

BD2 – Bazy Danych 2

dr inż. Tomasz Traczyk

9. Tworzenie aplikacji w s.i. z bazami danych

Copyright © Tomasz Traczyk
Instytut Automatyki i Informatyki Stosowanej Politechniki Warszawskiej
Materiały dydaktyczne przeznaczone są wyłącznie do indywidualnego użytku studiujących.
Rozpowszechnianie kopii bez pisemnej zgody autora jest zabronione.

Typy modułów aplikacji

Formularze

- Interaktywne
- Do przeglądania i edycji danych
- Sposób prezentacji zwykle ustalony

Raporty

- Wsadowe
- Sposób prezentacji sztywny lub mało elastyczny
- Wynik zwykle przystosowany do wydruku

Moduły analityczne

- Do interaktywnej analizy danych
- Sposób prezentacji elastyczny w ramach pewnego modelu (np. analizy wielowymiarowej)

Programy przetwarzające

- Przetwarzające dane z bazy i umieszczające wyniki tamże

Interfejsy do innych systemów

- Moduły importujące i eksportujące dane
- Interfejsy bezpośrednie (bramki itp.)

Zapytania *ad-hoc*

- Tworzone przez użytkownika za pomocą odpowiednich narzędzi

Pożądane cechy aplikacji

Cechy podstawowe

- Skuteczność i niezawodność
- Przyjazność dla użytkownika
- Przenośność
 - architektura klient-serwer
 - * różne GUI
 - * środowisko znakowe
 - architektury wielowarstwowe
- Łatwość utrzymania
 - prostota
 - modularność
 - czytelny i ustandaryzowany kod
 - dobre narzędzia do modyfikacji

Dostosowanie do typu użytkownika

- Intensywność korzystania z aplikacji
 - sporadyczna: GUI, myszka, *help*
 - rutynowa: klawiatura, *hints*
- Rodzaj dostępu
 - wewnętrzorganizacyjny: możliwość instalacji oprogramowania, dobre łącza
 - publiczny: przeglądarka WWW, protokół HTTP, typowe formaty informacji (HTML, XML, PDF)

Pożądane cechy aplikacji

Modularność

- Niezależność modułów
 - unikanie wspólnych elementów i ścisłych powiązań
 - zależności tylko przez bazę danych
- Ograniczone ponowne użycie
 - tylko elementy niezmiennie
 - bardzo duży wpływ na koszty testów regresyjnych

Prostota

- Moduły możliwie proste i zrozumiałe
- Ograniczony zakres modułów
 - zakres każdego modułu dobrze zdefiniowany
 - unikanie łączenia słabo związanych funkcji
- Ważne kryterium: łatwość utrzymania

Zalecenie projektowe 9.1 *Aplikacje powinny składać się z możliwie prostych, słabo wzajemnie uzależnionych modułów, komunikujących się ze sobą przez dane zapisane w bazie danych.*

Przyjazność aplikacji^a

GUI, myszka

- Spokojny „wystrój”
- Czytelne czcionki
- Możliwość 100% obsługi bez użycia myszki
- Możliwość obsługi myszką

Klarowny interfejs

- Jednakowe konwencje w całym systemie
- Niewielka liczba okien (może tylko jedno)
- Unikanie wyskakujących okienek
- Przejrzyste zasady nawigacji

Podpowiedzi

- Etykiety (*prompts, labels*): krótkie
- Podpowiedzi *hints*: jednozdaniowe, w linii komunikatów
- „Dymki” (*tooltips*): jednozdaniowe, dobre do przycisków
- Pomoc kontekstowa (*help*) – na poziomie
 - pola (kolumny),
 - bloku (tabeli),
 - modułu

Zalecenie projektowe 9.2 *Aplikacje dla użytkownika profesjonalnego powinny być bardziej efektywne niż efektowne. Decydujące znaczenie ma ergonomia i jednolitość konwencji.*

^aZ punktu widzenia użytkownika profesjonalnego

Typowy szablon formularza

Master-detail

- Część nadrzędna – *master*
 - układ formularzowy
- Część podrzędna – *detail*
 - układ tabelaryczny
 - automatyczna synchronizacja z częścią *master*
- Odpowiada strukturze typowego dokumentu (zamówienie, faktura itp.)

Przykład

The screenshot shows a window titled 'Publikacje' with a form layout. The top section contains input fields for ISBN (234567), Kod typu publikacji (MGR), Nazwa typu (Praca magisterska), Tytuł (Wybrane aspekty ogólnej wiedzy o Wszechświecie), Liczba stron (68), and Rok wydania (2008). Below this is a table titled 'Cechy publikacji' with columns for 'Kod cechy', 'Nazwa cechy', and 'Wartość'. The table contains the following data:

Kod cechy	Nazwa cechy	Wartość
KIER	Kierownik pracy	Dr Józef Wiśniewski
REC	Recenzent	prof. Adam Abacki
OCENA	Ocena pracy	4,0

Typowe elementy formularzy

Elementy sterujące

- Konieczne jednolite przypisanie zadań, np.
 - menu: nawigacja między modułami
 - przyciski: wywoływanie funkcji
 - * wspólnych dla wielu modułów – *toolbar*
 - * specyficznych dla danego kontekstu – przyciski na formularzu

Wyszukiwanie danych

- Przeglądarki z przejściami do szczegółów
- „Wyskakujące” listy do nawigacji
- Wyszukanie wg stałego warunku
- QBF (*Query By Form*)

Podstawianie kluczy obcych

- Za pomocą listy wyboru
- Z użyciem deskryptora
- Z wyświetleniem deskryptora na formularzu obok/zamiast wartości

Listy wyboru

- Krótkie i niezmiennie: przyciski radiowe, *checkboxes*
- Krótkie: *pop-lists*
- Długie: listy z wyszukiwaniem
 - dopasowanie do wzorca
 - autoredukcja

Zalecenie projektowe 9.3 Interfejsy użytkownika projektować należy w sposób logiczny i jednolity, zgodnie z typowymi rozwiązaniami przyjętymi w danym środowisku systemowym oraz uwzględniając intencje twórców wykorzystywanych narzędzi.

Typowe szablony raportów

Master-detail

- Nagłówki grup i tabelki

Master-detail

Tytuł Matematyczno-etyczne aspekty teorii metanaukowych

Kod typu DR

Kod cechy	Nazwa cechy	Wartość
PRO	Promotor	dr hab. Bohdan Babacki
REC	Recenzent	prof. Czesław Cabacki
REC2	Drugi recenzent	prof. Adam Abacki

Tytuł Wprowadzenie to teorii metanauki

Kod typu MONO

Kod cechy	Nazwa cechy	Wartość
REC	Recenzent	dr hab. Jan Kowalski

Tytuł Wstęp do mniemanologii stosowanej

Kod typu PODR

Kod cechy	Nazwa cechy	Wartość
REC	Recenzent	dr hab. Jan Kowalski

Z wcięciami

- Grupy oznaczone przez wcięcia

Raport z wcięciami

Kod typu	Licznik:	Tytuł
DR	1	Matematyczno-etyczne aspekty teorii metanaukowych
MGR	1	Wybrane aspekty ogólnej wiedzy o Wszeczświecie
MONO	2	Wstęp do mniemanologii stosowanej
		Wprowadzenie to teorii metanauki
Licznik:	4	

Macierzowy (zestawienie krzyżowe)

- Pokazuje zależność od dwóch wymiarów

Raport macierzowy

Tytuł	Cecha	Drugi recenzent	Promotor	Recenzent
	Wartość	Wartość	Wartość	Wartość
Matematyczno-etyczne aspekty teorii metanaukowych	prof. Adam Abacki	dr hab. Bohdan Babacki	prof. Czesław Cabacki	
Wprowadzenie to teorii metanauki			dr hab. Jan Kowalski	
Wstęp do mniemanologii stosowanej			dr hab. Jan Kowalski	
Wybrane aspekty ogólnej wiedzy o Wszeczświecie			prof. Adam Abacki	

Rozwiązania niefortunne

- Rozdzielenie wprowadzania nowych danych od modyfikacji istniejących
- Rozdzielenie przeglądania szczegółów od ich edycji
- Uniemożliwienie modyfikacji świeżo wprowadzonych danych (!)
- Liczne „wyskakujące” okna
- Za duże formularze, nie mieszczące się na ekranie
- Za bardzo „napakowane” ekrany
- Moduły obejmujące funkcje słabo zależne
- Raporty tylko do oglądania (nie nadające się do wydrukowania)
- Strony WWW zaśmiecone grafiką

Typy narzędzi do tworzenia aplikacji

Tradycyjne programowanie 3GL

- Języki: C, C++, Cobol, Java itp.
- Cechy
 - niemal nieograniczone możliwości
 - trudne – bardzo mała wydajność
 - programy bardzo trudne do utrzymania

RAD (*Rapid Application Development*)

- Narzędzia: Visual Basic, Delphi, JBuilder itp.
- Cechy
 - duże możliwości
 - znacznie większa wydajność
 - dość trudne w użyciu – typowe działania wymagają programowania
 - programy dość trudne do utrzymania – duża swoboda programisty

Typy narzędzi do tworzenia aplikacji

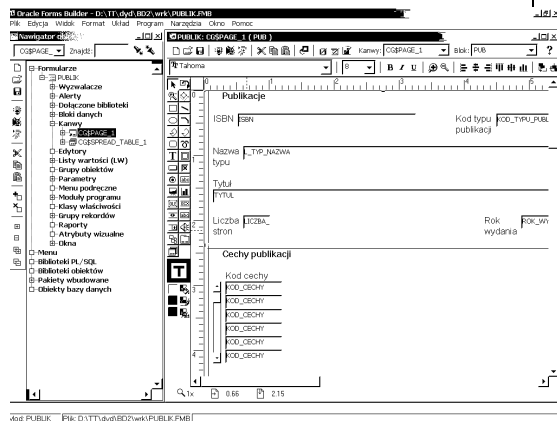
Specjalizowane narzędzia 4GL

- Cechy 4GL dla aplikacji z bazami danych
 - programowanie zdarzeniowe (*event-driven*)
 - programowanie wizualne
 - zintegrowane graficzne środowisko projektowo-programistyczne
 - konstrukcje wysokiego poziomu do realizacji typowych zadań, np. interakcji z użytkownikiem i bazą danych
 - silne wykorzystanie możliwości DBMS
 - typowe funkcjonalności aplikacji wbudowane w narzędzie
- Narzędzia 4GL: Oracle Forms+Reports, Powerbuilder itp.
- Cechy użycia 4GL
 - duża wydajność programisty
 - łatwe w użyciu – typowe funkcje realizowane domyślnie
 - łatwe utrzymanie – mało

programowania, dość mała swoboda programisty

- trudne uzyskanie nietypowego działania

Przykład Narzędzie 4GL – Oracle Forms

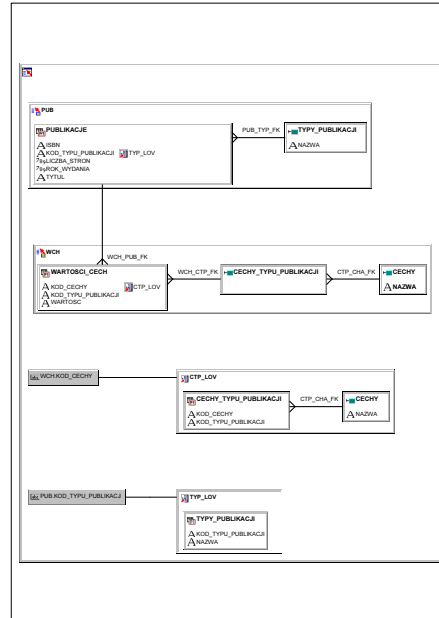


Typy narzędzi do tworzenia aplikacji

Generatory aplikacji

- Generowanie aplikacji na podstawie specyfikacji
- Narzędzia: systemy CASE, np. Oracle Designer
- Aplikacja generowana w 3GL lub 4GL, niekiedy wymaga niewielkich poprawek
- Cechy
 - bardzo duża wydajność pracy
 - powtarzalność wyników
 - stosunkowo łatwe utrzymanie
 - bardzo trudne uzyskanie nietypowego działania
 - narzędzia obszerne i trudne do opanowania

Przykład Projekt modułu formularza – Oracle Designer



Narzędzia 4GL – formularze

Zasady działania

- Graficzne projektowanie formularza
 - bloki dla tabel
 - pola dla kolumn
- Możliwość budowy formularzy *master-detail*
- Domyślna obsługa typowych akcji
- Programowanie
 - pełni rolę uzupełniającą – potrzebne tylko dla niestandardowych operacji
 - oparte na zdarzeniach
 - w specjalnym języku (PL/SQL, Access Basic itp.)

Postać programów

- Zapis
 - najczęściej binarny (grafika i elementy tekstowe razem)
 - nieliczne narzędzia dają postać tekstową (Progress 4GL)
- Wykonanie
 - zwykle wymaga *runtime* na kliencie lub odpowiedniego kontenera na serwerze aplikacyjnym
 - najczęściej istnieje postać pośrednia (półskompilowana)

Narzędzia 4GL – raporty

Typowa zasada działania

- Raport jest oparty o pojedyncze zapytanie (zwykle ze złączeniami)
- Grupy łąiące
 - grupa: kolejne wiersze z jednakową wartością wyróżnionych kolumn
 - wyróżnione na wydruku przez osobne nagłówki lub przez wcięcie
- Części raportu
 - nagłówek i stopka raportu
 - nagłówek i stopka strony
 - nagłówki i stopki grup łąiących
 - ciało
- Projektowanie raportu
 - definicja zapytania i grup łąiących
 - graficzne projektowanie wyglądu

Postać i wykonanie programów

- Podobnie jak dla formularzy

Postać wyników

- Wydruk na drukarkę
- Wyprowadzanie do pliku: tekstowego, Postscript, RTF, HTML, PDF
- Strony WWW serwowane na bieżąco: HTML, XML, PDF

Podejścia do tworzenia aplikacji

- Metodyki sformalizowane
 - projektowanie na podstawie wyników analizy
 - programowanie na podstawie specyfikacji
- Prototypowanie wytwórcze
- *Extreme programming* itp. „nowoczesne” podejścia

Zalecenie projektowe 9.4 *Prototypowanie może być bezpiecznie użyte tylko w projektach małych, prostych i dotyczących dziedziny dobrze znanej projektantowi.*

Zalecenie projektowe 9.5^a *Niesformalizowane sposoby programowania nie powinny być wykorzystywane do tworzenia systemów informacyjnych z bazami danych.*

Zalecenie projektowe 9.6 *Należy wybierać metodyki i technologie z możliwie dużym udziałem generowania aplikacji na podstawie specyfikacji i deklaratywnego określania jej działania, a z małym udziałem ręcznego programowania.*

^aKonserwatywne

Narzędzia do zapytań *ad hoc*

Konsole SQL

- Funkcje: przekazywanie zdań SQL do serwera i wyświetlanie wyników
- Przykłady: SQL*Plus, TORA

Narzędzia analityczne OLAP

- Funkcje: budowanie analiz wielowymiarowych
- Zwykle używają *business layer* obrazującego dane w sposób zrozumiały dla analityka biznesowego
- Przykłady: Business Objects, Oracle Discoverer

QBE (*Query By Example*)

- Odmiany
 - narzędzia do składania zapytań bez znajomości języka SQL
 - QBF (*Query By Form*) – do zapytań na formularzach

QBE (*Query By Example*)

Zasada działania

- System wyświetla pustą tabelkę-formularz, odpowiadającą strukturze wybranych tabel
- Użytkownik wpisuje wartości „wzorcowe”
 - stałe
 - warunki z użyciem operatorów, stałych i tzw. łączników
- Złączenia określa się przez
 - automatyczne wczytanie więzów *foreign key*
 - użycie łączników dla kolumn-kluczy
 - graficzne określenie powiązań
- System wypełnia formularz wierszami pasującymi do wzorca

Przykład Narzędzie QBE

