STUDIA INFORMATICA 2010
Volume 31 Number 2B (90)

Jan JAGIELSKI, Piotr WNĘK
Uniwersytet Zielonogórski, Instytut Metrologii Elektrycznej

JĘZYK NATURALNY W SYSTEMACH BAZ DANYCH

Streszczenie. W opracowaniu przedstawiono problem transformacji zapytań w języku naturalnym na równoważne zapytania w języku baz danych. Omówiono problem ekstrakcji informacji o modelu danych w bazie oraz przedstawiono system odpowiadający za ekstrakcję tychże informacji. Zostały poruszone również problemy, które pozostały do rozwiązania w przyszłości.

Słowa klucze: wyszukiwarka macierzowa, bazy danych, zapytania w języku naturalnym

NATURAL LANGUAGE IN DATABASES SYSTEMS

Abstract. This paper describes a problem of transformation from natural language queries into equivalent queries in databases. It discusses the problem of extracting information about the data model in database and provides a system responsible for extraction of such information. Authors also point out remaining problems to be solved in the future.

Keywords: matrix search engine, database, queries in natural language

1. Wprowadzenie

Ciągły rozwój systemów bazodanowych narzuca wymóg udoskonalania sposobów ekstrakcji wiedzy z baz danych. W pozycji [1] opisano rozwój systemów akwizycji wiedzy na przestrzeni ostatnich lat. Autor wskazuje, iż rozwój ten prowadzi do stworzenia systemów bazodanowych potrafiących udzielać odpowiedzi na zapytania postawione w języku naturalnym[1]. Znaczenie systemów informatycznych potrafiących wykorzystywać strukturę języków naturalnych stale rośnie, co widać w tendencji przechodzenia z „niskich" do „wysokich" po-

[1] Język, w którym użytkownik wypowiada się w sposób nieskrępowany pod względem naturalnym[2].

ziomów sterowania komputerem. Przykładem w tej kwestii są języki programowania, które w początkowych fazach swojego rozwoju polegały na zerojedynkowym kodowaniu poleceń. Dalszy rozwój doprowadził do powstania asemblera, z czasem ewolucja środowisk programistycznych stworzyła pojęcie obiektowych języków programowania [2, 12]. Dziś możemy spotkać się z próbami wykorzystania języka naturalnego w charakterze języka programowania [2, 13]. Podobnej ewolucji uległy systemy akwizycji wiedzy [1], obiektowe systemy baz danych mające za zadanie uwzględnić czynnik ludzki. Znaczy to, że ich celem jest lepsze niż dotychczas dopasowanie modeli pojęciowych i modeli realizacyjnych systemów do naturalnych zachowań [1, 3, 12].

2. Ekstrakcja wiedzy z baz danych

Systemy baz danych operują pojęciem języka zapytań bazy danych. Najczęściej stosowanym obecnie językiem tego typu jest język SQL, który ma konstrukcję zbliżoną do języka naturalnego. Jest niezależny od bazy danych i umożliwia realizację operacji określanych synonimem CRUD (ang. *CReate, Update, Delete*) [3]. Uniwersalność języka powoduje, iż może być on wykorzystywany w dowolnych systemach baz danych.

2.1. Języki zapytań do baz danych

Z założenia języki zapytań do baz danych pozwalają wyszukiwać informacje w bazie danych po ściśle określonych warunkach. System bazy danych realizując zapytanie, ma za zadanie przedstawić dane z bazy w sposób zrozumiały dla człowieka. Pomimo iż języki baz danych, takie jak SQL, są zbliżone do języka naturalnego, to dla osób niezaznajomionych z tematem konstrukcja zapytań w tym języku nastręcza sporych trudności.

Typowe języki zapytań do baz danych są w stanie realizować zapytania typu *znajdź pracownika zarabiającego 2000 złotych* natomiast nie potrafią realizować zapytań postaci *znajdź pracowników zarabiających około 2000 złotych*.

Zapytanie zawierające w swojej konstrukcji warunek niepewności, np. *około 2000*, nazywamy zapytaniem nieprecyzyjnym [4]. W praktyce każde zapytanie zawierające jawnie użyte wyrażenia języka naturalnego, zwane terminami lingwistycznymi, jak *wysoki*, *gruby*, *drogi*, *szybki* można zakwalifikować do tego typu zapytań. Terminy lingwistyczne określają [4]:

1. Nieprecyzyjne wartości, np. niskie zarobki.
2. Nieprecyzyjne porównania, np. zarobki znacznie niższe niż 2000 złotych.
3. Niestandardowe sposoby agregacji stopni spełnienia cząstkowych warunków zapytania.

W przypadku zapytań nieprecyzyjnych problemem jest ich realizacja w klasycznych relacyjnych bazach danych [5, 6]. W celu rozwiązania tego problemu i umożliwienia realizacji

zapytań zawierających w sobie pewien stopień niepewności buduje się systemy rozmytych baz danych, których konstrukcja bazuje na teorii zbiorów rozmytych i teorii możliwości jako podejściu do reprezentacji danych rozmytych. Pomimo istnienia działających rozwiązań rozmytych baz danych masowa migracja na te systemy nie następuje. Aby jednak udoskonalić działające i używane szeroko modele klasyczne baz danych, tworzy się modyfikacje języka SQL, które taką funkcjonalność będą w stanie zapewnić.

W pozycji [4] oraz [5] przedstawiono modyfikację języka SQL o nazwie SQLf, który jest w stanie realizować złożone zapytania do baz danych o charakterze rozmytym zawierające typowe terminy lingwistyczne jak *około, poniżej* lub *powyżej*. Twórcom języka SQLf zależało, aby dwa równoważne zapytania klasyczne, po wprowadzeniu do nich elementów rozmytych, nadal dawały identyczne wyniki [4].

2.2. Zapytania w języku naturalnym

W zakresie realizacji zapytań do systemów baz danych kolejnym krokiem będzie stworzenie systemu potrafiącego zwracać poprawne wyniki na zapytania formułowane w języku naturalnym. Pozwoli to na realizację systemów bazodanowych, obsługiwanych bezpośrednio przez człowieka nieznającego zagadnień technicznych. Człowiek będzie mógł prowadzić bezpośrednią rozmowę z systemem komputerowym i uzyskiwać od niego pożądane dane z bazy danych. Jednak zanim do tego dojdzie, należy pokonać wiele przeszkód związanych z rozumieniem przez maszyny, jakimi są komputery, języka naturalnego, jakim operuje człowiek [7]. Aby tego dokonać, należy zbudować modele otaczającego nas świata, które będą zrozumiałe dla komputerów. Te natomiast, aby mogły sprawnie działać, muszą dysponować algorytmami opisującymi sposób rozumienia tychże modeli.

Aktualnie systemy bazodanowe, aby móc odpowiedzieć na pytanie *pokaż pracowników zarabiających 2000 złotych*, muszą otrzymać zapytanie w postaci zgodnej z językiem zapytań do baz danych, np. SQL (rys. 1).

```
SELECT * FROM pracownicy WHERE zarobki=2000
```

Rys. 1. Zapytanie SQL do bazy danych
Fig. 1. Sample SQL query to database

Zapytanie takie jest transformowane na język zrozumiały dla komputera i ten, jeżeli jest w stanie, zwraca wynik na ekran w postaci zrozumiałej dla użytkownika. Język SQL jest sformalizowany i standaryzowany. Nie ma tu możliwości zbudowania zapytania w innej formie, niż według ustalonych z góry reguł. Nie można zapytania z rys. 1 zbudować w inny sposób, ponieważ system bazy danych wykryje niepoprawność zapytania i go nie zrealizuje albo zwróci niepoprawne wyniki.

Język naturalny nie jest tak sformalizowany i standaryzowany. Istnieją reguły określające konstrukcję zdania i sposób posługiwania się językiem, ale nie wymagają one od człowieka

tak dużej precyzji i konkretyzowania zapytań, jak język zapytań do baz danych. Aby móc zbudować SZBD[2] potrafiący realizować zapytania do baz danych w języku naturalnym, należałoby wcześniej zbudować model opisujący relacje pomiędzy językiem naturalnym a językiem SQL i bazą danych. Model ten powinien opisywać, w jaki sposób wypowiedzi realizowane w języku naturalnym powinny być konwertowane na równoważny język zapytań do baz danych.

3. Realizacja zapytań w języku naturalnym

W pozycji [8] opisano koncepcję systemu realizującego zapytania do systemu baz danych formułowanych w języku naturalnym. Całość systemu opiera się na zagadnieniach związanych z systemami konwersacyjnymi, czyli programami potrafiącymi naśladować ludzkiego rozmówcę. Dzisiejsze czat boty są już z powodzeniem wykorzystywane na stronach internetowych jako wirtualni eksperci zdolni pomóc przy wyborze towaru. Systemy konwersacyjne to kolejny etap rozwoju systemów ekspertowych.

3.1. System konwersacyjny jako translator

Jak wcześniej stwierdzono, do wykonania zapytania do baz danych w języku naturalnym, niezbędna jest translacja języka naturalnego na równoważny język zapytań do baz danych, na przykład SQL. System konwersacyjny jako warstwa realizująca proces konwersacji z użytkownikiem może wykonać proces tłumaczenia języka naturalnego na równoważny język zapytań. Użyty w badaniach system konwersacyjny bazuje na algorytmach systemu A.L.I.C.E (ang. *Artifical Lingusitic Internet Computer Entity*) wielokrotnie zdobywającego uznanie pod postacią nagrody Loebnera [10]. Wadą tego programu jest podejście algorytmiczne do problemu prowadzenia konwersacji z użytkownikiem. Oznacza to, że twórca bazy wiedzy takiego systemu musi przewidzieć wszelkie możliwe aspekty rozmowy, co w normalnych warunkach jest teoretycznie niewykonalne.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<aiml version="1.0.1" xmlns="http://alicebot.org/2001/AIML-1.0.1"
      xmlns:html="http://www.w3.org/1999/xhtml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://alicebot.org/2001/AIML-1.0.1
http://aitools.org/aiml/schema/AIML.xsd">
   <category>
      <pattern>Pokaz * z * </pattern>
      <template><srai>SELECT <star/> FROM <star index="2"/> </srai> </template>
   </category>
   <category>
      <pattern>Pokaz * z * gdzie *</pattern>
```

[2] SZBD[1] – System Zarządzania Bazą Danych, nazywany też serwerem bazy danych.

```
        <template>SELECT <star/> FROM <star index="2"/> WHERE <star index="3"/>
        </template>
    </category>
    <category>
        <pattern>Pobierz * z *</pattern>
        <template>SELECT <star/> FROM <star index="2"/> </template>
    </category>
</aiml>
```

Na powyższym listingu przedstawiono kod zawierający bazę wiedzy systemu translacyjnego. Baza wiedzy systemu została stworzona w oparciu o język AIML [11], który stanowi rdzeń wszystkich systemów konwersacyjnych bazujących na algorytmach systemu A.L.I.C.E. Programem, który przyjął funkcję systemu konwersacyjnego w badaniach, jest ProgramD, który jest implementacją interpretera języka AIML i pozwala na realizację systemów konwersacyjnych z wykorzystaniem algorytmów A.L.I.C.E. Sam ProgramD nie jest w stanie realizować innych zadań niż tylko umożliwienie dostępu do zawartości bazy wiedzy systemu konwersacyjnego. Problemem jest tu fakt, iż sam AIML jest językiem bardzo schematycznym i opiera się na zasadzie pytanie-odpowiedź. To znaczy, że każda wypowiedziana sentencja przez program konwersacyjny jest odpowiedzią na z góry określone pytanie, które musi być zawarte w bazie wiedzy. Przedstawiona na powyższym listingu baza wiedzy systemu translacyjnego przewiduje kilka zapytań realizowanych w języku naturalnym i określa sposób ich translacji na język SQL. Zapytanie postaci *Pokaż pracowników z tabeli pracownicy gdzie ich zarobki są równe 2000* jest transformowane do postaci:

```
SELECT pracowników FROM tabeli pracownicy WHERE ich zarobki są równe 2000
```

Jak widać, transformacja następuje niepoprawnie. Celem było uzyskanie zapytania postaci:

```
SELECT pracownik FROM pracownicy WHERE zarobki=2000
```

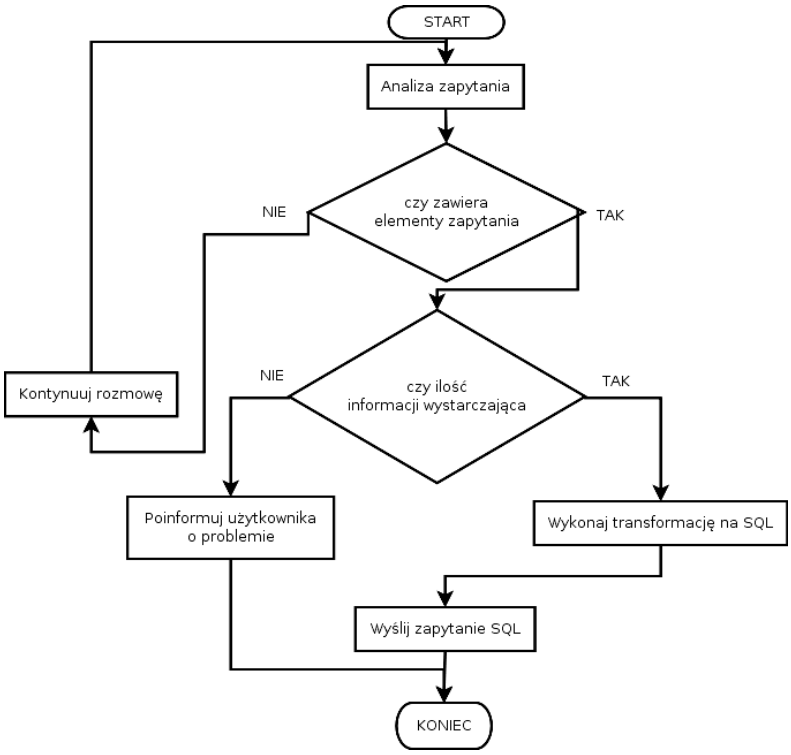
W obecnej postaci użytego systemu konwersacyjnego oraz używanej do celów badawczych bazy wiedzy nie jest możliwe zbudowanie w pełni funkcjonalnego systemu realizującego zapytania w języku naturalnym do baz danych. Jest to spowodowane problemami wynikającymi z elastyczności języka naturalnego. Języki sztuczne, takie jak języki programowania, podlegają z góry określonym regułom i ich budowa zazwyczaj jest podyktowana ogólnie przyjętymi zasadami ich używania.

By zbudować system o funkcjonalności pozwalającej na używanie naturalnego języka w celu wydobywania wiedzy z baz danych, należy określić reguły, które jednoznacznie będą wskazywać na chęć wydobycia wiedzy z baz danych. Podczas konwersacji użytkownik musi użyć słowa kluczowego, na przykład *baza danych*, wówczas zapytanie mogłoby wyglądać tak:

```
Pobierz z bazy danych przedsiebiorstwo informacje o pracownikach zarabiających
2000 złotych
```

Słowo kluczowe powinno wówczas uruchomić system ekstrakcji informacji na temat baz danych, zgodnie z przedstawionym na rys. 2 algorytmem [8]. System musi mieć możliwość ekstrakcji wszelkich niezbędnych danych związanych z informacją na temat baz danych, między innymi informacje o tabelach i kolumnach tych tabel [9].

Elementem systemu, który bezpośrednio odpowiada za ekstrakcję informacji, jest wyszukiwarka macierzowa.



Rys. 2. Algorytm działania modułu analizy zapytań
Fig. 2. Algorithm of queries analysis module

3.2. Idea wyszukiwarki macierzowej

Wyszukiwarka macierzowa rozkłada wprowadzony tekst na osobne elementy, następnie bazując na zawartości pliku konfiguracyjnego próbuje ekstrahować z pozyskanej sentencji wymaganą wiedzę. Dla przykładu przyjęto, że baza danych (tabela 1) zawiera jedną tabelę pracownicy, w której zawarte są dane o imieniu i nazwisku pracownika oraz jego stanowisko i wysokość zarobków.

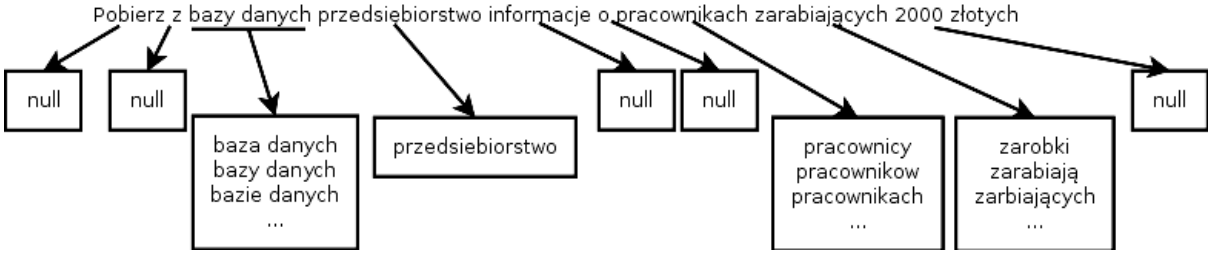
Użytkownik zadaje pytanie o pracowników zarabiających 2000 złotych. Wyszukiwarka macierzowa rozkłada zapytanie na kolejne elementy i przeszukuje, wcześniej stworzony przez administratora systemu, plik konfiguracyjny celem przekształcenia zapytania na równoważne zapytanie SQL.

Tabela 1

Tabela pracownicy w przykładowej bazie danych przedsiębiorstwo

Imie	Nazwisko	Stanowisko	Zarobki
Jan	Kowalski	Asystent	1000
Piotr	Kowalski	Stróż	1500
Ola	Kowalska	Asystentka	1300
Andrzej	Kowalski	Handlowiec	2000

Można stwierdzić, iż celem wyszukiwarki macierzowej jest naprowadzenie systemu na odpowiednie tabele i pola w bazie danych.



Rys. 3. Schemat działania wyszukiwarki macierzowej
Fig. 3. Schema of a matrix search engine

Na rys. 3 przedstawiono działanie wyszukiwarki macierzowej na podstawie zapytania postaci *pobierz z bazy danych przedsiębiorstwo informacje o pracownikach zarabiających 2000 złotych*. Wyszukiwarka macierzowa odnajduje w pliku konfiguracyjnym wszelkie odpowiedniki odmienionych wyrazów. Jeżeli takowe istnieją, jest w stanie przygotować na tej podstawie odpowiedni zestaw danych wymaganych przy konstrukcji poprawnego zapytania SQL. Należy przy tym zauważyć, że aby wyszukiwarka mogła wykonać to zadanie, musi mieć możliwość ekstrahowania wszelkich informacji z podanej sentencji. Oznacza to, że informacje te muszą być podane w sposób jawny w zdaniu.

W tabeli 2 umieszczono przykładową zawartość pliku konfiguracyjnego. Plik ten ma za zadanie naprowadzić system konwersacyjny na parametry do baz danych. W języku naturalnym to samo zdanie można wypowiedzieć na kilka sposobów. Mimo iż w zapisie i wymowie zdania te brzmią inaczej, w domyśle znaczą to samo. Zawartość pliku konfiguracyjnego przedstawiona w tabeli 2 jest próbą rozwiązania problemu odmiany wyrazów zgodnie z zasadami gramatyki języka polskiego. Jest to wymagane celem stworzenia poprawnego zapytania SQL dla danej bazy danych. Biorąc pod uwagę nazwę kolumny *zarobki*, należy zauważyć, iż w języku polskim można powiedzieć *pokaż informacje o zarobkach pracowników* lub *Ile zarabiają pracownicy*. Kontekst obydwu zdań jest identyczny, choć zapis zdania jest inny. Oba zapytania w języku sztucznym SQL będą miały postać:

```
SELECT zarobki FROM pracownicy
```

Nazwa kolumny *zarobki* nie może brzmieć inaczej, zadaniem wyszukiwarki macierzowej jest takie przetworzenie informacji, aby na jej podstawie możliwe było zbudowanie poprawnego składniowo zapytania SQL.

Tabela 2

Zawartość pliku konfiguracyjnego wyszukiwarki macierzowej – fragment

Tabela	Odmiana	Kolumna	Odmiana
Pracownicy	Pracownikach Pracowników Pracownikami Pracownice Pracownik …	Imie	Imiona Imionach Imieniu …
		Nazwisko	Nazwisku Nazwiskach Nazwiskiem Nazwisk …
		Stanowisko	Stanowisku Stanowisko Stanowisk Stanowiskach …
		Zarobki	Zarobków Zarobkami Zarobkiem Zarabiających …

4. Podsumowanie

W oparciu o wcześniejsze przemyślenia [8] przedstawiono rozwiązanie problemu ekstrakcji informacji o modelu bazy danych z sformułowanego zapytania w języku naturalnym. Informacje te są wymagane do dalszej transformacji języka naturalnego na równoważny język zapytań SQL. Problem, który wydaje się rozwiązany, w rzeczywistości wymaga jeszcze wielu badań. Między innymi przedstawiona wyszukiwarka jest na tyle prymitywna, iż nadal nie potrafi poprawnie ekstrahować informacji o warunkach zapytania. Przez to rezultaty zwracane przez system mogą być niepoprawne. Wymagane jest także udoskonalenie języka bazy wiedzy AIML odpowiedzialnego za przechowywanie odwzorowań języka naturalnego w język zapytań SQL. Przedstawiony w artykule prototyp wyszukiwarki macierzowej jest sugestią podstawy do dalszych rozważań i badań nad możliwością skonstruowania systemu baz danych, który będzie w stanie przetworzyć zapytanie w języku naturalnym i zwrócić na jego podstawie poprawne wyniki.

BIBLIOGRAFIA

1. Jagielski J., Wnęk P.: Ewolucja realizacji zapytań w systemach akwizycji wiedzy. Zeszyty naukowe Akademii Marynarki Wojennej, Rok XLX, No. 177B, Gdynia 2009, s. 109÷117.

2. Vetulani Z.: Komunikacja człowieka z maszyną. Komputerowe modelowanie kompetencji językowej. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2004.
3. Barczak A., Florek J., Sydoruk T.: Bazy danych. Wydawnictwo Akademii Podlaskiej, Siedlce 2007.
4. Zadrożny S.: Zapytania nieprecyzyjne i lingwistyczne podsumowania baz danych. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2006.
5. Myszkorowski K., Zadrożny S., Szczepaniak S. P.: Klasyczne i rozmyte bazy danych. Modele, zapytania i podsumowania. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008.
6. Myszkorowski K.: Związki wieloargumentowe w rozmytych bazach danych. Bazy Danych: Struktury, Algorytmy, Metody: Architektura, metody formalne i eksploracja danych, Vol. 1, Wydawnictwo Komunikacji i Łączności, Warszawa 2006, s. 117÷126.
7. Mazur Ł.: Systemy konwersacyjne. Software Developer's Journal, 07/2008.
8. Jagielski J., Wnęk P.: Odwzorowanie zapytań w języku naturalnym w język zapytań do baz danych. Conference Archives PTETiS, Vol. 26, Gliwice 2009, s. 241÷246.
9. Meng F., Chu W. W.: Database Query Formation Natural Language using Semantic Modeling and Statistical Keyword Meaning Disambiguation. Technical Report 990003, UCLA CS Dept., 16, 1999.
10. Home Page of the Loebner prize, http://www.loebner.net/Prizef/loebner-prize.html, 06.12.2009.
11. AIML – The Artifical Inteligence Markup Language, http://www.alicebot.org/aiml.html, 06.12.2009.
12. Eckel B.: Thinking in JAVA. Edycja polska, Wydawnictwo HELION, Gliwice 2006.
13. RunRev, http://www.runrev.com/, 07.12.2009.

Recenzenci: Dr inż. Małgorzata Bach
Dr hab. inż. Krzysztof Goczyła, prof. Pol. Gdańskiej

Wpłynęło do Redakcji 7 stycznia 2010 r.

Abstract

Databases systems still evolve. In recent years old inefficient databases systems (historical models) have been replaced by relational model, which simplified the process of implementing queries in database systems. Continuous development of technology leads to a simplification of communication between man and machine. This process can be observed

also in the informatics systems, when we analyze the development of programming languages in recent years. The first programming languages consisted of a simple zero-one coding of commands but development of technology has led to the creation of low-level programming languages, which became insufficient in the course of time and this was the direct cause of the creation of high-level languages [2, 13]. Object-oriented programming languages were designed to match the conceptual models and the realization models to natural human behavior [1, 3, 12]. Now we seek also to build a programming language based on natural language [13]. This article describes a problem of realization natural language queries in database systems, the goal is to present basis problems which developers of this kind of systems encounter and we want to show a way of resolving these problems. The studies on this kind of systems have been conducted since many years all over the world [9]. One of the biggest problems the scientists came across is a fact that artificial language is based on predetermined rules. Natural language is also based on certain kind of rules, but they are not as strict as it is in programming language.

This article is treated by authors as a development approach set out in item [8] and they are trying to present here the solution of basic problems which they encountered while working on the system. A presented algorithm of performing database queries in natural language (Fig. 2) requires full information about names of tables and columns for proper working. Natural language is a subject to the rules of declination, by which information about the listed data model explicitly may not be correctly extracted from natural language queries. Therefore, as a mechanism for solving this problem a matrix search engine (Fig. 3), was created. Its task is to extract the information from given sentence about data model in database. The whole idea is based on a matrix search engine configuration file, which includes information about the name of columns, tables, etc.

The issue presented in this article should be considered as a basis for further research on database systems implementing the queries in natural language. The most of problems are not still resolved. For example, the problem of extracting information about conditions of the database queries as well as the problem of correctly created SQL queries: even though the system solves some problems associated with information about the data model of database, a natural language query is still not equivalent to a query in database queries.

Adresy

Piotr WNĘK: Instytut Metrologii Elektrycznej, ul. Podgórna 50, 65-246 Zielona Góra, Polska, P.Wnek@weit.uz.zgora.pl .

Jan JAGIELSKI: Instytut Metrologii Elektrycznej, ul. Podgórna 50, 65-246 Zielona Góra, Polska, J.Jagielski@ime.uz.zgora.pl .