

Paradygmaty Programowania

Marcin Bączyk

Wykład 10

24 maja 2021

Treść dzisiejszego wykładu

- style architektury oprogramowania
- wzorce architektoniczne

Architektura oprogramowania

Architektura oprogramowania składa się ze struktury systemu, w połączeniu z parametrami architektury, które system musi wspierać decyzjami architektonicznymi i zasadami projektowania

M. Richards, N.Ford

Struktura systemu

Struktura systemu to rodzaj (styl) architektury, w której system jest realizowany.

Parametry architektury

Parametry architektury określają kryteria sukcesu systemu, które są ortogonalne do funkcjonalności systemu.

Decyzje architektoniczne

Decyzje architektoniczne definiują reguły określające sposób, w jaki system powinien być konstruowany.

Zasady projektowania

Zasada projektowania jest wytyczną mówiącą o preferowanych rozwiązaniach stosowanych w danym systemie.

Wybrane style architektury oprogramowania

- „bryła błota„
- styl architektury warstwowej
- styl architektury mikrojądra
- styl architektury bazującej na usługach
- styl architektury sterowanej zdarzeniami

bryła błota

- Brak jakiegokolwiek zauważalnej struktury architektonicznej określa się mianem bryły błotnej.
- Systemy wykazują oznaki nieuregulowanego wzrostu i powtarzających się doraźnych napraw.
- Informacje przekazywane są bezładnie pomiędzy odległymi elementami systemu.
- Często wszystkie ważne informacje stają się globalne lub są duplikowane.
- Praktycznie każda zmiana w klasach systemu powoduje trudne do przewidzenia skutki uboczne dla systemu.
- „spagetti code”.

architektura warstwowa

- Składniki zorganizowane są w logiczne poziome warstwy, przy czym każda warstwa pełni określoną rolę w aplikacji.
- Składniki w obrębie konkretnej warstwy mają ograniczony zakres, zajmują się logiką, która odnosi się do konkretnej warstwy.
- Podstawowe warstwy to:
 - prezentacji
 - biznesowej
 - trwałości
 - bazodanowa

Architektura warstwowa jest architekturą podzieloną technicznie (w odróżnieniu od architektury podzielonej dziedzinowo).

architektura warstwowa

Warstwy izolacji mogą być:

- zamknięte lub
- otwarta.

Warstwa zamknięta oznacza, że gdy żądanie przechodzi z góry na dół, z warstwy na warstwę nie może ono pominąć żadnej z nich.

lej architektoniczny

W przypadku gdy proste żądania przechodzą przez kilka warstw bez dodatkowej logiki (np. z warstwy prezentacji do warstwy bazodanowej) może to powodować nieuzasadniony wzrost obciążenia poprzez tworzenie nowych instancji i przetwarzanie.

architektura warstwowa

- Jest dobrym rozwiązaniem dla małych prostych aplikacji lub witryn internetowych.
- Dobry punkt wyjścia gdy jeszcze nie jest wiadomym jaka architektura zostanie ostatecznie wybrana.
- Ze względu na prostotę i znajomość wśród programistów jest jedną z najtańszych architektur.
- Wraz ze wzrostem systemu pogarszają się takie parametry jak utrzymywalność, zwinność, testowalność i wdrażalność.
- Niska wdrażalność

architektura mikrojądra

- Architektura złożona z dwóch składników: podstawowego systemu i dołączanych wtyczek.
- Logika podzielona jest na niezależne dołączane składniki i podstawowy system.
- System podstawowy może mieć strukturę warstwową (podzielony technicznie) lub strukturę modułową (podzielony dziedzinowo).
- Dołączane składniki to samodzielne niezależne wtyczki zawierające wyspecjalizowane mechanizmy przetwarzania , dodatkowe funkcje lub niestandardowy kod.

architektura warstwowa

- Prosta architektura o niskim koszcie wytworzenia i utrzymania.
- Niska skalowalność , odporność na błędy i elastyczność.
- Dobrze pasuje do problemów, której wymagają różnych konfiguracji.
- Funkcjonalność może być odizolowana od niezależnych dołączanych składników.

architektura bazująca na usługach

- Uważana za jeden z najbardziej pragmatycznych stylów architektonicznych.
- Architektura rozproszona o relatywnie niskim poziomie złożoności i kosztach wytworzenia.
- Opiera się na rozproszonej makrowarstwowej strukturze składającej się z osobno wdrażanego:
 - interfejsu użytkownika
 - usług o większym stopniu ogólności
 - monolitycznej bazy danych
- Usługi są niezależnymi i wdrażanymi osobno częściami aplikacji.

architektura bazująca na usługach

- Struktura jest wyznaczona przez dziedziny (podział dziedzinowy) a nie względy techniczne (jak np. logika prezentacji czy biznesowa).
- Zmiany dokonywane w jednej z dziedzin mają wpływ tylko na jedną konkretną usługę.
- Wszystkie usługi mogą współdzielić ten sam interfejs użytkownika i tę samą bazę danych.
- Wysoka odporność na błędy.
- Zapewnia dostatecznie dobry poziom modułowości bez konieczności wnikania w złożoność i pułapki związane z rozdrobnieniem usług.

architektura sterowana zdarzeniami

- Wysoka wydajność uzyskiwana dzięki asynchronicznej komunikacji i wysokiemu stopniowi przetwarzania równoległego.
- Wysoka skalowalność uzyskiwana poprzez programowe równoważenie obciążenia procesorów zdarzeń.
- Odporność na błędy osiągnięta dzięki wysoce rozprężonym i asynchronicznym procesorom zdarzeń.
- Dodawanie nowych funkcji jest stosunkowo proste.
- Niska prostota i testowalność ze względu na niedeterministyczny i dynamiczny przepływ zdarzeń.

wzorzec architektoniczny

Uznany i sprawdzony sposób rozwiązania danego problemu z zakresu architektury oprogramowania.

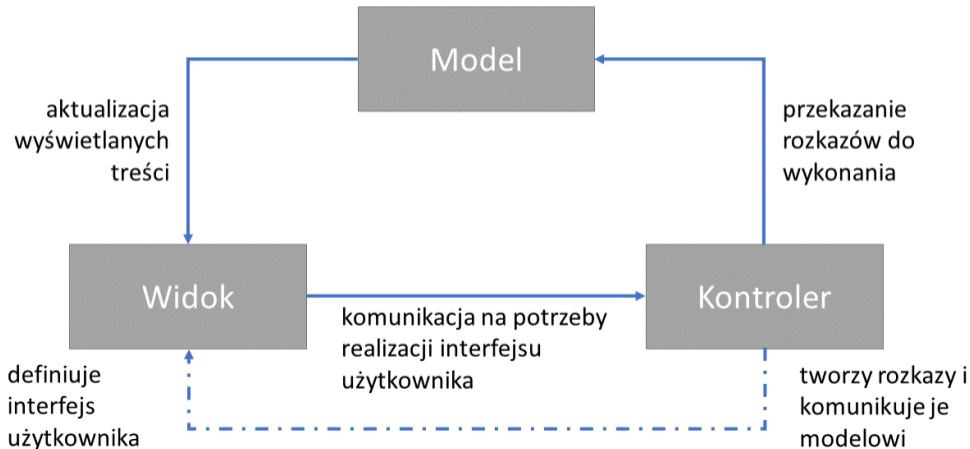
Wzorce architektoniczne, w przeciwieństwie do wzorców projektowych, dotyczą całego systemu informatycznego bądź jego modułów.

Przykłady wzorców architektonicznych

- Architektura trójwarstwowa
- Model-Widok-Kontroler
- Peer-to-peer
- Architektura zorientowana na usługi

definiuje między innymi:

- logikę biznesową aplikacji oraz
- strukturę danych



model - widok - kontroler

