

Narzędzia i metody automatyzacji inżynierii oprogramowania

dr hab. inż. Michał Śmiałek, prof. uczelni
dr inż. Kamil Rybiński



**Wydział
Elektryczny**
POLITECHNIKA WARSZAWSKA

Inżynieria oprogramowania



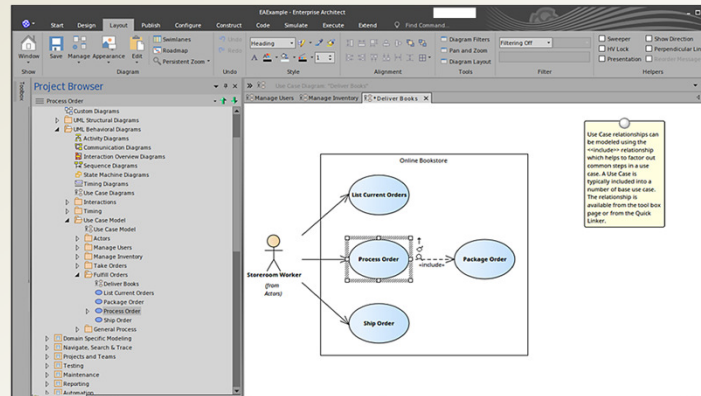
1

Narzędzia CASE

- Konieczność zarządzania wieloma produktami pracy
 - Dokumentacja współpracy z klientem
 - Modele, projekty, kod, testy
 - Dokumentacja techniczna, podręczniki
 - Wersje produktów, ...
- Komputerowe wsparcie inżynierii oprogramowania
 - Zarządzanie projektami informatycznymi
 - Zarządzanie wymaganiami i zmianami
 - Modelowanie systemów
 - Implementacja i zarządzanie konfiguracją systemu

2

Narzędzia wspierające modelowanie obiektowe



- Wsparcie dla tworzenia i zarządzania modelami (nie tylko obiektowymi)
 - UML, BPMN, ERD, ...

3

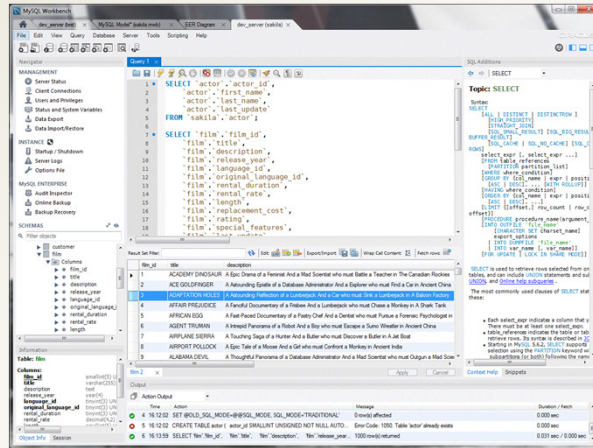
Cechy narzędzi do modelowania

- Łatwe tworzenie, przeglądanie i modyfikacja diagramów
 - Drzewo modelu, przyborniki z elementami, ...
- Pilnowanie poprawności notacji i spójności modeli
 - Zapobiega tworzeniu nieprawidłowych modeli
 - Umożliwia śledzenie relacji między modelami
- Generowanie kodu
 - Realizacja inżynierii okrężnej
- Generowanie dokumentacji
 - Automatyczne tworzenie dokumentów HTML, RTF, ...
- Możliwość wymiany modeli między narzędziami
 - Standardowy format wymiany danych (XMI)

4

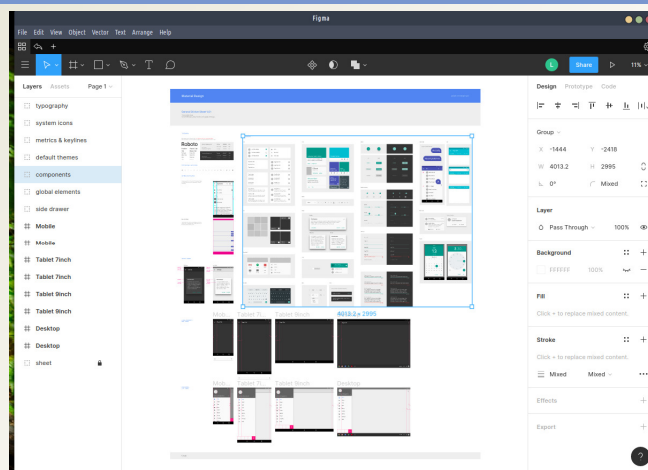
Narzędzia do zarządzania bazami danych

- Współpraca z narzędziami modelowania
- Przeglądanie struktury i zawartości BD
- Wydawanie poleceń do administracji BD
- Uruchamianie poleceń BD, odpuszkwanie
- Łatwe tworzenie zapytań



5

Narzędzia do modelowania interfejsu użytkownika



- Tworzenie „aktywnych” szkiców interfejsu użytkownika

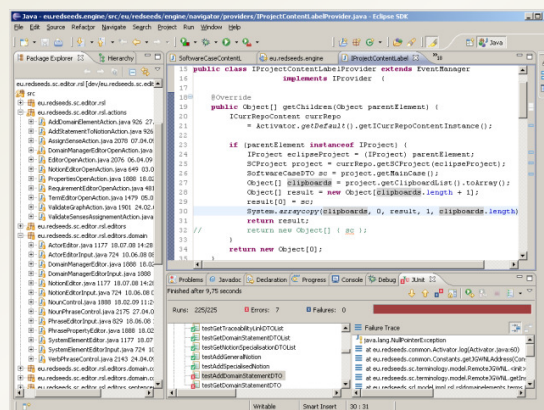
6

Cechy narzędzi do modelowania IU

- Cel: szybkie omówienie i zatwierdzenie wyglądu aplikacji
- Tworzenie interfejsów szkieletowych
 - Zastąpienie kartki i ołówka
 - Schematyczne rozmieszczenie elementów IU
 - Koncentracja na treści IU niż na jego wyglądzie
- Tworzenie makiet
 - Przedstawienie docelowego wyglądu IO
 - Dopracowanie szczegółów graficznych
 - Możliwa symulacja nawigacji między ekranami
- Generowanie kodu
 - Elementy kodu definiujące strukturę i wygląd IU

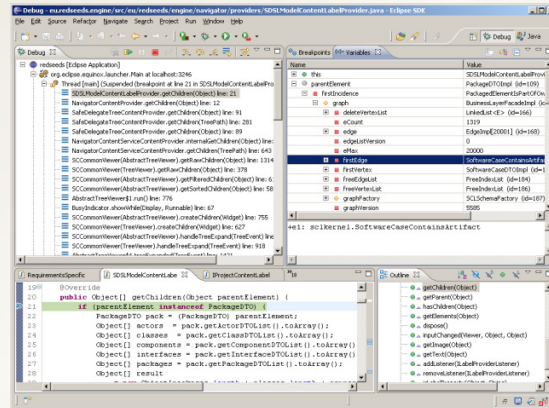
Zintegrowane środowiska programistyczne (IDE)

- Edycja kodu
 - Łatwe przeglądanie kodu
 - Kolorowanie kodu
 - Refaktoryzacja
- Kompilacja kodu
 - Szybkie wykrywanie i poprawianie błędów składni
 - Generowanie kodu wykonywalnego
 - Integracja z innymi narzędziami

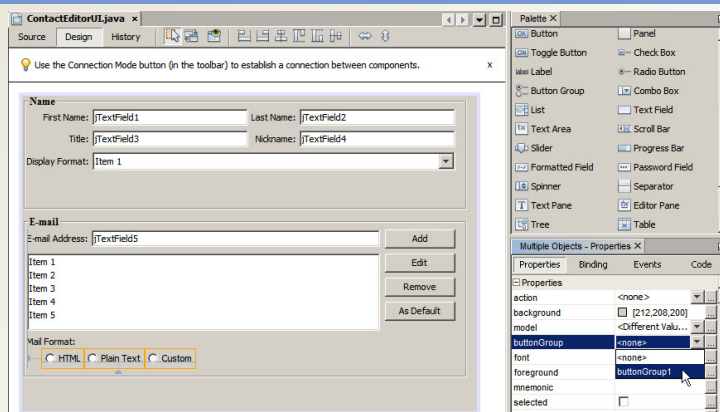


Narzędzia do odpluskwiania kodu

- Analiza działania programów
- Zatrzymywanie działania kodu
 - Podglądanie stanu zmiennych
- Wykonywanie kodu krokowo



Prototypowanie aplikacji



- Edytor IU zintegrowany ze środowiskiem IDE
- Szybkie prototypowanie aplikacji (RAD)

Narzędzia ciągłej integracji i wdrażania

- Automatyczne tworzenie działających wersji systemu
- Automatyczne instalowanie wersji w środowisku wykonawczym
- Automatyczne wykonywanie testów
- Integracja ze środowiskami IDE
 - „Jedno kliknięcie”, aby skompilować, zintegrować, zainstalować i uruchomić system

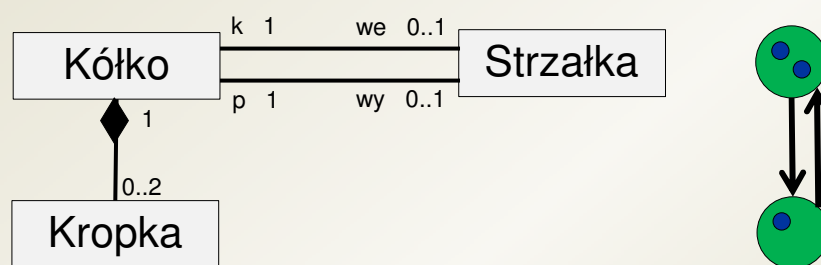
Narzędzia zarządzania zmianami i śledzenie błędów i problemów

- Narzędzia do zarządzania wersjami i zmianami
 - Realizacja procesu wersjonowania oraz obsługi zmian
 - Patrz poprzedni wykład
- Śledzenie błędów i problemów
 - Opisanie niepożądanego działania systemu i postulowanych zmian
 - Udostępnienie formularzy do zgłaszania problemów
 - Realizacja procesu obsługi problemów
 - Integracja z systemem wersjonowania

Narzędzia wspomagania testowania

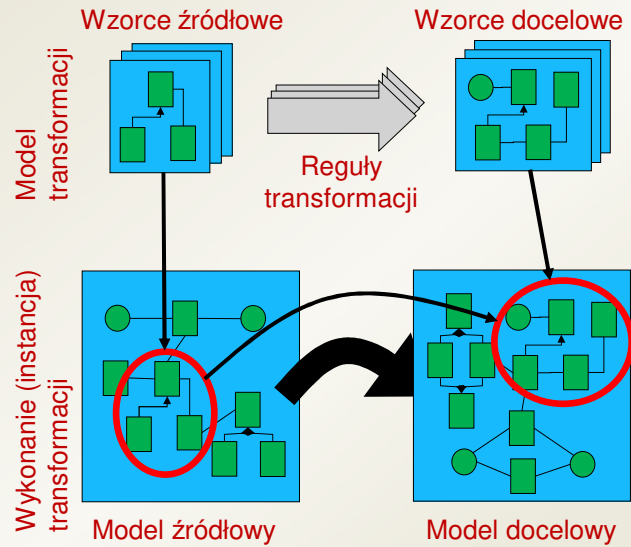
- Testowanie interfejsu użytkownika
 - Badanie poprawności struktury i działania IU
 - Zapisywanie i odtwarzanie typowych przebiegów przez IU
 - Możliwe łatwe wykonywanie testów regresyjnych
 - Zadawanie różnych zestawów danych wejściowych
- Testowanie jednostkowe
 - Generowanie szkieletów testów
 - Uruchamianie skryptów testowych
 - Analiza wyników testów i raportowanie
 - Łatwe definiowanie zaślepek oraz zestawów danych testowych
- Analiza kodu
 - Weryfikacja zgodności z przyjętymi standardami kodowani

Wytwarzanie oprogramowania sterowane modelami

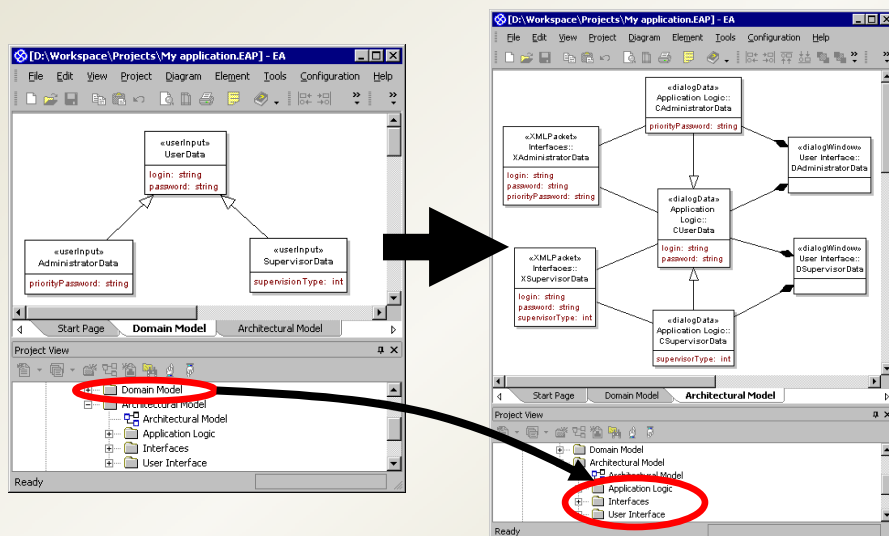


- Definiowanie języków modelowania oraz przekształceń między modelami
- Metamodel (składnia abstrakcyjna)
 - Definicja składni języka graficznego
 - Zazwyczaj wykonany w języku graficznym
 - Metamodels posiadają m.in. UML, BPMN, ERD, ...

Transformacja modeli



Transformacja modeli w narzędziach CASE



Metoda „Rozwój i Obsługa” (DevOps)



- Automatyzacja dyscyplin: implementacja, testowanie, wdrożenie, utrzymanie
 - Spójny zestaw narzędzi automatyzujących rutynowe czynności w cyklu życia kodu

Etapy DevOps (1)

- Planowanie
 - Narzędzia do zarządzania projektami software’owymi
 - Integracja z narzędziami do śledzenia błędów i problemów
- Kodowanie
 - Środowiska IDE
 - Centralne repozytorium kodu
 - Zarządzanie wersjami
 - Ścisła integracja z innymi narzędziami (IDE zarządza)
- Budowa
 - Narzędzia ciągłej integracji (budowa wydań)
 - Uruchamianie integracji bezpośrednio z IDE

Etapy DevOps (2)

- Testowanie
 - Automatyzacja testów jednostkowych
 - Automatyzacja testów akceptacyjnych
- Konfiguracja i wdrożenie
 - Narzędzia do ciągłej integracji i wdrożenia
 - Wykorzystanie środowisk do konteneryzacji
 - Automatyczne instalowanie kontenerów w środowisku wykonawczym
- Obsługa
 - Utrzymanie systemu w ruchu w środowisku wykonawczym (zazw. kontenerowym)
- Monitorowanie
 - Zbieranie i przekazywanie informacji na temat działania systemu i zauważonych problemów/rozszerzeń