

Sebastian ŁACHECIŃSKI
Uniwersytet Łódzki, Katedra Informatyki Ekonomicznej

NOTACJE MODELOWANIA W PROJEKTOWANIU RELACYJNYCH BAZ DANYCH

Streszczenie. W artykule zaprezentowano rozwój poszczególnych notacji modelowania danych w odniesieniu do modelu relacyjnego oraz została przedstawiona analiza porównawcza wybranych notacji modelowania. Wykazano również, które ze standardów modelowania najczęściej są implementowane w najbardziej popularnych narzędziach CASE do modelowania danych. Artykuł ten w pewien sposób systematyzuje wiedzę na temat istniejących notacji wykorzystywanych w modelowaniu danych za pomocą diagramów ERD i UML.

Słowa kluczowe: notacja modelowania, diagram ERD, modelowanie danych

MODELING NOTATIONS IN THE DESIGN OF RELATIONAL DATABASES

Summary. The article presents the development of the individual data modeling notation with reference to the relational model. Some comparative analysis of selected modeling notation has been shown. It turns out that the most common modeling standards are implemented in the most popular CASE tools for data modeling. The author tries to systematise existing knowledge about the notation which is used in modeling of the data by means of the ERD and UML diagrams.

Keywords: modeling notation, ERD diagram, modeling data

1. Wprowadzenie

Na przestrzeni ostatnich ponad 40 lat powstało wiele różnych standardów modelowania danych. Część z nich przetrwała do czasów obecnych i pozostaje w ciągłym użyciu, niektóre z nich zaś wyszły z powszechnego użycia. Jedne stały się bazą do budowy zupełnie nowych

standardów, ewoluując do nowych postaci, w celu uwzględnienia dodatkowych funkcjonalności oraz zwiększenia poziomu szczegółowości tworzonego modelu danych.

Notacja jest niczym innym jak swojego rodzaju specyfikacją określającą sposób opisu modelu danych przy wykorzystaniu umownych znaków i symboli graficznych. Pomimo wielu cech wspólnych notacje mają również pewne różnice. Każda z notacji ma swoje zalety i wady.

Jedne notacje lepiej nadają się do tworzenia modeli koncepcyjnych, inne z kolei znacznie lepiej spełniają swoją funkcję podczas tworzenia modeli logicznych, które później są transformowane do modelu fizycznego. Podział ten jest wyraźnie widoczny w niektórych narzędziach CASE do projektowania i modelowania danych wybranych producentów.

Dobór właściwej notacji znacznie ułatwia komunikację pomiędzy projektantem a klientem, jak również pomiędzy poszczególnymi członkami zespołu projektowego, a także pomiędzy samymi zespołami w przypadku tworzenia wielkich projektów. W zależności od wybranego pakietu do modelowania danych użytkownik ma możliwość wyboru różnego rodzaju notacji na danym etapie modelowania dla tworzonego modelu danych.

Modele koncepcyjne z założenia są wolne od szczegółów implementacyjnych modelu. Do najpopularniejszych modeli koncepcyjnych, tzw. modeli pojęciowych, można zaliczyć model związków encji ERD oraz diagram UML. Diagramy ERD pozwalają w postaci graficznej przedstawić model struktury danych oraz związków zachodzących pomiędzy poszczególnymi obiektami. Pomimo faktu, że standard UML nie został stworzony bezpośrednio z myślą o tworzeniu modeli danych, pozwala on na utworzenie takiego modelu za pomocą diagramu klas.

Diagram ERD, spopularyzowany przez Chena, z założenia miał być niezależny od wszelkich technik modelowania. Jego przeznaczeniem była prezentacja modelu na najwyższym poziomie abstrakcji modelowania, w zupełnym oderwaniu od konkretnego implementacyjnego modelu danych. W praktyce można spotkać dwa rodzaje diagramów: ogólny, gdzie celem jest związek, oraz szczegółowy, gdzie głównym elementem diagramu jest encja, najczęściej zawierająca informacje o atrybutach odbiegających od standardowej notacji Chena.

Do modeli implementacyjnych można zaliczyć model: relacyjny, obiektowy, relacyjno-obiektowy, semistrukturalny, sieciowy, hierarchiczny. Wybrany model danych jest modelowany w drugiej fazie projektowania – tworzenia modelu logicznego. Trzecim etapem jest utworzenie modelu fizycznego, określającego m.in. strukturę plików, a zatem i wybór środowiska, w którym zostanie zaimplementowana i wdrożona baza danych. Tak więc ważną rolę będzie odgrywać specyfika konkretnego serwera baz danych, której znajomością powinien także wykazać się projektant, aby wybór przez niego podjęty był optymalny z punktu widzenia wymagań klienta oraz wydajności aplikacji.

Wskutek rozwoju aplikacji baz danych o dużo większych wymaganiach wobec samej bazy danych niż dotychczas tworzone pojawiła się potrzeba wprowadzenia nowych semantycznych pojęć modelowania, uwzględniających specyfikę nowo tworzonych aplikacji baz danych, w które nie był wyposażony standardowy model ER. W odpowiedzi na nowe wyzwania pojawił się rozbudowany model ER zwany rozszerzonym modelem związków encji EER (ang. *Enhanced Entity Relationship*), zawierający dodatkowo opracowane nowe semantyczne pojęcia modelowania. Notacja EER jest rozszerzeniem podstawowego modelu ER o elementy obiektowości, takie jak: dziedziczenie, specjalizacja-generalizacja, nadklasa-podklasa, agregacja i kompozycja.

2. Notacje modelowania danych

Jedne z notacji są bardziej popularne w stosunku do innych, a co za tym idzie – częściej są używane w praktyce modelowania danych. Między innymi na skutek prac rozwojowych niektóre z dostępnych notacji mają kilka funkcjonujących równolegle nazw. Tym samym, w zależności od wybranego narzędzia CASE, niektóre z notacji występują w nich pod różnymi nazwami. Wśród najczęściej spotykanych notacji można wymienić notacje: relacyjną [18], IDEF1X, Martina (IE, Crows Foot, Finkelsteina – jako jedną z odmian standardu IE), Bachmana, Chena, Barkera (ORACLE), Merise, EXPRESS-G, UML, Jean-Raymond Abrial.

W zależności od użytego narzędzia CASE istnieje możliwość wybierania notacji z grupy przypisanej do konkretnego modelu koncepcyjnego, logicznego i fizycznego. Ponadto w zależności od tego, na jakim poziomie modelu pracuje modelarz danych, będą widoczne pewne różnice w obrębie tej samej notacji dostępnej na kilku poziomach analizy modelu danych.

Model koncepcyjny w swojej bezpośredniej postaci nigdy nie podlega wdrożeniu. Reprezentuje tylko ogólne idee oraz koncepcje. Ma bardzo ogólny charakter. Encje koncepcyjne prezentowane w tej fazie modelowania rzadko mają również zbiór dodatkowych cech w postaci atrybutów. Poziom analizy koncepcyjnej stanowi punkt wyjścia do zbudowania modelu logicznego, stanowiącego już szczegółowy opis modelu. Na tym etapie są identyfikowane i wyodrębniane wszystkie elementy danych niezbędne z punktu widzenia projektu, a więc potrzeb klienta, w postaci encji logicznych z definiującymi je atrybutami. Dokumentowane są również rodzaje związków zachodzących pomiędzy poszczególnymi encjami logicznymi (z określeniem obowiązkowości, kardynalności czy też identyfikacyjnego charakteru związku) oraz reguły biznesowe. Natomiast sam model logiczny nie jest w żaden sposób związany z jakąkolwiek platformą bazodanową. Model fizyczny dokonuje zamiany encji logicznych w fizyczne tabele, zatem musi on uwzględniać charakterystyczne właściwości i specyfikę platformy, na której będzie wdrażany.

Model danych powinien umożliwić przedstawienie struktury danych w postaci występujących w nim obiektów odgrywających kluczowe role z punktu widzenia funkcjonalności projektu oraz wydajności tworzonego systemu. Ponadto powinien pozwolić wyróżnić ważne cechy modelu, takie jak: silna (niezależna) i słaba (zależna) encja, encja asocjacyjna, związek identyfikujący i nieidentyfikujący, encje kategorii (zupełne, niezupełne, zawierające, wykluczające), atrybut kluczowy, atrybut złożony, atrybut wywiedziony (pochodny), rodzaj związku, kardynalność związku, opcjonalność związku. W zależności od wybranej notacji modelowania nie wszystkie wymienione cechy modelu mogą być dostępne modelarzowi danych.

2.1. Notacja Bachmana

Twórcą tej notacji jest Charles Bachman, który jako pierwszy opisał strukturę bazy danych w 1969 roku. Notacja Bachmana jest nazywana diagramem struktury danych. Była używana do modelowania danych przy użyciu sieciowego lub relacyjnego modelu danych. Encje są odwzorowane w postaci prostokątów, a związki w postaci łączących je linii. Groty strzałek na końcu linii określają kardynalność związku. Zdefiniowany przez Bachmana diagram opisujący strukturę danych został rozwinięty i spopularyzowany w 1973 roku przez Petera Chena, który sformalizował teorię, budując dla niej podstawy naukowe [9, 15, 17, 30].

2.2. Notacja Chena

Notacja ta została opracowana przez Petera Chena w 1976 roku [1, 2, 9-11, 30]. Jest jednym z najstarszych standardów modelowania. Największą popularność zyskała głównie w dziedzinie modelowania koncepcyjnego. Silne encje są przedstawiane za pomocą prostokątów zawierających nazwę danej encji. Słabe encje są reprezentowane przez prostokąty o podwójnej linii. W standardzie tym na poziomie modelowania koncepcyjnego nie są przedstawiane atrybuty. Związki w odróżnieniu do innych notacji są rzeczownikami, a nie czasownikami. Są definiowane przez użycie na diagramie symbolu rombu i odpowiednio podwójnego symbolu rombu dla związku powiązanego ze słabą encją. Ponadto notacja Chena w odróżnieniu od innych standardów pozwala na przedstawienie związków wieloznacznych $n:m$, związków wieloargumentowych oraz na zilustrowanie atrybutów związku. Atrybutem związku może być inny związek. Atrybuty są obrazowane w postaci pojedynczej elipsy. W przypadku atrybutu klucza głównego jego nazwa umieszczona w środku elipsy jest z podkreśleniem. Atrybut wielowartościowy jest prezentowany w postaci podwójnej elipsy, a atrybut pochodny określa elipsa w postaci przerywanej linii.

2.3. Notacja IE

Notacja IE (*Information Engineering*) została opracowana przez Jamesa Martina i Clive'a Finkelsteina w 1976 roku [1]. Prace zapoczątkował Clive Finkelstein w 1976 roku i były one przez niego rozwijane w latach 1976-1980 [11-14, 19, 20, 22, 24-26, 28]. Następnie zostały rozwinięte przez Jamesa Martina, który wprowadził do notacji własne modyfikacje. To właśnie metodologia Martina przyczyniła się do rozwoju branży narzędzi CASE, stanowiąc jej podwaliny. Obecnie jest to jeden z najbardziej popularnych standardów modelowania danych. Swoją popularność zawdzięcza niezwyklej prostocie i łatwości interpretacji diagramów ERD. Inną spotykaną nazwą jest notacja Martina. Istnieje kilka różnych wersji standardu IE charakteryzujących się kilkoma szczegółami składniowymi (m.in. niektóre z nich pozwalają na definiowanie związków wykluczających się w przeciwieństwie do innych wersji). Charakterystyczną cechą są symbole końcowe linii związków przypominające znak kurzej łapki w celu oznaczenia strony „wiele” związku wielokrotnego, stąd też potocznie nazywana bywa również notacją „kurze stopki” (*crow's foot*) bądź też „kurze łapki” (*crows feet*).

Silna encja jest reprezentowana w postaci prostokąta zawierającego nazwę będącą rzeczownikiem w liczbie pojedynczej. Słaba encja jest prezentowana za pomocą prostokąta rysowanego podwójną linią. Encja asocjacyjna jest prezentowana postaci prostokąta zawierającego wewnątrz umieszczony romb (w modelu koncepcyjnym). W modelu logicznym encje słabe są odwzorowywane w postaci prostokąta z zaokrąglonymi rogami podobnie jak w notacji IDEF1X. W jednej z odmian standardu IE opartej na klasycznej notacji Martina związki są odwzorowywane tylko za pomocą ciągłej linii, co nie daje tym samym możliwości jawnego określenia identyfikacyjnego lub też nieidentyfikacyjnego charakteru danego związku (wizualnie kopia klucza głównego encji nadrzędnej jest umieszczana jako atrybut klucza głównego encji podrzędnej i stanowi tym samym jej złożony klucz główny). W innej z odmian standardu związki w zależności od swojego rodzaju są modelowane za pomocą różnego stylu linii, co daje możliwość określenia związku nieidentyfikacyjnego za pomocą linii przerywanej. Na poziomie modelowania logicznego atrybuty klucza głównego są separowane od pozostałych atrybutów poziomą linią rozdzielającą. Notacja ta jest uważana za semantycznie silniejszą w odniesieniu do notacji IDEF1X i daje możliwość bardziej abstrakcyjnego podejścia do modelowania danych.

2.4. Notacja IDEF1X

Notacja IDEF1X (*Integration DEFINition Language 1 Extended*) wywodzi się z notacji IDEF opracowanej w latach 70. XX w. przez US Air Force [1, 2, 8, 14, 19, 22, 23]. W 1993 roku standard ten został rozszerzony przez Appletona. Notacja jest oparta na modelu relacyj-

nym opracowanym przez Codda oraz na modelu ERD zaproponowanym przez Chena. Umożliwia przedstawienie większości związków zachodzących pomiędzy danymi. Pozwala na łatwiejszą prezentację kluczy głównych i obcych. Jest używana w wielu narzędziach CASE do modelowania danych. Jest standardem używanym powszechnie w projektach rządowych USA. Standard ten jest nadal rozwijany na potrzeby związane z językiem UML oraz technikami obiektowymi, które zostały zaimplementowane w wersji IDEF1X97. W przeciwieństwie do notacji Barkera nie pozwala na prezentacje związków alternatywnych. Pozwala na przedstawienie diagramów ERD w postaci szczegółowej. Silna encja jest prezentowana w postaci prostokąta z ostrymi rogami, natomiast encja słaba jest odwzorowywana w postaci prostokąta z zaokrąglonymi rogami. Związki w zależności od ich charakteru są prezentowane za pomocą linii ciągłej lub przerywanej i określają odpowiednio rodzaj związku (identyfikującego lub nieidentyfikującego). Standard ten jest wyposażony w pokaźny zbiór symboli końcowych, które pozwalają uwzględnić w projektowanym modelu różne reguły licznosci oraz opcjonalności związków. Ponadto standard ten umożliwia określenie zupełności lub niezupełności kategorii encji. Atrybuty klucza głównego są oddzielone od pozostałych atrybutów danej encji za pomocą poziomej linii. Choć pierwotnie była przeznaczona dla modelowania fizycznego, notacja ta jest najbardziej przydatna podczas modelowania logicznego.

2.5. Notacja Barkera

Notacja została opracowana przez Richarda Barkera w 1990 roku. Jest wykorzystywana jako standard w narzędziach CASE firmy ORACLE. Wzorowana jest na notacji IE. Pozwala na prezentacje związków alternatywnych. Nazywana bywa również notacją Oracle's CASE Method [1, 3-7, 14, 19, 21, 23]. Jest oparta jest na koncepcji związków binarnych. Notacja ta w przeciwieństwie do notacji Chena nie pozwala na bezpośrednie przypisanie atrybutu danemu związkowi. Rozwiązanie tej sytuacji jest możliwe dzięki wybiegowi wprowadzenia dodatkowej encji pomiędzy dwiema uprzednio zdefiniowanymi encjami głównymi, tym samym związek $n:m$ zostanie zastąpiony dwoma związkami $1:n$. Encje są obrazowane w postaci prostokątów z zaokrąglonymi rogami. Przerywana linia związku symbolizuje jego opcjonalność – w przeciwieństwie do notacji IE nie ma symbolu końcowego umieszczonego na końcu linii definiującej związek (symbolizującego opcjonalność związku). Atrybut klucza głównego jest poprzedzany symbolem #, ale w przeciwieństwie do standardu IE czy IDEF1X nie jest separowany od pozostałych atrybutów za pomocą linii oddzielającej. Notacja Barkera pozwala także na modelowanie związków kategorii i podtypu-nadtypu w postaci zagnieżdżonych prostokątów. Umożliwia definiowanie związków wykluczających ilustrowanych za pomocą symbolu łuku obejmującego linie związków.

2.6. Notacja UML

Choć już w latach 70. zaistniało pojęcie modelowania obiektowego, a do pierwszej połowy lat 90. istniało kilkadziesiąt metod obiektowych (wśród których najbardziej znane były metody Shlaera-Mellora, Fusion czy Coad-Yourdon, a największe uznanie znalazły metody Boocha, OOSE, OMT), to dopiero w połowie lat 90. (1996 rok) przy kooperacji takich potentatów z sektora IT, jak IBM, Microsoft, ORACLE czy też HP, zostało utworzone konsorcjum UML. W wyniku tej współpracy powstał nowy standard w postaci zuniifikowanego języka modelowania UML 1.0. W 1997 roku opracowana specyfikacja została przekazana organizacji OMG (*Object Management Group*), która do dnia dzisiejszego zajmuje się jej rozwojem. Od tego czasu powstało wiele kolejnych wersji tego standardu. Obecnie najnowszą wersją jest UML 2.4, opublikowana w 2011 roku.

Notacja UML (*Unified Modeling Language*) nie jest oficjalną notacją modelowania danych. Jest graficznym językiem pozwalającym obrazować, kreować i dokumentować poszczególne elementy wchodzące w skład projektowanego systemu informatycznego. Na chwilę obecną standard UML pozwala zbudować wiele typów diagramów należących do kilkunastu przeróżnych kategorii diagramów. Najbardziej zbliżone do diagramów ERD są diagramy klas. To właśnie one, pomimo istniejących odmiennych opinii w kręgu specjalistów z branży IT zajmujących się projektowaniem baz danych, są w praktyce modelowania ostatnimi czasy również używane do tworzenia modelu baz danych. Model danych utworzony w UML może być przekształcony w postać tabel relacyjnych baz danych.

Encje są przedstawiane za pomocą prostokątów. Związki są przedstawiane za pomocą linii, na końcach której w postaci liczb (odzwierciedlających wartości minimalną i maksymalną) jest oznaczana krotność związku. Notacja UML pozwala na definiowanie atrybutów związków. Atrybuty takie są połączone z linią związku za pomocą linii przerywanej. W celu przedstawienia związków n-arnych o stopniu większym niż dwa jest używany symbol rombu. Nazwy atrybutów powinny rozpoczynać się od małej litery. Atrybut klucza głównego powinien być zdefiniowany jako pierwszy pośród pozostałych atrybutów – najczęściej jest oznaczany za pomocą symbolu PK. W przypadku atrybutu złożonego poniżej jego nazwy z lekkim wcięciem z lewej strony są umieszczane jego składowe. Atrybuty wielowartościowe obok swoich nazw mają umieszczone znaczniki powtórzeń wartości tych atrybutów. Atrybuty pochodne są poprzedzane symbolem „/” [11, 16, 19, 27].

2.7. Inne notacje modelowania

Istnieje wiele innych notacji, które nie zyskały takiej popularności bądź też wyszły z powszechnego użytku i przestały być stosowane w narzędziach CASE najnowszej genera-

cji. Jedną z notacji, o której warto wspomnieć, jest standard EXPRESS-G, po raz pierwszy wprowadzony na początku lat 90. Jest on graficznym składnikiem języka specyfikacji wymagań formalnych informacji – języka Express o sformalizowanej postaci w standardzie ISO 10303-11. Pozwala tworzyć diagramy na poziomie encji i schematu. Język Express i notacja Express-G są składnikami międzynarodowego standardu przedstawiania i wymiany danych dotyczących produktów w formacie możliwym do zinterpretowania przez komputery – STEP (*Standard for the Exchange of Product Model Data*) [29]. Notacja była zaimplementowana m.in. w Visio 2000, EDraw 6. Należy wymienić także notację Jean-Raymond Abriala z 1974 roku, czy też notację Merise (*Methode d'Etude et de Realisation Informatique pour les Systemes d'Enterprise*) opracowaną w 1992 roku. Notacja Merise pozwala na tworzenie związków n-arnych. Jest wykorzystywana w narzędziu Sybase Power Designer 16.1 na etapie tworzenia modelu koncepcyjnego.

Istnieje także wiele notacji związanych z modelowaniem innych struktur niż tradycyjne modele danych, chociażby takie, jak: modelowanie procesów biznesowych za pomocą notacji BPMN czy też architektura korporacyjna Archimate, TOGAF. Traktują one proces modelowania w dużo szerszym ujęciu na różnych poziomach. Ze względu na to, iż wspomniane notacje odnoszą się do innych wymiarów modelowania bądź też traktują proces modelowania w szerszym, pełnowymiarowym ujęciu, z punktu widzenia architektury korporacyjnej, odbiegają one od głównego tematu artykułu (który koncentruje się na porównaniu wybranych notacji modelowania danych na bazie modelu relacyjnego w projektowaniu baz danych), zostały one tutaj jedynie zasygnalizowane i nie będą dokładniej omówione ani też uwzględnione w analizie porównawczej.

Notacja BPMN (*Business Process Model and Notation*, wcześniej zwana notacją *Business Process Modeling Notation*) [37] umożliwia modelowanie procesów biznesowych niezależnych od specyfiki danej branży czy organizacji. Dzięki temu jest metodą uniwersalną, niezależną od kontekstu użycia.

TOGAF (*The Open Group Architecture Framework*) [33, 34] umożliwia kompleksowe podejście do projektowania, planowania, implementacji i zarządzania informacyjną architekturą przedsiębiorstwa. Architektura korporacyjna jest modelowana na poziomie procesów biznesowych, systemów informatycznych, danych i technologii.

Archimate [31, 32] jest niezależnym i otwartym językiem modelowania architektury korporacyjnej, który pozwala opisywać, wizualizować i analizować w ramach czterech domen architektonicznych: biznesowej, danych, aplikacji i technicznej. Opiera się na paradygmacie warstwowości (definiującym warstwy: biznesową, danych, aplikacji i techniczną) i paradygmacie usługowości (identyfikującym usługi biznesowe, aplikacyjne oraz infrastrukturalne).

Ponadto w przeciwieństwie do standardu UML czy też BPMN umożliwia prezentację związków pomiędzy poszczególnymi domenami architektonicznymi.

Narzędzie ER Studio firmy EMBARCADERO oprócz modułu do modelowania danych ER Studio Data Architect (zwierającego wybrane notacje modelowania danych, tj. IDEF1X, różne warianty standardu IE) zawiera również m.in. moduły do modelowania procesów biznesowych oraz architektury aplikacji. ER Studio Business Architect [35] umożliwia tworzenie modeli koncepcyjnych w projektowaniu baz danych, a także modelowanie procesów biznesowych na podstawie notacji BPMN. ER Studio Software Architect [36] wykorzystuje notację UML, język zapisu ograniczeń w modelu obiektowym OCL (*Object Constraint Language*), będący składową języka UML, oraz standardowy zestaw języków do przekształcania modelu QVT (*Query/Views/Transformation*) do obiektowego modelowania aplikacji.

3. Porównanie notacji














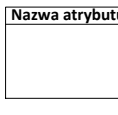
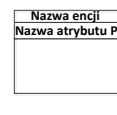


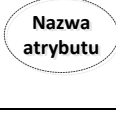
Poszczególne implementacje notacji modelowania danych mogą nieznacznie różnić się w poszczególnych narzędziach CASE w odniesieniu do postaci źródłowej opracowanej przez ich twórców. W tabelach 1-3 zestawiono graficzną reprezentację danych wykorzystywanych w diagramach ERD i UML przez poszczególne notacje modelowania. W tabeli 4 zaprezentowano zestawienie wybranych notacji modelowania zaimplementowanych w wybranych narzędziach CASE.

Niektóre z pakietów CASE, w przeciwieństwie do innych, mają wprowadzony wyraźny podział, jeśli chodzi o wykorzystanie wybranych notacji do tworzenia modelu danych na konkretnym poziomie modelowania. Oprogramowanie ERWin zawiera podział na notacje, które są dostępne w fazie projektowania logicznego (IDEF1x, IE) i fizycznego (IDEF1x, IE, Data Warehousing). Również pakiet Power Designer zawiera podział na notacje przypisane do modelu koncepcyjnego (Barker, Entity/Relationship, IDEF1x, Merise, E/R+Merise), logicznego (Barker, Entity/Relationship, IDEF1x) i fizycznego (IDEF1x, Conceptual, CODASYL, Relational).

Część spośród dostępnych na rynku narzędzi CASE przeznaczonych do modelowania danych pozwala użytkownikowi wybrać, w jaki sposób chce dokonać prezentacji związku $n:m$. Jednym z narzędzi oferujących taką możliwość jest Power Designer. Pozwala on na prezentację związku $n:m$ łączącego dwie encje za pomocą jednego związku lub automatyczne utworzenie logicznej encji asocjacyjnej. Podobnie pakiet ER Studio XE umożliwia utworzenie związku $n:m$ na poziomie modelu logicznego, w modelu fizycznym zaś automatycznie generuje dodatkową asocjacyjną tabelę fizyczną, za pomocą której związek $n:m$ jest rozkładany na dwa związki $1:n$.

Tabela 1

Zestawienie elementów składowych dla wybranych notacji

Składnia	Wybrane notacje modelowania					
	Bachmana	Chena	IE (Martina)	IDEF1X	Barkera	UML
Encja silna						
Encja słaba						
Atrybut kluczowy	PK Nazwa atrybutu		<u>Nazwa atrybutu</u>		# Nazwa atrybutu	
Atrybut	Nazwa atrybutu		Nazwa atrybutu	Nazwa atrybutu	Nazwa atrybutu	Nazwa atrybutu
Atrybut wielowartościowy	Brak		{Nazwa atrybutu}	Brak	Brak	Nazwa atrybutu Atr1 Atr2
Atrybut pochodny	Brak		Brak	Brak	Brak	/Nazwa atrybutu

W przypadku ER Studio XE standard UML jest obsługiwany w narzędziu ER Studio Software Architect 1.1, który wchodzi w skład całego pakietu.

Notacje Chena, Express-G, Bachmana można spotkać w starszych wersjach narzędzi CASE. Były one implementowane m.in. w Visio 2000.

Tabela 2
Zestawienie określonych kombinacji związków dla wybranych notacji

Liczność	Wybrane notacje modelowania					
	Bachmana	Chena	IE (Martina)	IDEF1X	Barkera	UML
1:1 ob,ob.						
1:1 op,op						
1:1 op,ob						
1:1 ob,op						
1:n, i ob,op						
1:n, i ob,op						
1:n op,op						
1:n ob,ob						
1:n ob,op						
m:n op,op						
m:n ob,ob						
m:n ob,op						
m:n op,ob						

ob – obowiązkowe uczestnictwo encji,

op – opcjonalne uczestnictwo encji,

i – związek identyfikujący

Tabela 3

Zestawienie specjalizacji – generalizacji dla wybranych notacji

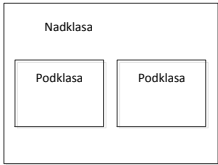
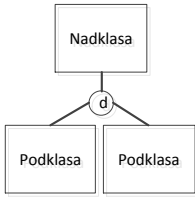
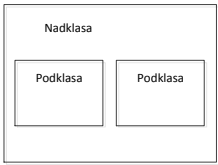
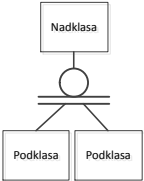
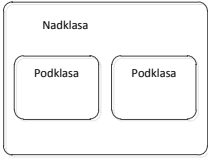
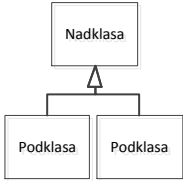
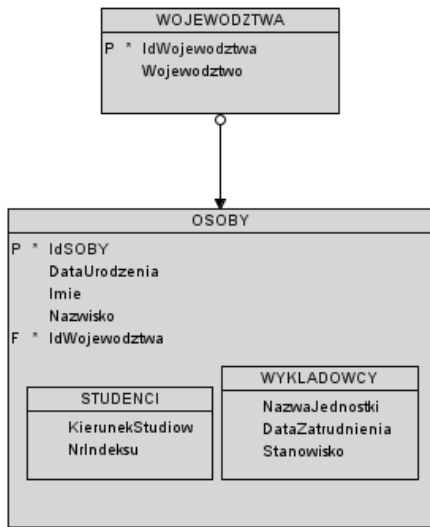
Notacja modelowania	Bachmana	Chena	IE (Martina)
Specjalizacja Generalizacja			
Notacja modelowania	IDEF1X	Barkera	UML
Specjalizacja Generalizacja			

Tabela 4

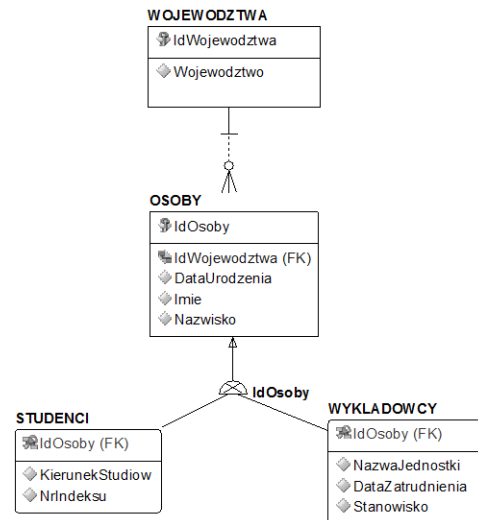
Zestawienie wykorzystania poszczególnych notacji w wybranych narzędziach CASE do modelowania danych

Notacje modelowania	Wybrane narzędzia CASE do modelowania danych					
	MS Visio 2010	ER Studio Data Architect 8.5.3	Oracle SQL Developer Data Modeler 3.1.0.700	CA ERwin Data Modeler 8.2	Sybase Power Designer 16.1	My SQL Workbench 5.2.38
Bachmana			✓			
Chena						
IE (Martina)	✓	✓	✓	✓	✓	✓
IDEF1X	✓	✓		✓	✓	✓
Barkera (Oracle)			✓		✓	
UML	✓	✓			✓	✓

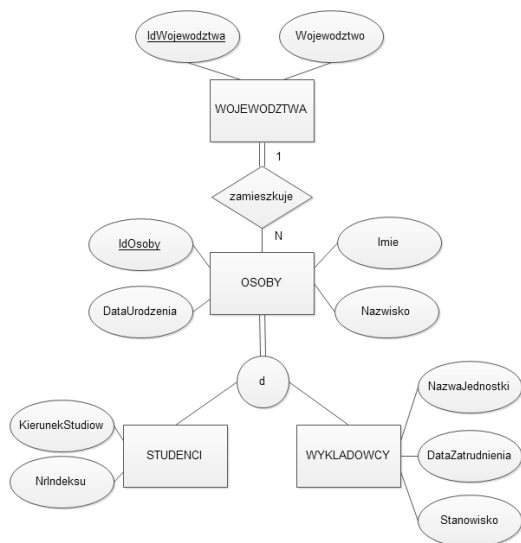
Na rysunkach 1-6 przedstawiono prosty model danych dla kilku różnych notacji modelowania, utworzony za pomocą wybranych narzędzi: Oracle SQL Data Modeler, EDraw Max, ER Studio XE, Sybase Power Designer, MS Visio.



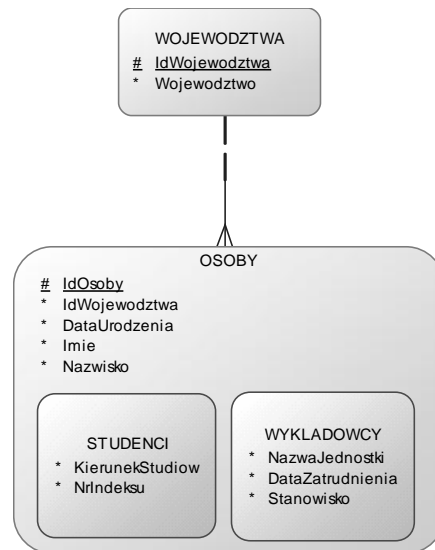
Rys. 1. Notacja Bachmana
Fig. 1. Bachman notation



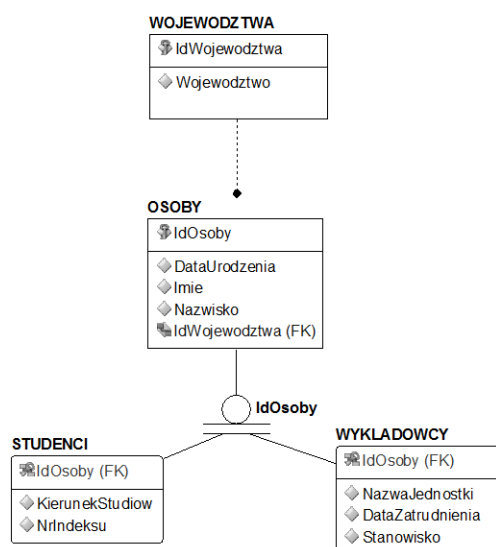
Rys. 3. Notacja IE
Fig. 3. IE notation



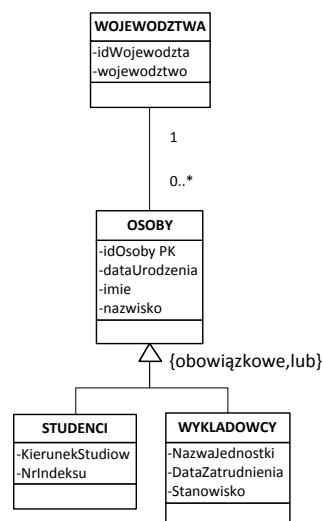
Rys. 2. Notacja Chena
Fig. 2. Chen notation



Rys. 4. Notacja Barkera
Fig. 4. Barker notation



Rys. 5. Notacja IDEF1X
Fig. 5. IDEF1X notation



Rys. 6. Notacja UML
Fig. 6. UML notation

4. Podsumowanie

Ze względu na fakt, iż poszczególne notacje modelowania różnią się mniej lub bardziej od siebie sposobem opisu modelu danych, a co za tym idzie – możliwościami opisanie modelu danych, są wykorzystywane na różnych poziomach modelowania. Jedne z nich znacznie lepiej będą spełniać swoje zadania na poziomie projektowania koncepcyjnego, a zupełnie nie będą przystosowane do tworzenia modelu danych na poziomie projektowania fizycznego (jak np. notacja Chena). Natomiast inne będą bardzo dobrze przystosowane do tworzenia modelu logicznego (np. IE, IDEF1X, Barkera), radząc sobie równie dobrze z tworzeniem modelu fizycznego. Ze względu na swoje możliwości opisu modelu danych jedne notacje zyskały sobie większą popularność w stosunku do pozostałych, tym samym część z nich jest o wiele częściej implementowana w najbardziej popularnych narzędziach CASE do modelowania danych w stosunku do pozostałych, a część z nich zupełnie nie jest w nich uwzględniana (bądź też była uwzględniana w starszych wersjach narzędzi CASE modelujących dane i wyszła już z powszechnego użytku). Drugim czynnikiem, który przyczynił się do spopularyzowania jednych notacji w stosunku do innych, była intuicyjność w posługiwaniu się nimi oraz w tworzeniu i interpretacji modelu danych. Do najbardziej intuicyjnych notacji zalicza się notację IE opracowaną przez Martina, która jest implementowana we wszystkich dominujących narzędziach CASE. W przypadku tej notacji można się spotkać z różnymi jej odmianami, implementowanymi w narzędziach CASE różnych producentów. Równie często imple-

mentowanymi notacjami modelowania są notacje IDEF1X, UML i notacja Barkera, która jest podstawowym standardem implementowanym w narzędziach firmy ORACLE.

Jak widać, nie ma jednej jedynie słusznej notacji modelowania. Podejmując decyzję o wyborze notacji modelowania, projektant powinien kierować się nie tylko znajomością danego standardu, lecz także powinien mieć świadomość możliwości, jakie oferuje dana notacja, w kontekście zarówno tworzenia, jak i samej prezentacji modelu danych dla konkretnego modelowanego problemu. Zatem oczywiste wydaje się, że osoby zawodowo trudniące się modelowaniem danych i projektowaniem baz danych powinny być zaznajomione z kilkoma czołowymi, najbardziej popularnymi notacjami modelowania wykorzystywanymi w praktyce modelowania danych w profesjonalnych narzędziach CASE.

BIBLIOGRAFIA

1. Allen S.: Modelowanie danych, Helion, Gliwice 2005.
2. Banachowski L. i in.: Relacyjne bazy danych. Wykłady i ćwiczenia. Wydawnictwo PJWSTK, Warszawa 2009.
3. Barker R.: Case Method. Modelowanie związków encji, WNT, Warszawa 2005.
4. Barker R.: Case Method: Entity Relationship Modeling, tom 1, Addison Wesley, Boston 1990.
5. Barker R.: Case Method: Tasks and Deliverables, Tom 2, Addison Wesley, Boston 1990.
6. Barker R.: Entity Relationship Modelling, Addison Wesley, Boston 1991.
7. Barker R.: Modelowanie związków encji, WNT, Warszawa 1997.
8. Bruce T.: Designing Quality Databases with IDEF1X Information Models, Dorset House Publishing, New York 1992.
9. Chen P.: Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. http://bit.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf, 2012.05.25.
10. Chen P.: The Entity-Relationship Model – Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1, 1976.
11. Connolly T., Begg C.: Systemy baz danych. Praktyczne metody projektowania, implementacji i zarządzania, tom I. RM, Warszawa 2004.
12. Finkelstein C.: An Introduction to Information engineering: From Strategic Planning to Information Systems. Addison-Wesley, Sydney 1989.
13. Finkelstein C.: Information engineering methodology. W: Bernus P., Mertins K., Schmidt G. (eds.): Handbook on Architectures of Information Systems. Springer-Verlag, Berlin 1998, pp. 405-427.

14. Halpin T., Morgan T.: Information modeling and relational database. Morgan Kaufmann Publishers, Burlington 2008.
15. <http://bit.csc.lsu.edu/~chen/chen.html>, 2012.05.12.
16. <http://brasil.cel.agh.edu.pl/~09sbfraczek/diagram-klas,1,11.html>, 2012.05.14.
17. http://it.toolbox.com/wiki/index.php/Entity_Relationship_Diagram, 2012.05.19.
18. <http://office.microsoft.com/pl-pl/visio-help/wybieranie-notacji-kardynalnosci-idef1x-relacyjnej-lub-kurzycz-apk-HA010115486.aspx#BM2>, 2012.05.12.
19. <http://www.agiledata.org/essays/dataModeling101.html>, 2012.05.14.
20. <http://www2.cs.uregina.ca/~bernatja/crowsfoot.html>, 2012.05.19.
21. Inguanez F.: Entity Relationship Modelling A short guide to designing Entity Relationship Models using Barker's notation, <http://frankieinguanetz.files.wordpress.com/2012/01/barkers-erd-notation.pdf>, 2012.05.30.
22. Johnson E., Jones J.: Modelowanie danych w SQL Server 2005 i 2008. Helion, Gliwice 2009.
23. Knowledge Based Systems, (2011, maj 12). IDEF1X. Data Modeling Method, <http://www.idef.com/IDEF1x.htm>, 2012.05.12.
24. Martin J., Finkelstein C.: Information engineering. Technical Report, Savant Institute, Carnforth, Lancs, UK 1981.
25. Martin J.. Information engineering, Prentice-Hall, Englewood Cliffs, NJ 1989.
26. Martin J.. Principles of Object Oriented Analysis and Design, Prentice Hall, Englewood Cliffs, NJ 1993.
27. Muller R. J.: Bazy danych - język UML w modelowaniu danych. MIKOM, Warszawa 2000.
28. Roszkowski J.: Analiza i projektowanie strukturalne, Helion, Gliwice 2004.
29. Schenck D., Wilson P.: Information Modeling the EXPRESS Way. Oxford University Press, 1994.
30. Song I. et al: A Comparative Analysis of Entity-Relationship Diagrams. Journal of Computer and Software Engineering, Vol. 3, No.4 (1995), pp. 427-459. <http://cin.ufpe.br/~in1008/aulas/A%20Comparative%20Analysis%20of%20Entity-Relationship%20Diagrams.pdf>, 2012.05.18.
31. <http://architekturakorporacyjna.pl/>, 2013.12.07.
32. <http://www.opengroup.org/subjectareas/enterprise/archimate>, 2013.12.07.
33. <http://www.opengroup.org/subjectareas/enterprise/togaf>, 2013.12.07.
34. <http://www.togaf.org/>, 2013.12.07.
35. <http://www.embarcadero.com/br/products/er-studio-business-architect>, 2013.12.07.

36. <http://www.embarcadero.com/br/products/er-studio-software-architect>, 2013.12.07.
37. <http://www.bpmn.org/>, 2013.12.07.

Abstract

The article presents the development of the individual data modeling notation with reference to the relational model. Some comparative analysis of selected modeling notation has been shown. It turns out that the most common modeling standards are implemented in the most popular CASE tools for data modeling. Graphical representation of data used in ERD and UML diagrams by different modeling notations are summarized in Tables 1-3. Summary of components for chosen notation ie strong and weak entity, attribute key, polyvalent and derivative attribute is presents in Table 1. Specified combinations of relationships for selected notations specifying: optionality and mandatory relationship and identifying relationship is shown in Table 2. The comparison of specialization - generalization for the selected notation is presented in Table 3. The implementation of various notations in selected CASE tools for data modeling is presented in Table 4. Simple data model for several different modeling notations created using selected tools: Oracle SQL Data Modeler, EDraw Max, ER Studio XE, Sybase Power Designer, MS Visio are shown in figures 1-6. Some notations gained more popularity than other notations due to their capabilities describe the data model. Therefore, some notations are often implemented in the most popular CASE tools for data modeling in relation to the other notation. One of the most intuitive notations is developed by Martin -IE notation. This notation is implemented in all the leading data modeling tools. Notation IE has many different varieties of his that are implemented in CASE tools different manufacturers. In addition, they are also often implemented notation: IDEF1X, UML notation and Barker's notation being the basic standard tools of ORACLE. There is no single correct data modeling notation. Choosing a modeling notation Database Designer, should take into account in addition to the knowledge of the standard, the possibilities offered by the notation in the context of the creation and presentation of the data model for a specific modeled problem.

Adres

Sebastian ŁACHECIŃSKI: Uniwersytet Łódzki, Katedra Informatyki Ekonomicznej, Wydział Ekonomiczno-Socjologiczny, ul. Rewolucji 1905 r. nr 37, 90-214 Łódź, Polska, slachecinski@uni.lodz.pl.