

Architektura nowoczesnych aplikacji internetowych

Lech Madeyski Michał Stochmiałek

Wydziałowy Zakład Informatyki
Wydział Informatyki i Zarządzania
Politechnika Wrocławska

Krajowa Konferencja Inżynierii Oprogramowania 2004



Plan prezentacji

- 1 Wprowadzenie
 - Motto i cel prezentacji
 - Wyzwania związane z wyborem architektury
- 2 Wybrane podejścia do architektury
 - Triada MVC a aplikacje internetowe
 - Szkielet architektoniczny PCMEF a aplikacje internetowe
- 3 Propozycja szkieletu architektonicznego XWA
 - Propozycja szkieletu architektonicznego XWA
 - XWA a wyzwania architektoniczne
 - Architektura e-Informatyki
- 4 Podsumowanie

Plan prezentacji

- 1 Wprowadzenie
 - Motto i cel prezentacji
 - Wyzwania związane z wyborem architektury
- 2 Wybrane podejścia do architektury
 - Triada MVC a aplikacje internetowe
 - Szkielet architektoniczny PCMEF a aplikacje internetowe
- 3 Propozycja szkieletu architektonicznego XWA
 - Propozycja szkieletu architektonicznego XWA
 - XWA a wyzwania architektoniczne
 - Architektura e-Informatyki
- 4 Podsumowanie

Plan prezentacji

- 1 Wprowadzenie
 - Motto i cel prezentacji
 - Wyzwania związane z wyborem architektury
- 2 Wybrane podejścia do architektury
 - Triada MVC a aplikacje internetowe
 - Szkielet architektoniczny PCMEF a aplikacje internetowe
- 3 Propozycja szkieletu architektonicznego XWA
 - Propozycja szkieletu architektonicznego XWA
 - XWA a wyzwania architektoniczne
 - Architektura e-Informatyki
- 4 Podsumowanie

Plan prezentacji

- 1 Wprowadzenie
 - Motto i cel prezentacji
 - Wyzwania związane z wyborem architektury
- 2 Wybrane podejścia do architektury
 - Triada MVC a aplikacje internetowe
 - Szkielet architektoniczny PCMEF a aplikacje internetowe
- 3 Propozycja szkieletu architektonicznego XWA
 - Propozycja szkieletu architektonicznego XWA
 - XWA a wyzwania architektoniczne
 - Architektura e-Informatyki
- 4 Podsumowanie

Motto

Any technology distinguishable from magic is insufficiently advanced.

— Gregory Benford

so there is some space for improvement

Motto

Any technology distinguishable from magic is insufficiently advanced.

— Gregory Benford

so there is some space for improvement

Jaki jest cel prezentacji?

Przekonać Państwa do wykorzystania i wspólnego rozwijania szkieletu architektonicznego XWA, który (jak mamy zamiar pokazać) jest:

- dostosowany do specyfiki aplikacji internetowych
- odpowiada na trudne wyzwania architektoniczne stawiane nowoczesnym aplikacjom internetowym

Jaki jest cel prezentacji?

Przekonać Państwa do wykorzystania i wspólnego rozwijania szkieletu architektonicznego XWA, który (jak mamy zamiar pokazać) jest:

- dostosowany do specyfiki aplikacji internetowych
- odpowiada na trudne wyzwania architektoniczne stawiane nowoczesnym aplikacjom internetowym

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Niektóre wyzwania stojące przed wyborem architektury aplikacji internetowych

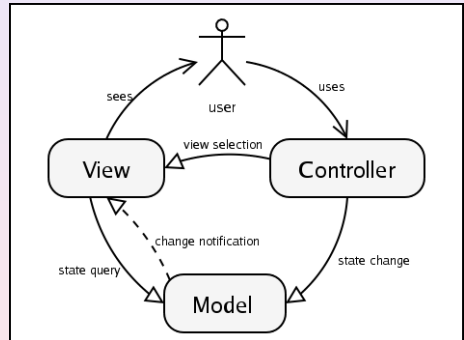
- łatwość pielęgnacji, utrzymywalność systemu
- możliwość obsługi szerokiego spektrum formatów publikacji
- integracja różnych źródeł danych i aplikacji
- umożliwienie efektywnej współpracy osób o różnych specjalizacjach (np. grafików, programistów, autorów treści)
- uwzględnienie specyfiki protokołu HTTP
- uproszczenie implementacji rozbudowanych interakcji użytkownika z systemem

Wybrane podejścia do architektury



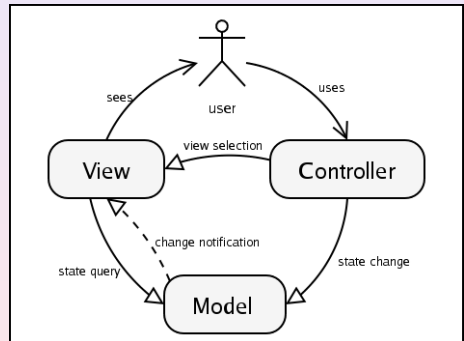
Triada MVC

- Model – rdzeń aplikacji
- Kontroler – wejście interfejsu użytkownika
- Widok – wyjście interfejsu użytkownika



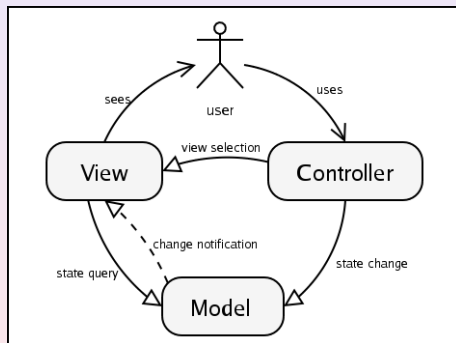
Model w szkielecie MVC

- określa działanie aplikacji oraz przetwarzane dane
- jest niezależny od interfejsu użytkownika
- *nic nie wie o widoku i kontrolerze*
- powiadamia widok o zmianach (*change notification*), luźne powiązanie z widokiem (*Obserwator, komunikaty*)



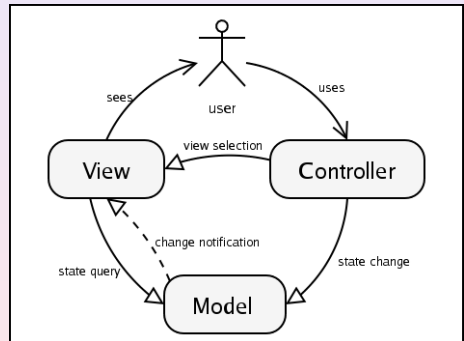
Model w szkielecie MVC

- określa działanie aplikacji oraz przetwarzane dane
- jest niezależny od interfejsu użytkownika
- *nic nie wie o widoku i kontrolerze*
- powiadamia widok o zmianach (*change notification*), luźne powiązanie z widokiem (*Obserwator, komunikaty*)



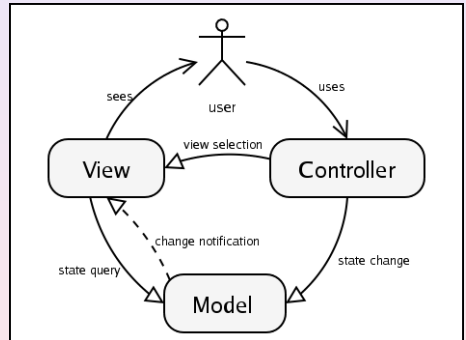
Model w szkielecie MVC

- określa działanie aplikacji oraz przetwarzane dane
- jest niezależny od interfejsu użytkownika
- *nic nie wie o widoku i kontrolerze*
- powiadamia widok o zmianach (*change notification*), luźne powiązanie z widokiem (*Obserwator, komunikaty*)



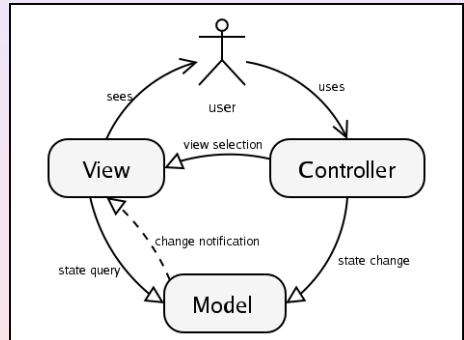
Widok w szkielecie MVC

- zarządza prezentacją modelu
- jest ściśle powiązany z konkretnym modelem
- musi *znać* specyfikę danych oraz operacji
- pobiera informacje z modelu (*state query*)



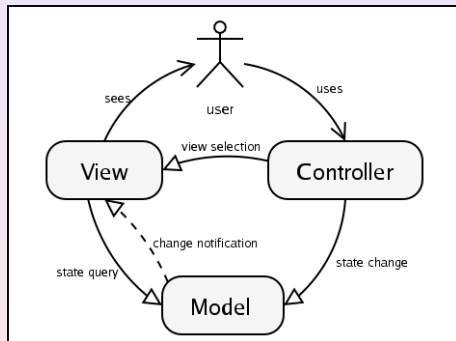
Widok w szkielecie MVC

- zarządza prezentacją modelu
- jest ściśle powiązany z konkretnym modelem
- musi *znać* specyfikę danych oraz operacji
- pobiera informacje z modelu (*state query*)



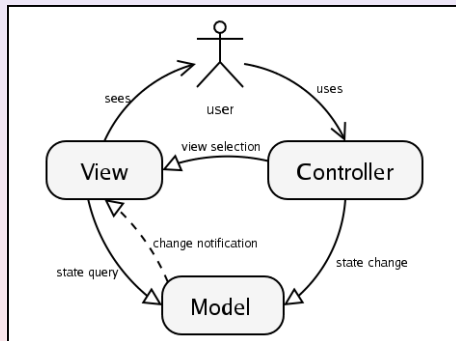
Widok w szkielecie MVC

- zarządza prezentacją modelu
- jest ściśle powiązany z konkretnym modelem
- musi *znać* specyfikę danych oraz operacji
- pobiera informacje z modelu (*state query*)



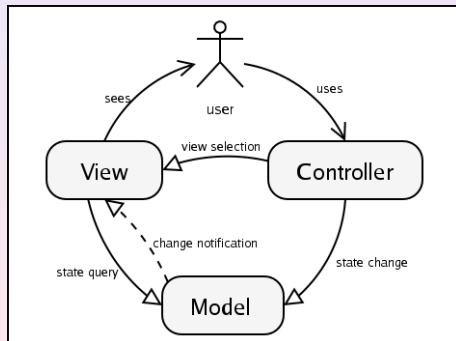
Kontroler w szkielecie MVC

- odpowiedzialny za reagowanie na akcje użytkownika
- odwzorowuje akcje użytkownika na:
 - operacje w modelu (*state change*)
 - zmiany w widoku (*view selection*)



Kontroler w szkielecie MVC

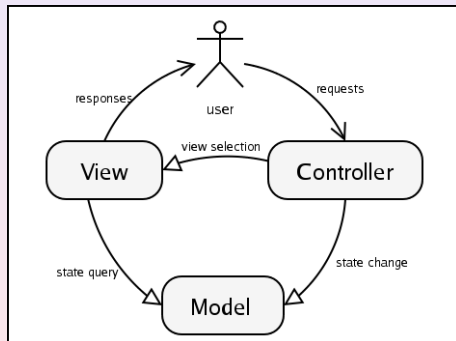
- odpowiedzialny za reagowanie na akcje użytkownika
- odwzorowuje akcje użytkownika na:
 - operacje w modelu (*state change*)
 - zmiany w widoku (*view selection*)



Szkielet MVC a protokół HTTP

- pasywność HTTP (*akcje są inicjowane przez użytkownika w cyklu żądanie-odpowiedź*) wymusza modyfikację MVC
- niemożliwe jest automatyczne uaktualnianie widoku przy zmianach modelu

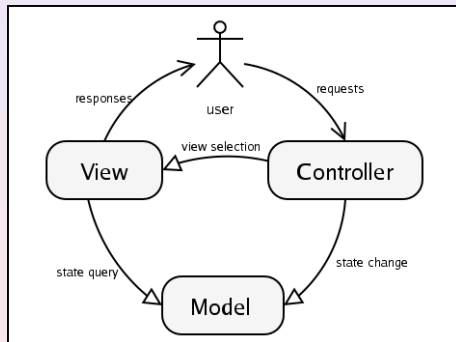
Nie jest to wada MVC, lecz dostosowanie do specyfiki HTTP.



Szkielet MVC a protokół HTTP

- pasywność HTTP (*akcje są inicjowane przez użytkownika w cyklu żądanie-odpowiedź*) wymusza modyfikację MVC
- niemożliwe jest automatyczne uaktualnianie widoku przy zmianach modelu

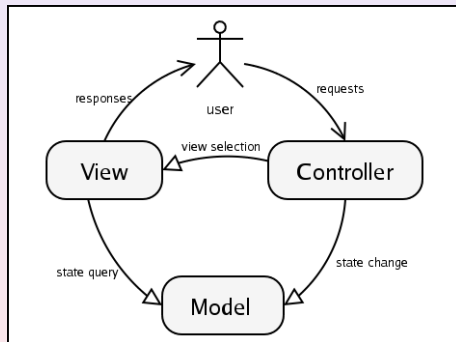
Nie jest to wada MVC, lecz dostosowanie do specyfiki HTTP.



Szkielet MVC a protokół HTTP

- pasywność HTTP (*akcje są inicjowane przez użytkownika w cyklu żądanie-odpowiedź*) wymusza modyfikację MVC
- niemożliwe jest automatyczne uaktualnianie widoku przy zmianach modelu

Nie jest to wada MVC, lecz dostosowanie do specyfiki HTTP.



Wartość szkieletu MVC

Wartość szkieletu MVC leży w dwóch zasadach:

- **separacja prezentacji i modelu** – umożliwia zmianę interfejsu użytkownika bez potrzeby modyfikacji modelu np. udostępnienie interfejsu klasycznego oraz internetowego
- **separacja widoku i kontrolera** – umożliwia zmianę kontrolera bez konieczności modyfikacji widoku np. użycie dwóch kontrolerów związanych z jednym widokiem, przy czym tylko jeden umożliwia edycję

Wartość szkieletu MVC

Wartość szkieletu MVC leży w dwóch zasadach:

- **separacja prezentacji i modelu** – umożliwia zmianę interfejsu użytkownika bez potrzeby modyfikacji modelu np. udostępnienie interfejsu klasycznego oraz internetowego
- **separacja widoku i kontrolera** – umożliwia zmianę kontrolera bez konieczności modyfikacji widoku np. użycie dwóch kontrolerów związanych z jednym widokiem, przy czym tylko jeden umożliwia edycję

Możliwe zagrożenia i konsekwencje

Zagrożenia:

- mylne rozumienie roli poszczególnych elementów triady
- złamanie zasad separacji widoku i kontrolera oraz prezentacji i modelu

Konsekwencje: rosnące problemy z pielęgnacją systemu

Przykład: kontroler bywa mylnie interpretowany jako element reprezentujący aspekt behawioralny (działanie aplikacji) podczas gdy model reprezentuje jedynie aspekt statyczny (dane)

Rezultatem jest brak możliwości zmiany warstwy prezentacji bez modyfikacji logiki zawartej w tym przypadku w kontrolerze.



Możliwe zagrożenia i konsekwencje

Zagrożenia:

- mylne rozumienie roli poszczególnych elementów triady
- złamanie zasad separacji widoku i kontrolera oraz prezentacji i modelu

Konsekwencje: rosnące problemy z pielęgnacją systemu

Przykład: kontroler bywa mylnie interpretowany jako element reprezentujący aspekt behawioralny (działanie aplikacji) podczas gdy model reprezentuje jedynie aspekt statyczny (dane)

Rezultatem jest brak możliwości zmiany warstwy prezentacji bez modyfikacji logiki zawartej w tym przypadku w kontrolerze.



Możliwe zagrożenia i konsekwencje

Zagrożenia:

- mylne rozumienie roli poszczególnych elementów triady
- złamanie zasad separacji widoku i kontrolera oraz prezentacji i modelu

Konsekwencje: rosnące problemy z pielęgnacją systemu

Przykład: kontroler bywa mylnie interpretowany jako element reprezentujący aspekt behawioralny (działanie aplikacji) podczas gdy model reprezentuje jedynie aspekt statyczny (dane)

Rezultatem jest brak możliwości zmiany warstwy prezentacji bez modyfikacji logiki zawartej w tym przypadku w kontrolerze.



Możliwe zagrożenia i konsekwencje

Zagrożenia:

- mylne rozumienie roli poszczególnych elementów triady
- złamanie zasad separacji widoku i kontrolera oraz prezentacji i modelu

Konsekwencje: rosnące problemy z pielęgnacją systemu

Przykład: kontroler bywa mylnie interpretowany jako element reprezentujący aspekt behawioralny (działanie aplikacji) podczas gdy model reprezentuje jedynie aspekt statyczny (dane)

Rezultatem jest brak możliwości zmiany warstwy prezentacji bez modyfikacji logiki zawartej w tym przypadku w kontrolerze.



Możliwe zagrożenia i konsekwencje

Zagrożenia:

- mylne rozumienie roli poszczególnych elementów triady
- złamanie zasad separacji widoku i kontrolera oraz prezentacji i modelu

Konsekwencje: rosnące problemy z pielęgnacją systemu

Przykład: kontroler bywa mylnie interpretowany jako element reprezentujący aspekt behawioralny (działanie aplikacji) podczas gdy model reprezentuje jedynie aspekt statyczny (dane)

Rezultatem jest brak możliwości zmiany warstwy prezentacji bez modyfikacji logiki zawartej w tym przypadku w kontrolerze.



W jakim stopniu MVC odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zaleta:** umożliwia klarowny podział odpowiedzialności pomiędzy osoby o różnych kompetencjach (graficy, programiści, projektanci stron WWW)
- **Wada:** brak wskazówek dotyczących zorganizowania modelu, a w szczególności:
 - rozbudowanej logiki biznesowej,
 - obsługi zewnętrznych źródeł danych czy usług sieciowych

W jakim stopniu MVC odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

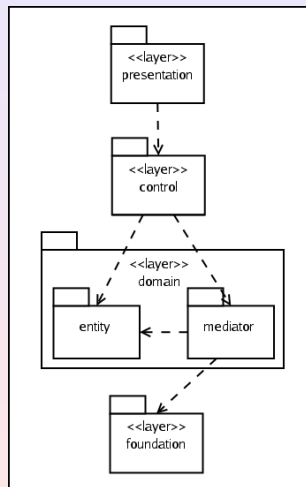
- **Zaleta:** umożliwia klarowny podział odpowiedzialności pomiędzy osoby o różnych kompetencjach (graficy, programiści, projektanci stron WWW)
- **Wada:** brak wskazówek dotyczących zorganizowania modelu, a w szczególności:
 - rozbudowanej logiki biznesowej,
 - obsługi zewnętrznych źródeł danych czy usług sieciowych

W jakim stopniu MVC odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zalety:** umożliwia klarowny podział odpowiedzialności pomiędzy osoby o różnych kompetencjach (graficy, programiści, projektanci stron WWW)
- **Wady:** brak wskazówek dotyczących zorganizowania modelu, a w szczególności:
 - rozbudowanej logiki biznesowej,
 - obsługi zewnętrznych źródeł danych czy usług sieciowych

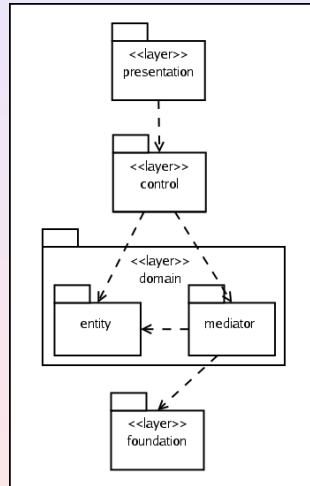
Szkielet architektoniczny PCMEF

- architektura warstwowa
- pakiety składowe:
 - presentation
 - control
 - mediator
 - entity
 - foundation
- źródło: Leszek Maciaszek,
Professional Software Engineering,
Addison-Wesley 2004



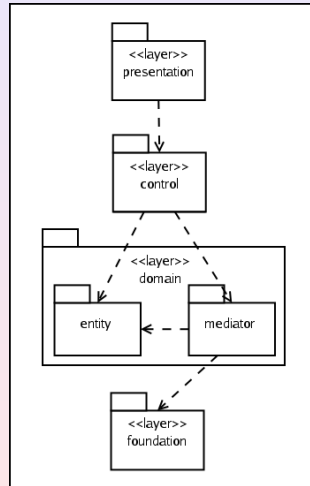
Pakiet presentation

- odpowiada za sposób prezentacji aplikacji oraz komunikację z użytkownikiem
- łączy w sobie Widok oraz Kontroler z MVC
- przykładem klas z pakietu presentation są klasy korzystające z biblioteki Java Swing



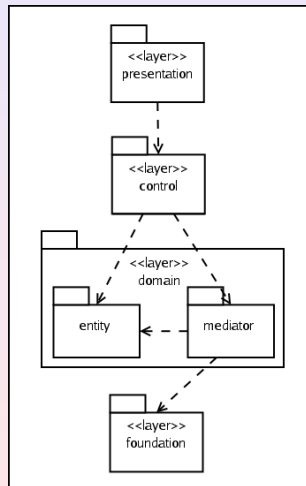
Pakiet presentation

- odpowiada za sposób prezentacji aplikacji oraz komunikację z użytkownikiem
- łączy w sobie Widok oraz Kontroler z MVC
- przykładem klas z pakietu presentation są klasy korzystające z biblioteki Java Swing



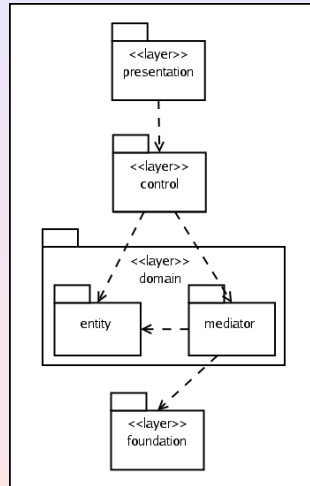
Pakiet presentation

- odpowiada za sposób prezentacji aplikacji oraz komunikację z użytkownikiem
- łączy w sobie Widok oraz Kontroler z MVC
- przykładem klas z pakietu presentation są klasy korzystające z biblioteki Java Swing



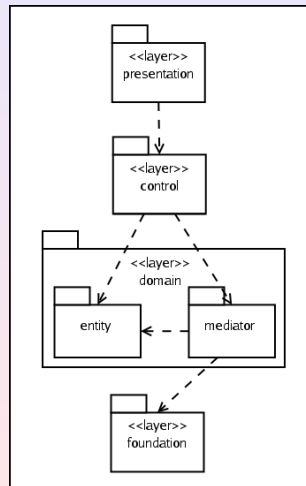
Pakiet control

- realizuje obsługę żądań użytkownika przekazywanych z warstwy presentation
- odpowiada za logikę programu, rozwiązania algorytmiczne, obliczenia oraz utrzymywanie stanu sesji użytkownika
- w MVC byłby elementem Modelu



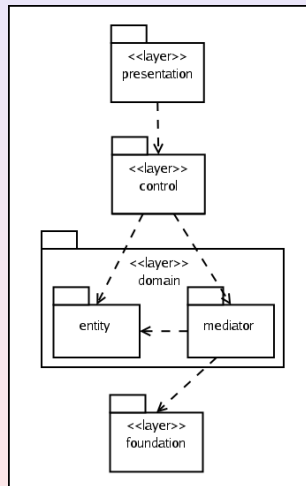
Pakiet control

- realizuje obsługę żądań użytkownika przekazywanych z warstwy presentation
- odpowiada za logikę programu, rozwiązania algorytmiczne, obliczenia oraz utrzymywanie stanu sesji użytkownika
- w MVC byłby elementem Modelu



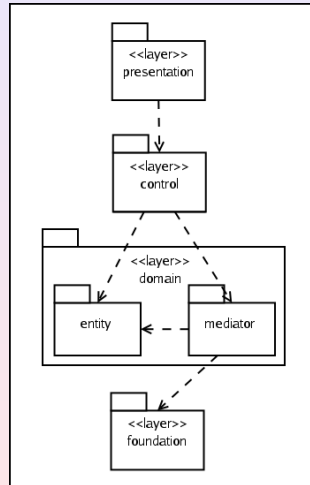
Pakiet control

- realizuje obsługę żądań użytkownika przekazywanych z warstwy presentation
- odpowiada za logikę programu, rozwiązania algorytmiczne, obliczenia oraz utrzymywanie stanu sesji użytkownika
- w MVC byłby elementem Modelu



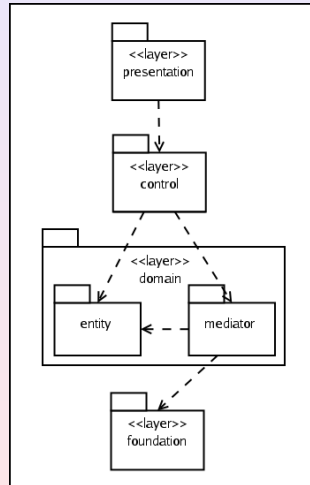
Pakiet entity

- zawiera obiekty biznesowe, które posiadają odzwierciedlenie w zewnętrznych źródłach danych
- w MVC byłby elementem Modelu



Pakiet entity

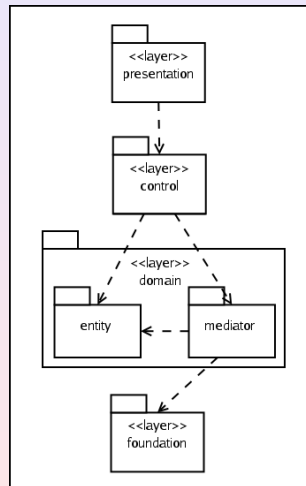
- zawiera obiekty biznesowe, które posiadają odzwierciedlenie w zewnętrznych źródłach danych
- w MVC byłby elementem Modelu



Pakiet mediator

Ma na celu usunięcie zależności:

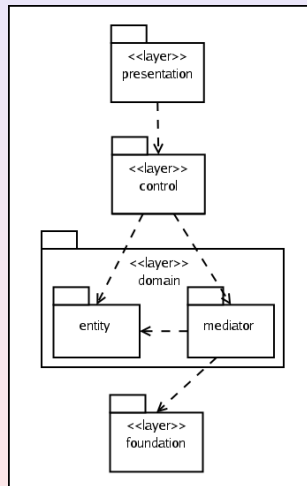
- pakietu `entity` od `foundation`
(likwiduje konieczność modyfikacji obiektów biznesowych przy zmianie mechanizmów trwałości danych)
- pakietu `control` od `foundation`
(pozwala na enkapsulację konstrukcji zapytań do zewnętrznych źródeł danych, oddzielając je od logiki aplikacji zawartej w warstwie `control`)
- w MVC byłby elementem Modelu



Pakiet mediator

Ma na celu usunięcie zależności:

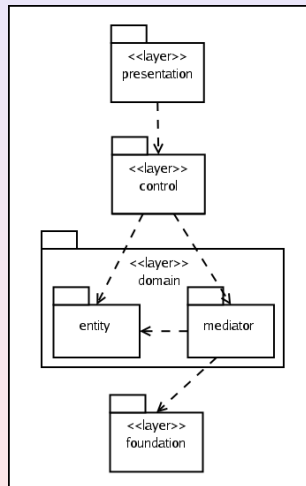
- pakietu entity od foundation
(likwiduje konieczność modyfikacji obiektów biznesowych przy zmianie mechanizmów trwałości danych)
- pakietu control od foundation
(pozwala na enkapsulację konstrukcji zapytań do zewnętrznych źródeł danych, oddzielając je od logiki aplikacji zawartej w warstwie control)
- w MVC byłby elementem Modelu



Pakiet mediator

Ma na celu usunięcie zależności:

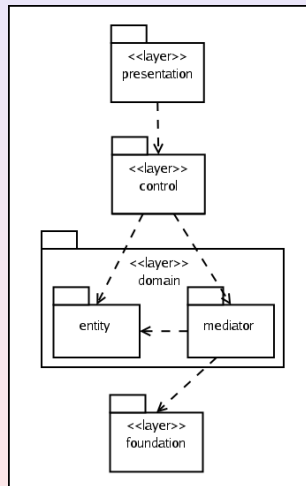
- pakietu entity od foundation
(likwiduje konieczność modyfikacji obiektów biznesowych przy zmianie mechanizmów trwałości danych)
- pakietu control od foundation
(pozwala na enkapsulację konstrukcji zapytań do zewnętrznych źródeł danych, oddzielając je od logiki aplikacji zawartej w warstwie control)
- w MVC byłby elementem Modelu



Pakiet foundation

Zawiera zazwyczaj gotowe implementacje odpowiadające za obsługę zewnętrznych źródeł danych:

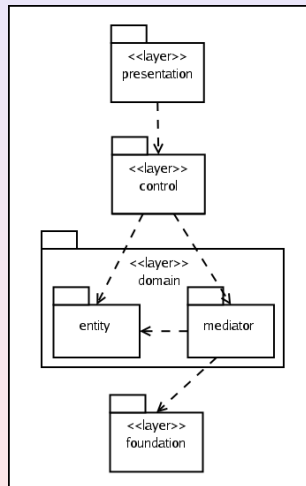
- baz danych
(np. Hibernate, JDBC, JDO)
- repozytoriów dokumentów
(np. klient WebDAV z Jakarta Slide)
- usług sieciowych (ang. *web services*)
(np. Apache Axis)
- systemów plików
(np. java.io z J2SE)



Pakiet foundation

Zawiera zazwyczaj gotowe implementacje odpowiadające za obsługę zewnętrznych źródeł danych:

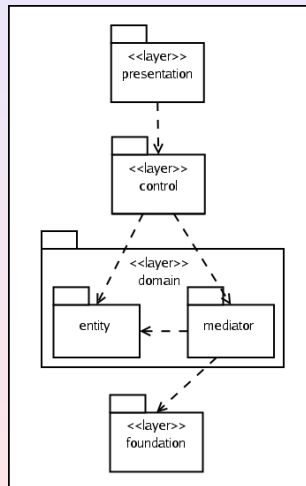
- baz danych
(np. Hibernate, JDBC, JDO)
- repozytoriów dokumentów
(np. klient WebDAV z Jakarta Slide)
- usług sieciowych (ang. *web services*)
(np. Apache Axis)
- systemów plików
(np. java.io z J2SE)



Pakiet foundation

Zawiera zazwyczaj gotowe implementacje odpowiadające za obsługę zewnętrznych źródeł danych:

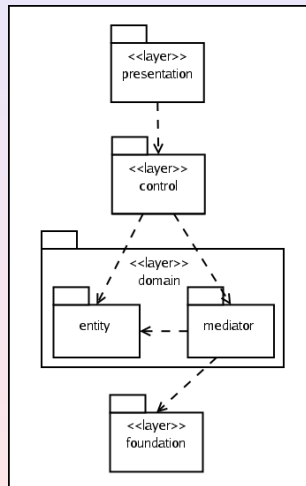
- baz danych
(np. Hibernate, JDBC, JDO)
- repozytoriów dokumentów
(np. klient WebDAV z Jakarta Slide)
- usług sieciowych (ang. *web services*)
(np. Apache Axis)
- systemów plików
(np. `java.io` z J2SE)



Pakiet foundation

Zawiera zazwyczaj gotowe implementacje odpowiadające za obsługę zewnętrznych źródeł danych:

- baz danych
(np. Hibernate, JDBC, JDO)
- repozytoriów dokumentów
(np. klient WebDAV z Jakarta Slide)
- usług sieciowych (ang. *web services*)
(np. Apache Axis)
- systemów plików
(np. java.io z J2SE)



W jakim stopniu PCMEF odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zalety:**

- minimalizacja zależności pomiędzy pakietami
- zdefiniowana semantyka poszczególnych warstw
- modularyzacja ułatwiająca pielęgnację systemu

- **Wada:** mocno powiązany kontroler z widokiem w warstwie presentation

W przypadku aplikacji internetowych **oddzielenie kontrolera od widoku jest dobrą praktyką**, która powinna znaleźć swoje odzwierciedlenie w architekturze.

W jakim stopniu PCMEF odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zalety:**

- minimalizacja zależności pomiędzy pakietami
- zdefiniowana semantyka poszczególnych warstw
- modularyzacja ułatwiająca pielęgnację systemu

- **Wada:** mocno powiązany kontroler z widokiem w warstwie presentation

W przypadku aplikacji internetowych **oddzielenie kontrolera od widoku jest dobrą praktyką**, która powinna znaleźć swoje odzwierciedlenie w architekturze.

W jakim stopniu PCMEF odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zalety:**

- minimalizacja zależności pomiędzy pakietami
- zdefiniowana semantyka poszczególnych warstw
- modularyzacja ułatwiająca pielęgnację systemu

- **Wada:** mocno powiązany kontroler z widokiem w warstwie presentation

W przypadku aplikacji internetowych **oddzielenie kontrolera od widoku jest dobrą praktyką**, która powinna znaleźć swoje odzwierciedlenie w architekturze.

W jakim stopniu PCMEF odpowiada na wyzwania stawiane architekturze aplikacji internetowych?

- **Zalety:**

- minimalizacja zależności pomiędzy pakietami
- zdefiniowana semantyka poszczególnych warstw
- modularyzacja ułatwiająca pielęgnację systemu

- **Wada:** mocno powiązany kontroler z widokiem w warstwie presentation

W przypadku aplikacji internetowych **oddzielenie kontrolera od widoku jest dobrą praktyką**, która powinna znaleźć swoje odzwierciedlenie w architekturze.

Propozycja szkieletu architektonicznego XWA

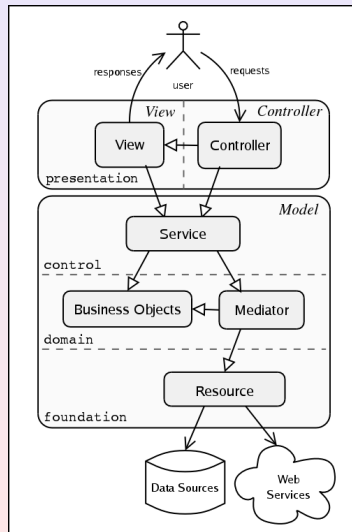


Propozycja szkieletu architektonicznego XWA

Architektura XWA (ang. *eXtensible Web Architecture*) jest architekturą warstwową z wyraźnie wyodrębnioną triadą MVC. Łączy zalety:

- szkieletu MVC – rozdzielając widok od kontrolera
- szkieletu PCMEF – organizując model w architekturę warstwową

Dodatkowo modyfikacji ulegnie kontroler.

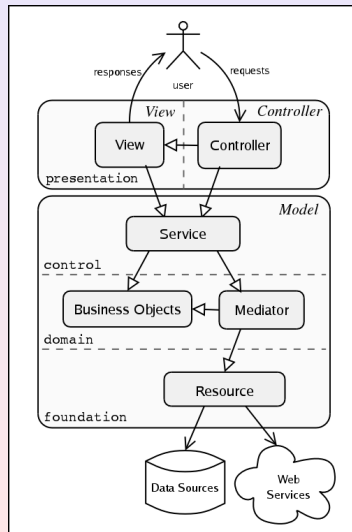


Propozycja szkieletu architektonicznego XWA

Architektura XWA (ang. *eXtensible Web Architecture*) jest architekturą warstwową z wyraźnie wyodrębnioną triadą MVC. Łączy zalety:

- szkieletu MVC – rozdzielając widok od kontrolera
- szkieletu PCMEF – organizując model w architekturę warstwową

Dodatkowo modyfikacji ulegnie kontroler.

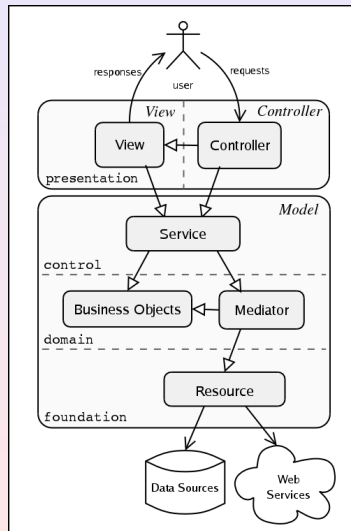


Propozycja szkieletu architektonicznego XWA

Architektura XWA (ang. *eXtensible Web Architecture*) jest architekturą warstwową z wyraźnie wyodrębnioną triadą MVC. Łączy zalety:

- szkieletu MVC – rozdzielając widok od kontrolera
- szkieletu PCMEF – organizując model w architekturę warstwową

Dodatkowo modyfikacji ulegnie kontroler.

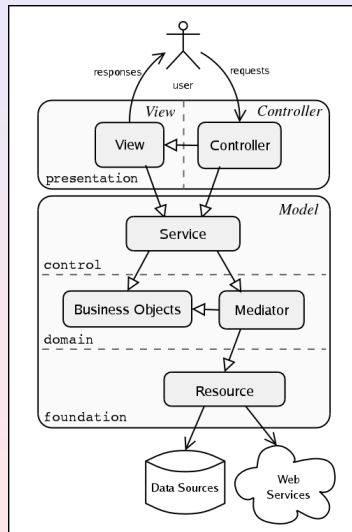


Szkielet architektoniczny XWA

Składa się z sześciu pakietów:

- View
- Controller
- Service
- Bussiness Objects
- Mediator
- Resource

Pakiety są zorganizowane w czteropoziomą hierarchię, w której warstwy wyższe zależą od niższych.



XWA a wyzwania architektoniczne



XWA a pielęgnacja systemu

Jak XWA próbuje radzić sobie z problemem pielęgnacji systemu?

- organizuje system w hierarchię warstwową
- precyzyjnie specyfikuje semantykę poszczególnych pakietów i warstw w kontekście aplikacji internetowych
- zachowuje kluczowe zasady MVC: **separacji modelu od prezentacji** oraz **separacji widoku od kontrolera**
- sugeruje zastosowanie technologii komponentowych (np. kontenery obiektów Avalon czy Spring ładnie wpasowują się w model XWA)

XWA a pielęgnacja systemu

Jak XWA próbuje radzić sobie z problemem pielęgnacji systemu?

- organizuje system w hierarchię warstwową
- precyzyjnie specyfikuje semantykę poszczególnych pakietów i warstw w kontekście aplikacji internetowych
- zachowuje kluczowe zasady MVC: **separacji modelu od prezentacji** oraz **separacji widoku od kontrolera**
- sugeruje zastosowanie technologii komponentowych (np. kontenery obiektów Avalon czy Spring ładnie wpasowują się w model XWA)



XWA a pielęgnacja systemu

Jak XWA próbuje radzić sobie z problemem pielęgnacji systemu?

- organizuje system w hierarchię warstwową
- precyzyjnie specyfikuje semantykę poszczególnych pakietów i warstw w kontekście aplikacji internetowych
- zachowuje kluczowe zasady MVC: **separacji modelu od prezentacji** oraz **separacji widoku od kontrolera**
- sugeruje zastosowanie technologii komponentowych (np. kontenery obiektów Avalon czy Spring ładnie wpasowują się w model XWA)

XWA a pielęgnacja systemu

Jak XWA próbuje radzić sobie z problemem pielęgnacji systemu?

- organizuje system w hierarchię warstwową
- precyzyjnie specyfikuje semantykę poszczególnych pakietów i warstw w kontekście aplikacji internetowych
- zachowuje kluczowe zasady MVC: **separacji modelu od prezentacji** oraz **separacji widoku od kontrolera**
- sugeruje zastosowanie technologii komponentowych (np. kontenery obiektów Avalon czy Spring ładnie wpasowują się w model XWA)

XWA a szerokie spektrum formatów publikacji

Jak XWA próbuje radzić sobie z problemem obsługi szerokiego spektrum formatów publikacji?

- proponuje wykorzystanie XML jako interfejsu pomiędzy logiką aplikacji a prezentacją umożliwiając prostą rozbudowę systemu o obsługę szerokiego spektrum formatów prezentacji (np. HTML, PDF, PS, WML) czy urządzeń dostępowych po stronie użytkownika systemu

XWA a integracja różnych źródeł danych i aplikacji

Jak XWA próbuje radzić sobie z problemem integracji źródeł danych i aplikacji?

- wydzielenie pakietów Mediator i Resource znacznie ułatwia proces integracji zewnętrznych źródeł danych i aplikacji (np. obsługa zapytań do nowych źródeł danych oddzielona jest od logiki aplikacji i nie wymusza modyfikacji obiektów biznesowych).

XWA a efektywna współpraca osób o różnych specjalizacjach

W jaki sposób XWA umożliwia efektywną współpracę osób o różnych specjalizacjach?

- proponuje precyzyjny podział odpowiedzialności pomiędzy programistów logiki biznesowej, warstwy prezentacji, grafików a projektantów stron WWW, dzięki zachowaniu obu kluczowych zasad MVC



Uwzględnienie specyfiki protokołu HTTP

Jak XWA uwzględnia specyfikę protokołu HTTP?

XWA proponuje wprowadzenie **kontrolera kontynuacji**, który:

- w sposób niewidoczny dla programisty utrzymuje stan interakcji użytkownika z systemem (rozwiązując problem *bezstanowości* protokołu HTTP)
- w momencie przyjścia kolejnego żądania automatycznie wznawia przerwana interakcję (uwzględniając w ten sposób *pasywność* protokołu HTTP)
- znacznie upraszcza implementację rozbudowanych interakcji użytkownika z systemem (interakcja jest modelowana za pomocą sekwencyjnie realizowanych skryptów lub programów)



Uwzględnienie specyfiki protokołu HTTP

Jak XWA uwzględnia specyfikę protokołu HTTP?

XWA proponuje wprowadzenie **kontrolera kontynuacji**, który:

- w sposób niewidoczny dla programisty utrzymuje stan interakcji użytkownika z systemem (rozwiązując problem *bezstanowości* protokołu HTTP)
- w momencie przyjścia kolejnego żądania automatycznie wznawia przerwana interakcję (uwzględniając w ten sposób *pasywność* protokołu HTTP)
- znacznie upraszcza implementację rozbudowanych interakcji użytkownika z systemem (interakcja jest modelowana za pomocą sekwencyjnie realizowanych skryptów lub programów)



Uwzględnienie specyfiki protokołu HTTP

Jak XWA uwzględnia specyfikę protokołu HTTP?

XWA proponuje wprowadzenie **kontrolera kontynuacji**, który:

- w sposób niewidoczny dla programisty utrzymuje stan interakcji użytkownika z systemem (rozwiązując problem *bezstanowości* protokołu HTTP)
- w momencie przyjścia kolejnego żądania automatycznie wznawia przerwana interakcję (uwzględniając w ten sposób *pasywność* protokołu HTTP)
- znacznie upraszcza implementację rozbudowanych interakcji użytkownika z systemem (interakcja jest modelowana za pomocą sekwencyjnie realizowanych skryptów lub programów)



Projekt e-Informatyka

Projekt akademicki realizowany przez studentów mający na celu:

- skupienie społeczności informatycznej
- stworzenie nowoczesnej platformy publikacji elektronicznych



e-Informatyka.pl

Projekt e-Informatyka

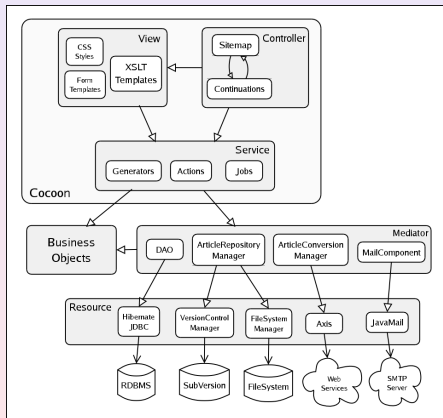
Projekt akademicki realizowany przez studentów mający na celu:

- skupienie społeczności informatycznej
- stworzenie nowoczesnej platformy publikacji elektronicznych



e-Informatyka.pl

Architektura e-Informatyki – implementacja XWA



Podsumowanie

Zaproponowaliśmy szkielet architektoniczny XWA uwzględniający specyfikę nowoczesnych aplikacji internetowych, a w szczególności:

- precyzyjnie wyspecyfikowaliśmy semantykę poszczególnych pakietów i warstw co ułatwi zastosowanie XWA w praktyce
- zapewniliśmy klarowną separację **widoku**, **kontrolera** i **modelu**, jednocześnie precyzując organizację modelu
- dostrzegając wyraźną potrzebę wyodrębniliśmy nowy element architektury – **kontroler kontynuacji**, który znacznie upraszcza implementację skomplikowanych interakcji
- zaproponowaliśmy użycie interfejsu XML w warstwie prezentacji co pozwoli na prostą rozbudowę systemu o obsługę szerokiego spektrum formatów prezentacji



Podsumowanie

Zaproponowaliśmy szkielet architektoniczny XWA uwzględniający specyfikę nowoczesnych aplikacji internetowych, a w szczególności:

- precyzyjnie wyspecyfikowaliśmy semantykę poszczególnych pakietów i warstw co ułatwi zastosowanie XWA w praktyce
- zapewniliśmy klarowną separację **widoku, kontrolera i modelu**, jednocześnie precyzując organizację modelu
- dostrzegając wyraźną potrzebę wyodrębniliśmy nowy element architektury – **kontroler kontynuacji**, który znacznie upraszcza implementację skomplikowanych interakcji
- zaproponowaliśmy użycie interfejsu XML w warstwie prezentacji co pozwoli na prostą rozbudowę systemu o obsługę szerokiego spektrum formatów prezentacji



Podsumowanie

Zaproponowaliśmy szkielet architektoniczny XWA uwzględniający specyfikę nowoczesnych aplikacji internetowych, a w szczególności:

- precyzyjnie wyspecyfikowaliśmy semantykę poszczególnych pakietów i warstw co ułatwi zastosowanie XWA w praktyce
- zapewniliśmy klarowną separację **widoku, kontrolera i modelu**, jednocześnie precyzując organizację modelu
- dostrzegając wyraźną potrzebę wyodrębniliśmy nowy element architektury – **kontroler kontynuacji**, który znacznie upraszcza implementację skomplikowanych interakcji
- zaproponowaliśmy użycie interfejsu XML w warstwie prezentacji co pozwoli na prostą rozbudowę systemu o obsługę szerokiego spektrum formatów prezentacji



Podsumowanie

Zaproponowaliśmy szkielet architektoniczny XWA uwzględniający specyfikę nowoczesnych aplikacji internetowych, a w szczególności:

- precyzyjnie wyspecyfikowaliśmy semantykę poszczególnych pakietów i warstw co ułatwi zastosowanie XWA w praktyce
- zapewniliśmy klarowną separację **widoku, kontrolera i modelu**, jednocześnie precyzując organizację modelu
- dostrzegając wyraźną potrzebę wyodrębniliśmy nowy element architektury – **kontroler kontynuacji**, który znacznie upraszcza implementację skomplikowanych interakcji
- zaproponowaliśmy użycie interfejsu XML w warstwie prezentacji co pozwoli na prostą rozbudowę systemu o obsługę szerokiego spektrum formatów prezentacji



Podsumowanie

Przetestujmy i wspólnie rozwijajmy zaproponowane rozwiązanie!

Tym bardziej, że jest ono wciąż *distinguishable from magic*.

Your improvements are welcome!

Prezentacja i serwis e-Informatyka powstały za pomocą narzędzi Open Source.

