

KIWIEL



**POLSKA AKADEMIA NAUK**  
**Instytut Badań Systemowych**

# **WSPOMAGANIE DECYZJI**

# **SYSTEMY EKSPERCKIE**

pod redakcją

**Romana Kulikowskiego i Lucyny Bogdan**

Warszawa 1995

# **WSPOMAGANIE DECYZJI**

## **SYSTEMY EKSPERCKIE**

pod redakcją

**Romana Kulikowskiego i Lucyny Bogdan**

Warszawa 1995

Wydano z wykorzystaniem dotacji  
KOMITETU BADAŃ NAUKOWYCH

Materiały konferencji: "Analiza Decyzyjna, Systemy Ekspertckie, Zastosowania Systemów Komputerowych",  
Warszawa, 25-27 maja 1994r.

Komitet Programowy Konferencji:

Andrzej Ameljańczyk, Zdzisław Bubnicki, Wiesław Grudzewski, Olgierd Hryniewicz, Janusz Kacprzyk, Lech Kruś, Roman Kulikowski (przewodniczący), Kazimierz Mańczak, Ireneusz Nykowski, Zdzisław Pawlak, Roman Słowiński, Andrzej Straszak, Andrzej Weryński, Andrzej Wierzbicki.

Wykonano z oryginałów tekstowych dostarczonych przez autorów

© Instytut Badań Systemowych PAN, Warszawa 1995

ISBN 83-85847-85-5

# PUSTE ODPOWIEDZI W RELACYJNYCH BAZACH DANYCH A ICH RELAKSACJA Z UŻYCIEM LOGIKI ROZMYTEJ

Dariusz Łazarz  
Instytut Badań Systemowych PAN  
Newelska 6, 01-447 Warszawa

## 1. Wstęp

Wyszukanie pożądanej informacji z komputerowej bazy danych to proces, który w wydatny sposób wspomaga podjęcie właściwej, najbardziej racjonalnej decyzji. Obecne systemy informatyczne wspomagające ten proces, mimo, że stają się coraz bardziej zaawansowane i wyrafinowane, są jednak w dalszym ciągu niedostateczne i to z wielu punktów widzenia. Jednym z istotnych elementów, który wydatnie ogranicza szybkie uzyskanie informacji jest tradycyjny, mało elastyczny język zapytań. Ponieważ człowiek rozumie i komunikuje się w kategoriach lingwistycznych (języka naturalnego), więc należałoby tak konstruować języki zapytań, aby były one jak najbardziej zbliżone do języka naturalnego. Jednak ich rozwój poszedł w nieco innym kierunku. Podjęto mianowicie wiele wysiłków i starań w zakresie ich optymalizacji pod względem algorytmu wykonania. Jednocześnie niewiele poczyniono, aby polepszyć jakość odpowiedzi na zapytanie sformułowane przez użytkownika. Składnia większości języków zapytań nie przewiduje sformułowań (klauzul), które precyzowałyby jak zachować się ma system, aby wpłynąć na lepsze zrozumienie otrzymanej odpowiedzi.

Jednym z przejawów niedoskonałości odpowiedzi otrzymywanej na zapytanie skierowane do bazy danych jest odpowiedź pusta. Celem niniejszej pracy jest analiza zagadnienia *pustych odpowiedzi*, głównych podejść i kierunków badawczych, ze szczególnym uwzględnieniem koncepcji powstałych z wykorzystaniem logiki rozmytej oraz zarysowanie koncepcji systemu FRQUERY.

## 2. Systemy odpowiedzi bezpośredniej

W większości istniejących systemów relacyjnej bazy danych odpowiedzi z systemu sprowadza się do dwóch skrajnych sytuacji. W pierwszej, w odpowiedzi na postawione zapytanie użytkownik otrzymuje niepusty zbiór rekordów pochodzący z bazy danych (niepusta odpowiedź). Druga, sytuacja wystąpi wtedy, gdy odpowiedź jest

pusta. Oznacza to, że zapytanie użytkownika zostało spełnione przez zerową liczbę rekordów. Systemy generujące takie sytuacje określa się mianem systemów odpowiedzi *bezpośredniej*. Systemy te w zasadzie nie proponują nic ponadto co zostało określone w warunkach zapytania i znajduje się w bazie danych. Ta forma odpowiedzi, jest często postrzegana jako nieefektywna i niejasna, a także jest wieloznaczna. Odpowiedź bezpośrednia z systemu może bowiem:

- nie zawierać w sobie istotnych dla użytkownika informacji;
- przedstawiać użytkownikowi pustą odpowiedź;
- być niezrozumianą przez użytkownika;
- zawierać informacje w formie mało czytelnej dla użytkownika, grup użytkowników;
- nie posiadać mechanizmu różnicowania zawartych w niej informacji, itp.

Odpowiedź bezpośrednia u użytkowników, szczególnie początkujących, może budzić niechęć i przerażenie. Jednakże dopiero pusta odpowiedź bezpośrednia staje się prawdziwie nieprzyjemna nawet dla doświadczonych użytkowników. Więcej, forma ta często niesie za sobą negatywne konsekwencje. Wystarczy wymienić tylko niektóre takie jak możliwość:

- wprowadzenia użytkownika w błąd;
- wzbudzenia w nim wątpliwości poprzez wieloznaczność odpowiedzi;
- uniemożliwienia podjęcia właściwej decyzji.

Te negatywne konsekwencje wynikają z faktu istnienia dwóch stanów *pustej* bezpośredniej odpowiedzi. Po pierwsze, możemy mieć do czynienia z *pożądanym* stanem pustej odpowiedzi bezpośredniej. Oznacza to, że rzeczywiście zapytanie użytkownika, odpowiadające jego intencjom, jest skonstruowane prawidłowo pod względem syntaktycznym, a baza danych odpowiada negatywnie na zapytanie i taki jest stan faktyczny jej zasobów. Taka odpowiedź jest z punktu widzenia użytkownika jak najbardziej poprawna, chociaż pusta. Istnieje natomiast drugi stan pustej odpowiedzi bezpośredniej, który określamy mianem *niepożądanego*. Jest to również odpowiedź pusta aczkolwiek nie wynika to z rzeczywistego stanu zasobów bazy danych. Sytuacja niepożądanego pustej bezpośredniej odpowiedzi powstaje w wyniku popełnienia "błędu" przez użytkownika w jakimś punkcie podczas konstrukcji zapytania. "Błąd" taki może przyjmować różne formy i może mieć dalekosiężne skutki na zawartość otrzymanej odpowiedzi. W tym miejscu warto podać prosty przykład.

#### *Przykład*

Załóżmy, że mamy zbiorowość pracowników (*pracownicy*) firmy opisanych danymi takimi jak *wiek, zarobki, stanowisko pracy* itd. Załóżmy dalej, że w tej zbiorowości mamy dwóch wicedyrektorów w wieku 36 lat, dyrektora w wieku 37 lat i jednego kierownika działu w wieku 38 lat. Załóżmy teraz, że kierujemy do tej bazy danych następujące zapytanie (SQL):

```
SELECT * FROM Pracownicy
WHERE Wiek < 35 AND Stanowisko pracy >= kierownik
```

Na zapytanie o "młodych przedstawicieli kadry kierowniczej" przy tradycyjnym rozumieniu, że "młody" odpowiada wiekowi do 35 lat, odpowiedź będzie pusta. Ewidentnie jest to niezgodne z naszym odczuciem i oczekiwaniem, które wskazuje, że w rzeczywistości kadra kierownicza w tej firmie jest faktycznie "młoda". W tym przypadku "błąd" użytkownika i w konsekwencji pusta odpowiedź wynika ze zbyt sztywnego sformułowania warunków na "młody" w zapytaniu. Zakładając ponadto, iż ów użytkownik otrzymał wcześniej informację wskazującą na istnienie "młodej kadry kierowniczej", dochodzi do wzbudzenia wątpliwości.

### 3. Systemy neutralizacji bezpośrednich pustych odpowiedzi

Powyższe rozważania utwierdzają nas również w przekonaniu, że puste odpowiedzi bezpośrednie są nieprzyjazną formą odpowiedzi z systemu, oraz, że istnieje potrzeba ciągłego udoskonalania języków zapytań, w które wyposaża się obecne systemy zarządzania bazami danych w celu neutralizacji takich zachowań. Udoskonalenia winny iść w kierunku, który zapewni im między innymi takie cechy odpowiedzi jak informatywność, czytelność, wyeliminuje puste niepożądane bezpośrednie odpowiedzi, wskaże szybki sposób uzyskania dodatkowych informacji, itd.

W związku z tą niedoskonałością systemów odpowiedzi bezpośredniej zaczęły powstawać systemy, których zadaniem jest odejście od bezpośredniej odpowiedzi lub jej przewyciężenie w momencie powstania, poprzez zaproponowanie pośredniej odpowiedzi. Proces ten nazywamy *neutralizacją* bezpośrednich pustych odpowiedzi. W oparciu o neutralizację powstają *systemy odpowiedzi pośredniej* zwane również *systemami przyjaznej odpowiedzi*. Podstawowym zadaniem tych systemów jest wspomoczenie użytkownika (w różnych formach) w procesie zdobywania niezbędnych i właściwych informacji określonych w jego zapytaniu do bazy danych. Istnieją różne rodzaje i formy tych systemów, tak jak różne są podejścia i metody generowania odpowiedzi pośrednich. Na przykład, wśród systemów odpowiedzi pośrednich istnieją systemy, które są ukierunkowane na niwelowanie pewnych przypadków odpowiedzi bezpośrednich, jak i też systemy, które mogą w miarę możliwości odnosić się do wszystkich przypadków bezpośrednich odpowiedzi. Ogólnie rzecz biorąc, system odpowiedzi pośrednich powinien charakteryzować się między innymi tym, że nie poda użytkownikowi informacji, która mogła by go wprowadzić w błąd. Podpowiedź powinna nieść za sobą możliwie dużo informacji, ale nie za dużo. Podpowiedź powinna zwracać jasne, przejrzyste i ściśle związane z zapytaniem informacje. Z istniejących w tym zakresie podejść, technik i metod można wymienić te najpopularniejsze, odnoszące się bezpośrednio do głównych źródeł pustych odpowiedzi w relacyjnym SZBD. Do technik tych należą:

- (a) *korekta błędnych założeń stanu bazy danych,*
- (b) *korekta błędnych intencji*

(c) *poszerzanie spektrum odpowiedzi na bazie modyfikacji kwerendy (relaksacja).*

Chcemy zaznaczyć, że nie znajdują się w naszym polu zainteresowania sytuacje pustych odpowiedzi wynikające z błędnej syntaktyki kwerendy, czyli ze źle sformułowanego zapytania pod względem syntaktycznym. W naszej opinii jest to zadanie dla systemów korekcyjnych, których przykładem jest system zaproponowany przez Motro [1].

#### 4. Technika relaksacji zapytania

W dalszej części rozpatrzmy technikę (relaksacji) generalizacji. Istotą podejścia generalizacji jest transformacja początkowego zapytania, produkującego pustą odpowiedź, tak aby w rezultacie mogło powstać nowe zapytanie, które będzie miało szersze spektrum odpowiedzi, a jednocześnie będzie semantycznie powiązane z początkowym zapytaniem. Odpowiedź takiego zrelaksowanego zapytania musi zawierać w sobie spektrum odpowiedzi zapytania początkowego. Dzięki takiemu warunkowi nałożonemu na tą transformację, mamy pewność, że jeżeli powstanie niepusta odpowiedź ze zrelaksowanego zapytania będzie ona częściowo (pośrednią) odpowiedzią na zapytanie początkowe. Wróćmy do naszego początkowego zapytania o "młodych przedstawicieli kadry kierowniczej" w podanym powyżej przykładzie. Dzięki mechanizmowi generalizacji, możemy zaproponować rozluźnienie początkowego warunku (wiek < 35), aby poszukać wartości najbliższej spełniających warunki. Jest to jak najbardziej zgodne naturalnym zachowaniem człowieka. Ponadto, odpowiedź ze zrelaksowanego zapytania, nie tylko dostarcza nam dodatkowych informacji, ale też może nas upewnić czy odpowiedź pusta (powstała z zapytania początkowego) jest prawidłową odpowiedzią, czyli inaczej czy mamy do czynienia z pożądaną czy też niepożądaną pustą odpowiedzią. Motro [2] stwierdza, że jeśli kilka kolejnych generalizacji tego samego początkowego zapytania produkuje zbiór pusty, to odpowiedź na zapytanie początkowe jest pożądaną (prawidłową) pustą odpowiedzią. Relaksacja czyli poszerzanie spektrum odpowiedzi na bazie modyfikacji kwerendy różni się zdecydowanie od innych technik wykorzystywanych w systemach pośrednich odpowiedzi. Cechą wyróżniającą tę technikę (niezależnie od zastosowanego narzędzia) od innych jest fakt, że w swoim działaniu nie koncentruje się ona na jednym przypadku pustych odpowiedzi, ani nie zakłada a priori żadnej specjalnej klasy czy przypadków pustych odpowiedzi. Jakość tego działania zależy głównie od dwóch parametrów, jakimi są stopień transformacji początkowego zapytania (w jakim stopniu można je jeszcze poszerzać), oraz ilość i jakość informacji w bazie danych na daną chwilę. Są to ograniczenia, ale jesteśmy zdania, że bez tej wiedzy nawet człowiekowi jest trudno skonstruować satysfakcjonującą odpowiedź na pytanie.

Dotychczas pojawiło się kilka rozwiązań stosujących poszerzanie spektrum odpowiedzi na bazie modyfikacji kwerendy (generalizację) w relacyjnych systemach zarządzania bazą danych. Jako kryterium klasyfikacji można przyjąć kryterium rodzaju narzędzia wykorzystywanego do transformacji pojęć, jakie wybrano do implementacji techniki generalizacji w kwerendzie. Właśnie ta część mechanizmu jest najistotniejszym elementem sprawnego i wydajnego systemu transformacji zapytania i to właśnie ona w głównej mierze decyduje o powodzeniu i jakości zastosowanej metody. Na tej podstawie wyróżniliśmy trzy klasy systemów. Są nimi modele wykorzystujące:

- (a) hierarchię pojęć;
- (b) reguły;
- (c) pojęcia rozmyte.

### 5. System rozmytych pojęć

Skoncentrujemy się obecnie na modelach *rozmytych pojęć*. Systemy z tego nurtu generalizacji nakierowane są na skonstruowanie mechanizmu transformacji, który pozwoliłby na zaistnienie w relacyjnych SZBD syntaktyki i semantyki rozmytych pojęć. Przykładem takiego modelu, który dalej rozpatrujemy, jest model nazwany *systemem rozmytych kwerend*. W *systemie rozmytych kwerend* definiowanie rozmytych pojęć odbywa się poprzez zastosowanie teorii zbiorów rozmytych Zadeh'a [3]. Zastosowanie teorii zbiorów rozmytych w definiowaniu pojęć rozmytych dostarcza całościowej metody (w jednym mechanizmie) realizacji generalizacji kwerendy. Reprezentowany w pracach między innymi Bosca [4], Kacprzyka [5] Larsena [6] i innych *system rozmytych kwerend* zakłada, że baza danych pozostaje niezmienna, podczas gdy rozszerzeniu o rozmyte pojęcia ulega tylko język zapytań. Jest to mechanizm zaimplementowany jako nadbudowa do standardowego, istniejącego systemu. Współpracując wraz z podstawowym językiem zapytań, jest on w stanie formułować rozszerzone (rozmyte) zapytania, bez konieczności zmiany pierwotnego systemu składni zapytań. Mechanizmy relaksacji zaimplementowane w tym systemie, pozwalają na skuteczną, neutralizację pustych odpowiedzi. W odróżnieniu do innych podejść, w tej transformacji użytkownikowi dano możliwość wykorzystania niemalże wszystkich elementów występujących w kwerendzie w celu poszerzenia zakresu jej odpowiedzi. Tymi elementami są predykaty, kwantyfikatory i agregatory. Mechanizmy relaksacji występujące w *systemach rozmytych kwerend* możemy pogrupować na:

Generalizację predykatów:

- proste rozmywanie precyzyjnych i rozmytych predykatów
- rozmywanie precyzyjnych i rozmytych predykatów z użyciem modyfikatorów

Generalizację kwerendy (globalna):

- sterowanie agregatorami kwerendy
- agregacja sterowana z wykorzystaniem mechanizmu wag
- agregacja sterowana kwantyfikatorem

Analiza zalet i wad różnych podejść doprowadziła do opracowania koncepcji systemu FRQUERY, w którym zastosowano następujące rodzaje relaksacji zapytań z wykorzystaniem logiki rozmytej:

- proste rozmywanie precyzyjnych i rozmytych predykatów
- sterowanie agregatorami kwerendy
- agregacja sterowana z wykorzystaniem mechanizmu wag



System FRQUERY zawiera najogólniej biorąc, pewne nowe elementy, głównie w zakresie sterowania agregacją kwerendy. System FRQUERY jest implementowany w środowisku relacyjnej bazy danych FoxPro for Windows 2.5 i będzie przedmiotem szczegółowej analizy w następnych pracach.

#### LITERATURA

- [1] MOTRO A. "FLEX: A tolerant and cooperative user interface to databases" IEEE Transactions on Knowledge and Data Engineering, vol.2, No.2, June 1990.
- [2] MOTRO A. "SEAVE: A Mechanism for Verifying User in Query Systems", ACM Transactions on Office Information Systems, Vol. 4, No. 4, October 1986.
- [3] ZADEH L.A. "Fuzzy sets", Information and control Vol.8, 1965.
- [4] BOSCH P., GALIBOURG M., HAMON G., "Fuzzy querying with SQL: extensions and implementation aspects, Fuzzy Sets and Systems, vol.28, 1988, pp. 333-349.
- [5] KACPRZYK J., ZADROŻNY S. and ZIOLKOWSKI A. "FQUERY III+: A human-consistent database querying system based on fuzzy logic with linguistic quantifiers" Information Systems Vol.14, No. 6, pp.443-453, 1989.
- [6] LARSEN H., YAGER R.R. "Retrieving Information by fuzzification of queries" Journal of Intelligent Information Systems, vol.2, No. 4, November 1993.

**ISBN 83-85847-85-5**

---

**W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy  
prosimy o kontakt  
z Instytutem Badań Systemowych PAN  
ul. Newelska 6, 01-447 Warszawa  
tel. 36-19-01 w. 241 e-mail: kotuszew@ibspan.waw.pl**