciej generacji mają stanowić rozwinięcie dla istniejących systemów relacyjnych opartych o SQL. Dotychczasowe rozwiązania miałyby być rozbudowane o szereg nowych cech, wśród których znalazłyby się cechy obiektowego modelu danych.

Trzeci manifest – *The third manifesto* [3] – został zaprezentowany w 1995 roku. W opracowaniu tym autorzy podjęli próbę opisanie „przyszłych” systemów baz danych. Autorzy odrzucają wcześniejsze postulaty obiektowego modelu danych i proponują pozostanie przy modelu relacyjnym. Opracowanie nawiązuje m.in. do problemu niedopasowania w dotychczasowych rozwiązaniach pomiędzy relacyjnym modelem bazy danych oraz obiektowym modelem oprogramowania aplikacji. Autorzy wskazują jednak, że niedopasowanie nie wynika z cech modelu relacyjnego, ale z zastosowania języka SQL jako metody dostępu do danych. W pracy zaproponowano alternatywę do SQL określaną jako „D”. „D” jest pojęciem stworzonym przez autorów manifestu i stanowi specyfikację języka danych. Innymi słowy specyfikacja ta określa wymagania, które wg autorów powinny spełniać przyszłe sposoby dostępu do danych. Większość argumentów wniesionych przez autorów była jednak trudna do zaakceptowania przez specjalistów z dziedziny systemów baz danych

3. Standard ODMG

Obserwacje sukcesu, który relacyjne systemy baz danych zawdzięczają m.in. szeroko rozpropagowanym, dobrze opisanym standardom, zmotywowały do dalszych prac nad wprowadzeniem standardów dla obiektowych

baz danych. W roku 1991 odbyły się pierwsze dyskusje grupy przedstawicieli głównych dostawców rozwiązań bazodanowych, która później przekształciła się w formalną grupę do prac nad standardem – ODMG (*Object Database Management Group*). Prace zaowocowały zapisaniem pierwszego szkicu standardu dla obiektowych baz danych nazwanego *ODMG-93: The Object Database Standard*. Publikacja tego opracowania miała miejsce w 1996 roku. W 1993 roku nastąpiło stowarzyszenie ODMG z grupą OMG (*Object Management Group*).

W 1997 roku w wyniku dalszych prac powstała kolejna wersja standardu *ODMG 2.0: A Standard for Object Storage*. Końcowa wersja standardu *The Object Data Standard: ODMG 3.0* powstała w roku 1999 (opublikowana w roku 2000). Głównym celem standardu było zawarcie specyfikacji przenośności aplikacji opartych o obiektowe bazy danych. Przenośność rozumiana jest tu jako możliwość pracy z różnymi produktami bazodanowymi. Pożądany efekt to umożliwienie twórcom oprogramowania utrwalania obiektów bezpośrednio z poziomu języków programowania (JAVA, C++ czy Smalltalk) do systemów baz danych, zgodnych ze standardem ODMG 3.0), bez potrzeby zarządzania właściwym procesem utrwalania. Standard ODMG 3.0 został oparty na czterech pod-stawowych komponentach: Model obiektowy (Object Model), Język specyfikacji obiektów (Object Specification Language), Obiektowy język zapytań (Object Query Language), Związanie z językami programowania (Object Binding: C++, Java, Smalltalk).

4. Obiektowe bazy danych

Obiektowe bazy danych wspierają model danych oparty o paradygmaty modelowania obiektowego. Jego podstawową strukturą są obiekty. Jednak w przeciwieństwie do otwartości modelu relacyjnego, obiekty wraz z atrybutami i metodami w obiektowych bazach danych podlegają założeniom enkapsulacji.

Obiekty mogą posiadać zarówno proste, jak i złożone (nieatomowe) atrybuty, takie jak: listy, zbiory, tablice czy inne struktury złożone. Metody w obiektach nadają im nową rolę – odwzorowują dynamikę bytów rzeczywistych, które reprezentują. W odróżnieniu od tradycyjnego modelu relacyjnego, zamiast relacji opartych o klucze *primary key* i *foreign key*, wykorzystywane są tu atrybuty referencyjne, które wskazują zwykle na inne obiekty.

5. Cechy charakterystyczne

Jedną z cech charakterystycznych obiektów w modelu obiektowym jest to, że ich atrybuty podlegają enkapsulacji. Użytkownicy, czy też programy komputerowe, nie mają możliwości bezpośredniego dostępu do atrybutów obiektów utrwalonych w obiektowej bazie danych.

Jedynym sposobem komunikacji z obiektami jest wykorzystanie metod posiadających jasno sprecyzowany sposób dostępu do danych.

Połączenie opisu statyki w formie atrybutów z opisem dynamiki zachowań w formie definicji metod sprawia, że obiekty, jako podstawowe struktury obiektowego modelu danych, umożliwiają o wiele bardziej intuicyjne modelowanie bytów rzeczywistych w procesie projektowania modelu danych.

W obiektowym modelu danych atrybuty i metody obiektów podlegają dziedziczeniu w obiektach klas specjalizujących, zgodnie ze strukturą hierarchii dziedziczenia.

W takich systemach bazodanowych zwykle wspierany jest również polimorfizm oraz wielodziedziczenie.

Obiekty dobrze odwzorowują cechy bytów rzeczywistych również dlatego, że utrzymują one bezpośrednio odwzorowania właściwości obiektów rzeczywistych i relacji między nimi w taki sposób, że obiekty nie tracą swojej integralności oraz tożsamości, tak jak może to mieć miejsce w modelowaniu relacyjnym.

Pewne założenia integralności realizowane są tu z definicji modelowania obiektowego i są domyślnie utrzymywane. Przykładem może być hierarchia dziedziczenia, w której obiekt przykładowej klasy *Bus* jest pojazdem, i jak każdy inny podtyp klasy *Pojazd* zachowuje cechy wspólne dla wszystkich pojazdów.

Inne ograniczenia integralności można jasno definiować jako część jawnej specyfikacji obiektu. Przykładem może być założenie, że dopuszczalna ładowność pojazdu nie może przekroczyć dopuszczalnej masy całkowitej pojazdu, pomniejszonej o masę własną pojazdu. Najczęściej ograniczenia integralności danych realizowane są poprzez anulowanie transakcyjnych operacji, które produkują niepoprawne dane.

Alternatywnym podejściem jest zapewnienie automatycznych mechanizmów naprawiających dane niespełniające nałożonych ograniczeń. W rozwiązaniu takim dla każdego ograniczenia wykorzystywana jest reguła produkcji, sprawdzająca, czy nie doszło do naruszenia ograniczeń, a jeśli tak, to inicjowana

jest operacja przywracająca integralność danych w obiektowej bazie danych.

Rolę sposobu dostępu do danych pełni nawigacja obiektowa oraz metody obiektów udostępniające enkapsulowane atrybuty.

Język programowania, służący do definiowania metod w obiektach, uzależniony jest z kolei od konkretnych implementacji systemów zarządzania obiektowymi bazami danych.

Kosztami łatwości modelowania, utrzymania i zachowania integralności bazy danych jest brak możliwości doraźnego definiowania i wykorzystania dostępu do danych, charakterystycznego przy wykorzystaniu języka SQL w relacyjnych bazach danych.

6. Efektywność stosowania

Obiektowe bazy danych rozwinęły się z potrzeby lepszego wsparcia oprogramowania tworzonego w oparciu o paradygmaty obiektowości. Skuteczne metody abstrakcji w obiektowych bazach danych zapewniają enkapsulację danych i operacji wewnątrz obiektów, jasno określając sposób dostępu do nich. Pozwalają one również na przechowywanie złożonych struktur danych, które ciężko przechowywać w modelach relacyjnych. W rezultacie stanowią dobrze dopasowaną warstwę bazy danych podczas pracy z obiektowymi językami programowania. Zastosowanie systemu bazy danych, opartego o obiektowy model danych, sprawia, że dane przetwarzane w systemie nie muszą przechodzić procesu mapowania obiektowo-relacyjnego podczas utrwalania. Ogranicza to zbędne obciążenie obliczeniowe mogące być również przyczyną utraty integralności.

7. Zagadnienia funkcjonalne

Rozwój obiektowych baz danych stanowi również przyczynę rozwoju technologii korespondujących z zagadnieniami systemów baz danych. Innowacyjne paradygmaty modelowania wprowadzają innowacje w projektowaniu. Rozwojem funkcjonalnym objęte są m.in. wszelkie narzędzia wspierające procesy projektowania obiektowych baz danych, modelowania danych jak i procesy wdrażania czy utrzymania systemów obiektowych baz danych.

W obszarze zainteresowań rozwoju funkcjonalnego znajdują się m.in. prace nad wdrażaniem autoryzacji wysokiego poziomu w obiektowych bazach danych [4]. Od samego początku z rozwiązaniami utrwalania danych

związane były zagadnienia ochrony i kontroli dostępu do danych. W pierwszych realizacjach systemów bazodanowych mechanizmy ograniczające dostęp nie stanowiły wewnętrznych modułów systemów zarządzania bazami danych. Opierały się one raczej na zabezpieczeniach dostępu, wbudowanych w system operacyjny stanowiący podłoże środowiska pracy dla osadzonego systemu zarządzania bazą danych. Wraz z rozwojem technologii powstało zapotrzebowanie na zwiększenie roli systemów zarządzania bazami danych w kontroli dostępu do danych. Powstały aktywne rozwiązania zabezpieczeń jako wbudowane moduły zarządzające dostępem do danych, z uwzględnieniem ról użytkowników, lokalizacji, przydzielonych uprawnień i grup dozwolonych operacji, czy wybranych fragmentów schematu bazy danych w postaci zdefiniowanych widoków.

Wraz z rozwojem obiektowych baz danych pojawiły się nowe zagadnienia do rozważenia, mianowicie należało przemyśleć nowe pojęcia, np. definicje klas modelu bazy danych w kontekście kontroli dostępu do nich i ograniczenia uprawnień odczytu i modyfikacji. Rozważeniu podlega to, jakie zagrożenia niesie ze sobą znajomość definicji klas i możliwość ich modyfikacji, w przypadku wykorzystania przez niepożądanych użytkowników.

Omawiane są też zagadnienia formalnej definicji i wdrażania autoryzacji wysokiego poziomu w obiektowych bazach danych. Oparte są one o logiczną specyfikację zabezpieczeń, bazującą z kolei na syntaktyce i semantyce zapytań bazodanowych.

Prowadzone są również prace nad sformalizowaniem paradygmatów, znanych w dotychczasowych rozwiązaniach jako postacie normalne, i zastosowaniem w modelowaniu obiektowych baz danych. Powstało tu pojęcie postaci normalnej hierarchii klas (*Class Hierarchy Normal Form*) [5]. W modelowaniu obiektowym normalizacja modelu może być realizowana z wykorzystaniem specyficznych właściwości tego modelowania, takich jak polimorfizm, enkapsulacja czy dziedziczenie klas. Zastosowanie normalizacji obiektowego modelu danych może wpłynąć na efektywność realizacji zapytań, gdyż z definicji może ograniczyć przetwarzanie zbędnych informacji o złożonej strukturze hierarchii klas dla docelowych obiektów, do których występują odwołania.

Zagadnienia modelowania danych powiązane są również z problemami formalnej reprezentacji modelu, umożliwiającą integra-

cję lub relokację modeli danych. Wraz z paradygmatami obiektowego modelowania danych pojawiły się nowe koncepcje reprezentacji schematów baz danych [6]. Należy tu zwrócić uwagę na definicyjne podobieństwo cech modelowania obiektowego i notacji XML. Notacja ta elastycznie odwzorowuje elementy modelu, które mogą w sobie zawierać zarówno proste jak i złożone elementy, oraz odniesienia referencyjne. Znaczenie tych zagadnień podnosi fakt, że do tej pory istnieją już dobrze ugruntowane metody konwersji modeli relacyjnych do ich reprezentacji w notacji XML. Rozważana jest tu konwersja schematu obiektowej bazy danych do schematu w notacji XML oraz reprezentacja schematów notacji XML w postaci modelu obiektowej bazy danych. Może to być rozpatrywane jako element pośredni w integracji między modelami relacyjnymi i obiektowymi. Nawiązując do integracji modeli danych, omawiana jest również teoria abstrakcyjnej transformacji modeli i migracji danych pomiędzy modelami relacyjnymi i obiektowymi w ujęciu niezależnym od konkretnych systemów baz danych [7]. Można sobie wyobrazić migrację danych na przykładzie konstrukcji hierarchii klas obiektowej bazy danych na podstawie schematu relacyjnej bazy danych.

Ciekawym aspektem aktualnego rozwoju w dziedzinie obiektowych baz danych są propozycje reprezentacji niedoskonałych informacji przestrzennych w rozmytych obiektowych bazach danych [8]. W tym celu do systemów obiektowych baz danych wprowadzane są założenia teorii zbiorów rozmytych. Poruszane są tu m.in. zagadnienia takie jak wprowadzanie pojęć dziedziczenia rozmytego w hierarchii klas oraz propozycje architektury rozmytych obiektowych baz danych.

Ważnym pojęciem w rozwoju obiektowych baz danych są więzy integralności. W opracowaniu [9] omawiane są propozycje specyfikacji ograniczeń integralności spełniających stawiane wymagania w modelach obiektowych baz danych oraz propozycje restrukturyzacji obiektowych baz danych w celu uproszczenia procesu weryfikacji definicji więzów integralności. W procesie takiej weryfikacji sprawdzeniu podlegać powinna spójność definicji oraz jej kompletność i realizowalność na zadanym modelu danych.

Cechą wspólną modeli danych w systemach bazodanowych jest zmienność. Dotyczy to również modeli zorientowanych obiektowo. Schemat obiektowej bazy danych może podlegać zmianom w cyklu życia zaprojektowanego

rozwiązania. Zmiany mogą być związane zarówno z korekcją nieprawidłowości popełnionych na etapie projektowania, jak i modyfikacjami działającego rozwiązania związanymi z żądaniami zmian funkcjonalności. Zmiany definicji klas w obiektowym modelu danych niosą ze sobą ryzyko nieoczekiwanych i niepożądanych skutków pracy z taką bazą danych. Zjawisko to dotyczy zarówno relacyjnych jak i obiektowych modeli danych. W takiej sytuacji można zastosować znane do tej pory proste rozwiązanie, tj. utrzymywanie wspólnego źródła informacji o aktualnym stanie modelu danych. W rozwiązaniu tym przechowywane informacje mają ustawiony okres aktualności i zdefiniowaną procedurę cyklicznego uaktualniania obowiązującą klientów bazy danych. Można zauważyć, że charakterystyki takiego rozwiązania są dopasowywane do przybliżonych, oczekiwanych czasów modyfikacji definicji modelu danych. Zastosowanie aproksymacji może nieść ze sobą niepożądane efekty. Z jednej strony istnieje prawdopodobieństwo, że odpytania do uaktualnianego zasobu informacji o modelu będą następowały zbyt często, generując niepotrzebny ruch i zajmując zasoby obliczeniowe hostów. Z drugiej strony niedopasowanie współczynników odświeżania do częstych modyfikacji może nieść ze sobą ryzyko braku informacji o aktualnej definicji modelu danych. Wraz z rozwojem obiektowych baz danych zaproponowano rozwiązanie powiadamiania o zmianach w wersjonowaniu definicji w obiektowym modelu danych [10]. Rozwiązanie to zakłada wykorzystanie popularnego modelu *Publisher-Subscriber* we współpracy aplikacji klienckich z obiektowymi bazami danych. W rozwiązaniu takim proponuje się wykorzystanie technologii agentów odpowiadających za monitorowanie zmian w modelu bazy danych. Technologia ta uwzględnia współpracę dwóch typów agentów:

- Agent *Publisher* – monitoruje stan bazy danych i w przypadku modyfikacji publikuje powiadomienia zawierające informacje o zmianach
- Agent typu *Subscriber* – odbiera powiadomienia, do których jest przypisany jako subskrybent, i aktualizuje informacje po stronie klienta.

Działanie takich agentów zapewnia koordynację między aplikacjami klienckimi korzystającymi z zasobów baz danych. Zagadnienia te zyskują nową odsłonę w związku z pracami nad rozwiązaniami funkcjonalnymi obiektowych baz danych

8. Zagadnienia wydajnościowe

Jak wspomniano wcześniej, wybrane zagadnienia będące przedmiotem prac rozwojowych w dziedzinie obiektowych baz danych zostały podzielone na dwie kategorie. Poza pracami nad zagadnieniami funkcjonalnymi opisanymi wcześniej, wybrano również zagadnienia, których rozwój może wpłynąć na poprawę charakterystyk wydajnościowych systemów bazodanowych zorientowanych obiektowo.

Prace tego typu najczęściej koncentrują się na nowych sposobach realizacji istniejących mechanizmów, z naciskiem na poprawę charakterystyk działania baz danych.

Poprawie podlegają między innymi takie cechy jak: minimalizowana złożoność obliczeniowa realizacji żądań, minimalizowana złożoność (ilość)ostępów do pamięci podczas realizacji zapytań, minimalizowany czas przygotowania odpowiedzi na napływające żądania dostępu do danych czy maksymalizowanie niezawodności systemów obiektowych baz danych. Poniżej omówione zostaną wybrane zagadnienia, wokół których można zauważyć koncentrację dotychczasowych prac.

9. Wertykalna fragmentacja klas

Wertykalna fragmentacja klas jest techniką projektowania mającą na celu ograniczenie ilości odczytów ze źródła danych, przy wykonywaniu zadanego zbioru zapytań, poprzez minimalizację liczby odwołań do zbędnych atrybutów instancji obiektów. Osiąga się to poprzez grupowanie atrybutów o wysokiej częstotliwości dostępu w wertykalne fragmenty klas [11]. Złożoność modeli obiektowych baz danych wynikająca z hierarchii dziedziczenia oraz hierarchii kompozycji klas wpływa na komplikację definicji fragmentacji wertykalnej stosowanej w obiektowych bazach danych. Ta grupa rozwiązań odnosi się do konkretnych rozwiązań tego problemu.

10. Horyzontalna fragmentacja klas z wykorzystaniem technik sztucznej inteligencji

Obok wertykalnej fragmentacji klas proponowane są też sposoby przekształcania obiektowego modelu danych, opisywane jako fragmentacja horyzontalna. Jest to reorganizacja danych w modelu obiektowym, oparta

o klasteryzację danych z wykorzystaniem funkcji k-średnich [12]. Metoda ta ma na celu pogrupowanie danych w podzbiory obiektów, przyporządkowane do konkretnej klasy modelu obiektowego, według wspólnych cech charakterystycznych. Wykorzystanie technik sztucznej inteligencji oraz modeli sieci neuronowych umożliwia dynamiczne uczenie się i rozpoznawanie cech wspólnych obiektów. Proces ten realizowany jest na bieżąco, w trakcie przetwarzania. Dane wejściowe, podlegające fragmentacji, są tu modelowane w przestrzeń wektorową z nadanymi różnymi miarami podobieństwa wraz z ich geometryczną interpretacją. W rezultacie obiekty tej samej klasy są grupowane w podzbiory tej klasy według zdefiniowanych szczegółowo cech i miar podobieństwa. Wpływa to na optymalizację realizacji zapytań do danych, powiązanych cechami wspólnymi w obrębie obiektowej bazy danych.

11. Indeksowanie

Indeksowanie jest jedną z metod ograniczenia przestrzeni przeszukania podczas realizacji zapytań w obiektowych bazach danych. Ma ono znaczący wpływ na skrócenie czasu realizacji zapytań. Indeksowanie jednowymiarowe to grupowanie obiektów według indeksów z wykorzystaniem jednowymiarowego wektora podzbiorów zbioru obiektów. Jedną z technik indeksowania jednowymiarowego jest tzw. metoda *key-clustering*, tworząca jednowymiarowe struktury indeksowane. Struktury te grupują obiekty według wartości identyfikatora, pomijając przynależność obiektów do poszczególnych klas. Metoda ta sprawdza się najlepiej w zapytaniach oczekujących jako wyniku dokładnie jednego obiektu odpowiadającego zadanej wartości identyfikatora, niezależnie od przynależności klasowej.

Inną metodą indeksowania jest tzw. metoda *class-clustering*. Metoda ta najpierw dzieli zbiór obiektów na partycje przynależności do klas, a następnie w każdej partycji tworzy jednowymiarowe struktury indeksowe według wartości zadanego klucza. Powstaje w ten sposób wiele zbiorów, ale ciągle nieuporządkowanych i jednowymiarowych. Metoda ta sprawdza się najlepiej dla zapytań, które nastawione są na wydobycie grupy obiektów przynależących do konkretnej klasy w definicji modelu obiektowego oraz spełniających pewne zadane ograniczenia.

Proponowana metoda indeksowania dwuwymiarowego [14] zakłada organizację

przestrzeni pamięci w dwuwymiarową strukturę grupującą dane oraz porządkującą je wg dwóch kryteriów jednocześnie. Porządkowanie odbywa się w dziedzinie wartości atrybutu klucza oraz w dziedzinie przynależności do klasy obiektów.

12. Prefetching

Prefetching – to koncepcja minimalizacji ilości odwołań pomiędzy klientem a systemem zarządzania bazą danych. Istniejące strategie odwołań między klientem a serwerem można podzielić na dwie grupy: *On-demand fetching* i *Prefetching*.

Nawiązując do podstawowej cechy systemów baz danych, jaką jest utrwalanie danych i obsługa żądań dostępu do danych, można opisać typową współpracę między aplikacjami klienckimi i serwerami usług persystencji. Aplikacje posługujące się w odwołaniach do danych nawigacją obiektową bazują na referencjach między obiektami. Mamy tu do czynienia z wykorzystywaniem kolejnych atrybutów referencyjnych w celu uzyskania dostępu do kolejnych instancji obiektów. Charakterystyki nawigacyjne takiego działania odpowiadają za generację znaczącej ilości odwołań między klientem a serwerem bazy danych. Działanie takie wpływa na znaczne obciążenie zasobów wydajnościowych serwera. W rozwiązaniu opisywanym jako *On-demand fetching* obiekty są ściągane z serwera bezpośrednio przy występowaniu kolejnych żądań dostępu do danych. Na korzystną ocenę takiego podejścia wpływa fakt, że przetwarzane są tylko i wyłącznie obiekty, co do których istnieje faktyczne zapotrzebowanie na dostęp.

W rozwiązaniu opisywanym jako *Prefetching*, obiekty, do których dostęp nie jest pewny, ale spodziewany, przygotowywane są z góry [15]. Wyznaczenie tymczasowego obszaru pamięci *cache* po stronie klienta umożliwia przechowywanie przygotowanych wcześniej obiektów i ograniczenie liczby odwołań do serwera bazy danych. Wymaga to oczywiście opracowania i wdrożenia metod synchronizacji wszystkich kopii danych. Należy również podkreślić, że sytuacja, w której zostały ściągnięte dodatkowe obiekty, do których nie nastąpiło odwołanie, będzie rozpatrywana jako niepotrzebne zaangażowanie zasobów wydajnościowych systemu bazy danych.

W literaturze dość szeroko opisywane są prace nad różnymi metodami skutecznej predykcji odwołań do konkretnych grup obiektów.

13. Przetwarzanie współbieżne

Przetwarzanie współbieżne w systemach bazodanowych występuje w kilku odsłonach. Pokróćce zostaną omówione wybrane zagadnienia odnoszące się do współbieżności obiektowych baz danych.

Współbieżność w systemach bazodanowych można rozpatrywać m.in. w kontekście równoległych baz danych. Obszar ten skupiony jest na rozwoju sposobów wyboru strategii nawigacji obiektowej. Strategia nawigacji jest tu kluczowym elementem mogącym wpłynąć na poprawę wydajności realizacji zapytań. Dlatego też w literaturze proponowane i oceniane są różne algorytmy i podejścia do realizacji tego typu zadań [16]. Do technik współbieżności nawiązują również prace nad rozwojem współbieżnych języków programowania na potrzeby przetwarzania danych w systemach perstystencji zorientowanych obiektowo [17].

Prace te skupione są również na zagadnieniach wpływających na poprawę efektywności przetwarzania w rozproszonych obiektowych bazach danych. Można sobie wyobrazić sytuację, w której mamy dane przechowywane w rozproszonej pamięci zarządzanej przez zbiór równoległe działających procesorów. Proponowane rozwiązania algorytmiczne odnoszą się do relokacji danych i poprawy stopnia zrównoleglenia operacji wykonywanych podczas przygotowania odpowiedzi na zapytania. Działania takie mają na celu skrócenie czasu obsługi żądań dostępu do danych przechowywanych w systemach obiektowych baz danych. Wraz ze współbieżnością przetwarzania rozpatrzeniu podlegają również zagadnienia blokowania dostępu do zasobów [18]. Nawiązują one do technik znanych z tradycyjnego programowania współbieżnego. W przetwarzaniu współbieżnym zachodzi potrzeba blokowania dostępu do zasobów współdzielonych na czas realizacji operacji transakcyjnych oraz ewentualnego wywłaszczania zasobów. Techniki te omawiane są jednak w nowej odsłonie, tzn. w odniesieniu do blokowania obiektów w obiektowych modelach baz danych. Nowe rozwiązania proponowane są z reguły w formie protokołów, określających, które instancje obiektów i jakie fragmenty obiektowego modelu danych współdzielonych podlegają blokowaniu tak, aby zminimalizować wzajemne blokowanie współbieżnych działań.

14. Optymalizacja zapytań

Na poprawę efektywności obsługi żądań dostępu do danych może również wpłynąć analiza i reorganizacja napływających zapytań [19]. Znajomości logiki i semantyki zapytań może być wykorzystana do ich semantycznej optymalizacji. Proponowane rozwiązania wykorzystują wiedzę semantyczną na temat modelu danych, zdefiniowaną głównie w formie więzów integralności (Integration Constraints) zapisanych w definicji modelu danych. Na bazie tej wiedzy możliwe jest ponowne sformułowanie zapytania do bazy danych w taki sposób, aby otrzymać zapytanie semantycznie odpowiadające pierwotnemu, które jednak może zostać przetworzone efektywniej. Nieformalnie ujmując, dwa zapytania można uznać za semantycznie tożsame, jeżeli dla tego samego modelu danych w wyniku zwrócą tożsame zbiory wynikowe, oraz spełniają te same zbiory więzów integralności.

Napływające zapytania można również analizować pod kątem możliwości zrównoleglenia realizacji. Wśród najnowszych prac proponowane są ogólne algorytmy zrównoleglenia przetwarzania zapytań, bazujące na identyfikowaniu niezależnych fragmentów zapytań dla obiektowych modeli danych. Napływające zapytania podlegają analizie i są dzielone na szereg zapytań niezależnych, a następnie przetwarzane współbieżnie.

Techniki takie charakteryzują się dużym stopniem skomplikowania, ale i znaczącym wzrostem efektywności przetwarzania zapytań w obiektowych bazach danych.

15. Pozostałe zagadnienia

Poza powyższymi grupami zagadnień odnoszących się do poprawy efektywności przetwarzania w obiektowych bazach danych, proponowane są również bardziej innowacyjne podejścia. Jednym z ciekawszych projektów badawczych są propozycje zorientowania obiektowej bazy danych na wzór modelu sieci *peer-to-peer* (*p2p*) [20]. Stanowi to oderwanie od tradycyjnych modeli systemów bazodanowych i ma na celu wykorzystanie cech charakterystycznych działania architektury *p2p*, tzn. zorientowania komponentów na równorzędnych, niezależnych od siebie węzłach oraz dynamicznej komunikacji. Poprzez formułowanie rozwiązań opartych o tę architekturę dąży się do uzyskania lekkości, elastyczności i skalowalności działania obiektowych baz danych. Dodatkową, ważną cechą takiego

podejścia jest wysoka tolerancja błędów działania przy zachowaniu wysokiej niezawodności systemu. Działanie takiej sieci mogłoby się opierać o typowe dla sieci *peer-to-peer* mechanizmy indeksowania, dzięki którym użytkownicy mogliby wyszukiwać i odwoływać się do interesujących fragmentów danych, wykorzystując efektywne algorytmy routingu *p2p*.

Inne ciekawe prace skupiają się na współpracy z obiektowymi systemami bazodanowymi w warunkach przerywanej komunikacji [21]. Założenia te odnoszą się głównie do pracy z urządzeniami mobilnymi. Można sobie wyobrazić aplikacje klienckie korzystające z dostępu do serwera obiektowej bazy danych osadzone na urządzeniach mobilnych. Podstawowym problemem jest umożliwienie pracy urządzenia, niezależnie od tego, czy komunikacja sieciowa jest dostępna, czy nie.

Proponowane rozwiązania będą odpowiedzialne za zachowanie lokalnych zmian w warunkach przerywanej łączności. Zmiany te zostaną z powodzeniem zintegrowane z pozostałymi danymi, gdy połączenie sieciowe stanie się znów dostępne i zostanie wznowiona komunikacja. Mechanizmy te przełożą się bezpośrednio na konstrukcje bazodanowe odpowiedzialne za zarządzanie procesem reintegracji danych przechowywanych w przygotowanej pamięci *cache* urządzenia. Konstrukcja mechanizmów pozwalających na pracę w warunkach przerywanej łączności ma znaczenie z punktu widzenia zachowania integralności danych przy utrzymaniu wysokiego stopnia niezawodności pracy.

16. Bibliografia

- [1] M. Atkinson et al., "The Object-Oriented Database System Manifesto", *Proceedings of 1st International Conference on Deductive and Object-Oriented Databases*, Kyoto, Japan, December 1989.
- [2] Committee for Advanced DBMS Function. Third generation database system manifesto. SIGMOD Record 19,3, 1990.
- [3] Hugh Darwen and C.J. Date, *The third manifesto. Technical report*, <http://www.acm.org/sigmod/record/issues/9503/manifesto.ps>, 1995
- [4] N. Xianjun, "A Logic Specification and Implementation Approach for Object Oriented Database Security", *Workshop on Knowledge Discovery and Data Mining*, IEEE, 461–464 (2008).
- [5] M. Ubaid, "A Pattern for the Effective Use of Object Oriented Databases", *IEEE*, (2009).
- [6] T. Naser, "Transforming Object-Oriented Databases into XML", *IEEE*, 600–605, (2007).
- [7] R. Alhajj, "Reengineering Relational Databases to Object-Oriented: Constructing the Class Hierarchy and Migrating the Data", *IEEE*, 335–344 (2001).
- [8] G. Bordogna, "Linguistic Representation of Imperfect Spatial Information in a Fuzzy Object Oriented Database", 855–860, *IEEE*, (2002).
- [9] B. Zaqaibeh, "Restructuring Object-oriented Database Model to Ease Checking the Integrity Constraints", *Proceedings of Student Conference on Research and Development*, 171–176 (2003).
- [10] K. Palanivel, "Publisher-Subscriber: An Agent System for Notification of Versions in OODBs", *IEEE* (2009).
- [11] Ch. Fung, "An Evaluation of Vertical Class Partitioning for Query Processing in Object Oriented Databases", *IEEE Transactions on knowledge and data engineering*, 14, 1095–1118 (2002).
- [12] A. Darabant, "Semi supervised learning techniques: k-means clustering in OODB Fragmentation", *IEEE* (2004).
- [13] M. Faheem, "Fragmentation and allocation of object-oriented databases for simple attributes and complex methods: a cost-based technique", *Proceedings of the Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on Information and Communications Technology*, 731–749 (2005).
- [14] J-H. Lee, "2D-CHI: A Tunnable Two-Dimensional Class Hierarchy Index for Object-Oriented Databases", *Proceedings of the Computer Software and Applications*, 598–607, (2000).
- [15] W-S. Han, "A Formal Framework for Prefetching Based on Type-Level Access Pattern in Object-Relational DBMSs", *IEEE Transactions on knowledge and data engineering*, 17, 1436–1448 (2005)
- [16] G. Wang, "Comparision of Parallel Algorithms for Path Expression Query in Object Database Systems", *Proceedings of Database Systems for Advanced Applications Conference*, 250–257 (2001).
- [17] K. Kimura, "Dynamic Performance Optimalization Mechanism for Parallel Object-Oriented Database Programming

- Languages”, *Proceedings of the International Database Engineering and Applications Symposium*, 405–409 (2000).
- [18] W. Jun, ”A multi-granularity locking-based concurrency control in object-oriented database systems”, *The Journal of Systems and Software*, 54, 201–217 (2000).
- [19] J. Grant, ”Logic-Based Query Optimization for Object Databases”, *IEEE Transactions on knowledge and data engineering*, 12, 529–531 (2000).
- [20] G. Chiao, ”A P2P Object-Oriented Database System that Supports Multi-Attribute and Range Queries with Improved Query Response Time”, *Proceedings of the Information Technology Symposium*, 1250–1255 (2010).
- [21] S. Chang, ”An Approach to Disconnected Operation in an Object-Oriented Database”, *Proceedings of the Third International Conference on Mobile Data Management*, (2002).

Object-Oriented Databases – Review of Development Directions

K. TOMASZEWSKI

Database systems are one of the key sectors for the development of modern information systems. They are one of the basic elements of architecture and have a critical impact on the functioning of all solutions. There is constantly ongoing development work in the areas of database systems of all types. At first, this article summarizes the standardization of OODB attempts. Next, few proposals for major improvements has been described. Selected issues have been divided into innovation in terms of functionality as well as new implementations of existing mechanisms, which improve the efficiency of their operations. For each group a number of issues, which are the subject of the latest developments, have been listed with short description.

Keywords: OODB, review, development.