


Wydział Informatyki PB

Inżynieria oprogramowania

Wykład 5:
UML: Diagramy fizyczne

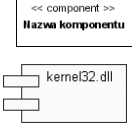
Marek Krętowski
pokój 206
e-mail: mkret@ii.pb.bialystok.pl
http://aragorn.pb.bialystok.pl/~mkret



Wersja 1.02

Komponenty - wprowadzenie

- Komponent to fizyczna, wymienna część systemu, która wykorzystuje i realizuje pewien zbiór interfejsów; na diagramie może być przedstawiany jako prostokąt z bolcami
- Służą do modelowania elementów fizycznych, które mogą być umieszczane na węzłach; są to m. in. programy wykonalne, biblioteki, tabele, pliki i dokumenty
- Komponent to fizyczne opakowanie bytów logicznych takich jak klasy, interfejsy i kooperacje
- Nazwę komponentu podaje się zwykle w formie krótkiego rzeczownika lub wyrażenia rzeczownikowego, pochodzącego ze słownictwa implementacji; nazwa komponentu obejmuje też rozszerzenia pliku, zależne od systemu operacyjnego (np. .java czy .dll)



Inżynieria oprogramowania (Wyk. 5) Slajd 2 z 26

Komponenty a klasy

- Podobieństwa komponentów i klas:
 - mają nazwy i realizują pewien zbiór interfejsów;
 - mogą brać udział w zależnościach, uogólnieniach i powiązaniach;
 - mogą być zagnieżdżone, mieć egzemplarze i uczestniczyć w interakcjach
- Różnice:
 - klasy reprezentują abstrakcje logiczne, a komponenty - elementy fizyczne (komponenty mogą być umiejscowione w węzłach a klasy nie)
 - komponenty reprezentują fizyczne opakowanie składników logicznych i znajdują się na innym poziomie abstrakcji
 - klasy bezpośrednio posiadają atrybuty i operacje, komponenty w zasadzie mają jedynie operacje, które są udostępniane przez ich interfejsy
- Komponent jest fizyczną implementacją zbioru innych składników logicznych takich jak klasy i kooperacje (związek pomiędzy komponentem i klasami może być zobrazowany za pomocą zależności)

Inżynieria oprogramowania (Wyk. 5) Slajd 3 z 26

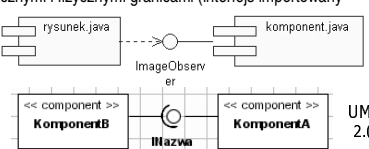
Komponenty i interfejsy

- Interfejs to zestaw operacji, które wyznaczają usługi oferowane przez klasę lub komponent
- We wszystkich popularnych udogodnieniach komponentowych (np. COM+, CORBA czy Enterprise Java Beans) używa się interfejsów do łączenia komponentów
- Umożliwia to podzielenie fizycznej implementacji na części: określane są interfejsy, które reprezentują podstawowe szwy w systemie; dostarczane są komponenty realizujące te interfejsy oraz komponenty, które korzystają z usług przez te interfejsy
- Interfejs realizowany przez komponent nazywamy eksportowanym (komponent udostępnia usługi innym poprzez ten interfejs); komponent może mieć wiele interfejsów eksportowanych

Inżynieria oprogramowania (Wyk. 5) Slajd 4 z 26

Komponenty i interfejsy (2)

- Interfejs z którego komponent korzysta nosi nazwę importowanego (komponent jest zgodny z tym interfejsem i na jego podstawie buduje swoje usługi); komponent może korzystać z wielu interfejsów importowanych
- Jeden komponent może zarówno importować jak i eksportować interfejsy
- Istnienie interfejsu pomiędzy komponentami rozbija ich bezpośrednią zależność (komponent korzystający z danego interfejsu będzie działał poprawnie niezależnie od tego, jaki komponent realizuje ten interfejs)
- Interfejsy żyją ponad logicznymi i fizycznymi granicami (interfejs importowany (eksportowany) przez komponent jest także używany (realizowany) przez klasy implementowane w komponencie



UML 2.0

Inżynieria oprogramowania (Wyk. 5) Slajd 5 z 26

Zastępstwo na poziomie kodu binarnego

- Wszystkie udogodnienia komponentowe mają za zadanie umożliwienie złożenie systemu z części wymiennych na poziomie kodu binarnego; oznacza to, że można zbudować system z komponentów a następnie rozwijać go poprzez dodawanie nowych komponentów i wymianę starych
- Komponent - **fizyczna** i **wymienna część systemu**, która wykorzystuje i realizuje pewien zbiór interfejsów
 - fizyczny - istnieje w świecie bitów, a nie pojęć
 - wymienny - może zostać zastąpiony przez inny korzystający z tych samych interfejsów
 - część systemu - rzadko występuje samodzielnie; współpracuje z innymi komponentami i w ten sposób wplata się w otoczenie architektoniczne lub technologiczne; jest fizycznie i logicznie spójny - strukturalna lub czynnościowa porcja systemu; potencjalnie wielokrotnie wykorzystywany w wielu systemach; reprezentuje podstawowy blok konstrukcyjny

Inżynieria oprogramowania (Wyk. 5) Slajd 6 z 26

Rodzaje komponentów

- **Komponenty wdrożenia** - niezbędne a zarazem wystarczające do scalenia działającego systemu, takie jak biblioteki dociągane dynamicznie (DLL) i pliki wykonywalne (EXE); Definicja komponentu w UML jest na tyle szeroka, że dotyczy klasycznych modeli obiektowych (np. COM+, CORBA, ...) a także innych jak dynamiczne strony WWW, tabele baz danych oraz programy wykonywalne korzystające z nietypowych mechanizmów komunikacji
- **Komponenty procesu wytwórczego** - powstałe w trakcie opracowywania oprogramowania, składające się z takich elementów jak pliki z kodem źródłowym i pliki danych na podstawie których generuje się komponenty wdrożenia. Nie są one bezpośrednio składnikiem działającego systemu: są produktami opracowanymi w procesie wytwórczym, niezbędnym do utworzenia systemu wykonywalnego
- **Komponenty wykonania** - powstają w wyniku działania systemu, np. obiekty COM+ ładowane z DLL

Inżynieria oprogramowania (Wyk. 8)

Slajd 7 z 26

Byty standardowe

Wszelkie mechanizmy rozszerzania mogą być zastawane do komponentów, w szczególności metki definiujące nowe własności (np. określające numer wersji) i stereotypy wprowadzające nowe rodzaje komponentów (np. specyficzne dla systemu operacyjnego)

Standardowe stereotypy komponentów (UML 1.5):

- `executable` - komponent, który można wykonać na węźle
- `library` - dynamiczna lub statyczna biblioteka obiektów
- `table` - komponent reprezentujący tabelę bazy danych
- `file` - komponent reprezentujący dokument zawierający kod źródłowy lub dane
- `document` - komponent reprezentujący dokument

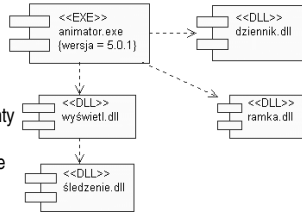
Inżynieria oprogramowania (Wyk. 8)

Slajd 8 z 26

Modelowanie plików wykonywalnych i bibliotek

- Jeżeli system składa się z wielu plików wykonywalnych i licznych bibliotek obiektowych to przy użyciu komponentów można zobrazować decyzje projektowe dotyczące systemu fizycznego
- Rola tego procesu jest jeszcze większa, gdy chce się panować nad wieloma wersjami systemu i zarządzać konfiguracją jego składników

- Zobrazowanie zależności między komponentami jest w istocie skróconą formą przedstawienia prawdziwego związku (komponenty zwykle są niezależne, jedynie importują interfejsy eksportowane przez inny komponent)

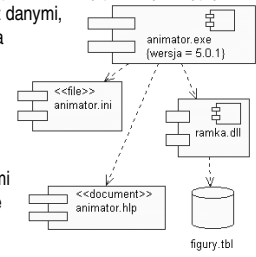


Inżynieria oprogramowania (Wyk. 8)

Slajd 9 z 26

Modelowanie tabel, plików i dokumentów

- Istnieją inne rodzaje (niż programy wykonywalne i biblioteki) pomocniczych komponentów, które są niezbędne w działającym systemie
- Składnikami implementacji mogą być np. dokumenty pomocy, skrypty oraz pliki dzienników, inicjalizacyjne, z danymi, instalacyjne i z procedurami kasowania
- Modelowanie takich komponentów jest istotną częścią procesu zarządzania konfiguracją systemu
- Najczęściej występującymi związkami pomiędzy komponentami pomocniczymi i właściwymi są zależności wskazujące wpływ potencjalnych zmian jednego składnika na drugi

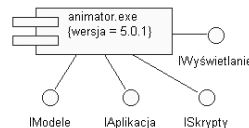


Inżynieria oprogramowania (Wyk. 8)

Slajd 10 z 26

Modelowanie interfejsu programowego (API)

- Programista budujący system z gotowych komponentów musi znać ich interfejsy programowe, umożliwiające łączenie ich ze sobą.
- Interfejsy te reprezentują szwy systemu, który modelujemy za pomocą interfejsów i komponentów
- Operacje wchodzące w skład niebanalnego API, są zwykle bardzo liczne; ich listę warto zapamiętać wewnątrz modelu i używać interfejsów jako wygodnych uchwytów, za pomocą których można mieć dostęp do tych zbiorów operacji
- Należy obrazować jedynie te elementy interfejsu, które są istotne w danym otoczeniu



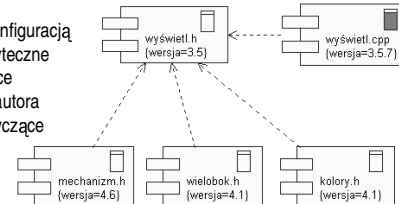
Inżynieria oprogramowania (Wyk. 8)

Slajd 11 z 26

Modelowanie kodu źródłowego

- Graficzne modelowanie kodu źródłowego jest szczególnie użyteczne do obrazowania zależności kompilacyjnych między plikami
- Ułatwia to panowanie nad podziałem i łączeniem grup tych plików, gdy konieczne jest np. rozwidlenie i scalenie ścieżek procesu wytwórczego

- Do zarządzania konfiguracją i kontroli wersji użyteczne są metki określające np. numer wersji, autora lub informacje dotyczące pobrania

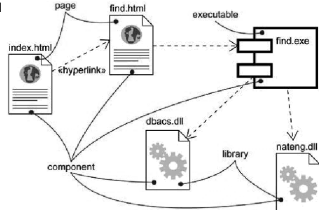


Inżynieria oprogramowania (Wyk. 8)

Slajd 12 z 26

Diagramy komponentów

- Pierwszy z dwóch rodzajów diagramów przedstawiających fizyczne aspekty systemów obiektowych
- Przedstawiają uporządkowanie komponentów i zależności między nimi; służy do modelowania statycznych aspektów perspektywy implementacyjnej systemu, w których bierze się przede wszystkim pod uwagę zarządzanie konfiguracją poszczególnych części systemu
- Zawierają:
 - komponenty (lub ich egzemplarze)
 - interfejsy
 - zależności, uogólnienia, powiązania i realizacje
- Podobne do diagramu klas (komponenty w roli klas)



Inżynieria oprogramowania (Wyk. 8) Slajd 15 z 26

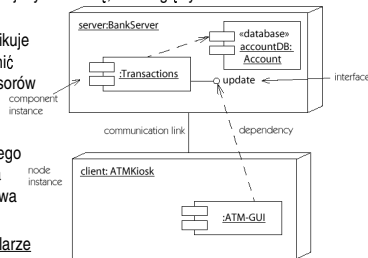
Węzły - wprowadzenie

- Wszystkie komponenty systemu informatycznego są wdrażane na sprzęcie komputerowym - niezależnie od tego czy napisano je od nowa, czy wykorzystano od nowa => SI ze swej natury obejmuje zarówno oprogramowanie jak i sprzęt
- Opracowując architekturę SI należy rozważać zarówno wymiar logiczny (klasy, interfejsy, ...) jak i fizyczny (komponenty reprezentujące fizyczne opakowanie bitów logicznych oraz węzły reprezentujące sprzęt na którym te komponenty są posadowiane)
- Węzeł** to fizyczny składnik działającego systemu; reprezentuje zasoby obliczeniowe; ma zwykle pewną ilość pamięci i zdolność przetwarzania
- Węzłów używa się do modelowania układu sprzętu komputerowego, na którym działa system; zwykle reprezentują procesory lub urządzenia, na których wdrażane są komponenty

Inżynieria oprogramowania (Wyk. 8) Slajd 14 z 26

Wdrożenie

- Węzeł przedstawiany jest w postaci sześciianu z nazwą (prostą lub ścieżkową); zwykle symbol węzła zwraca jedynie nazwę, ale mogą być metki i dodatkowe sekcje
- Stosując stereotypy modyfikuje się ten symbol, aby wyróżnić specyficzne rodzaje procesorów i urządzeń
- W praktyce nazwę podaje się na ogół w formie krótkiego rzeczownika lub wyrażenia pochodzącego ze słownictwa implementacji
- Węzły mogą mieć egzemplarze
- Zbiór obiektów i komponentów przypisanych węzłowi nazywamy jednostką rozproszenia



Inżynieria oprogramowania (Wyk. 8) Slajd 15 z 26

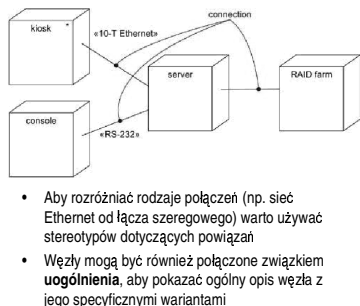
Węzły a komponenty

- Pod wieloma względami węzły przypominają komponenty, np.:
 - mogą brać udział w zależnościach, uogólnieniach i powiązaniach
 - mogą być zagnieżdżone
 - mogą uczestniczyć w interakcjach
- Ale są między nimi podstawowe różnice:
 - komponenty uczestniczą w działaniach systemu, a węzły realizują działania komponentów (komponenty są elementami działającymi na węzłach)
 - komponenty reprezentują fizyczne opakowanie el. logicznych a węzły reprezentują fizyczne wdrożenie komponentów
- Węzły są podobne w pewien sposób do klas, gdyż można określić ich atrybuty i operacje:
 - atrybuty: `prędkośćProcesora` czy `rozmiarPamięci`
 - operacje: `włącz`, `wyłącz` i `wstrzymaj`

Inżynieria oprogramowania (Wyk. 8) Slajd 16 z 26

Powiązania

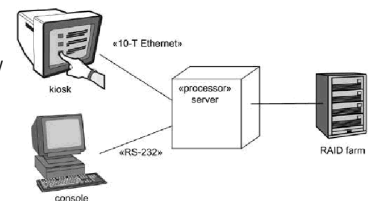
- Najczęściej występującym związkiem pomiędzy węzłami jest **powiązanie**, które w tym przypadku oznacza połączenie fizyczne (np. sieć Ethernet, łącze szeregowe lub wspólna szyna)
- Powiązania można użyć też do modelowania połączeń pośrednich (np. komunikacja satelitarna między odległymi maszynami)
- W przypadku powiązania węzłów (analogicznie jak dla powiązania klas) mogą być wykorzystane role, liczebność i ograniczenia



Inżynieria oprogramowania (Wyk. 8) Slajd 17 z 26

Modelowanie procesorów i urządzeń

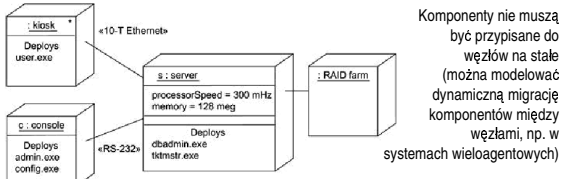
- Standardowe konfiguracje: wolnostojący komputer, system wbudowany, system klient-serwer i system rozproszony
- Procesor** to węzeł, który ma zdolność przetwarzania, czyli może realizować działania komponentów
- Urządzenie** to węzeł, który nie ma zdolności przetwarzania (lub nie jest ona modelowana) i zwykle reprezentuje sprzęt ze światem zewnętrznym
- Stereotypy mogą i powinny służyć do specyfikowania nowych typów procesorów i urządzeń (wprowadzenie specjalnych symboli graficznych)



Inżynieria oprogramowania (Wyk. 8) Slajd 18 z 26

Modelowanie rozproszonych komponentów

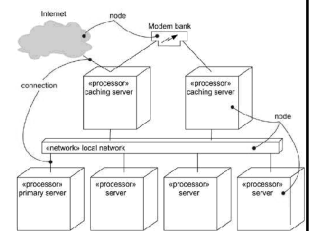
- Modelując układ systemu, można zobrazować fizyczne rozproszenie jego komponentów po procesorach i urządzeniach wchodzących w jego skład
- Komponent zwykle jest przypisany do jednego węzła ale może być też umieszczony na kilku (np. specyficzne programy wykonywalne i biblioteki)
- Położenie komponentów jest zwykle obrazowane przez ich wymienienie w dodatkowej sekcji symbolu węzła



Inżynieria oprogramowania (Wyk. 5) Slajd 19 z 26

Diagramy wdrożenia

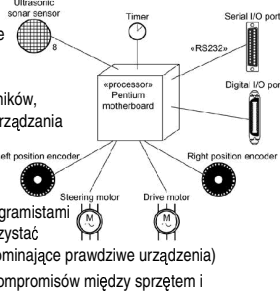
- Drugi z dwóch rodzajów diagramów przedstawiających fizyczne aspekty systemów obiektowych
- Obrazuje konfigurację węzłów działających w czasie wykonania i zainstalowane na nich komponenty
- Bierze pod uwagę przede wszystkim rozproszenie, dostarczenie i instalację części systemu
- Zawiera:
 - węzły
 - zależności i powiązania
- Komponenty mogą występować na diagramie wdrożenia o ile są przypisane do jakiegoś węzła



Inżynieria oprogramowania (Wyk. 5) Slajd 20 z 26

Modelowanie systemów wbudowanych

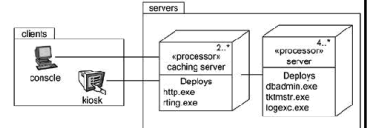
- System wbudowany to zestaw sprzętu oprogramowanego, który kontaktuje się ze światem rzeczywistym
- Obejmuje oprogramowanie, które steruje takimi urządzeniami jak silniki i wyświetlacze, a na jego działania mają wpływ bodźce zewnętrzne (dane z mierników, ruch i zmiany temperatury); Wymaga zarządzania światem realnym, w którym ruchome części ulegają awariom, sygnały są zakłócone, a zachowanie nieliniowe
- Aby ułatwić porozumienie między programistami a specjalistami od sprzętu należy wykorzystać stereotypowanie węzłów (symbole przypominające prawdziwe urządzenia)
- Diagramy wdrożenia ułatwiają analizę kompromisów między sprzętem i oprogramowaniem



Inżynieria oprogramowania (Wyk. 5) Slajd 21 z 26

Modelowanie systemu typu klient-serwer

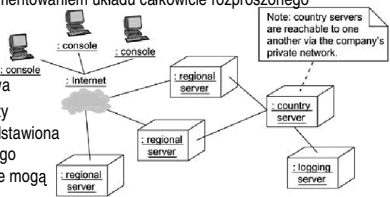
- Pytania podczas budowy systemu: Jak najlepiej przydzielić poszczególne komponenty programowe do węzłów? Jak komponenty będą się porozumiewać? Jak radzić sobie z awariami i zakłóceniami?
- Architektura klient-serwer kładzie nacisk na oddzielenie pojęć dotyczących interfejsu (na stacji klienta) i danych trwałych (na serwerze); kluczową kwestią jest zwykle umieszczenie komponentów obejmujących tzw. logikę biznesową
- Cienki klient (niewielka moc obliczeniowa, zadania ograniczone do zarządzania interfejsem użytkownika i wyświetlanie informacji; nie jest na ogół właścicielem komponentów)
- Klient gruby (większa moc, więcej czynności - zgodnie z regulami logicznymi systemu i przyjętymi regulami przedsiębiorstwa)



Inżynieria oprogramowania (Wyk. 5) Slajd 22 z 26

Modelowanie całkowicie rozproszonego systemu

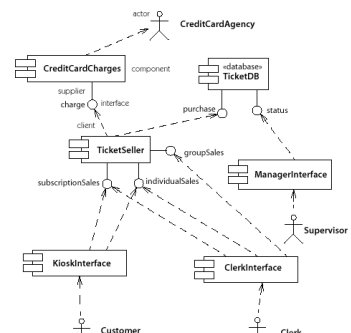
- System rozproszony to zarówno dwuprocesorowy jak i zainstalowany na wielu węzłach rozsianych po całym świecie
- Duże systemy rzadko są statyczne (w wyniku zmian obciążenia i awarii pojawiają się nowe i znikają stare węzły; powstają nowe kanały komunikacyjne; zmianie ulega też rozłożenie komponentów programowych)
- Specyfikowaniem i dokumentowaniem układu całkowicie rozproszonego systemu zajmuje się administrator panujący nad zasobami informacyjnymi przedsiębiorstwa
- Sieć (lokalna, rozległa czy Internet) może być przedstawiona w postaci stereotypowanego węzła (atrybuty i operacje mogą posłużyć do określenia właściwości sieci)



Inżynieria oprogramowania (Wyk. 5) Slajd 23 z 26

Przykład diagramu komponentów

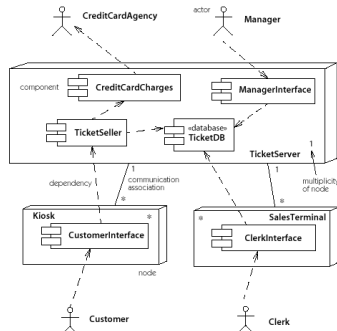
Box office system (sprzedaż biletów)



Inżynieria oprogramowania (Wyk. 5) Slajd 24 z 26

Przykład diagramu wdrożenia - poziom opisowy

Box office system



Przykład diagramu wdrożenia - poziom egzemplarzy

Box office system

