Raport Badawczy
Research Report

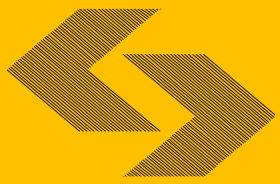
RB/64/2010

Budowa systemu porównywania obiektów złożonych

Ł. Sosnowski

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



BUDOWA SYSTEMU PORÓWNYWANIA OBIEKTÓW ZŁOŻONYCH

Łukasz Sosnowski
Studia doktoranckie IBS PAN

Artykul pokazuje implementację systemu porównującego obiekty i wyszukującego podobieństwa. Przedstawia podstawy teoretyczne, jak również aspekty analityczne i projektowe. Artykuł przedstawia analizę strukturalną za pomocą diagramów przepływu danych oraz analizę związków encji. Encje dokumentują zakres informacyjny części transakcyjnej systemu (OLTP) jak również część ROLAP - projekty kostek. Pokazano tutaj praktyczne problemy i propozycję implementacji. Pokazano praktyczny przykład implementacji komparatorów w oparciu o zbiory rozmyte.

Słowa kluczowe: *komparatory, zbiory rozmyte, bazy danych, hurtownie danych, analiza strukturalna, diagramy przepływu danych, diagramy encji, UML, diagram czynności*

Wstęp

Dzisiejsze zastosowania generują bardzo duże zapotrzebowanie na systemy dokonujące selekcji opartej na porównywaniu oraz klasyfikacji obiektów do grup, czy też zbiorów charakteryzujących się pewnymi z góry określonymi cechami. Systemy muszą potrafić szybko i trafnie odnajdywać obiekty, które są podobne do ustalonego wzorca lub też wykluczać obiekty posiadające pewne zabronione cechy. Takie zapotrzebowanie generują przykładowo systemy wyszukiwania informacji oraz systemy kontroli. Przykładem może być system kontroli i zliczania osób (http://www.airport-technology.com/contractors/ access/iee/), czy też systemy kontroli dostępu oparte o algorytmy biometryczne rozpoznające cechy indywidualne osoby typu: tęczówka, siatkówka, linie papilarne, twarz, głos i inne [10].

Niniejsza praca to zarys propozycji systemu umożliwiającego dokonywanie wyboru obiektów podobnych oraz ich klasyfikacji i interpretacji. Ideą systemu jest możliwość porównywania obiektów w obrębie danej klasy. Porównywanie ma na celu znalezienie najbardziej podobnych obiektów do zadanego. Podobieństwo definiowane jest poprzez wprowadzanie różnego rodzaju miar [4, 17, 6, 2] różnego rodzaju obiektów, jednakże miary te implementowane

są wewnątrz uniwersalnej struktury nazywanej komparatorem [11]. Można powiedzieć, że komparator to abstrakcyjna jednostka służąca porównywaniu obiektów oraz zwracająca pewien wynik. Wynik to najczęściej wskazanie na obiekt najbardziej podobny lub inna wartość określona przez konkretny typ komparatora. Komparator można przyrównać do funkcji matematycznej, która posiada argumenty wejściowe i dokonuje pewnego przekształcenia, w wyniku którego otrzymujemy wartość możliwą do dalszego wykorzystania i interpretacji w systemie.

Obiekt to najogólniej mówiąc byt, który możemy zmierzyć, zbadać, opisać, tudzież sklasyfikować lub porównać z innymi obiektami. Obiekt może posiadać cechy, które będziemy mogli użyć do jego klasyfikacji bądź analizy porównawczej. Obiekt może oznaczać zjawisko, sytuację, proces, sygnał, rzecz [17]. Definicja obiektu zbliżona jest do definicji encji, używanej w teorii baz danych [20]. W niniejszej pracy używam pojęcia obiektu na poziomie możliwie najbardziej abstrakcyjnym, jednak na poziomie projektowania systemu warto wyróżnić pewne specyficzne klasy obiektów, takie jak obrazy, teksty lub ciągi znaków. Specyfika poszczególnych klas obiektów nie stoi jednak w sprzeczności z uniwersalną architekturą proponowanego systemu.

Rozwiązania pokrewne

Podobne rozwiązania bazują na ekstrakcji cech z obiektu i przechowywaniu informacji w bazie danych relacyjnych. Podejście takie jest mało elastyczne i nie przystosowane do łatwego uzupełniania wiedzy w miarę rozwoju algorytmów przetwarzających informacje o obiektach. Jednakże takie podejście jest najczęściej stosowane. Oryginalne dane obiektu nie zasilają bazy danych lecz składowane są w postaci binarnej, zakodowanej, bez bezpośredniego dostępu z interfejsu manipulującego danymi. Spotykane dotąd systemy ograniczały się do wybranego rodzaju obiektów, np. obrazów i specjalizowały się w implementacji algorytmów dedykowanych, które nie były przenoszalne do innych klas obiektów. Przykładowe systemy porównywania obrazów, typowo składają się z kilku faz przetwarzania, takie jak: przetwarzanie wstępne i segmentacja (ekstrakcja cech, wyszukiwanie zależności, określenie miar), zapis w postaci odpowiedniej reprezentacji, adaptacja oraz decyzja [8]. W ramach ekstrakcji cech pozyskiwane są informacje o obiektach, które następnie służą do porównywania obiektów między sobą. Znane techniki ekstrakcji cech obiektów to badanie histogramu, rozpoznawanie krawędzi, rozpoznawanie kształtów, badanie tekstury, analiza widmowa i analiza falkowa.

Proponowany w niniejszym artykule system, różni się w kilku zasadniczych kwestiach. Przede wszystkim zakłada, przetwarzanie różnych klas obiektów nie skupiając się jedynie na jednej wybranej klasie. System zakłada rów-

nież innowacyjne podejście do składowania i reprezentacji obiektów, poprzez przechowywanie zdekodowanej informacji o obiekcie w postaci kostek ROLAP. W ten sposób cały system stanowi hurtownię danych, która przystosowana jest do składowania bardzo dużych wolumenów danych. Podejście to umożliwia swobodny dostęp do dowolnego fragmentu danych obiektu, pozwala na swobodną analizę, poprzez agregacje, grupowania i inne operacje na danych. Dodatkowo daje możliwość w dowolnym momencie łatwego uzupełnienia danych opisujących obiekt i pochodzących z ekstrakcji. Taka architektura nie zamyka listy atrybutów interesujących przy porównywaniu obiektów, lecz w łatwy sposób umożliwia modyfikacje i dodawania nawet po zasileniu systemu danymi.

Reprezentacja obiektów i rozpoznanych cech, zakłada przechowywanie w specjalnie zaprojektowanych kostkach ROLAP dla danej klasy obiektu. Chcąc przetwarzać dowolną inną klasę, należy przygotować odpowiednie struktury danych. Przykładowo dla kasy obiektów typu obrazy, będą przechowywane pojedyncze piksele w kostce, przez co pojedynczy piksel będzie stanowił rekord w tablicy faktów.

Zawartość artykułu

W pozostałej części artykułu, przedstawione zostały podstawowe elementy systemu w postaci diagramów encji wraz z opisem części transakcyjnej systemu jak również części stanowiącej hurtownię danych. Przedstawiono też ogólny schemat przepływu danych zobrazowany na diagramie DFD. Artykuł zawiera przykład praktycznej realizacji rozwiązania problemu porównywania obrazów na bazie analizy histogramowej. Pokazuje jak takie podejście zaimplementować w proponowanej architekturze i zaproponowanych strukturach danych. Ponadto znajduje się krótka informacja na temat dalszych badań i prac oraz informacja o wykorzystanej bibliografii.

1. Zarys architektury systemu

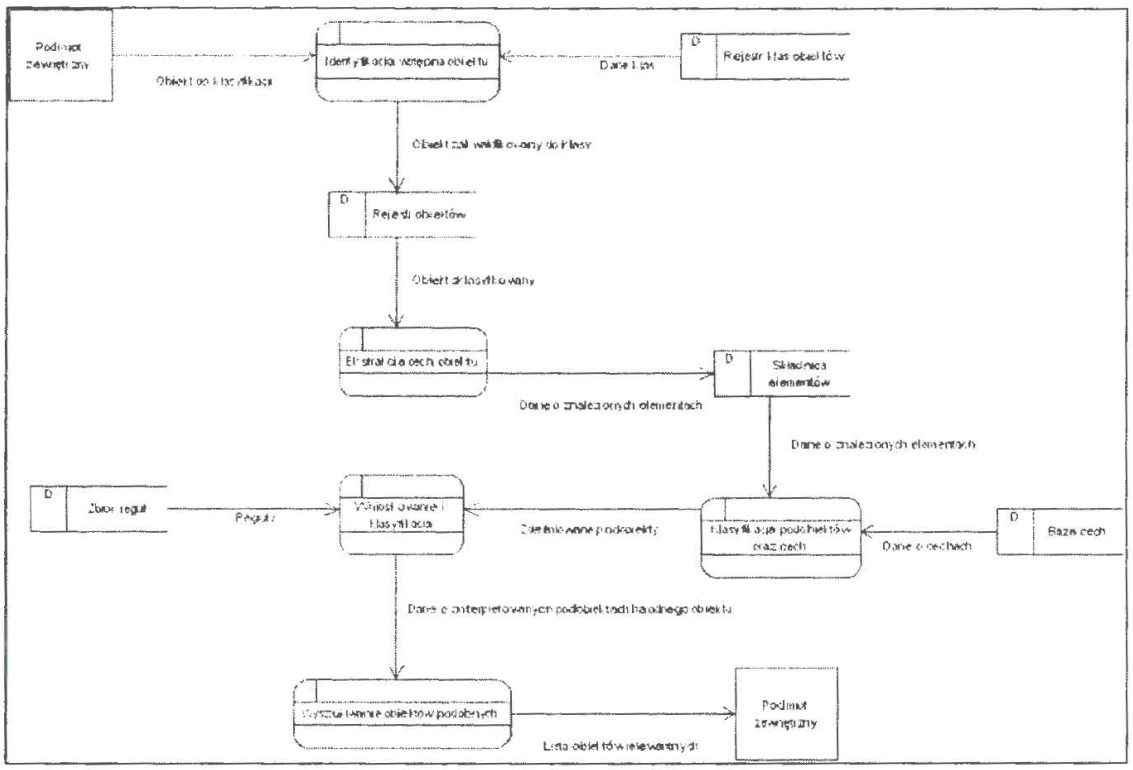
Analiza przepływu danych pokazuje podstawowe procesy przetwarzające dane, składnice danych (rejestry) przechowujące opisy obiektów, cechy obiektów (przechowujące wiedzę, która będzie używana do przetwarzania obiektów wejściowych) oraz przepływy danych wskazujące kierunek przepływu [13].

Rejestry to elementy, które będą uzupełnianie nową wiedzą. Zawartości tych elementów będą stanowiły o jakości działania budowanego systemu. Na rysunku 1 przedstawiono ogólny schemat przepływu danych w systemie porównywania obiektów złożonych.

W tabeli 1 przedstawiono opis poszczególnych elementów diagramu.

Tabela 1: Opis elementów diagramu DFD

NAZWA	OPIS
Podmiot zewnętrzny	Użytkownik lub system inicjujący oraz przekazujący obiekt do identyfikacji i klasyfikacji (najlepszej) do istniejących zbiorów referencyjnych
Identyfikacja wstępna obiektu	Rozpoznanie klasy obiektu, w celu dobrego wybrania zbioru cech do porównywania oraz zbioru reguł, np. obraz, wideo, dźwięk, ciąg znakowy, dokument tekstowy, etc.
Rejestr klas obiektów	Składnica typów klas obiektów, która dostarcza wymaganych danych do procesu identyfikacji
Rejestr obiektów	Składnica przechowująca tymczasowo przetwarzany obiekt.
Ekstrakcja cech obiektu	Wydobywanie cech obiektów przy pomocy technik dostępnych dla danej klasy obiektów, np. dla obrazów będą to operacje wyszukiwania krawędzie, histogramów kanałów kolorów, etc,
Składnica elementów	Tymczasowa składnica znalezionych elementów (obiektów). Elementy nie są jeszcze cechami gdyż nie są zinterpretowane.
Wnioskowanie i klasyfikacja	Proces interpretacji przynależności obiektu badanego do zbiorów elementów o danych cechach głównych . Interpretacja odbywa się poprzez reguły rozmyte [15][18], których spełnienie jest mierzone poprzez operatory i operacje logiki rozmytej
Zbiór reguł	Baza reguł definiujących ostateczną klasyfikację
Klasyfikacja pod-obiektów oraz cech	Proces klasyfikacji cech, a więc badania podobieństwa między znalezionym elementem a cechami opisanymi w bazie cech. Ten proces pozwoli na odrzucenie cech fałszywych (szumów) oraz nazwanie (interpretację) pozostałych
Baza cech	Rejestr cech, którą są opisane i scharakteryzowane posiadające swoje wartości (mierzone poprzez specjalistyczne miary), np. histogram barwy czerwonej
Podmiot zewnętrzny	Użytkownik lub system inicjujący oraz przekazujący obiekt do identyfikacji i klasyfikacji (najlepszej) do istniejących zbiorów referencyjnych
Wyszukiwanie obiektów podobnych	Znalezienie obiektów najbardziej podobnych, na podstawie klasyfikacji i wnioskowania



Rysunek 1: Diagram przepływu danych w systemie porównywania obiektów

Rysunek 1 wyjaśnia szczegółowe znaczenie poszczególnych elementów.

Na rysunku 2 pokazano ogólny diagram aktywności [21] komparatora, który stanowi podstawowy węzeł decyzyjny na poziomie atomowym. Komparator oparty jest o teorię zbiorów i relacji rozmytych [22]. Przedstawiony algorytm jest uogólnieniem algorytmu przedstawionego w [16].

Algorytm zakłada sprawdzenie podobieństwa obiektu wejściowego, ze zbiorem referencyjnym obiektów (w szczególności z każdym jego elementem). Do określenia podobieństwa wybranych obiektów, obliczamy funkcję przynależności do relacji. W zależności od klasy obiektów, ta funkcja będzie różnie dobierana, tak aby najlepiej mierzyła podobieństwa obiektów tej klasy lub danej cechy.

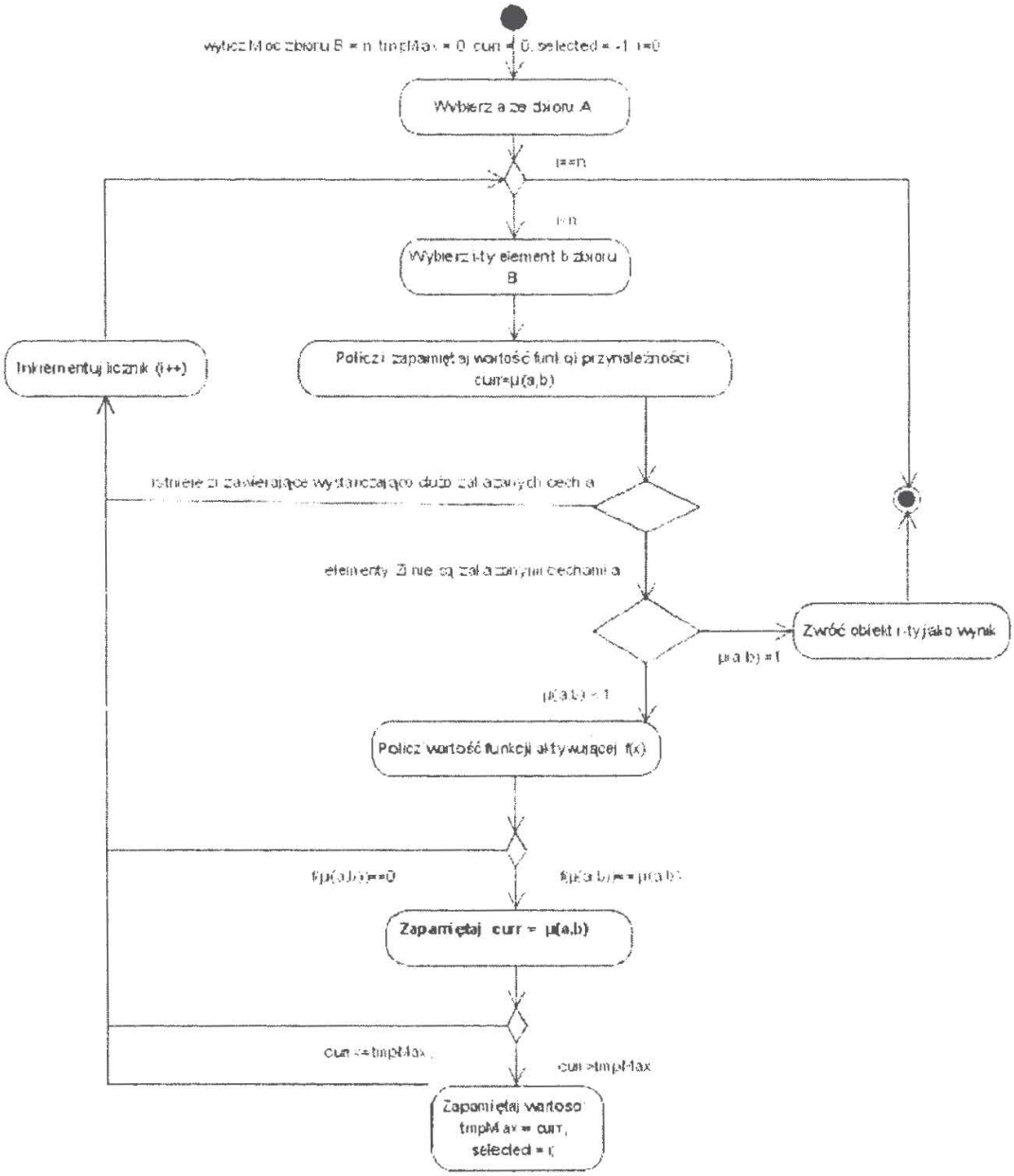
Zgodnie z definicją podaną w [9] jak również w źródłowej pracy [22], jest to funkcja $f : RxR \rightarrow [0,1]$, gdzie wartości skrajne przeciwdziedziny wskazują odpowiednio na całkowity brak podobieństwa („0") (przynależności do relacji) lub całkowite podobieństwa a zatem identyczność („1"). Następnie będziemy rozważać funkcję aktywacji, która na poziomie ogólnym ma spełniać zależność taką, iż jest to funkcja, która powyżej pewnej wartości „p" lub rów-

nej, jest funkcją identyczności, gdzie "p" jest z przedziału [0,1], a poniżej jest równa 0. Będzie to funkcja określająca minimalną jakość naszego rozwiązania.

Podobnie jak w przypadku funkcji przynależności, funkcja aktywacji - a dokładnie jej parametr "p" - będzie indywidualnie dopasowywany do konkretnej klasy obiektów, a być może do konkretnych obiektów referencyjnych. Do jego wyboru może być użyta wiedza ekspercka. Można również użyć algorytmów ewolucyjnych w celu próby optymalizacji doboru parametru (tak aby "p" był jak najmniejszy, a jednocześnie gwarantował dobrą jakość rozwiązania). Cechy zakazane zdefiniowane są w rodzinie zbiorów z_i takiej że dla każdego elementu zbioru referencyjnego istnieje zbiór z. Zbiór ten może być zbiorem pustym lub posiadać zdefiniowane elementy (cechy) stanowiące o wykluczeniu podobieństwa miedzy obiektem badanym posiadającym ową cechę, a tym obiektem referencyjnym. Jeśli cechy te występują to badanie podobieństwa traci sens i przechodzimy do wyboru kolejnego obiektu referencyjnego. Realizacja klasyfikacji cech zakazanych, została zaplanowana dla klasyfikatora rozmytego [12]. Dzięki temu klasyfikatorowi będzie można rozpatrywać nieostre występowanie cech, tzn. po pierwsze takie, że nie wszystkie cechy na raz występują i to nie będzie powodowało automatycznego niespełnienia warunku, jak również spełnienie występowania tylko w pewnym stopniu.

W końcowej fazie algorytmu następuje sprawdzenie wyliczona wartość podobieństwa jest wyższa od dotychczas sprawdzonych. Jeśli tak, to zapamiętujemy dane o obiekcie jako rozwiązanie kandydujące do rozwiązania końcowego. W wyniku działania algorytmu możemy dostać kilka różnych rodzajów wyników: a) brak wyników - brak przynajmniej jednego obiektu, dla którego relacja byłaby spełniona wraz zachowaniem warunku niewystępowania cech zakazanych oraz minimalnej jakości rozwiązania. b) dokładnie jedno rozwiązanie – jest dokładnie jeden obiekt najbliższy badanemu obiektowi c) wiele rozwiązań równoważnych – jest wiele obiektów tak samo podobnych.

W podanej wersji algorytm jako wynik zwróci pierwszy obiekt z tego zbioru wyników. Można również zmodyfikować algorytm tak, aby wszystkie wartości były zwracane lub np. ostatni obiekt ze zbioru wyników. Przyjmując możliwość zwrócenia jednego wyniku ze zbioru wyników, można optymalizować algorytm poprzez przyjęcie, iż osiągnięcie funkcji przynależności wartości "1" kończy poszukiwanie.



Rysunek 2: Diagram aktywności komparatora

2. Implementacja systemu

2.1. Reprezentacja obiektów

Obiekty przewidziane do przetwarzania przez system zostały podzielone na klasy (uprzednio zdefiniowane). Obiekty mogą być udostępniane w różnych formatach i postaciach. Przeważnie są to formaty kompresujące obiekty, aby łatwiej składować i przenosić obiekty miedzy systemami. Takie podejście, choć korzystne z punktu widzenia składowania, nie jest wygodne do przetwarzania.

Przetwarzanie wymaga szybkiego dostępu do danych właściwych, zdekodowanych. Przy projektowaniu systemu, została dopuszczona możliwość wydobywania informacji z obiektów składowanych (referencyjnych) w późniejszym czasie (nie podczas akwizycji obiektu). Wynika to stąd, iż zakładam ewolucyjność systemu, gromadzonej wiedzy i metod ekstrakcji. System ma być jedynie platformą ułatwiającą składowanie i przetwarzanie danych, może zmieniać zastosowanie i obszar badanych obiektów, tylko poprzez zmianę bazy wiedzy o obiektach lub regułach je definiujących.

Dlatego też przyjęta reprezentacja, musi być elastyczna ze względu na zdefiniowane wyżej aspekty. Zapewniając spełnienie tych postulatów, przyjąłem, iż każdy obiekt w postaci zdekodowanej (niezależnie od klasy) będzie przechowywany w bazie danych. Nie będą do tego jednak służyły specjalne typy pól dostępne w RDBMS'ach, takie jak BLOB, IMAGE, BYTE. Takie pola nie dawałyby swobody selekcji i manipulowania danymi. Przyjąłem, iż dla poszczególnych typów obiektów, będzie przygotowana struktura tabel przechowująca składowe obiektu. Dla typu obiektu "obraz", przechowywane będą pojedyncze piksele w postaci rekordów. Oznacza to, iż dla średniej rozdzielczości obrazu, w bazie danych pojawi się względnie dużo rekordów. Aby sprostać takiemu wyzwaniu, będzie potrzebne specjalne środowisko to składowania danych i zarządzania nimi. Dodatkowym problemem jest objętość jaką dane zdekodowane zajmują. Należy zatem również zwrócić uwagę na to, aby oprócz swobody dostępu, zapewnić realną możliwość przechowywania danych obiektów.

Moim wyborem jest RDBMS MySQL z dedykowanym silnikiem BRIGHTHOUSE, dzięki któremu będę w stanie spełnić wymogi postawione przed budowanym systemem.

2.2. Środowisko ICE

ICE (Infobright Community Edition) to darmowy silnik firmy Infobright zintegrowany z RDBMS MySQL. Służy on do budowy dużych hurtowni danych oraz analitycznych baz danych. Silnik posiada wiele sprzęgów połączeniowych oraz narzędzi klasy Business Intelligence. Ideą silnika jest podział kolumn z danymi na paczki danych (z ang. data packs) zawierające 64K wartości każda, a następnie analizy i przechowywania w sposób skompresowany zawartości paczki. Podstawowe informacje są składowane w postaci tzw. węzłów danych (ang. data pack nodes), zawierających wartości minimalne, maksymalne oraz agregaty dla każdej z paczek danych. Węzły danych, jak również bardziej zaawansowane węzły wiedzy (ang. knowledge nodes), służą dynamicznemu wspomaganiu optymalizacji i wykonania kwerend SQL'owych. ICE daje bardzo duży stopień kompresji danych. Określa się go jako większy niż

10:1. Poza tym, umożliwia w efektywny sposób budowę i zarządzanie hurtownią danych o wielkości do 30 Tb danych. Zapewnia to rozwiązanie postawionego przeze mnie problemu, potrzeby wyboru silnika wydajnego, a jednocześnie dbającego o wielkość przechowywanych danych.

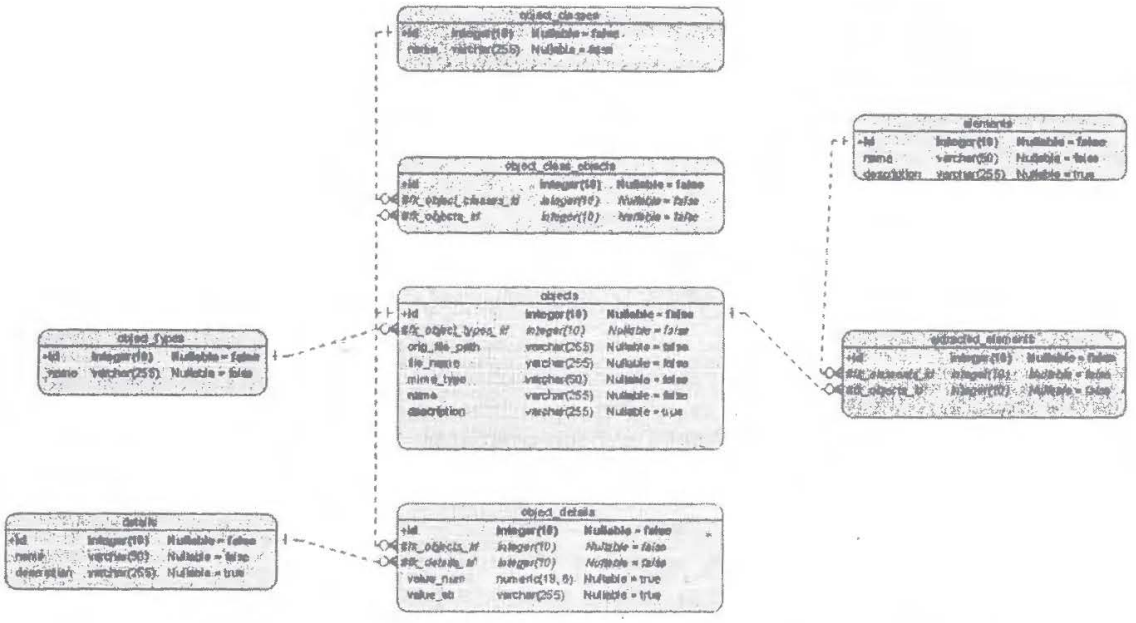
2.3. Składowanie danych

Projektowany system został podzielony na dwie podstawowe części: OLTP - część transakcyjną oraz ROLAP - dedykowane kostki do przechowywania odpowiednich danych. W części pierwszej, zapisywane będą metadane obiektów, ich podstawowe cechy, przynależności do klas, itp. Dane te mają służyć do łatwiejsze zarządzania obiektami, nie będą jednak miały bezpośredniego wpływu na jakość rozwiązań zwracanych przez system. Druga część systemu będzie zbudowana na zasadzie hurtowni danych, oparta na kostkach. Struktury te będą dedykowane dla poszczególnych klas obiektów.

W ramach tych klas kostki będą tworzone dla specjalizowanych danych. Takie podejście ma na celu wykorzystanie całego potencjału hurtowni danych, do analizy danych. W ramach klas obiektów, tworzone kostki będą miały niewątpliwie wspólne wymiary (choć niekoniecznie wszystkie). Dzięki temu, będą mogły być połączone i będą tworzyły tzw. konstelacje [19, 1]. Dzięki konstelacjom analiza danych staje się łatwiejsza poprzez porównania i agregowanie danych po wspólnych wartościach wymiarów. Dzięki temu można dane pokazywać jeszcze bardziej przekrojowo i w sposób uogólniony. Zakładam istnienie kostek dedykowanych dla obiektów zdekodowanych (ich danych) oraz tworzenie kostek do przechowywania danych o pewnych elementach pozyskanych z obiektów głównych. Mogą to być fragmenty obiektów, jak również różnego rodzaju dane przetworzone, statystyki lub agregaty. Kostki takie nie będą z góry zdefiniowane. System dopuszcza tworzenie takich kostek w dowolnym momencie i w zależności od potrzeb.

2.4. Analiza związków encji

W celu pokazania praktycznej realizacji, przedstawię diagramy związków encji wraz z pełnym zakresem informacyjnym. W części dotyczącej kostek, zgodnie z opisem w poprzednim rozdziale, przedstawię jedynie propozycję najprostszych kostek, które wraz z rozwojem systemu będą dodawane i łączone w konstelacje.

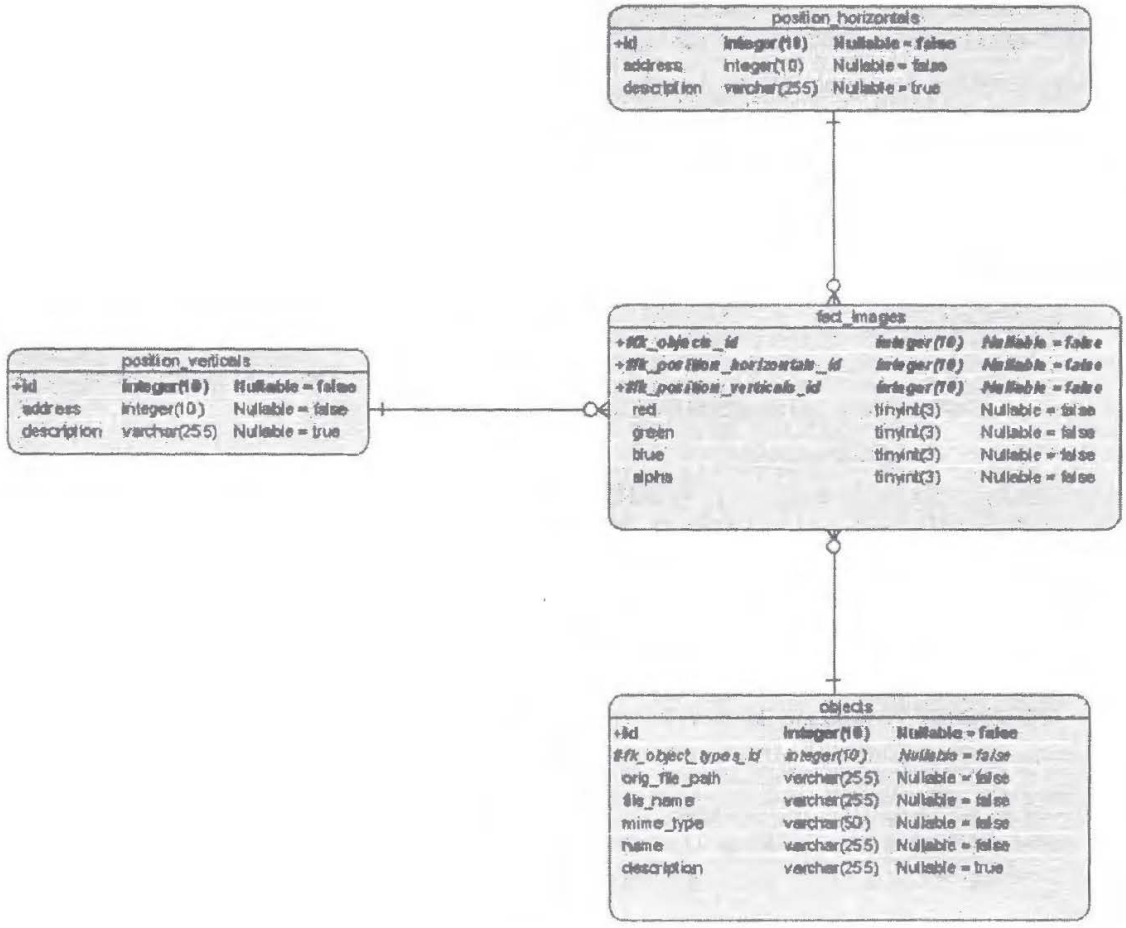


Rysunek 3: Diagram związków encji części OLTP

Schemat transakcyjny został przedstawiony na rysunku 3 obrazujący powiązania pomiędzy poszczególnymi encjami tej części systemu. W tabeli 2 podano dokładny opis poszczególnych encji. Główną ideą tej części projektu, jest zapewnienie tam gdzie to możliwe standaryzacji i projektowania systemu przynajmniej w trzeciej postaci normalnej (3PN) [20, 7] celem łatwego poszerzania danych systemu.

W celu łatwego zarządzania schematem wprowadziłem dedykowaną notację dla atrybutów encji a docelowo dla pól tabel. W przypadku atrybutów, które następnie będą stanowiły klucze obce, stosuję prefiks "FK" (od ang. Foreign Key), w celu łatwiejszej identyfikacji atrybutów, po których może następować złączenie. Po prefiksie pojawia się nazwa tabeli, do której zdefiniowana jest referencja oraz na końcu atrybut (pole) w tabeli docelowej do którego referencja się odnosi. Taki schemat nazewnictwa działa dla referencji opartych na pojedynczych atrybutach. W praktyce jednak takie przypadki stanowią zdecydowaną większość ze względów wydajnościowych. W tym projekcie tego rodzaju związki są jedynymi stosowanymi.

Kolejne aspekty systemu dotyczą kostek ROLAP, które powstają do analizowania poszczególnych typów zagadnień. Mając na uwadze bardzo dużą ilość danych, które będą przetwarzane, zakładam słuszność podejścia do tego problemu jako do problemu analizy danych w hurtowniach danych.



Rysunek 4: Kostka ROLAP przechowująca dane o obiektach typu obraz

W projektach został użyty schemat gwiazdy [1], w celu zapewnienia odpowiedniej efektywności wykonywania analiz. Poniżej znajdują się jedynie wybrane projekty kostek, które dotyczą klasy obiektów - obrazy, jednakże nie są kompletnym zbiorem kostek ROLAP, a jedynie podstawowym zestawem. W miarę postępu badań, struktur kostkowych będzie przybywać. Poszczególne kostki będą połączone wspólnymi wymiarami, dzięki czemu będą tworzyły wspomniane wcześniej konstelacje.

Rysunek 4 przedstawia główną kostkę dla typu obiektów "obraz". Kostka ta realizuję przechowywanie danych obiektów tego typu w sposób zdekodowany, tzn. pojedyncze piksele w postaci rekordów w tablicy faktów. Wymiarami są jedynie współrzędnie na osi X oraz osi Y oraz wymiar obiektu. Dzięki tej kostce uzyskujemy dostęp do dowolnego obszaru badanego obrazu przy pomocy jednego zapytania. Do analizy dodatkowo będą nam potrzebne specjalne funkcje realizujące operatory jednopunktowe oraz operacje sąsiedztwa do przetwarzania obrazów [6, 5]. Dlatego też stosowne funkcje realizujące odpowied-

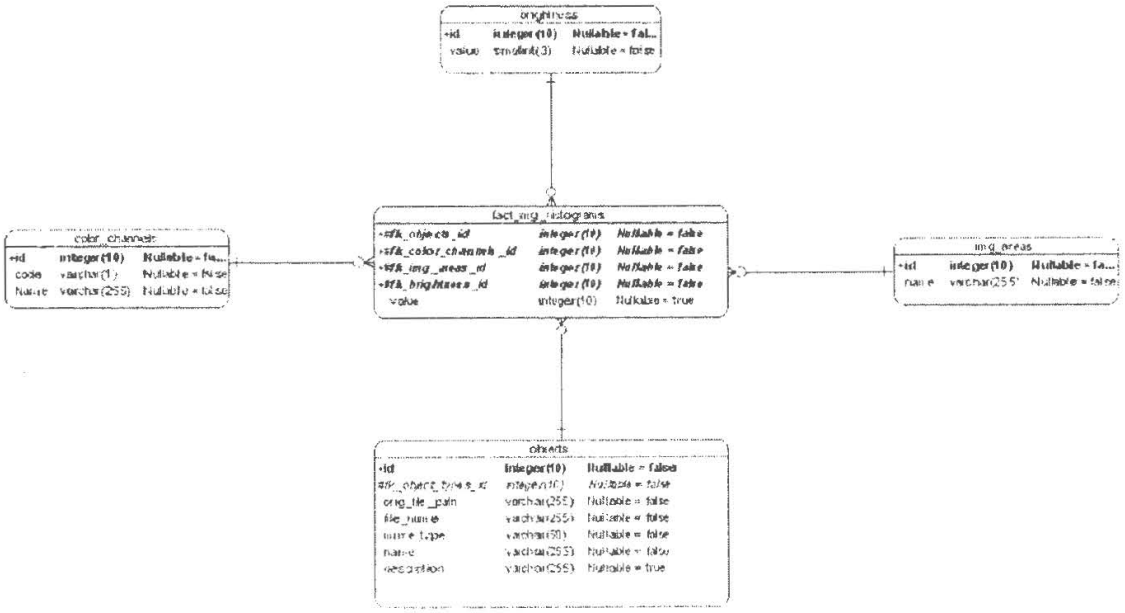
nie operacje, zdefiniowałem z postaci funkcji składowanych. Przykładem zaimplementowanych operatorów jest: operator odwrotności, operator binaryzacji, operator dodawania, operator odejmowania, operator mnożenia, filtr dolnoprzepustowy (dla sąsiedztwa ośmiospójnego). Dzięki temu podstawowe operacje przetwarzania obrazów, mogą być wykonywane bezpośrednio w bazie danych, przy pomocy wydajnego silnika BRIGHTHOUSE. Tabela 3 przedstawia opis tabeli faktów oraz wymiarów prezentowanej kostki ROLAP.

Tabela 2: Opis encji części OLTP

NAZWA	OPIS
object_classes	Klasy obiektów, byt niezależny od typu. Klasa określa pewien podzbiór związany z tematyką, rodzajem, itp., np. ludzie, architektura, zdjęcia z wakacji. Klasa to pewnego rodzaju kategoria związana z merytorycznym znaczeniem obiektu a nie formą obiektu (zdjęcie, dźwięk, etc)
elements	Słownik elementów (cech) wydobytych z obiektów
object_class_objects	Intersekcja ewidencjonująca przynależność obiektu do danej klasy (może przynależeć do wielu)
objects	Encja opisująca obiekty zarejestrowane w bazie danych. Jest to główna encja która posiada dane dotyczące typu obiektu, ścieżek, etc. Stanowi również wymiar dla kostek ROLAP poszczególnych typów obiektów
object_types	Słownik typów obiektów
extracted_elements	Przypisanie elementów (cech) do obiektu
object_details	Wartości szczegółów obiektów
details	Słownik szczegółów, które mogą dotyczyć obiektów

Tabela 3: Opis kostki ROLAP dotyczącej danych obrazów

NAZWA	OPIS
position_horizontals	Wymiar dotyczący pozycji poziomej piksela
position_verticals	Wymiar dotyczący pozycji pionowej piksela
objects	Encja opisująca obiekty zarejestrowane w bazie danych. Jest to główna encja która posiada dane dotyczące typu obiektu, ścieżek, etc. Stanowi również wymiar dla kostek ROLAP poszczególnych typów obiektów
fact_images	Tablica faktów związana z obiektem typu obraz. Jeden rekord będzie reprezentował jeden piksel obrazu



Rysunek 5: Diagram związków encji kostki histogramowej

Kolejną strukturą ROLAP skonstruowaną na potrzeby przetwarzania obrazów jest kostka dedykowana dla histogramów [14]. Zadaniem tej kostki i udostępnienie informacji o histogramie poszczególnych obiektów lub fragmentów obiektów. Dzięki tej kostce będziemy operowali na danych zagrego-

wanych, wcześniej przeliczonych, co powoduje, iż nie ma konieczności liczenia histogramów dla każdego obiektu w celu porównania z obiektem wejściowym za każdym razem. Takie wyliczenie może być wykonane przy pierwszym odwołaniu do danych. Rysunek 5 przedstawia diagram ERD schematu gwiazdy tej kostki, natomiast tabela 4 opisy encji wymiarów i faktów.

3. Przykład

Załóżmy, iż w systemie istnieją dane trzech obiektów od ID = {1,2,3} Obiekt o ID=1 i ID=2 to obiekty referencyjne a obiekt o ID=3 obiekt badany. Obiekty należną do klasy "obrazy" oraz za pomocą ich danych zostały zasilone ww. kostki. A zatem dysponujemy trzema histogramami obiektów (dla każdego kanału niezależnie). Sprawdzimy jak zastosować komparator w celu diagnozy czy obiekt o ID=3 jest podobny i w jakim stopniu do pozostałych. Praktycznym zastosowaniem może być algorytm "odsiewania pustych stron".

W celu zmniejszenia liczby danych do porównywania histogram zostanie skwantyfikowany, tak że wartości pikseli nie będących wielokrotnością "n" zostaną zaokrąglone w dół do najbliższej takiej liczby. Tutaj przyjąłem n=10. W celu kwantyzacji histogramu wykonuję podaną poniżejinstrukcję SQL na kostce histogramowej:

```
SELECT CC.CODE,B.VALUE - (B.VALUE mod 10) QB,SUM(FIH.VALUE)
AS HV FROM FACT_IMG_HISTOGRAMS FIH INNER JOIN BRIGHT-
NESS B ON FIH.FK_BRIGHTNESS_ID=B.ID INNER JOIN
COLOR_CHANNELS CC ON FIH.FK_COLOR_CHANNELS_ID=CC.ID
WHERE FIH.FK_OBJECTS_ID=1 AND FIH.FK_IMG_AREAS_ID=1
GROUP BY CC.CODE,(B.VALUE - (B.VALUE mod 10))
```

Dzięki kwantyzacji histogramu zamiast 256 wierszy zapytanie zwróci maksymalnie 26, co znacznie uprości porównywanie i nie wpłynie ujemnie na wyniki. W tabeli 5 znajdują się skwantyfikowane histogramy dla trzech badanych obiektów.

Kostki histogramowe zawierają niewiele rekordów, ponieważ dla każdego obiektu znajduję się tam maksymalnie 256 * 4 rekordy. Wynika to z faktu, iż przestrzeń jasności ma 256 elementów i zapisuje 4 kanały (czerwony, zielony, niebieski oraz kanał alfa). Dlatego też zapytanie kwantujące wykonuje się z kilka milisekund i jest całkowicie niezauważalne z punktu widzenia czasu przetwarzania. Wartości histogramów dla niektórych kwantów jasności są duże. Wynika to z rozdzielczości obrazów oraz samych obrazów. Obraz o ID =1 to obraz pustej białej strony, natomiast obraz o ID = 2 to czarna strona. Wszystkie obiekty są w rozdzielczości 1024x1408, dlatego też jak łatwo policzyć tablica faktów kostki odpowiedzialnej za ewidencje samych obiektów (w tym przypad-

ku obrazów), ma dużo więcej elementów. Dokładnie jest to 1024*1408*3 = 4325376 rekordów.

W celu realizacji algorytmu skonstruuję komparatory K1, K2 i K3 odpowiadające kolejno za kanały barw: czerwony, zielony, niebieski. Komparator K_i zbada podobieństwo obiektu o ID=3 do obiektów zbioru referencyjnego o ID=1 i ID=2. Zakładamy, że liczba pikseli poszczególnych obrazów jest identyczna. Jako funkcję przynależności do relacji przyjmuję

$$\mu(a,b) = 1 - \frac{\sum_{j=0}^{j=n-1} | a_j - b_j |}{2},$$

gdzie „n" - to wymiarowość przestrzeni jasności kolorów (tutaj 256), a „j" - j-ty indeks unormowanego wektora histogramu. Unormowanego ponieważ wektory histogramów obiektów mają postać

$$b=[(\frac{hist(0)}{\sum_{i=0}^{i=n-1} hist(i)});(\frac{hist(1)}{\sum_{i=0}^{i=n-1} hist(i)});...;(\frac{hist(n-1)}{\sum_{i=0}^{i=n-1} hist(i)})],$$

gdzie funkcja hist(x) zwraca liczbę pikseli o jasności "x" w badanym kanale. Następnie wyniki zwrócone przez komparatory zasilą regułę rozmytą, która ostatecznie zinterpretuje otrzymane wyniki.

Komparator K1 bada podobieństwo dla kanału barwy czerwonej. Zbiory wyjątków z_i dla wszystkich obiektów referencyjnych są zbiorami pustymi. Jakość minimalnego rozwiązania dopuszczalnego określono za pomocą parametru p na "0.999999".

Mając obliczone wartości histogramów obiektów możemy wyliczyć wektory b1,b2,b3. Badamy podobieństwo b3 z

b1 dla K1. $\mu(b_3,b_1) = 1 - \frac{\frac{55856}{1441792}}{2} = 1 - 0.19370 = 0.980630,$

$$\mu(b_3,b_2) = 1 - \frac{\frac{2883568}{1441792}}{2} = 1 = 0.999994 = 0.000006.$$

Widać, iż oba obiekty są zbyt mało podobne, choć b1 jest zdecydowanie bardziej podobny do b3 niż b2. W tym przypadku wyliczanie kolejnych kompara-

torów nie ma już sensu, gdyż całość reguły i tak nie będzie spełniona ze względu na K1.

W innym przypadku należało by obliczyć pozostałe komparatory. Na końcu zbadać spełnienie reguły: JEŚLI K1=b1 AND K2=b1 AND K3=b1 TO b3 jest podobne do b1 lub JEŚLI K1=b2 AND K2=b2 AND K3=b2 TO b3 jest podobne do b2.

Tabela 4 Opis encji kostki histogramowej

NAZWA	OPIS
color_channels	Wymiar definiujący kanały,np. czerwień (R), zieleń (G), niebieski (B), alfa (A)
fact_img_histograms	Tablica faktów związana z kostką ROLAP dotyczącą analizy histogramów obrazów
img_areas	Wymiar definiujący obszar którego dotyczy histogram. W szczególności może to być cały obraz, bądź jakiś jego fragment (obiekt na obrazie, etc)
Brigthness	Wymiar dotyczący jasności pikseli
Objects	Encja opisująca obiekty zarejestrowane w bazie danych. Jest to główna encja która posiada dane dotyczące typu obiektu, ścieżek, etc. Stanowi również wymiar dla kostek ROLAP poszczególnych typów obiektów

Tabela 5: Skwantyfikowane histogramy dla obiektów o ID=1 (lewa górna tabela), ID=2 (lewa środkowa tabela) oraz ID=3 (lewa dolna tabela oraz prawa tabela)

ID1	R	G	B
40	1	1	1
70	1	1	1
160	3	3	3
190	0	0	2
200	2	2	5
210	5	5	9
220	14	16	14
230	25	26	34
240	332	331	387
250	1441409	1441407	1441336

ID2	R	G	B
0	1441264	1441263	1441260
10	524	524	520
20	2	3	10
40	1	1	1
50	1	1	1

ID3	R	G	B
10	4	1	1
20	34	1	1
30	337	39	24

ID3	R	G	B
40	1006	251	166
50	1497	851	610
60	1800	1456	1208
70	1835	1804	1680
80	1292	1727	1344
90	934	1577	864
100	767	1114	655
110	702	829	625
120	699	746	567
130	705	753	639
140	606	627	714
150	672	650	865
160	664	643	1050
170	678	694	1148
180	765	775	1108
190	834	837	1018
200	800	777	911
210	972	948	925
220	1180	1097	1128
230	1772	1392	1394
240	7756	6060	7805
250	1413481	1416143	1415342

Zakończenie

W artykule pokazano podstawowe części systemu, stanowiące podstawę do rozbudowy i dalszej analizy. Pokazano również przykład użycia struktur ROLAP do analizy obiektów. Dalsze kroki w rozwoju systemu, to dalsze prace nad sposobami porównywania obiektów i zapisu wyników oraz reguł interpretacyjnych. Prace będą również obejmowały dane z pozostałych wybranych klas obiektów. Dla tych klas będą również projektowane specjalistyczne struktury do analizy - kostki ROLAP, tworzące konstelacje. Kostki zostaną wypełnione danymi w celu możliwości sprawdzenia systemu w działaniu. Zostaną również wyodrębnione części wspólne jako elementy ułatwiające analizę, np. histogram zastosowany do obrazów będzie można również przenieść do klasy "teksty". Całość prac będzie obejmować kompletny system z przykładową baza wiedzy i reguł dostosowaną do wybranego obszaru badanych obiektów.

Literatura

[1] Agosta L. (2000): *The Essential Guide to Data Warehousing.*

[2] Cantu-Paz E., Cheung S-C,Kamath C. (2003): *Retrieval of Similar Objects in Simulation Data Using Machine Learning Techniques.*

[3] Choraś R.(2005): *Komputerowa wizja. Metody interpretacji i identyfikacji obiektów.*

[4] Cohen W., Ravikumar P.,Fienberg S. (2003): *A Comparison of String Distance Metrics for Name-Matching Tasks.*

[5] Cytowski, Gielecki, Gola (2008): *Cyfrowe przetwarzanie obrazów medycznych.*

[6] Doros M. (2000): *Przetwarzanie obrazów*, Materiały pomocnicze.

[7] Jaszkiewicz A. (1997): *Inżynieria oprogramowania.*

[8] Jaworska T., *Analysis of Object Features in Terms of the Dissimilarity of Pattern Recognition.*

[9] Kacprzyk J. (2001): *Wieloetapowe sterowanie rozmyte.*

[10] Kowalczuk Z., Wiszniewski B. (2007): *Inteligentne wydobywanie informacji w celach diagnostycznych.*

[11] Lorenz A., Blüm M., Ermert H., Senge Th., *Comparison of Different Neuro-Fuzzy Classification Systems for the Detection of Prostate Cancer in Ultrasonic Images.*

[12] Nowicki R. (2009): *Rozmyte systemy decyzyjne w zadaniach z ograniczoną wiedzą.*

[13] Roszkowski, J. (2004): *Analiza i projektowanie strukturalne.*

[14] Russ J. (2006): *The Image processing Handbook.*

[15] Rutkowski L. (2006): *Metody i techniki sztucznej inteligencji.*

[16] Sosnowski Ł. (2009): *Inteligentne dopasowanie danych przy użyciu teorii zbiorów rozmytych w systemach przetwarzania danych.* W: Analiza systemowa w finansach i zarządzaniu T.1 1 pod redakcją prof. J. Hołubca.

[17] Stąpor K. (2005), *Automatyczna klasyfikacja obiektów.*

[18] Szczepaniak P. (2004): *Obliczenia inteligentne, szybkie przekształcenia i klasyfikatory.*

[19] Todman C. (2005): *Projektowanie hurtowni danych.*

[20] Ullman J., Garcia-Molina H., Widom J. *Systemy baz danych - pełny wykład.*

[21] Wrycza S. (2005): *Język UML 2.0 w modelowaniu systemów informatycznych.*

[22] Zadeh L. (1965): *Fuzzy Sets, Information and Control*, vol. 8.