

Lech TUZINKIEWICZ, Bartosz PIECHOTA  
Politechnika Wrocławska, Instytut Informatyki

## OCENA NARZĘDZI WSPIERAJĄCYCH PROCES PROJEKTOWANIA BAZ DANYCH

**Streszczenie.** W artykule przedstawione zostało podejście do porównania narzędzi typu CASE, w kontekście możliwości transformowania modeli danych w procesie projektowania baz danych. Zaproponowany został model jakości do oceny narzędzi CASE. Zdefiniowana została funkcja oceny, która uwzględniając wyniki przeprowadzonych eksperymentów, wyraża jakość ocenianych narzędzi w kontekście wybranych charakterystyk modelu jakości.

**Słowa kluczowe:** narzędzia CASE, projektowanie baz danych, transformacja modeli danych, model jakości

## QUALITY EVALUATION OF THE SOFTWARE TOOLS SUPPORTING DATABASE DESIGN

**Summary.** In this paper an approach to CASE tools comparison in the context of data models transformation capabilities in database design is presented. Quality model for CASE tools assessment has been introduced. Proposed quality function, basing on results of conducted experiments, determines the quality of the tools in chosen characteristics context.

**Keywords:** CASE tools, database design, quality model, quality in use

### 1. Wstęp

Projektowanie baz danych jest procesem złożonym i polega na modelowaniu danych na różnych poziomach abstrakcji [2]. W praktyce wykorzystuje się narzędzia wspierające ten proces, dzięki czemu możliwe jest skrócenie czasu, niezbędnego na zaprojektowanie bazy danych i w przypadku osób o małym doświadczeniu w tej dziedzinie – uzyskanie produktu,

który będzie poprawny składniowo. Zapewnienie wierności odwzorowania modelowanej dziedziny przedmiotowej nie jest na razie wspierane przez tego typu narzędzia.

Z uwagi na to, że istnieje różnorodna i bogata oferta narzędzi wspierających projektowanie baz danych [3, 4, 5, 6, 7, 8], nasuwają się pytania jak oceniać te narzędzia i jakie kryteria stosować przy podejmowaniu decyzji o ich wyborze? W związku powyższym został zaproponowany model jakości użytkowej, który może być wykorzystany zarówno do oceny porównawczej narzędzi typu CASE, jak i, biorąc pod uwagę oczekiwania projektanta w stosunku do tej klasy narzędzi, możliwość ich oceny w kontekście potrzeb użytkownika.

W artykule przedstawiono podstawowe cele związane z projektowaniem baz danych, które stanowią kontekst przyjętych kryteriów oceny narzędzi typu CASE. Na podstawie modelu jakości użytkowej z normy ISO 25010 [1] zaproponowano model jakości, w którym wskazano charakterystyki mające zastosowanie przy ocenie jakości narzędzi wspierających projektowanie baz danych. Całość została zilustrowana przykładem oceny jakości użytkowej wybranych, popularnych narzędzi typu CASE, w kontekście tak dobranego modelu konceptualnego danych, by uwzględnić typowe przypadki zależności występujące w rzeczywistych projektach. Wnioski z przeprowadzonych badań i eksperymentów zostały ujęte w podsumowaniu.

## 2. Model jakości użytkowej

Podstawą oceny narzędzi CASE jest możliwość zrealizowania za ich pomocą następujących celów:

- Zamodelowanie dziedziny przedmiotowej (wycinka rzeczywistości) zgodnie z potrzebami wynikającymi z wymagań funkcjonalnych i нефункциональных, określonych dla projektowanego systemu.
- Utworzenie modelu fizycznego danych, na podstawie którego można wygenerować instancję bazy, pozwalającą zrealizować przyjęte wymagania funkcjonalne i нефункциональные.

Jakość ocenianego produktu, w tym przypadku narzędzia typu CASE wspierającego proces projektowania baz danych, będzie oceniana w kontekście możliwości realizacji następujących zadań:

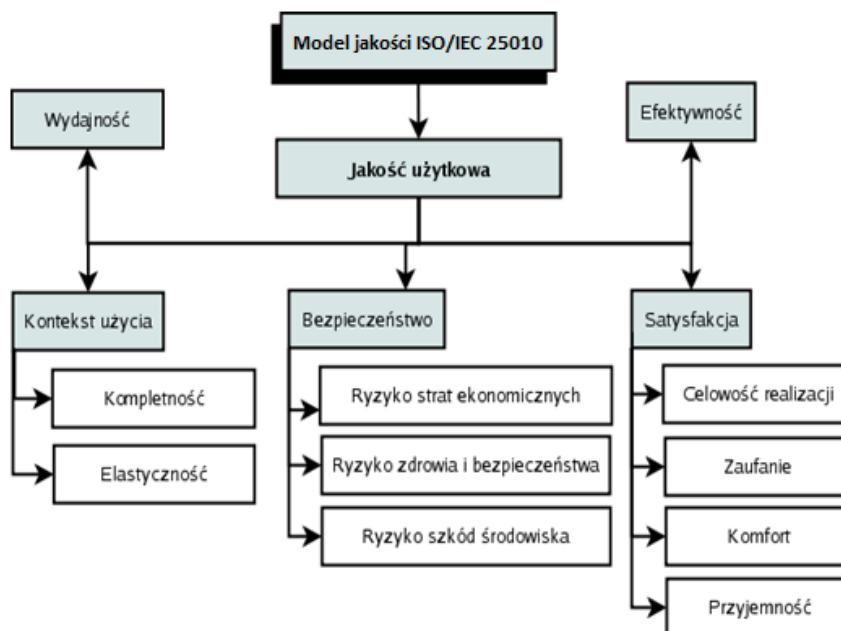
- Definicji modelu konceptualnego, który będzie odzwierciedleniem modelowanego wycinka rzeczywistości. Narzędzie powinno umożliwiać reprezentowanie bytów świata rzeczywistego wraz z istniejącymi ograniczeniami. Ograniczenia mogą dotyczyć własności wewnętrznych (dotyczących określonego bytu) i zewnętrznych (relacji pomiędzy bytami) danego elementu wycinka rzeczywistości oraz własności statycznych i dynamicznych.

- Transformacji modeli pomiędzy poszczególnymi poziomami abstrakcji (model konceptualny -> model logiczny -> model fizyczny).
- Modyfikacji modeli powstałych w wyniku transformacji.
- Kreowania instancji bazy danych na podstawie zdefiniowanego modelu danych.
- Wygenerowania danych testowych z uwzględnieniem specyfiki danego środowiska bazodanowego (SZBD) oraz zapewnienie jakości tych danych, tzn. uwzględnienie ograniczeń zdefiniowanych w modelu danych.
- Wykonanie inżynierii odwrotnej, tzn. utworzenia modelu konceptualnego na podstawie istniejącej bazy danych.

Na podstawie wniosków z analizy modelu jakości użytkowej z normy ISO 25010 [1], w kontekście wyżej wymienionych zadań, został zaproponowany model jakości do oceny środowisk wspierających proces projektowania baz danych. Spełnienie wymienionych powyżej wymagań jest podstawą oceny jakości produktów.

### 2.1. Model jakości użytkowej w normie ISO 25010 Quality in Use

Model jakości użytkowej przeznaczony do oceny narzędzi wspierających proces projektowania baz danych został opracowany na podstawie modelu zaproponowanego w normie ISO/IEC 25010 (rys. 1).



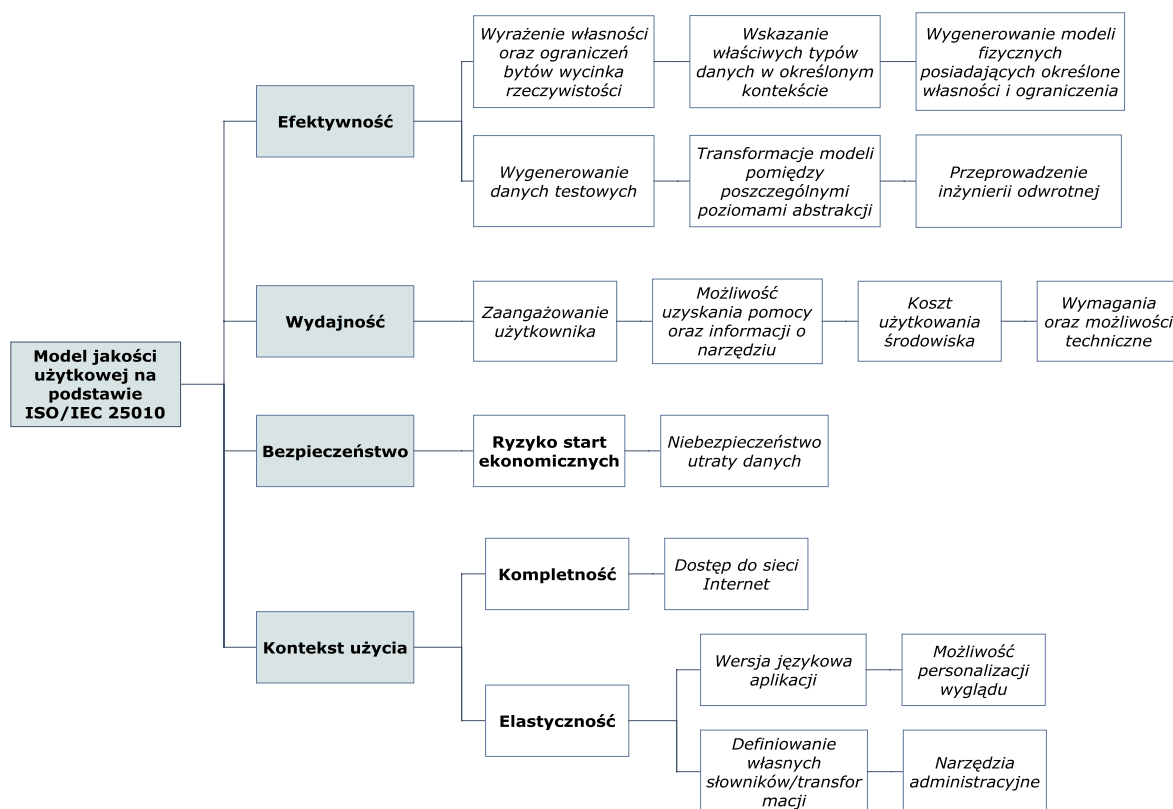
Rys. 1. Norma ISO 25010 – Model jakości użytkowej  
Fig. 1. ISO 25010 standard – Quality in use model

Z modelu jakości użytkowej (rys. 1) zostały wybrane charakterystyki, które mają zastosowanie w kontekście jakościowej oceny artefaktów tworzonych w procesie projektowania

baz danych. Definicja modelu jakości oraz dyskusja zaproponowanego rozwiązania zostały przedstawione w kolejnym podrozdziale niniejszego artykułu.

## 2.2. Model jakości użytkowej dla oceny narzędzi wspierających projektowanie baz danych

Na podstawie normy ISO/IEC 25010 zaproponowany został model jakości użytkowej dla oceny narzędzi CASE. Charakterystyki tego modelu wywodzą się z modelu jakości użytkowej z normy ISO/IEC 25010. Poszczególne charakterystyki zawierają atrybuty służące ocenie jakości poszczególnych elementów/funkcjonalności badanych narzędzi. Na diagramie (rys. 2.) pominięte zostały te charakterystyki z normy, które nie mają zastosowania przy ocenie jakości użytkowej baz danych. Pogrubioną czcionką wyróżnione są charakterystyki z normy ISO, natomiast kursywą oznaczono zdefiniowane atrybuty. Zaproponowany model jakości ma charakter generyczny i instancja tego modelu pozwala oceniać jakość narzędzi CASE w kontekście potrzeb użytkownika (ograniczając zbiór charakterystyk).



Rys. 2. Model jakości użytkowej do oceny narzędzi wspierających projektowanie baz danych (na podstawie normy ISO/IEC 25010)

Fig. 2. Proposed quality model consistent with ISO/IEC 25010

### 2.2.1. *Charakterystyki oraz atrybuty zaproponowanego modelu jakości do oceny narzędzi CASE*

- Efektywność
  - Wyrażenie własności oraz ograniczeń bytów wycinka rzeczywistości:
    - Języki/notacje, za pomocą których można opisać byty wycinka rzeczywistości (UML, ERD).
    - Zestaw ograniczeń, które powinny być uwzględniane (zakres wartości, typ wyliczeniowy, zależności referencyjne, obligatoryjność, unikalność).
  - Wskazanie właściwych typów danych w określonym kontekście:
    - Liczba dziedzin (typów) wartości dostępnych przy definiowaniu własności statycznych.
    - Możliwość zdefiniowania typów użytkownika.
    - Możliwość zdefiniowania ograniczeń na predefiniowane dziedziny (typy) wartości.
  - Wygenerowanie modeli fizycznych, które mają zdefiniowane własności i ograniczenia na poziomie konceptualnym:
    - Model fizyczny dla wybranych systemów baz danych.
    - Model fizyczny zgodny ze standardem SQL.
  - Wygenerowanie danych testowych:
    - Zgodność z ograniczeniami modelu.
    - Możliwość wykorzystania słowników, w celu uzyskania możliwie najbardziej rzeczywistych danych.
  - Transformacje modeli pomiędzy poszczególnymi poziomami abstrakcji:
    - Transformacja pomiędzy poziomami: konceptualnym, logicznym i fizycznym.
    - Możliwość wprowadzenia ręcznych modyfikacji modeli przez użytkownika.
  - Przeprowadzenie inżynierii odwrotnej:
    - Odtworzenie modeli na wyższych poziomach abstrakcji z niższych poziomów.
    - Zachowanie własności oraz ograniczeń.
- Wydajność
  - Zaangażowanie użytkownika:
    - Czas potrzebny użytkownikowi na opanowanie narzędzia.
    - Popularność narzędzia (liczba niezależnych for internetowych, grup dyskusyjnych dotyczących danego narzędzia).
  - Możliwość uzyskania pomocy oraz informacji o narzędziu:
    - Istnienie pomocy kontekstowej.
    - Dostępność szkoleń z zakresu obsługi narzędzia.
    - Dostępność dokumentacji technicznej.

- Wsparcie on-line ze strony producenta.
- Możliwość uzyskania wsparcia od innych użytkowników.
- Koszt użytkowania środowiska:
  - Płatna/bezplatna licencja, zakres wykorzystania licencji.
- Wymagania oraz możliwości techniczne:
  - Systemy operacyjne, dla których istnieją wersje danych narzędzi.
  - Możliwość integracji z zewnętrznymi narzędziami wytwórczymi /projektowymi (Eclipse, NetBeans, Visual Studio).
- **Satysfakcja**

*Charakterystyka dotycząca satysfakcji jest subiektywna, a celem niniejszego opracowania jest uwzględnienie aspektu technicznego oraz ustalenie merytorycznej zawartości narzędzi, dlatego też w zaproponowanym modelu charakterystyka ta została pominięta. W przyszłości można wziąć pod uwagę tę charakterystykę, bazując na ankietach skierowanych do użytkowników różnych środowisk.*
- **Bezpieczeństwo**
  - Ryzyko start ekonomicznych:
    - Niebezpieczeństwo utraty danych.
    - Niebezpieczeństwo zniekształcenia danych.
  - **Ryzyko zdrowia i bezpieczeństwa**
  - **Ryzyko szkód środowiska**

*Obydwie pogrubione charakterystyki zostały pominięte w opracowanym modelu jakości, ponieważ w ramach żadnej z nich nie zdefiniowano atrybutów, które mogłyby być istotne z punktu widzenia ocenianych narzędzi.*
- **Kontekst użycia**
  - **Kompletność**
    - Dostęp do sieci Internet
      - Wymagany dostęp do sieci podczas uruchamiania i pracy z wykorzystaniem narzędzia, możliwość pracy w trybie off-line.
  - **Elastyczność:**
    - Wersja językowa aplikacji.
    - Możliwość personalizacji wyglądu.
    - Definiowanie własnych słowników/transformacji.
    - Narzędzia administracyjne:
      - Wykorzystanie inżynierii odwrotnej do utrzymania oraz diagnostyki produktu przez administratora.

### 3. Przykład oceny jakościowej narzędzi CASE

W kontekście przyjętego modelu jakości przeprowadzono wiele eksperymentów, których wyniki posłużyły do ostatecznej oceny wybranych narzędzi. Eksperymenty zostały przeprowadzone bazując na modelu konceptualnym, zawierającym elementy reprezentujące typowe przypadki bytów i ich relacji, występujące w praktycznych rozwiązaniach. Model konceptualny danych został wyspecyfikowany w języku UML. W dalszej części przedstawiono wybrane (do oceny jakościowej) narzędzia typu CASE oraz wyniki eksperymentów, uzyskanych dla przykładowej funkcji oceny.

#### 3.1. Wybrane narzędzia CASE i środowiska docelowe

Podczas przeglądu istniejących narzędzi CASE zostały wytypowane następujące narzędzia, które mogłyby zostać poddane eksperymentom w kontekście opracowanego modelu jakości:

- **Visual Paradigm DB Visual Architect 6.1,**
- Sybase PowerDesigner 15.3,
- **Microsoft Visual Studio 2010 Ultimate,**
- **astah\* professional 6.3,**
- IBM InfoSphere Data Architect 7.5.3,
- Sparx Enterprise Architect 8.0 Professional.

Nie wszyscy producenci wyrazili jednak zgodę na udostępnienie swoich produktów w eksperymencie oceny jakościowej, w związku z czym, w zbiorze ocenianych narzędzi znalazły się tylko wyróżnione produkty (pogrubione) z powyższej listy.

Ponadto, w zakresie oceny uwzględniony został aspekt możliwości generowania modeli fizycznych danych. W związku z powyższym, wybrano popularne produkty komercyjne oraz dwa produkty bezpłatne, często wykorzystywane w projektach informatycznych:

- Microsoft SQL Server 2008,
- Oracle 11g,
- PostgreSQL 9.0.3,
- MySQL 5.5.9.

#### 3.2. Ocena narzędzi CASE w kontekście wybranych atrybutów modelu jakości

W celu dokonania oceny wybranych narzędzi w zakresie możliwości modelowania konceptualnego w języku UML oraz możliwości automatycznej transformacji z zaproponowanego modelu jakości użytkowej do modelu fizycznego zostały wybrane charakterystyki wraz z atrybutami. Wyniki eksperymentów znajdują się w kolejnych punktach artykułu.

### 3.2.1. Języki/notacje, za pomocą których można opisać byty wycinka rzeczywistości – UML

Stworzona została lista kontrolna elementów języka UML, które zostały uznane jako istotne przy modelowaniu, w kontekście przygotowanego modelu jakości i które można wykorzystać do modelowania na poziomie konceptualnym.

Tabela 1

Podstawowy zbiór elementów języka UML – implementacja w wybranych narzędziach typu CASE

	VP DB Visual Architect 6.1	Microsoft Visual Studio 2010	astah* professional 6.3
klasy (wraz ze stereotypami)	x	x	x
typy danych atrybutów klas	x	x	x
asocjacja	x	x	x
własności asocjacji (nazwa, kierunek, liczności końców asocjacji, nazwy ról)	x	x	x
agregacja słaba	x	x	x
kompozycja (agregacja silna)	x	x	x
generalizacja (dziedziczenie)	x	x	x
zbiór generalizacji	x	-	-
ograniczenia zbioru generalizacji: overlapping/disjoint	x	-	-
ograniczenia zbioru generalizacji: complete/incomplete	-	-	-
ograniczenie bag	x	-	-
ograniczenie ordered	x	x	-
ograniczenie sequence	-	-	-
klasy asocjacyjne	x	-	x
asocjacje n-arne	x	-	-
typ wyliczeniowy <<enum>>	x	x	x

x – możliwe do zamodelowania

-- brak

### 3.2.2. Generowanie modeli fizycznych w kontekście przyjętego zbioru elementów modelu konceptualnego zdefiniowanego w tabeli 1

Elementy listy kontrolnej, będące zbiorem podstawowym języka UML, wykorzystywane do specyfikacji modeli konceptualnych danych, były analizowane w kontekście możliwości ich automatycznej transformacji do modeli danych na poziomie fizycznym. Wyniki transformacji przeprowadzonych w wybranych narzędziach CASE zostały umieszczone w tabeli 2.

W trakcie transformacji elementów z poziomu konceptualnego danych do poziomu fizycznego zostały zauważone następujące odstępstwa od oczekiwanych rezultatów (numery odpowiadają odsyłaczom w tabeli 2.):

1. Klucz obcy tabeli agregowanej nie jest elementem klucza głównego, może być NULL.



2. Można wybrać sposób transformowania generalizacji – tabela tylko dla klasy rodzica, tylko dla klasy potomka albo tabele dla wszystkich klas w hierarchii.
3. *Disjoint, Bag, Ordered* nie są odzwierciedlone w modelu fizycznym.
4. Dla kluczy obcych w implementacji klasy asocjacyjnej nie ma ograniczenia UNIQUE.
5. Związki asocjacji n-arnej nie są transformowane.
6. Typ wyliczeniowy transformowany jako kolejna tabela, zły typ danych.
7. Visual Studio nie wspiera automatycznej transformacji modelu danych bez dodatkowych wtyczek/konfiguracji, dlatego transformacja za pomocą tego narzędzia zostanie sprawdzona w przyszłości, w ramach prac związanych z poruszaną tematyką. Narzędzie to nie zostanie również uwzględnione w kontekście możliwości generowania modeli fizycznych dla wybranych SZBD.
8. Nieprawidłowo przetransformowane licznosci końców asocjacji.
9. Oddzielne tabele dla wszystkich klas w hierarchii.

Zauważone odstępstwa mają wpływ na jakość transformacji i w związku z tym zostały uwzględnione w funkcji oceny.

Tabela 2

Wyniki transformacji przyjętego zbioru elementów modelu koncepcyjnego

	VP DB Visual Architect 6.1	Microsoft Visual Studio 2010	astah* professional 6.3
klasy (wraz ze stereotypami)	x	- <sup>7</sup>	x
typy danych atrybutów klas	x	- <sup>7</sup>	x
asocjacja	x	- <sup>7</sup>	x
własności asocjacji (nazwa, kierunek, licznosci końców asocjacji, nazwy ról)	x	- <sup>7</sup>	x* <sup>8</sup>
agregacja słaba	x	- <sup>7</sup>	x*
kompozycja (agregacja silna)	x* <sup>1</sup>	- <sup>7</sup>	x
generalizacja (dziedziczenie)	x <sup>2</sup>	- <sup>7</sup>	x <sup>9</sup>
zbiór generalizacji	x	- <sup>7</sup>	-
ograniczenia zbioru generalizacji: overlapping/disjoint	- <sup>3</sup>	- <sup>7</sup>	-
ograniczenia zbioru generalizacji: complete/incomplete	-	- <sup>7</sup>	-
ograniczenie bag	- <sup>3</sup>	- <sup>7</sup>	-
ograniczenie ordered	- <sup>3</sup>	- <sup>7</sup>	-
ograniczenie sequence	-	- <sup>7</sup>	-
klasy asocjacyjne	x* <sup>4</sup>	- <sup>7</sup>	x
asocjacje n-arne	- <sup>5</sup>	- <sup>7</sup>	-
typ wyliczeniowy <<enum>>	x* <sup>6</sup>	- <sup>7</sup>	x* <sup>6</sup>

x – transformacja zgodna z oczekiwaniami

x\* – transformacja mająca odstępstwa od oczekiwanych rezultatów

-- transformacja niepoprawna/brak transformacji

### 3.2.3. Obsługiwane docelowe SZBD modeli fizycznych

W tabeli 3. przedstawiono listę kontrolną wybranych docelowych środowisk bazodanowych, dostępnych w poszczególnych narzędziach.

Tabela 3

Możliwość wygenerowania modeli fizycznych dla wybranych SZBD

	VP DB Visual Architect 6.1	astah* professional 6.3
Microsoft SQL Server 2008	x	-
Oracle 11g	x	-
PostgreSQL 9.0.3	x	-
MySQL 5.5.9	x	-

x – możliwość wygenerowania modelu fizycznego dla środowiska

- – brak możliwości wygenerowania modelu fizycznego dla środowiska

### 3.3. Funkcja oceny

Przyjęto, że pomiar oparty na listach kontrolnych będzie wykonywany przez zliczanie pozycji implementowanych przez poszczególne narzędzia CASE. W przypadku poprawnej implementacji elementu w danym narzędziu (oznaczono znakiem x) pozycja brana jest z wartością 1, w przypadku częściowej implementacji (oznaczono znakiem x\*) przyjmowana jest wartość 0,5, a w przypadku braku implementacji elementu (oznaczono znakiem -) wartość 0. Zatem, liczba wszystkich pozycji listy kontrolnej z uwzględnieniem wag staje się miarą dla atrybutu z modelu jakości.

Definiując funkcję jakości brano pod uwagę zdefiniowany zbiór celów użycia narzędzi CASE:

- Kluczowym artefaktem w procesie projektowania baz danych jest model konceptualny, bowiem jego spójność z modelowaną dziedziną decyduje o użyteczności bazy danych, uzyskanej w wyniku transformacji tego modelu.
- Istotną rolę odgrywają również możliwości transformacji elementów modelu konceptualnego, aczkolwiek w przypadku braku możliwości tę niedogodność można skorygować poprzez ręczną korektę modelu uzyskanego na poziomie fizycznym.
- Brak reguł transformacji modeli konceptualnych do określonych środowisk bazodanowych nie przekreśla użyteczności narzędzia, bowiem w praktyce z reguły wykorzystujemy jedno, ewentualnie dwa systemy zarządzania bazami danych. W przypadku braku wymaganego systemu zarządzania bazami danych narzędzie będzie wykorzystywane jedynie w początkowej fazie projektowania bazy danych (co nie przekreśla użyteczności narzędzia typu CASE).

Biorąc pod uwagę wyżej wymienione kryteria, funkcja oceny (QF) zostanie zdefiniowana w następujący sposób:

$$QF = 0,6 * m_1 + 0,3 * m_2 + 0,1 * m_3, \quad (1)$$

gdzie  $m_1$ ,  $m_2$ ,  $m_3$  to odpowiednio miary atrybutów, obliczone na podstawie trzech list kontrolnych. Przy czym im większa wartość funkcji oceny, tym dane narzędzie jest lepsze w kontekście zaproponowanego modelu jakości. W tabeli 4. zostały zamieszczone obliczone wartości funkcji oceny dla porównywanych narzędzi CASE.

Tabela 4

Obliczone wartości funkcji oceny dla poszczególnych narzędzi CASE

Nazwa narzędzia CASE	Wartość funkcji oceny QF
Visual Paradigm DB Visual Architect 6.1	22,5
Microsoft Visual Studio 2010 Ultimate	9 <sup>1</sup>
astah* professional 6.3	16,5

Na podstawie obliczonych wartości funkcji oceny można bezpośrednio porównać wybrane narzędzia CASE, w kontekście wybranych charakterystyk modelu jakości. Jednoznacznie można stwierdzić, które z ocenianych narzędzi reprezentuje najwyższą jakość w wybranym kontekście.

#### 4. Podsumowanie

W artykule został przeanalizowany model jakości użytkowej, zaproponowany w normie ISO/IEC 25010 i uznany za model bazowy do opracowania modelu jakości, do oceny narzędzi typu CASE. W przyjętym modelu jakości użytkowej zostały wybrane charakterystyki, które określają perspektywę oceny, uwzględniającą możliwość definiowania modeli konceptualnych z użyciem języka UML, transformację modeli konceptualnych do poziomu fizycznego oraz implementację tych modeli w wybranych systemach zarządzania bazami danych.

Zaproponowany model należy traktować jako propozycję służącą dyskusji nad potrzebą i możliwością oceny jakościowej narzędzi typu CASE. Doświadczenia z przeprowadzonych eksperymentów na wybranej grupie użytkowników (w początkowej fazie prac - trzech projektantów systemów informatycznych wykorzystujących różne narzędzia) mogą być podstawą potwierdzenia celowości dalszych prac nad zaproponowanym modelem, tzn.:

- zaproponowania atrybutów oraz miar jakości dla wszystkich charakterystyk,

---

<sup>1</sup> Należy zauważyć, że ocena globalna narzędzia Visual Studio 2010 poprawi się w przypadku uwzględnienia przy ocenie wtyczek/dodatków, umożliwiających transformację modelu konceptualnego do poziomu fizycznego, co zostanie wykonane w dalszym etapie prac, związanych z niniejszą tematyką.

- zdefiniowania funkcji oceny, biorących pod uwagę większą liczbę charakterystyk (funkcja oceny (zaproponowana w artykule) oparta jest tylko na trzech podstawowych atrybutach),
- opracowania systemu wspierającego ocenę jakościową narzędzi typu CASE, umożliwiającego definiowanie własnych funkcji oceny oraz wybór podzbioru charakterystyk z modelu jakości,
- przeprowadzenia kolejnych eksperymentów na reprezentatywnej grupie użytkowników narzędzi typu CASE.

## BIBLIOGRAFIA

1. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. International Organization for Standardization/International Electrotechnical Commission, 2011.
2. Dubielewicz I., Hnatkowska B., Huzar Z., Tuzinkiewicz L.: Metodyka QUAD, Sterowane jakością wytwarzanie aplikacji bazodanowych. Oficyna wydawnicza Politechniki Wrocławskiej, Wrocław 2010.
3. astah\* professional 6.3, <http://astah.change-vision.com/en/product/astah-professional.html>.
4. Visual Paradigm DB Visual Architect 6.1, <http://www.visual-paradigm.com/product/dbva/>.
5. Microsoft Visual Studio 2010 Ultimate, <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/ultimate>.
6. Sybase PowerDesigner 15.3, <http://www.sybase.com/products/modelingdevelopment/powerdesigner>.
7. IBM InfoSphere Data Architect 7.5.3, <http://www-142.ibm.com/software/products/pl/pl-ibminfodataarch/>.
8. Sparx Enterprise Architect 8.0, <http://www.sparxsystems.com/products/index.html>.

Recenzenci: Dr inż. Piotr Bajerski  
Dr inż. Dariusz Mrozek

Wpłynęło do Redakcji 31 stycznia 2011 r.

**Abstract**

This paper covers the problem of comparing available CASE tools supporting database design process in terms of transformation capabilities between data models on different levels of abstraction. Main goals connected with database design are presented in the first paragraph. These goals establish the context of assessment criteria of selected CASE tools. On the basis of ISO/IEC 25010 [1] Quality in use model the quality model for CASE tools assessment has been introduced. Characteristics contained in the quality model can be utilized to assess the quality of the software tools supporting database design. An example of the quality evaluation of popular, widely used CASE tools in the context of prepared quality model gives the idea on how to conduct the experiments. Conceptual data model used in the experiments considers typical features and dependencies existing in real IT projects. A quality function (1) that has been defined shows how to measure software tools quality in practice. The conclusions from conducted studies and experiments have been included in the last paragraph.

**Adresy**

Lech TUZINKIEWICZ: Politechnika Wrocławska, Instytut Informatyki, ul. Wybrzeże Wyspiańskiego 27, 50-327 Wrocław, Polska, lech.tuzinkiewicz@pwr.wroc.pl.

Bartosz PIECHOTA: Politechnika Wrocławska, Instytut Informatyki, ul. Wybrzeże Wyspiańskiego 27, 50-327 Wrocław, Polska, 157638@student.pwr.wroc.pl.