

*Hubert Szczepaniuk*

Szkoła Główna Gospodarstwa Wiejskiego w Warszawie

## **Zastosowania grafowych baz danych w systemach informatycznych zarządzania**

### **Applications of graph databases in Management Information Systems**

**Synopsis.** Efektywność procesów informacyjnych w systemach informatycznych zarządzania jest bezpośrednio determinowana przez dobór odpowiedniej architektury bazy danych. W artykule autor podejmuje próbę rozpoznania zastosowań grafowych baz danych na płaszczyźnie architektury technicznej systemów informatycznych zarządzania. Analizie poddane zostaną teoretyczne podstawy grafów w informatyce, uwarunkowania modelowania baz danych na podstawie struktur grafowych oraz płaszczyzny zastosowań grafowych baz danych.

**Słowa kluczowe:** grafowe bazy danych, bazy danych klasy NoSQL, systemy informatyczne zarządzania

**Abstract.** The efficiency of information processes in Information Management Systems is directly determined by the selection of the appropriate database architecture. In the article, the author attempts to identify the applications of graph databases on the level of technical architecture of Management Information Systems. The theoretical applications of graphs in computer science, conditions of modeling databases based on graph structures and planes of application of graph databases will be subject to analysis.

**Key words:** graph databases, NoSQL databases, Management Information Systems

## **Wstęp**

Realizacja funkcji zarządzania w organizacjach gospodarczych wymaga efektywnych procesów informacyjnych. Systemy informatyczne zarządzania (SIZ) są wykorzystywane do przetwarzania zasobów informacyjnych organizacji, przez co bezpośrednio wspierają realizację kluczowych funkcji zarządzania. Na płaszczyźnie architektury technicznej SIZ oparte są na systemach zarządzania bazami danych (SZBD).

W zakresie SZBD informatyka oferuje różnorodne technologie do których zaliczyć można przede wszystkim rozwiązania oparte na hierarchicznym, sieciowym, relacyjnym lub obiektowym modelu danych. Obecnie najbardziej popularny jest relacyjny model danych. Według rankingu popularności silników baz danych opracowanego przez DB-Engines na grudzień 2017 roku, cztery pierwsze miejsca należą do rozwiązań opartych na relacyjnym modelu danych [DB-Engines].

Od pewnego czasu można zauważyć stopniowy rozwój baz danych opierających się na paradygmatach NoSQL. Do klasy takich rozwiązań można zaliczyć przede wszystkim: grafowe bazy danych, bazy danych typu klucz-wartość oraz bazy danych zorientowane na dokument. Szczególnie istotne właściwości w kontekście efektywności SZBD oraz modelowania warstwy semantyki danych wykazują koncepcje opierające się na strukturach grafowych. Popularność grafowych baz danych nie jest obecnie imponująca. Według rankingu popularności opracowanego przez DB-Engines pierwsza baza danych oparta na grafowym modelu danych znajduje się poza pierwszą piętnastką [DB-Engines]. Przyczyn takiego stanu rzeczy należy szukać w długiej historii modelu relacyjnego sięgającej lat 70. XX wieku oraz mocno ugruntowanej na rynku pozycji SZBD takich jak: Oracle, MySQL, Microsoft SQL Server i PostgreSQL.

Rozwój koncepcji zarządzania procesowego, odchodzenie od integracji procesów gospodarczych w środowisku on-line w kierunku organizacji wirtualnych, dynamiczny rozwój zjawiska Big Data oraz wzrost popularności koncepcji WEB 3.0 opartej na sieciach semantycznych stawia przed systemami informatycznymi wspierającymi zarządzanie nowe wymagania zwłaszcza w zakresie wykorzystywanego przez nie modelu bazy danych. Istnieje pilna potrzeba badań naukowych w zakresie implementacji baz danych klasy NoSQL w systemach informatycznych zarządzania.

## **Cel i metodyka badań**

Przedmiotem badań w niniejszym artykule są grafowe bazy danych. Celem badań jest analiza zastosowań struktur grafowych na płaszczyźnie architektury baz danych w systemach informatycznych zarządzania, które obsługują procesy informacyjne w organizacjach gospodarczych. Przyjęta w niniejszym artykule metodyka badań obejmuje interdyscyplinarne podejście badawcze uwzględniające analizę i syntezę literatury przedmiotu, modelowanie matematyczne, teorię grafów i programowanie.

## **Teoretyczne podstawy struktur grafowych w informatyce**

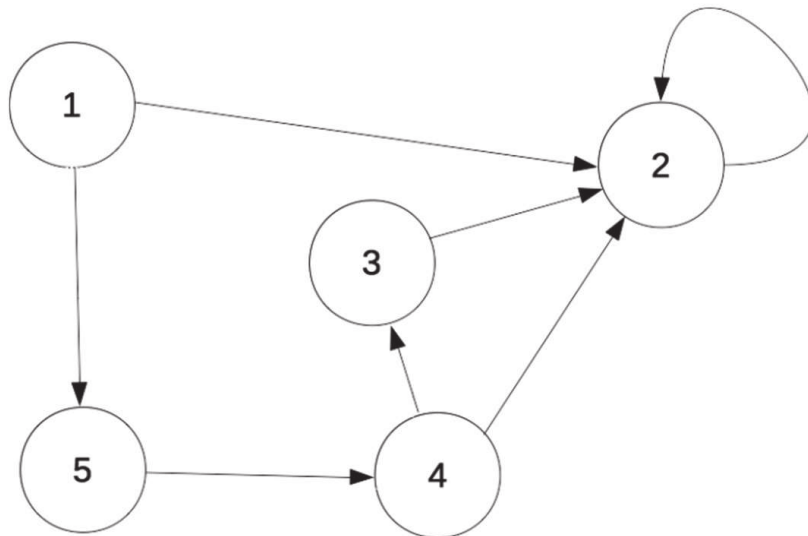
W matematyce graf można zdefiniować formalnie jako parę  $G = (V, E)$ , gdzie  $V$  to niepusty i skończony zbiór reprezentujący wierzchołki, natomiast  $E$  to skończony zbiór z powtórzeniami jedno- i dwuelementowych podzbiorów  $V$ , który reprezentuje krawędzie grafu [Łachwa 2013].

Przytoczona definicja grafu zawiera ograniczenia na skończoność zbioru  $V$  i  $E$ . W matematyce można spotkać koncepcje zakładające grafy o nieskończonej liczbie wierzchołków i krawędzi. Co do zasady w informatyce zasoby systemu komputerowego wyrażone w postaci czasu procesora, objętości pamięci wolnego dostępu i pamięci ma-

sowej są skończone. Dlatego na potrzeby analiz w niniejszym artykule autor przyjmuje założenie o skończoności zbiorów  $V$  i  $E$ .

Szczególnym przypadkiem w strukturach grafowych są grafy skierowane, które wykazują zastosowanie w informatyce. Graf skierowany zwany również digrafem można zdefiniować formalnie jako parę  $D = (V, E)$ , gdzie  $V$  jest zbiorem reprezentującym wierzchołki, natomiast  $E$  jest zbiorem z powtórzeniami reprezentującym krawędzie skierowane w taki sposób, że  $E \subseteq V \times V$  [Łachwa 2013].

Na rysunku 1 przedstawiono graf skierowany o pięciu wierzchołkach i określonych krawędziach.



Rysunek 1. Digraf o pięciu wierzchołkach i określonych krawędziach

Figure 1. Digraf with five vertices and defined edges

Źródło: opracowanie własne.

Graf skierowany przedstawiony na rysunku 1 opisany jest jako uporządkowana para zbiorów  $D = (V, E)$  w której zbiór  $V = V \{1, 2, \dots, 5\}$ , zbiór  $E$  natomiast jest podzbiorem iloczynu kartezjańskiego  $V \times V$  postaci  $E = \{(1, 2), (1, 5), (2, 2), (3, 2), (4, 2), (4, 3), (5, 4)\}$ .

Struktury grafowe mogą być implementowane w programach komputerowych na rozmaite sposoby. Do najważniejszych algorytmów wykorzystywanych w informatyce przy implementacji grafów skierowanych w wybranych językach programowania zaliczyć można m.in. [Wałaszek]:

- macierz sąsiedztwa,
- listę sąsiedztwa,
- macierz incydencji.

Algorytm oparty na matematycznej koncepcji macierzy sąsiedztwa implementuje dwuwymiarową tablicę kwadratową o rozmiarze  $v$  równym liczbie wierzchołków w grafie [Wałaszek]. Algorytm w sposób ogólny zakłada, że indeksy wierszy w macierzy sąsiedztwa odpowiadają numerom wierzchołków stanowiących początek krawędzi, indeksy kolumn natomiast odpowiadają numerom wierzchołków stanowiących koniec

poszczególnych krawędzi grafu<sup>1</sup>. Dla grafu skierowanego przedstawionego na rysunku 1 macierz sąsiedztwa  $A$  może mieć postać:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

Algorytm implementujący listę sąsiedztwa przechowuje dla każdego wierzchołka zbiór węzłów sąsiadujących z nim przez krawędzie skierowane<sup>2</sup>. W celu jego implementacji można wykorzystać rozmaite konstrukcje programistyczne, np. typ danych typu lista. Dla grafu skierowanego przedstawionego na rysunku 1 lista sąsiedztwa może być określona następująco:  $1 \rightarrow 2,5$ ;  $2 \rightarrow 2$ ;  $3 \rightarrow 2$ ;  $4 \rightarrow 2,3$ ;  $5 \rightarrow 4$ .

Algorytm implementujący matematyczną koncepcję macierzy incydencji buduje tablicę dwuwymiarową o rozmiarach wyznaczonych przez liczbę wierzchołków i krawędzi grafu [Wałaszek]. Algorytm zakłada, iż w przypadku, gdy dana krawędź rozpoczyna się od określonego wierzchołka, to wartość komórki tablicy przyjmuje  $-1$ , w przypadku, gdy krawędź kończy się w określonym wierzchołku komórka przyjmuje wartość  $1$ <sup>3</sup>. Pętla na danym wierzchołku jest zazwyczaj reprezentowana przez wartość  $2$ <sup>4</sup>. W pozostałych przypadkach, gdzie nie zachodzi incydencja, komórka tablicy przyjmuje wartość  $0$ <sup>5</sup>. Dla grafu skierowanego przedstawionego na rysunku 1 macierz incydencji  $I$  może być określona następująco:

$$I = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (2)$$

Przy doborze odpowiedniego algorytmu implementującego reprezentację grafu należy wziąć pod uwagę efektywność danego rozwiązania w zakresie wykorzystania zasobów systemu komputerowego. Algorytmy oparte na macierzy sąsiedztwa posiadają złożoność pamięciową na poziomie  $\Omega(V^2)$ , co w przypadku grafów rzadkich może

<sup>1</sup> Szerzej o implementacji w języku programowania algorytmu macierzy sąsiedztwa, listy sąsiedztwa i macierzy incydencji pisze J. Wałaszek, [źródło elektroniczne] [https://eduinf.waw.pl/inf/alg/001\\_search/0124.php](https://eduinf.waw.pl/inf/alg/001_search/0124.php) [dostęp: 08.01.2018].

<sup>2</sup> Tamże.

<sup>3</sup> Tamże.

<sup>4</sup> Tamże.

<sup>5</sup> Tamże.

oznaczać dużą liczbę niewykorzystanych zasobów systemowych rezerwowanych przez algorytm. Algorytmy oparte na macierzy incydencji posiadają złożoność pamięciową na poziomie  $\Omega(V \times E)$ , Z kolei algorytmy oparte na listach sąsiedztwa posiadają złożoność pamięciową na poziomie  $\Omega(E)$  i mogą opierać się na typie danych typu lista, który jest predefiniowany w nowoczesnych językach programowania. Należy wskazać, że oprócz złożoności pamięciowej występuje również złożoność obliczeniowa wykonania podstawowych operacji na grafie tj.: dodanie/usunięcie wierzchołka, dodanie/usunięcie krawędzi, przeszukiwanie grafu. Złożoność pamięciowa dotyczy wykorzystania pamięci operacyjnej systemu komputerowego, a złożoność obliczeniowa dotyczy obciążenia procesora. W zależności czy algorytm opiera się na koncepcji macierzy sąsiedztwa, listy sąsiedztwa czy macierzy incydencji to złożoność pamięciowa i obliczeniowe wymienionych operacji na strukturze grafu może być znacząco różna. W zależności od konkretnego problemu algorytmicznego należy dobrać odpowiedni sposób implementacji grafów.

Współczesne języki programowania dostarczają gotowe interfejsy ze zdefiniowanych klas umożliwiające efektywną implementację struktur grafowych w kodzie programu. Programistyczna implementacja grafu w postaci macierzy sąsiedztwa jest stosunkowo prosta. Zorientowana obiektowo implementacja w języku programowania C++ reprezentacji grafu za pomocą macierzy sąsiedztwa przedstawiona jest na listingu 1.

```
class CGraph
{
    unsigned V,E;
    bool DiGraph;
    bool MultiGraph;

    unsigned **GMatrix;

public:
    CGraph(unsigned _V, bool _di, bool _mu);
    ~CGraph();
    bool Insert(unsigned v,unsigned u);
    bool Delete(unsigned v,unsigned u);
    unsigned Degree(unsigned v);
    bool Search(unsigned v,unsigned u);
};
```

Listing 1. Obiektowa implementacja grafu w postaci klasy CGraph w języku C++

Listing 1. Object-oriented implementation of a graph in the form of CGraph class in C ++ language

Źródło: [Wikipedia... 2016].

Implementacja grafu przedstawiona na listingu 1 definiuje klasę *CGraph* składającą się ze zmiennych *V* i *E* w ciele klasy, które reprezentują liczbę wierzchołków i krawędzi w grafie. Zmienna *DiGraph* oraz *MultiGraph* są zmiennymi typu logicznego i określają odpowiedni rodzaj grafu. Dynamicznie alokowana w pamięci tablica *GMatrix* posłuży do przechowywania macierzy sąsiedztwa. Następnie zaimplementowany jest konstruktor obiektu klasy *CGraph* i zapowiedź podstawowych metod manipulacji i dostępu na danych tj.: operacji wstaw, usuń i wyszukaj.

## Relacyjne a grafowe modele danych

Współczesne rozwiązania bazodanowe zdominowane są przez relacyjne bazy danych, których koncepcje nakreślił E.F. Codd, opierając się na założeniach rozszerzonego iloczynu kartezjańskiego dla  $n$  zbiorów oraz  $n$ -członowej relacji na nim określonej [Codd 1970]. Baza danych opierająca się na modelu formalnym Codda obejmuje zespół relacji powiązanych związkami 1–1, 1– $n$ ,  $m$ – $n$ . Relacyjne bazy danych oparte na modelu formalnym Codda zapewniają spójność, integralność na poziomie encji i odwołań oraz redukują redundancje danych. Współczesne rozwiązania informatyczne z zakresu systemów zarządzania relacyjnymi bazami danych (SZRBD) opierają się na systemach informatycznych działających w architekturze klient–serwer. Do najbardziej popularnych systemów komercyjnych zaliczyć można m.in. Microsoft SQL Server i Oracle Database, a do dystrybuowanych na licencjach otwartych MariaDB.

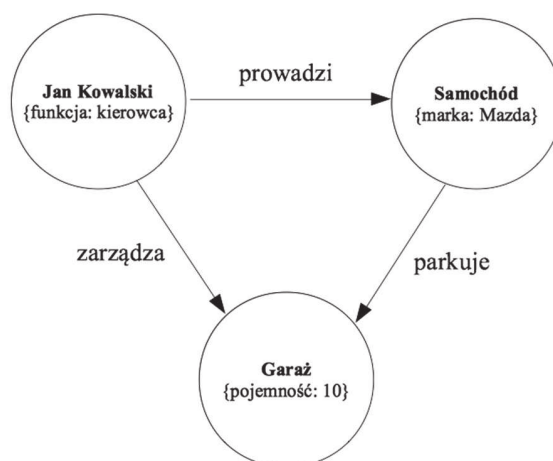
Systemy informatyczne zarządzania, które na poziomie architektury bazodanowej opierają się na relacyjnych bazach danych, wymagają opracowania schematu danych na etapie projektowania systemu i implementacji. Po realizacji procesu projektowania i implementacji schemat danych jest statyczny i najczęściej nie ulega zmianie. Modelowanie relacyjnej bazy danych na etapie projektowania systemu informatycznego wspierającego zarządzanie obejmuje co najmniej zespół następujących czynności [Grunkowski 2013]:

- opracowanie diagramu relacji;
- opracowanie diagramu związków encji;
- opracowanie struktury tabel, określenie związków, kluczy głównych i obcych;
- przeprowadzenie procesu normalizacji bazy danych przez kolejne postacie normalne 1NF, 2NF, 3NF;
- strojenie i optymalizacja bazy pod kątem efektywności zapytań dostępu i manipulacji na danych.

Etap modelowania bazy danych jest jednym z najbardziej istotnych elementów projektu SIZ. Wdrożenie zmian w koncepcji relacyjnej bazy danych już po zaimplementowaniu kluczowych modułów systemu wymaga znacznych zasobów czasowych i kosztowych. Statyczna struktura schematu relacyjnej bazy danych nie sprzyja wykorzystywaniu jej w organizacjach, których potrzeby informacyjne obejmują przetwarzanie dynamicznych, strukturalnie zróżnicowanych i dużych zbiorów danych.

Struktury grafowe mogą być efektywnie wykorzystywane w bazach danych, ponieważ wykazują zdolności do przechowywania heterogenicznych zbiorów danych o dynamicznej strukturze. Struktura grafowa wykorzystana do przechowania zbiorów danych zakłada, że wierzchołkom grafu skierowanego oraz krawędziom przypisywane są zdefiniowane atrybuty. Encje danych reprezentowane są przez atrybuty przypisywane wierzchołkom grafu, atrybuty przypisywane krawędziom natomiast reprezentują relacje w bazie danych. Na rysunku 2 przedstawiono przykładowy model danych reprezentowany w strukturze grafowej z określonym zbiorem atrybutów dla wierzchołków i krawędzi.

Warto w tym miejscu wskazać, że reprezentacja danych na strukturze grafowej wykazuje zdolność do realizacji na niej algorytmów odnoszących się do semantyki danych, dzięki relacjom określonym przez zbiór atrybutów krawędzi.



Rysunek 2. Model danych opisany na strukturze grafowej

Figure 2. The data model described on the graph structure

Źródło: opracowanie własne.

Rozpoczęcie projektowania bazy danych dla SZB od opracowania diagramu encji pozwala na zredukowanie procesu modelowania schematu grafowej bazy danych do następujących czynności [Gruntkowski 2013]:

- opracowanie diagramu relacji,
- implementacja wyznaczonych związków w grafowej bazie danych.

Oprócz funkcji umożliwiającej modelowanie schematu danych SZB realizują również funkcje pozwalające na dostęp do danych. Dostęp do danych realizuje się poprzez język zapytań do bazy danych. Zapytania w grafowej bazie danych można podzielić na dwa rodzaje [Słotwiński 2010]:

- zapytania strukturalne – zapytania o strukturę danych opisanych na grafie (np. przeszukiwanie grafu w głąb do określonego poziomu i wyznaczenie liczby dostępnych węzłów;
- zapytania dostępu do danych – zapytania o dane opisane na grafie według zdefiniowanych kryteriów (np. wypisanie wszystkich atrybutów węzłów grafu, które spełniają określone kryteria).

Silnik bazy danych w celu realizacji zapytania użytkownika wykonuje matematyczne algorytmy przeszukiwania grafu. Efektywność takich algorytmów w porównaniu do zapytań realizowanych na relacyjnej bazie danych często może być wyższa. Dla przykładu efektywność mierzona czasem zwrócenia wyników zapytania w języku SQL do relacyjnej bazy danych, które wymaga wielu złączeń typu JOIN, może być o wiele rzędów wartości niższa niż efektywność zapytania o te same dane do grafowej bazy danych. Wynika to przede wszystkim z matematycznych właściwości algorytmów wyszukiwania realizowanych na strukturach grafowych. Warto w tym miejscu dodać, że w przypadku relacyjnych baz danych istnieje ogólnie przyjęty standard w postaci strukturalnego języka zapytań SQL i jego rozszerzeń tj. PL/SQL czy Transact-SQL. W przypadku grafowych baz danych w zależności od implementacji SZB wykorzystuje się różnorodne języki zapytań, które często mają odmienną syntaktykę, dla przykładu Neo4J wykorzystuje język Cypher.

## Zastosowania grafowych struktur danych

Wymagania stawiane na etapie projektowania, implementacji i użytkowania systemów informatycznych zarządzania wyznaczyły płaszczyzny kluczowych obszarów, w których możliwe jest szukanie efektywnych zastosowań grafowych struktur danych. Jedno z najpełniejszych zestawień dotyczących obszarów zastosowań grafowych baz danych opracowali C. Angles i R. Gutierrez. Warto przytoczyć wybrane obszary [Angles i Gutierrez 2008, Słotwiński 2010]:

- uogólnienie klasycznego modelu danych w celu ułatwienia reprezentacji złożonych obiektów i analizy związków pomiędzy danymi;
- systemy logistyczne wspierające zarządzanie sieciami transportowymi i systemy komunikacji drogowej;
- systemy reprezentacji i analizy wiedzy;
- modele grafowe reprezentacji danych oraz manipulacji danymi;
- wspomaganie funkcjonalności oferowanej przez obiektowe modele danych;
- systemy multimedialne;
- optymalizacja modeli danych dla stron WWW.

Warto w tym miejscu dokonać analizy zastosowania struktur grafowych na płaszczyźnie koncepcji sieci semantycznej. Twórca standardu WWW, T. Berners-Lee zauważając ograniczenia płaskich struktur danych w dokumentach hipertekstowych, zaproponował koncepcje Internetu opartego na tzw. sieci semantycznej [Berners-Lee i in. 2001]. Wspominania koncepcja zakłada wdrożenie standardów opisu treści internetowych pozwalających na jej przetwarzanie na poziomie semantyki. Założenia sieci semantycznej nie integrują w dotychczasowe protokoły komunikacji warstwy aplikacji modelu ISO OSI takie jak HTTP i HTTPS czy też protokoły warstwy transportowej (TCP, UDP) i sieciowej (IPv4, IPv6).

Przetwarzanie zasobów internetowych na poziomie semantyki przez specjalistyczne oprogramowanie agentowe może przyczynić się do ułatwienia automatyzacji usług internetowych sterowanych zbiorem zadań zleczanych przez człowieka. W celu realizacji koncepcji sieci semantycznej konieczne jest wykorzystanie nowych standardów opisu treści internetowych takich, które pozwalają na modelowanie nie tylko danych, ale również określenie i analizę różnorodnych relacji na zbiorze danych [Berners-Lee i in. 2001].

Naturalnym rozwiązaniem w sieciach semantycznych może być wykorzystanie struktur grafowych, które poprzez zdefiniowane zbiory atrybutów na krawędziach grafu umożliwiają łatwe określenie skomplikowanych relacji na zbiorach danych. Umożliwia to z kolei implementację oprogramowania agentowego zdolnego do realizacji algorytmów wnioskowania na warstwie danych i warstwie semantyki określonej przez zbiór relacji. Na podstawie założeń struktur grafowych oraz składni języka XML powstał język RDF umożliwiający opisywanie zasobów internetowych w postaci ontologii i budowanie w ten sposób warstwy semantyki danych.

## Podsumowanie i wnioski

Architektura SZBD determinuje efektywność procesów informacyjnych realizowanych przez SIZ, co bezpośrednio przekłada się na sprawność realizacji funkcji zarządzania w organizacji. Najpopularniejszy obecnie relacyjny model danych nie zawsze stanowi



optymalne rozwiązanie dla organizacji, które przetwarzają duże, dynamiczne zbiory danych o zmiennej strukturze. Statyczność schematu relacyjnej bazy danych może utrudniać dokonanie zmian wynikających np. z przeorganizowania procesów biznesowych w organizacji. Z kolei dynamiczna struktura schematu grafowej bazy danych umożliwia opisanie złożonych zbiorów danych oraz określenie rozmaitych relacji na danych. Zbiór relacji określony przez atrybuty krawędzi grafu reprezentuje dodatkową, semantyczną warstwę bazy danych, dzięki której możliwa jest implementacja algorytmów realizujących przetwarzanie maszynowe na poziomie wiedzy.

Autor postuluje, iż w przypadku organizacji, których procesy informacyjne funkcjonują w warunkach heterogenicznych zbiorów danych o dynamicznej i zmiennej strukturze, warto poddać analizie na etapie projektowania SIZ efektywność zastosowania grafowych baz danych, uwzględniając co najmniej [szerzej na ten temat Szczepaniuk i Szczepaniuk]:

- adekwatność modelu danych w zakresie dopasowania do realizowanych w organizacji procesów informacyjnych;
- elastyczność i skalowalność schematu bazy danych w zakresie przyszłych potrzeb reorganizacji procesów biznesowych organizacji;
- efektywność w zakresie wydajności silnika bazy danych wartościowanej przez czas realizacji zapytań manipulacji i dostępu do danych.

Na szczególną uwagę zasługuje również zastosowanie grafowych baz danych w systemach informatycznych wspomagających zarządzanie procesami logistycznymi w transporcie. Wyznaczenie trasy na podstawie lokalizacji zapisanych w relacyjnej bazie danych wymaga często implementacji algorytmu o wyższej złożoności obliczeniowej niż w przypadku algorytmu operującego bezpośrednio na danych zorganizowanych w struktury grafowe.

Z ekonomicznego punktu widzenia kluczowy jest również czynnik finansowy inwestycji w SIZ. Systemy zarządzania bazami danych oparte na strukturach grafowych dystrybuowane są często na licencjach otwartych, co oznacza bezpłatny dostęp do oferowanej przez nie technologii.

## Literatura

- Angles R., Gutierrez C., 2008: Survey of Graph Database Models, [źródło elektroniczne] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.190.7888&rep=rep1&type=pdf> [dostęp: 20.12.2017], 7–9.
- Berners-Lee T., Hendler J., Lassila O., 2001: The Semantic Web, [źródło elektroniczne] [https://www.sop.inria.fr/acacia/cours/essi2006/Scientific%20American\\_%20Feature%20Article\\_%20The%20Semantic%20Web\\_%20May%202001.pdf](https://www.sop.inria.fr/acacia/cours/essi2006/Scientific%20American_%20Feature%20Article_%20The%20Semantic%20Web_%20May%202001.pdf) [dostęp: 20.12.2017], 1–4.
- Codd E.F., 1970: A relational model of data for large shared data banks, [źródło elektroniczne] <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> [dostęp: 20.12.2017], 337–388.
- DB-Engines, [źródło elektroniczne] <https://db-engines.com/en/ranking> [dostęp: 20.12.2017].
- Gruntkowski A., 2013: Grafowe bazy danych, [źródło elektroniczne] <http://trug.pl/archive/20130918/Grafowe-bazy-danych.pdf> [dostęp: 20.12.2017].
- Łachwa A., 2013: Matematyka dyskretna, [źródło elektroniczne] [http://zpgk.fais.uj.edu.pl/documents/2349539/7815838/MD14%20\(grafy1\).pdf](http://zpgk.fais.uj.edu.pl/documents/2349539/7815838/MD14%20(grafy1).pdf) [dostęp: 20.12.2017].

- Słotwiński D., 2010: Grafowe bazy danych – przegląd technologii, [źródło elektroniczne] [https://ai.ia.agh.edu.pl/wiki/\\_media/pl:dydaktyka:ztb:2010:projekty:gdb:grafowe\\_bazy\\_danych.pdf](https://ai.ia.agh.edu.pl/wiki/_media/pl:dydaktyka:ztb:2010:projekty:gdb:grafowe_bazy_danych.pdf) [dostęp: 20.12.2017], 5–8.
- Szczepaniuk E., Szczepaniuk H.: Effectiveness of selected database models in Management Information Systems (w druku).
- Właszek J., Reprezentacja grafów w komputerze, [źródło elektroniczne] [http://eduinf.waw.pl/inf/alg/001\\_search/0124.php](http://eduinf.waw.pl/inf/alg/001_search/0124.php) [dostęp: 20.12.2017].
- Wikipedia. Wolna encyklopedia, 2016: Reprezentacja grafu, [źródło elektroniczne] [https://pl.wikipedia.org/wiki/Reprezentacja\\_grafu](https://pl.wikipedia.org/wiki/Reprezentacja_grafu) [dostęp: 20.12.2017].

Adres do korespondencji:

**Dr Hubert Szczepaniuk**

Szkoła Główna Gospodarstwa Wiejskiego w Warszawie

Wydział Nauk Ekonomicznych

Katedra Logistyki

Zakład Badań Operacyjnych

ul. Nowoursynowska 166

02-787 Warszawa

e-mail: [hubert\\_szczepaniuk@sggw.pl](mailto:hubert_szczepaniuk@sggw.pl)