

Projektowanie systemów internetowych

Dr inż. Marek MIŁOSZ

Plan przedmiotu (1)

- Modelowanie wymagań do systemu (scenariusze, modele przypadków użycia, diagramy interakcji)
- Obiektość w projektowaniu
- Wprowadzenie do UML (*Unified Modeling Language*)
- Modelowanie klas i ich związków
- Diagramy implementacyjne: komponentów i wdrożeniowe

Plan przedmiotu (2)

- Mapowanie modelu klas na relacyjny model BD
- Specyfika aplikacji internetowych
- Projektowanie architektury systemów internetowych
- Metodyki projektowania systemów internetowych
- Model MVC i język WebML
- Problemy eksploatacji serwisów internetowych (pozycjonowanie)
- Bezpieczeństwo aplikacji internetowych

Literatura (1)

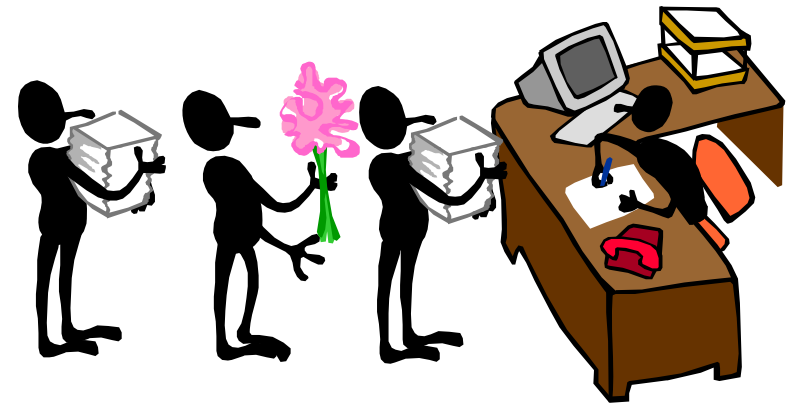
- Booch G., Rumbaugh J., Jacobson I.: ***UML przewodnik użytkownika***. WNT, Warszawa 2001
- Graessle P., Baumann H., Baumann Ph.: **UML 2.0 w akcji. Przewodnik oparty na projektach**. Helion, Gliwice 2006
- Piwko Ł.: **UML 2 Almanach**. Helion, Gliwice 2006
- Sommerville I.: **Inżynieria oprogramowania**. WNT, 2003

Literatura (2)

- Ian Graham: ***Metody obiektowe w teorii i w praktyce.*** WNT, Warszawa 20 04
- Cockburn A.: ***Jak pisać efektywne przypadki użycia.*** WNT, Warszawa 2004
- Schneider G., Winters J.P.: ***Stosowanie przypadków użycia.*** WNT, Warszawa 2004
- Internet

Organizacja

- Wymiar godzin (30W + 30L)
- Umiejscowienie
- Sprawy organizacyjne
- Zajęcia laboratoryjne (IBM Rational Rose i inne)
- Forma zaliczenia:
 - wykład (egz.)
 - laboratorium



Analiza i projektowanie obiektowe

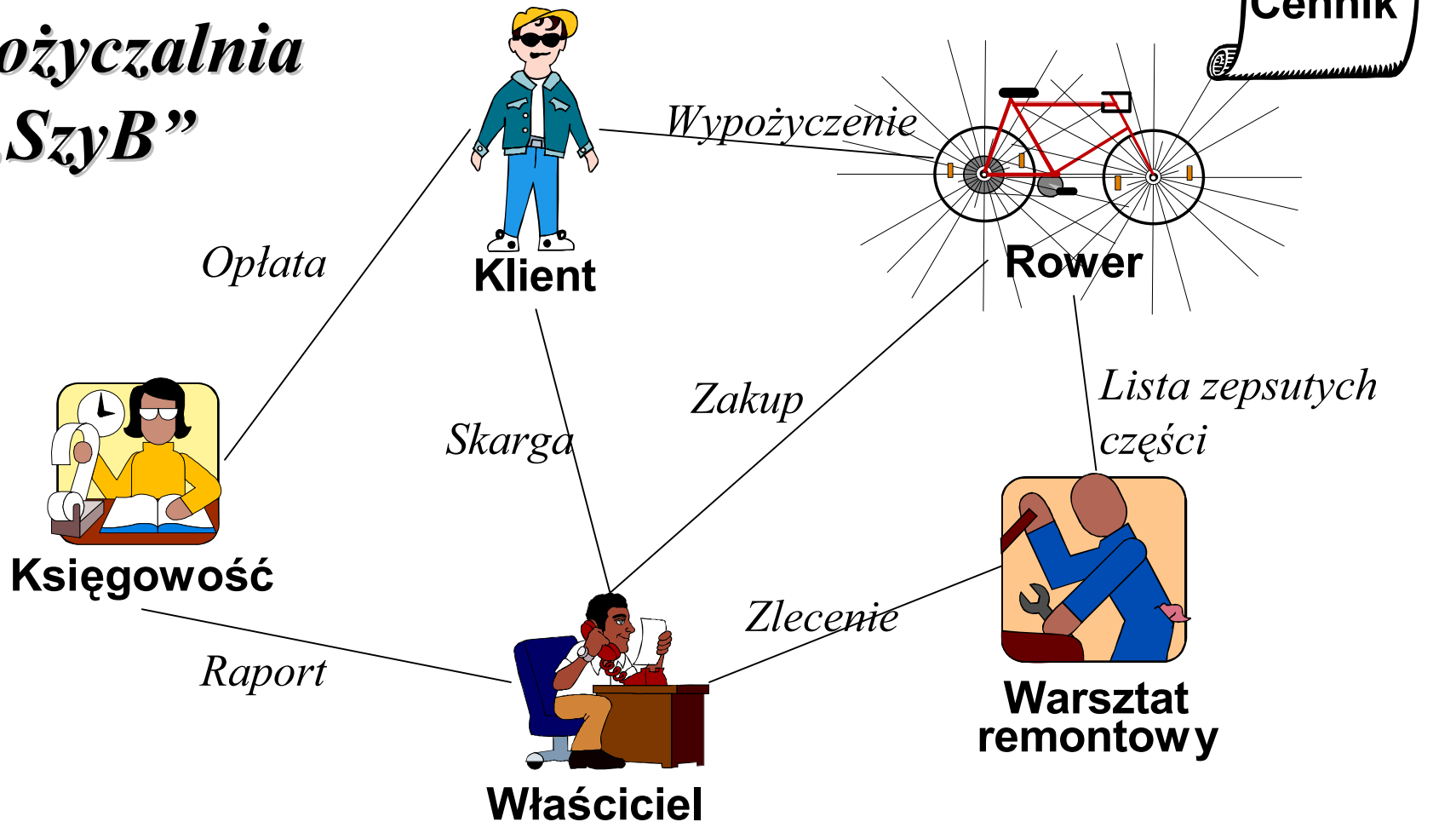
- Nowe podejście do projektowania
- Obiektowość
- Wieloaspektowość (widoków, struktur) systemu i jego projektu
- Różne metodyki (OMT, Booch, UML)
- UML - Unified Modeling Language
- Rational Software – IBM (www.rational.com www.ibm.com)
- Visual UML (www.visualobject.com)
- Inne (Visual Paradigm)

Rzeczywistość - model obiektowy (obiekt+komunikat)



Wypożyczenie

*Wypożyczalnia
„Szyb”*



Model wymagań

Abstrakcja wymagań użytkownika SI - opis obiektowy wymagań poprzez działania

- Scenariusze przypadków użycia SI
- Model przypadków użycia (ang. *Use Case*)
- Diagramy interakcji (sekwencji) dla przypadków użycia (ang. *Sequence Diagrams*) (dynamika) - komunikacja pomiędzy poszczególnymi obiektami

Scenariusz przypadku użycia

- Określa pojedynczą sekwencję zdarzeń i interakcji obiektów zewnętrznych (użytkownika) z systemem
- Scenariusz rozpoczyna się **interakcją** (np. użytkownik chce wydrukować listę obecności)
- Scenariusz kończy się konkretnym **rezultatem**, uzyskanym dzięki systemowi (np. system drukuje listę)

Przebiegi scenariusza

- Przebiegi:
 - główne (np. wydruk listy studentów)
 - alternatywne (np. przed wydrukiem listy użytkownik wyszukuje grupę w której jest konkretny student)
 - wyjątkowe (np. wydruk nie powiódł się)
- Poziomy opis:
 - biznesowy (procedury biznesowe)
 - implementacyjny (funkcjonalność systemu)

Scenariusz - skład (1)

1. Opis
2. Wspierane procedury i procesy biznesowe
3. Użytkownicy
4. Warunki początkowe
5. Warunki końcowe
6. Przebieg główny (podstawowy)
7. Przebiegi alternatywne

Scenariusz - skład (2)

- 8. Sytuacje wyjątkowe
- 9. Reguły
- 10. Wymagania niefunkcjonalne
- 11. Uwagi i pytania otwarte

Szczegółowość scenariusza i język opisu
dostosowany jest do poziomego opisu
systemu

Ilość i czytelność

- Średniej wielkości system zawiera kilkadziesiąt scenariuszy z kilkoma przebiegami alternatywnymi i wyjątkowymi każdy
- Każdy scenariusz to jedna-dwie strony opisu
- Trudności w analizie scenariuszy
- Diagram (rysunek) czyni model bardziej czytelnym






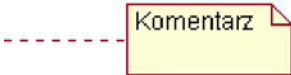

Model przypadku użycia (*Use Case Model/View*)

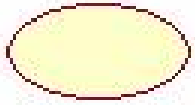
- Cel: analiza wykorzystania systemu informatycznego
- Spojrzenie na system z zewnętrznego punktu widzenia
- Wyszukanie i zdefiniowanie użytkowników (bezpośrednich i pośrednich) oraz sposobów użycia systemu
- Wykrycie identycznych działań, realizowanych przez różnych użytkowników (unikanie redundancji funkcjonalnej)

Główne elementy modelu przypadków użycia

- **Aktor** - użytkownik SI (spójny zbiór ról konkretnej osoby: klient, kierownik, pracownik); wchodzi w interakcję z SI. Każda interakcja jest inna.
- **Przypadek użycia** - sposób interakcji aktora z SI (konkretne zadanie do realizacji w SI); nazwana sekwencja operacji wykonywanych przez aktora (zakup, rejestracja, wydruk dokumentu, założenie); funkcja systemu

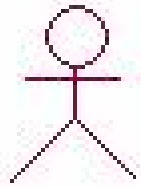
Notacja modelu przypadków użycia

 Ustalenie cennika	Przypadek użycia	<i>Use Case</i>
 Klient	Aktor	<i>Actor</i>
	Interakcja (powiązanie)	<i>Association</i>
	Generalizacja (uogólnienie)	<i>Generalization</i>
	Relacja (zależność, użycie)	<i>Dependency</i>
	Komentarz (notatka)	<i>Note</i>
	Pakiet (blok wielokrotnego/ ponownego użycia)	<i>Package</i>



Przypadek użycia (*Use Case*)

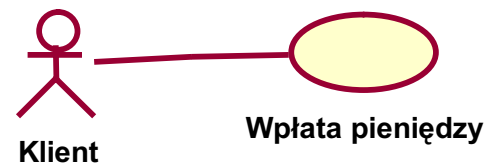
- Unikalna nazwa (ramach konkretnego pakietu), opisująca przypadek z punktu widzenia jego zasadniczych celów
- Nazwa: czynność (np. Rezerwacja roweru) lub polecenie (np. Wypożycz rower)
- Możliwość grupowania w pakiety (nazwane)
- Nazwa ścieżkowa - unikalna
(NazwaPakietu::NazwaPrzypadku)



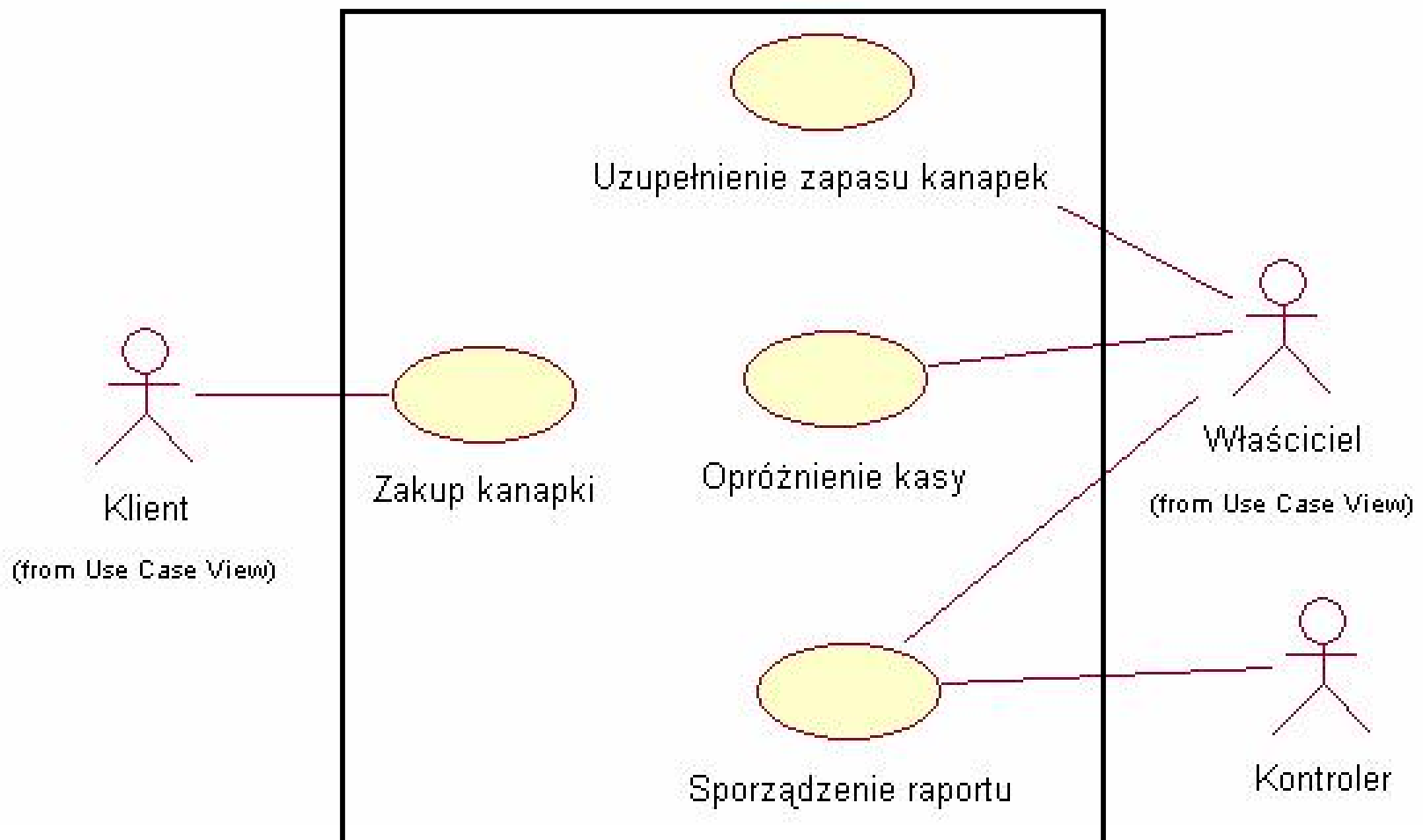
Klient

Aktor (*Actor*)

- Aktor = przyszły użytkownik systemu
- Aktor - byt z otoczenia systemu, **rola**
- Unikalna nazwa
- Aktor = sprawca zdarzeń powodujących uruchomienie przypadku użycia (niekoniecznie bezpośredni użytkownik)
 - wchodzi w interakcję z przypadkiem użycia

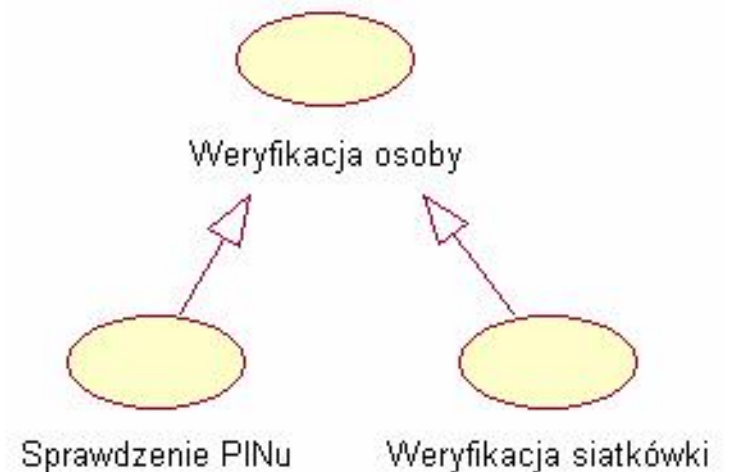
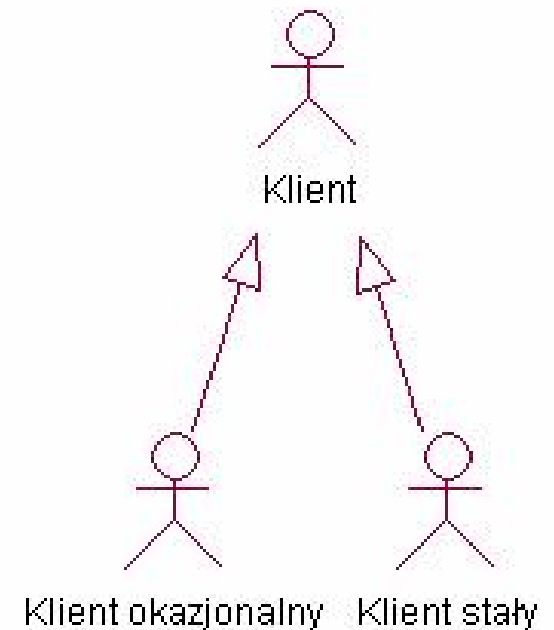


Przykład - automat do sprzedaży kanapek



Generalizacja

- Uogólnianie (w górę)
- Specjalizacja - w dół
- Możliwość zdefiniowania ogólnych jak i uszczegółowionych aktorów i przypadków użycia
- Dziedziczenie dostępu do poszczególnych przypadków

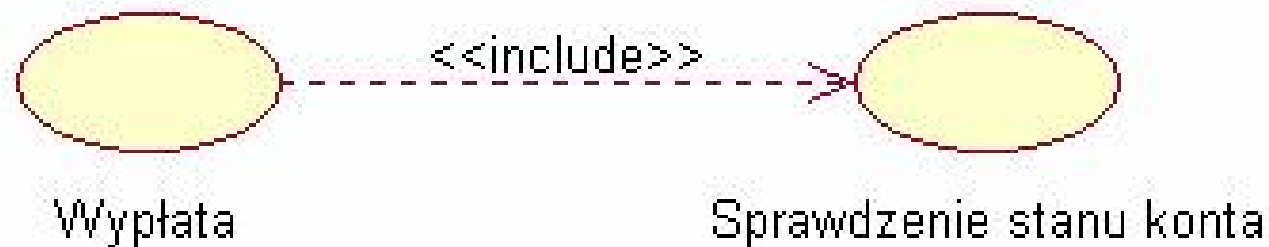


Zależności między przypadkami użycia (*Dependency*)

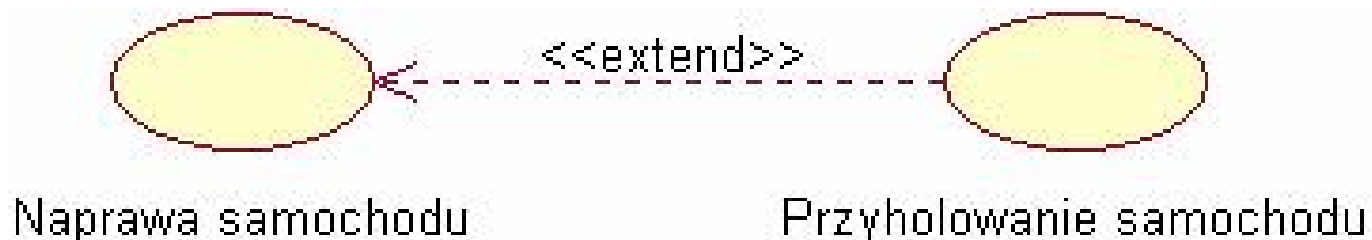
- Przypadki mogą być wykonane w dowolnej kolejności przez aktorów, chyba że są związane między sobą zależnościami (relacjami)
- Typy (stereotypy) zależności:
 - <<include>> - wywołanie przypadku (obowiązkowe) - <<uses>>
 - <<extend>> - rozszerzenie przypadku (opcjonalne)

Notacja zależności

<<include>> - przypadek bazowy **zawsze** wykorzystuje inny przypadek



<<extend>> - przypadek bazowy **czasami** wykorzystuje inny przypadek (rozszerza się)



Zależności i przypadki użycia a scenariusze

<<include>> - przebieg
główny (podstawowy);
zawsze realizowany

<<extend>> - przebieg
alternatywny

***Przebieg scenariusza -
udokumentowanie
przypadku użycia***

Use Case Specification for Ewidencja wypożyczenia

General | Diagrams | Relations | Files

Name: Package: Use Case View

Stereotype:

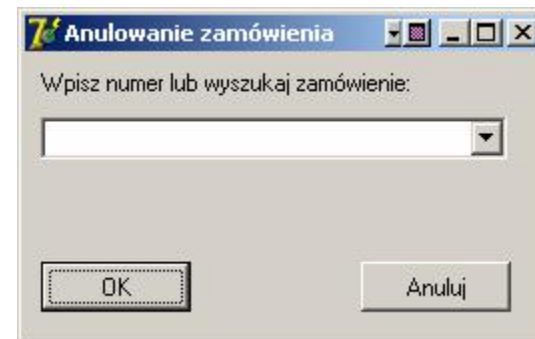
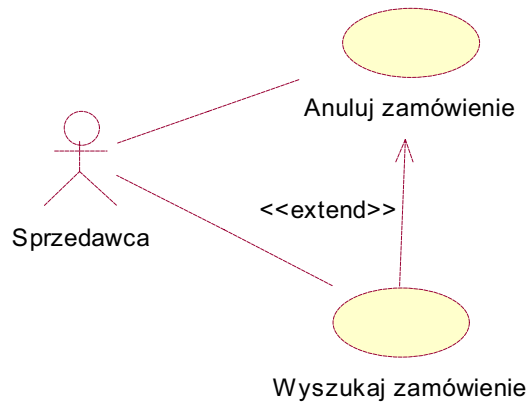
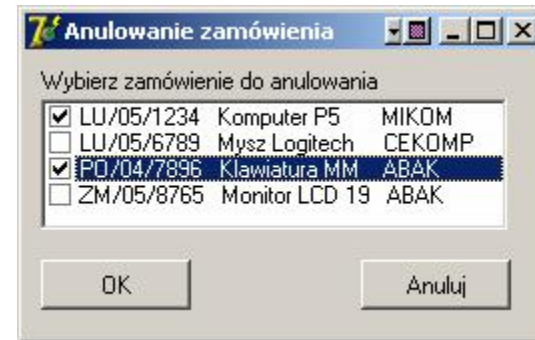
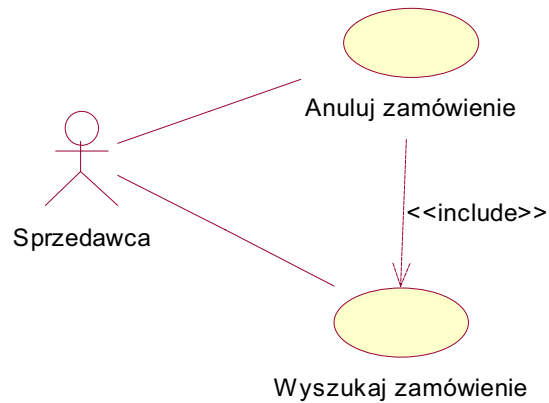
Rank: ☐ Abstract

Documentation:

1. Podaj dane o wypożyczeniu (klient, rower, czas)
2. Wyznacz cenę wypożyczenia
3. Odnotuj fakt wypożyczenia
4. Wydrukuj kwit klienta

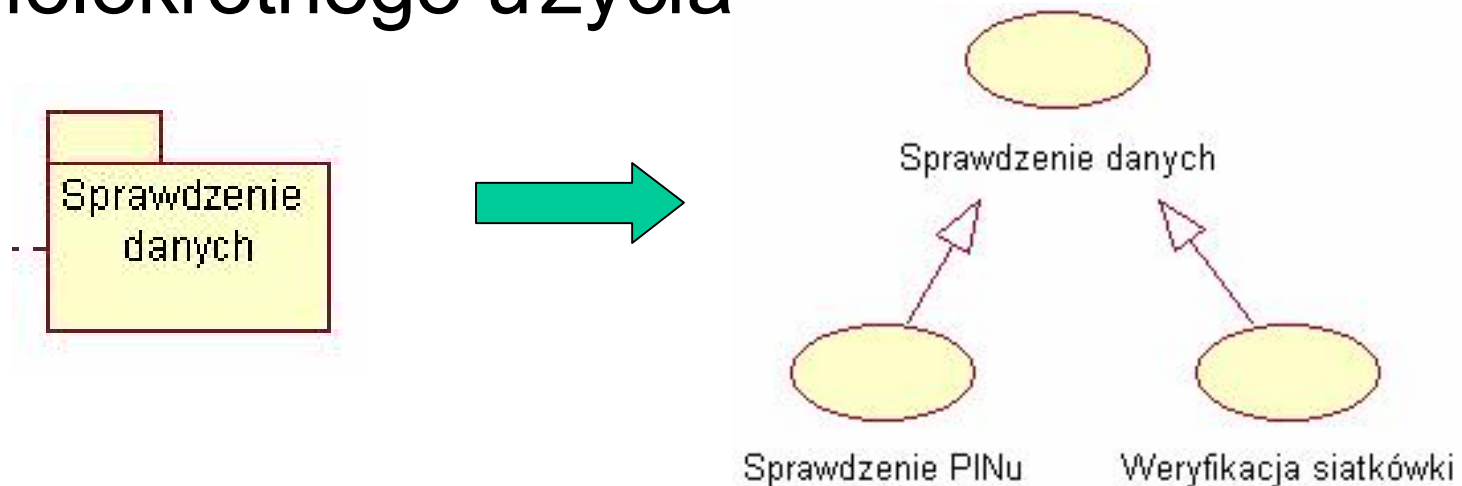
OK Cancel Apply Browse Help

Implementacyjny aspekt zależności



Pakiety (*Package*)

- Grupują przypadki użycia i inne byty
- Pakiet - narzędzie do grupowania, ukrywania wewnętrznej struktury (hermetyzacji), hierarchizacji modelu i wielokrotnego użycia

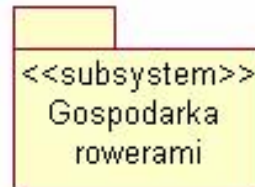


Pakiet - przykład wykorzystania (1)

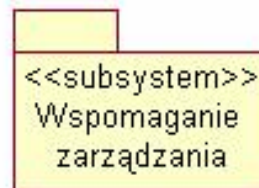
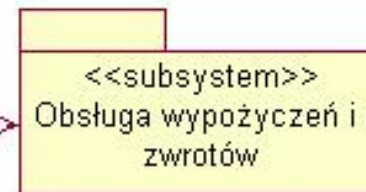
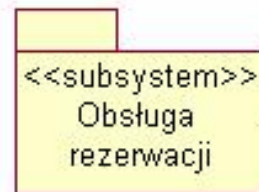
Szyb



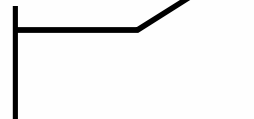
Klient



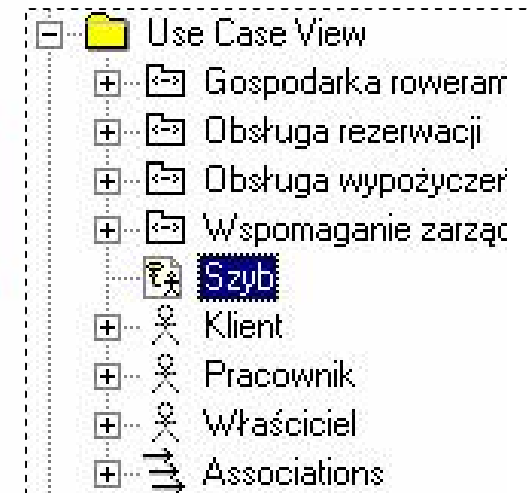
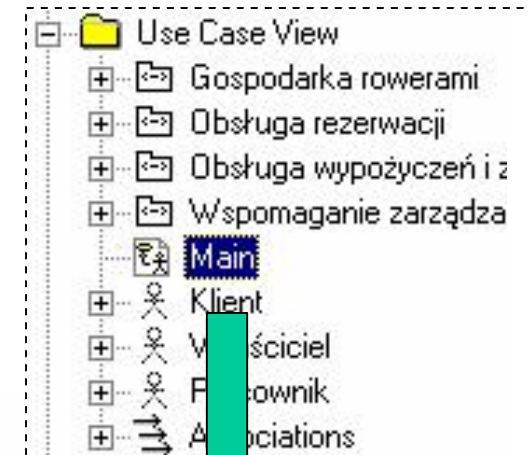
Pracownik



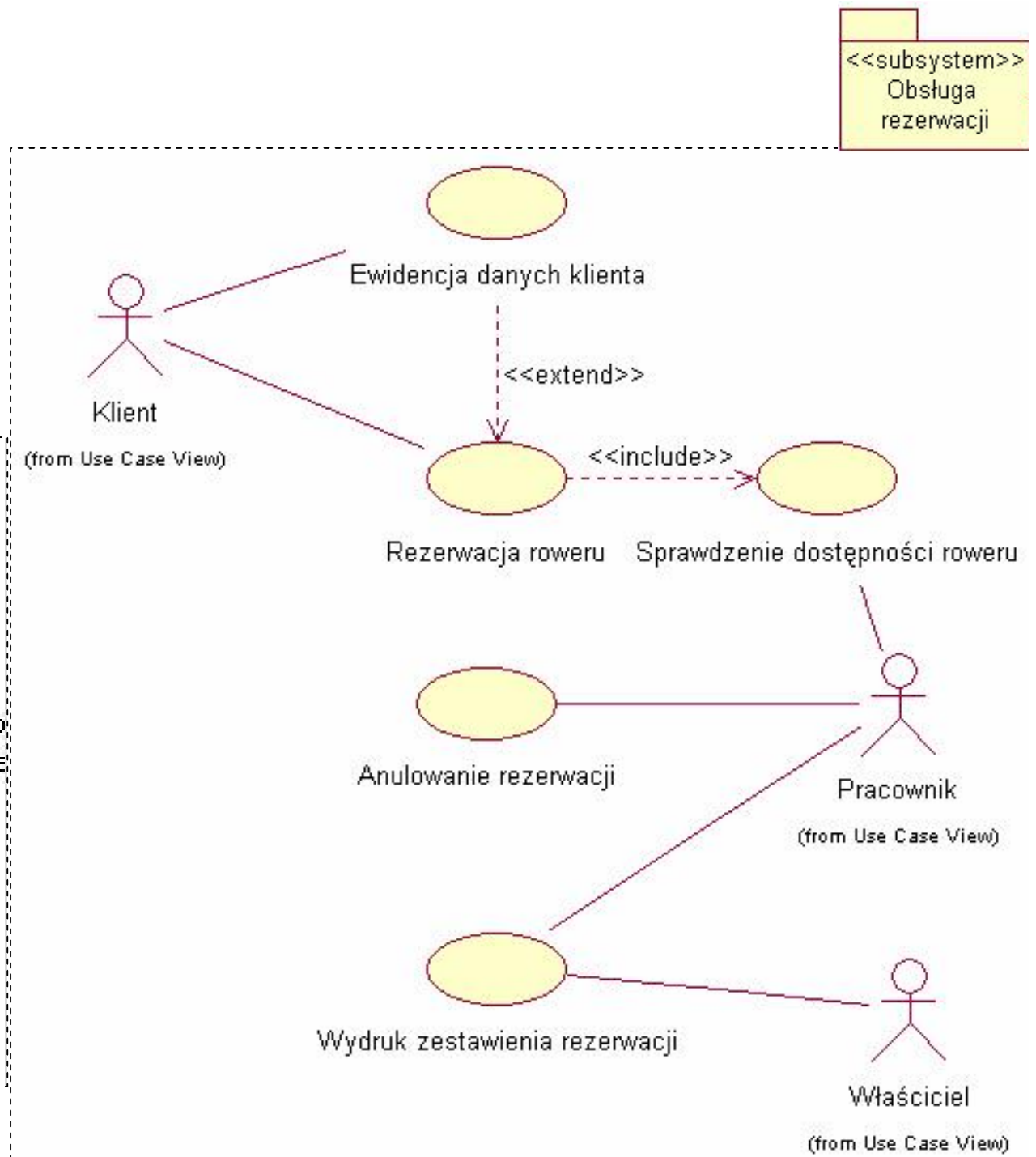
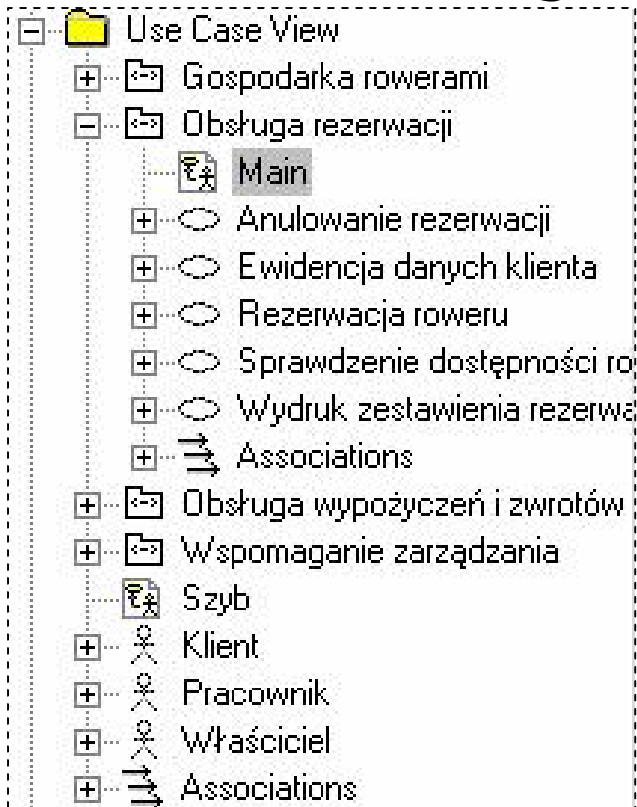
Stereotyp



Właściciel

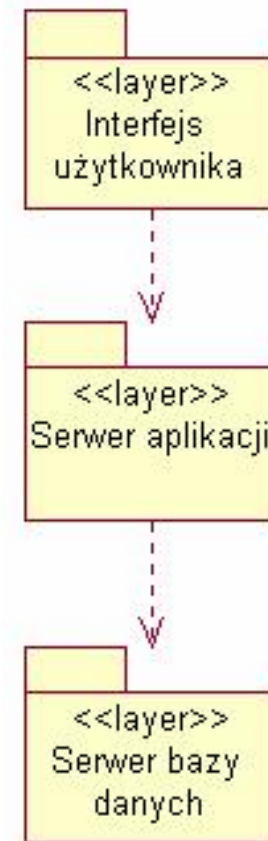
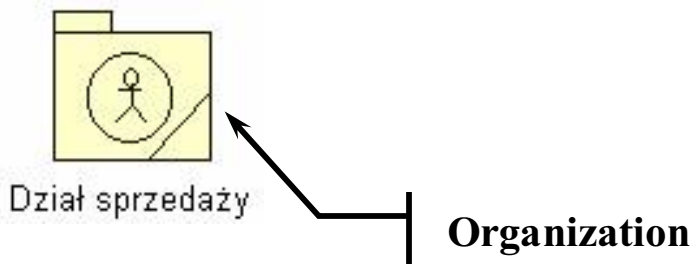
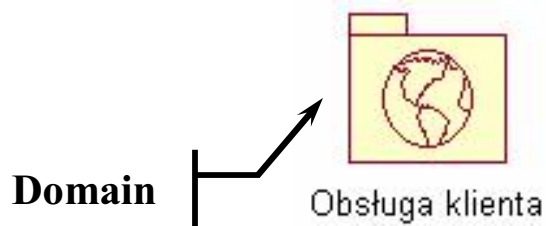
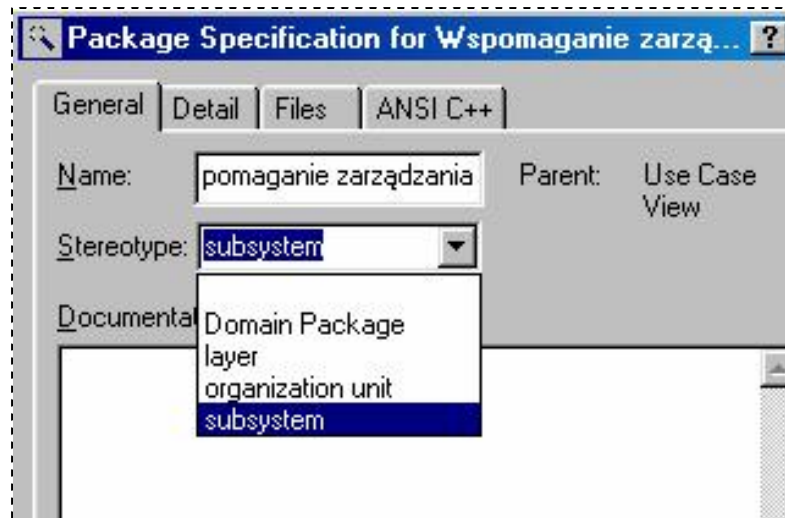


Pakiet: Obsługa rezerwacji

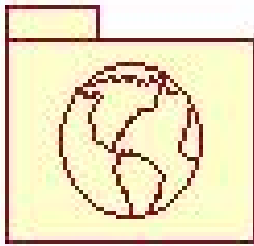


Stereotypy w pakietach

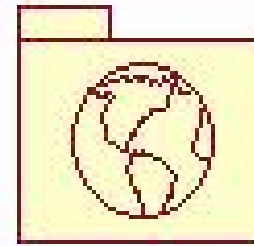
- Udokładniają (klasyfikują) opis pakietu



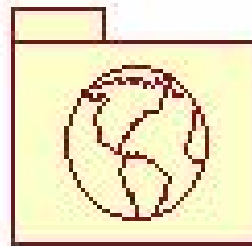
Widoki architektury systemu (struktury, przekroje SI)



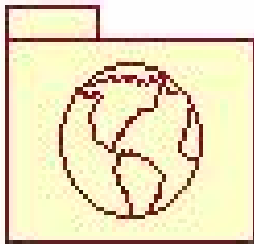
Projektowa



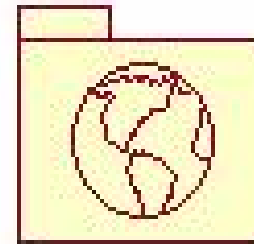
Implementacyjna



Przypadków użycia

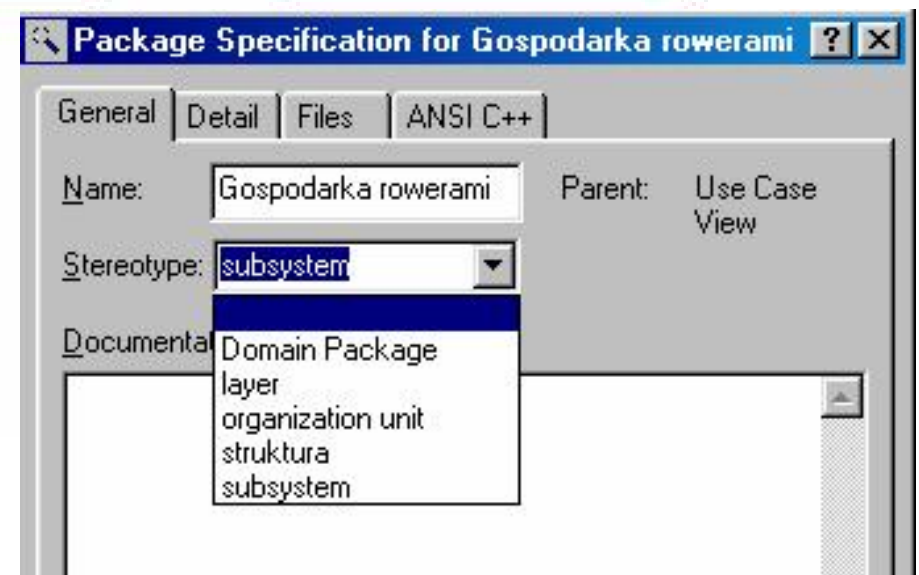
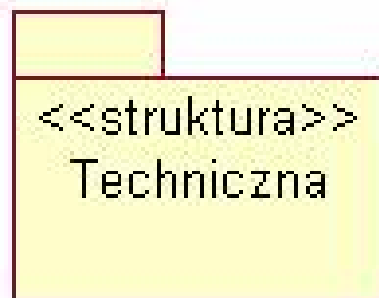
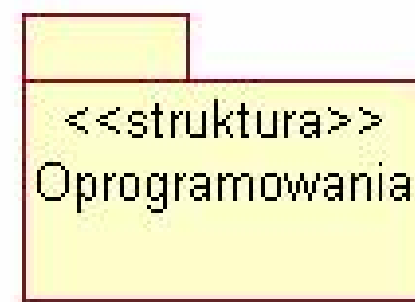
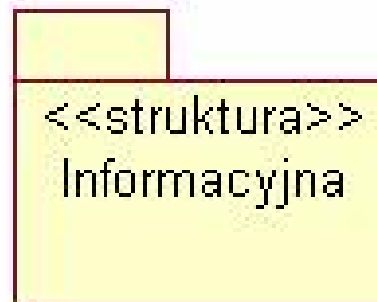
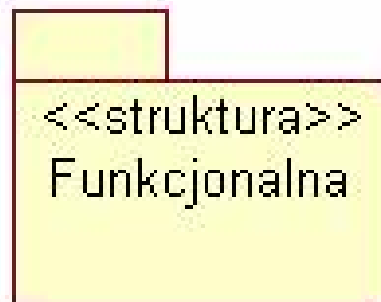


Procesowa



Wdrożeniowa

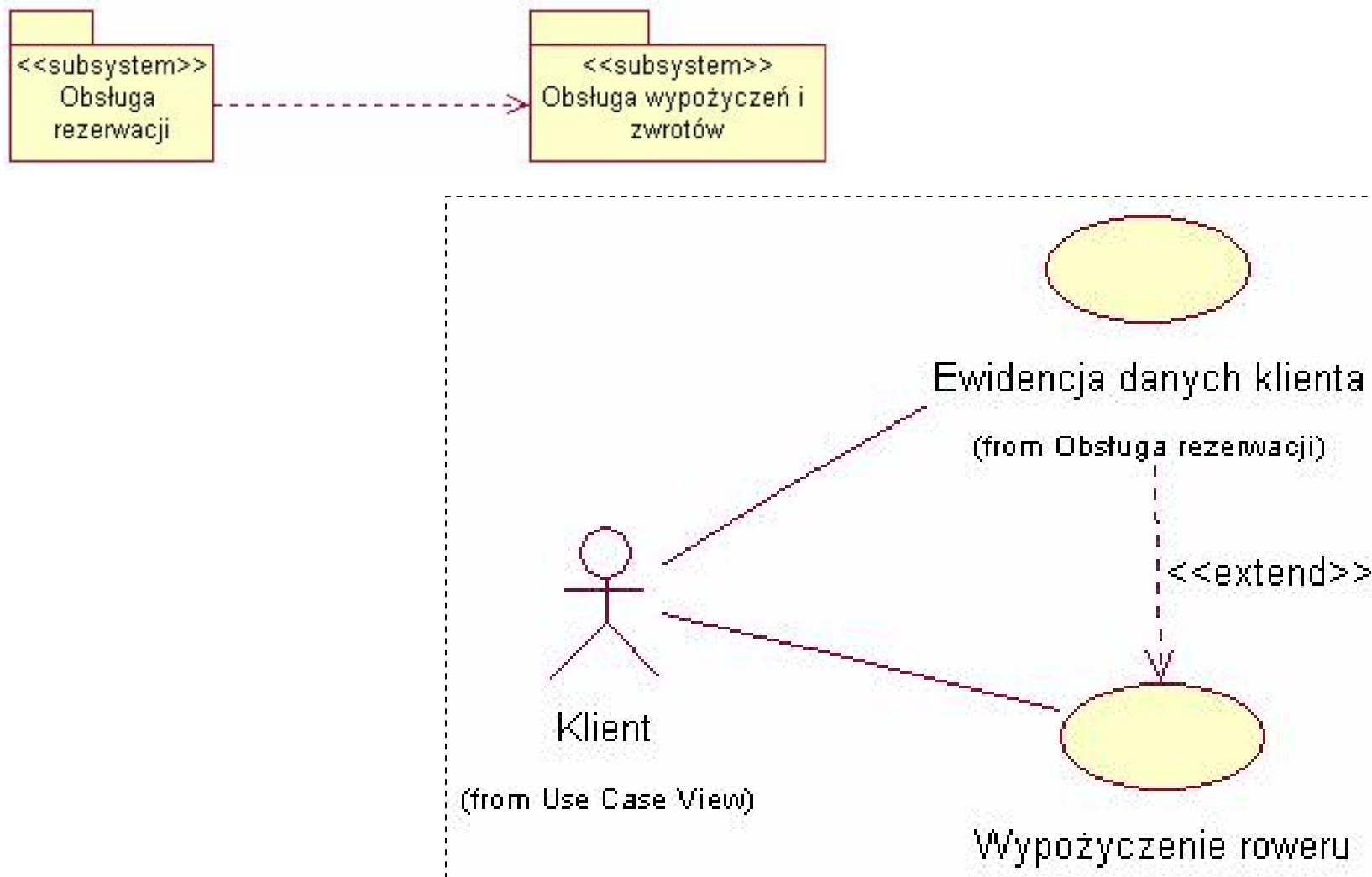
Własne stereotypy



Wskazówki do tworzenia pakietów

- Pakiet ma systematyzować, grupować
- Nie jest odpowiednikiem bytu z rzeczywistości, jest **bytem projektowym**
- Cechy pakietów:
 - spójność (wyraźna granica)
 - luźno związany z innymi pakietami
 - niezbyt wiele poziomów zagnieżdżenia
 - zrównoważenie zawartości i dekompozycji

Zależności między pakietami i wspólne przypadki użycia

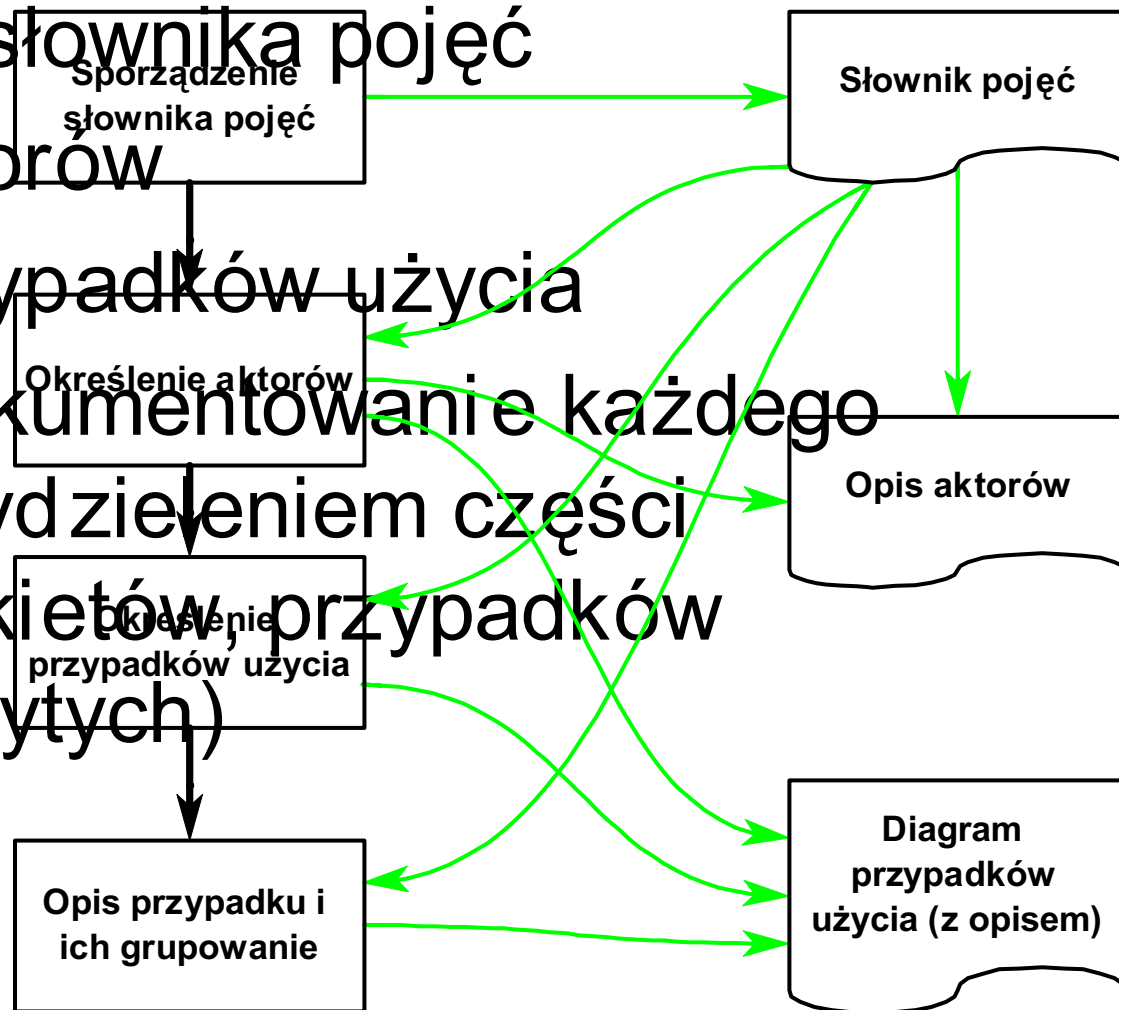


Tworzenie przypadków użycia

- Model służy do spojrzenia na system z punktu widzenia aktorów, którzy go używają
- Nie powinno być zbyt dużej liczby szczegółów (zbyt szczegółowy - utrudnia analizę, zbyt ogólny - wykrycie elementów wielokrotnego użycia)
- Model przypadków użycia - rezultat pracy twórczej (niedeterministycznej, subiektywnej)

Algorytm budowy modelu przypadków użycia

- Sporządzenie słownika pojęć
- Określenie aktorów
- Określenie przypadków użycia
- Tworzenie i dokumentowanie każdego przypadku z wydzielaniem części wspólnych (pakietów, przypadków wielokrotnie użytych)



Słownik pojęć

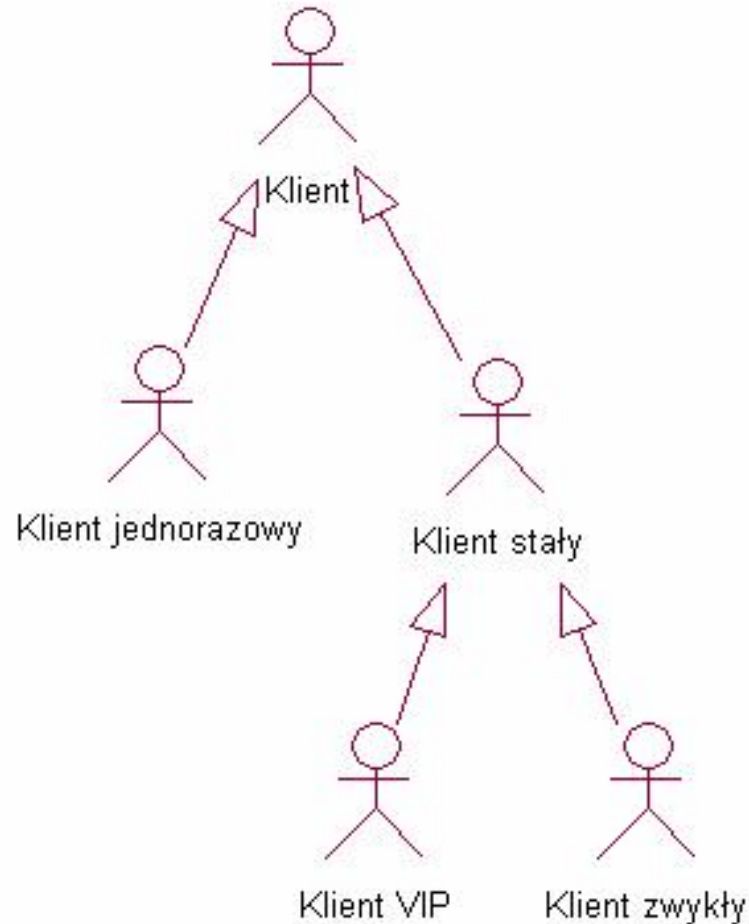
- Precyzuje pojęcia z dziedziny problemowej analizy i modelowania systemu
- Terminy odnoszą się do aktorów, przypadków użycia, obiektów, związków itp.
- Posługiwanie się zdefiniowanymi terminami powinno być regułą w całym projekcie i realizacji systemu
- Rzeczowniki - aktorzy, czasowniki - przypadki użycia (funkcje, akcje, zachowanie się)

Aktorzy - określanie

- Osoby potrzebujące wspomagania ze strony systemu (np. księgowy)
- Osoby niezbędne do działania systemu (np. administratorzy)
- Inne zewnętrzne elementy dostarczające lub pobierające dane z systemu (np. czujniki, inne systemy)

Ustalenie aktorów wiąże się z ustaleniem granic systemu (zakresu odpowiedzialności)

Dziedziczenie dla aktorów



Umożliwia wykorzystanie własności dziedziczenia w definiowaniu zależności aktor-przypadek użycia.

Klient VIP dziedziczy dostęp do wszystkich przypadków Klienta, ale można mu zdefiniować jego własne przypadki użycia

Przypadki użycia

- konstruowanie (1)

- Dla każdego aktora należy wyszukać przypadki użycia związane z nim
- Należy unikać definiowania zbyt prostych (elementarnych) przypadków - prowadzi to do znacznego wzrostu ich liczby
- Nazwy - krótkie, jednoznaczne, odzwierciedlenie czynności od strony aktora a nie systemu („wpłacenie pieniędzy” a nie „przyjęcie pieniędzy od klienta”)
- Przypadki należy zdefiniować (opisać)

Przypadki użycia

- konstruowanie (2)

- Wydzielenie przypadków bazowych, podstawowych
- Ustalenie wzajemnych zależności pomiędzy przypadkami (obowiązkowa, alternatywna)
- Dodanie przypadków skrajnych, wyjątkowych, uzupełniających i opcjonalnych
- Wyizolowanie pakietów (bloków wielokrotnego użycia)
- Zasada Millera (niezbyt dokładnie i niezbyt ogólnie)

Przypadki użycia

- konstruowanie (3)

- Należy używać pakietów w celu grupowania przypadków - uwypuklenie jednego aspektu modelowanego systemu na jednym diagramie
- Tworzenie diagramu:
 - diagram ma mieć nazwę
 - właściwa topologia (kolejność, przecięcia linii)
 - działania i role zbliżone powinny być umieszczone blisko siebie
 - należy wykorzystywać notatki w celu wyjaśniania

Dokumentacja przypadku użycia

- Diagram (*Use Case View Model*)
- Opis (dokumentacja):
 - początek i koniec przypadku
 - opis interakcji z aktorami (kiedy i co jest przesyłane)
 - dane użyte przez przypadek a zgromadzone w systemie
 - dane zapamiętywane w systemie przez przypadek użycia
 - wyjątki
 - specjalne wymagania, odpowiedzialność
- Scenariusze i/lub diagramy interakcji

Opis przypadku - przykład

- **Nazwa:** Rejestracja wypożyczeń
 - Cel: obsługa klienta, który chce wypożyczyć książkę
- **Aktor:** Klient
- **Zdarzenie inicjujące:** Zgłoszenie chęci
- **Komunikat inicjujący:** Chcę wypożyczyć (*Dane: nr karty bibliotecznej, dane książki*)
- **Komunikat do aktora:**
 - Nie mamy takiej książki
 - Wszystkie egzemplarze wypożyczone
 - Nie możesz wypożyczyć – wypożyczyłeś za dużo książek (*Dane: liczba książek*)
 - Nie możesz wypożyczyć – przekroczony termi zwrotu (*Dane: dane książki*)
 - Książka zostaje wypożyczona (*Date do zapisu: nr karty, sygnatura książki, data zwrotu*)
- **Scenariusz**

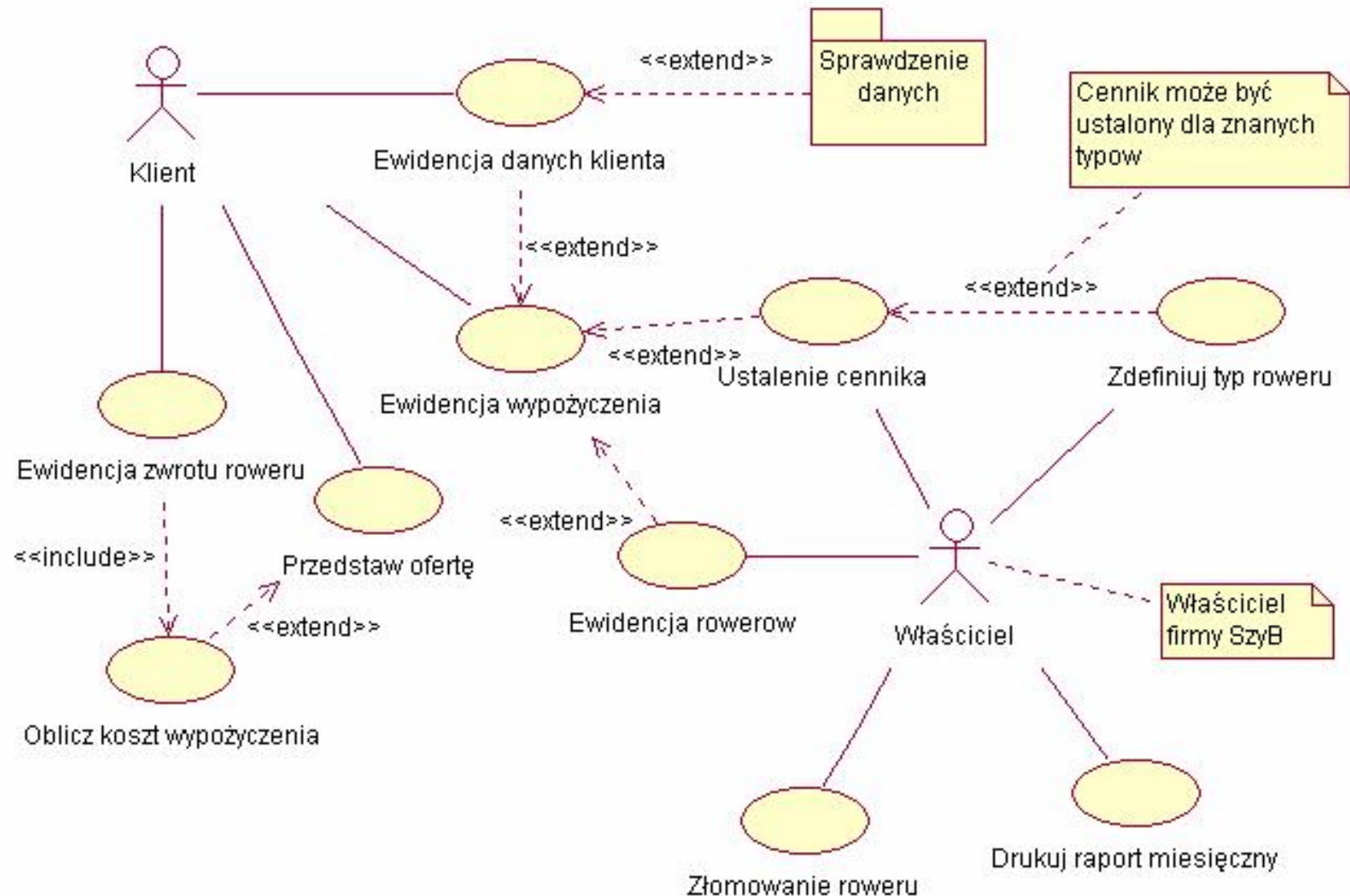
Zadania modelu przypadków użycia (1)

- Odwzorowanie systemu informatycznego z punktu widzenia użytkownika
- Prawidłowe ustalenie wymagań funkcjonalnych, w tym i wyjątków
- Lepsze zrozumienie możliwości wykorzystania systemu, zwiększenie świadomości celów wśród analityków i decydentów
- Ustalenie praw dostępu do funkcji

Zadania modelu przypadków użycia (2)

- Możliwość zdefiniowania przypadków testowych dla systemu (przebiegi podstawowe, alternatywne i wyjątki)
- Model przypadków użycia jest statyczny - nie pokazuje kolejności użycia - dynamika uwzględniania jest na diagramach interakcji

Przykład - fragment Szyb



Diagramy interakcji (*Interaction Diagrams*)

- Cel: modelowanie dynamicznych aspektów systemu
- Przedstawia zbiór obiektów systemu i związków między nimi
- Diagramy interakcji:
 - diagram sekwencji, przebiegu (*Sequence Diagram*)
 - diagram kooperacji (*Collaboration Diagram*)

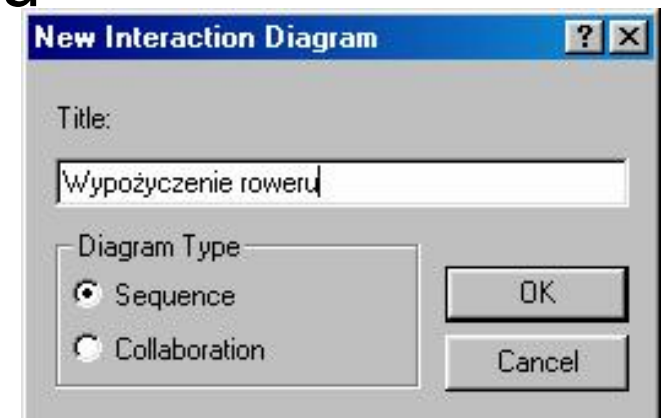
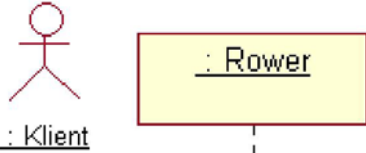
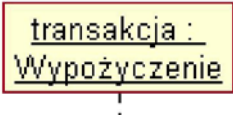
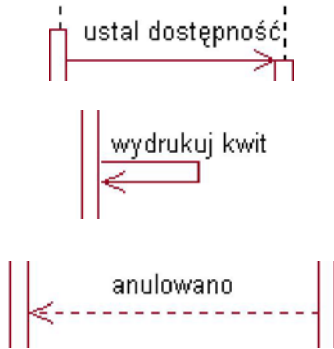




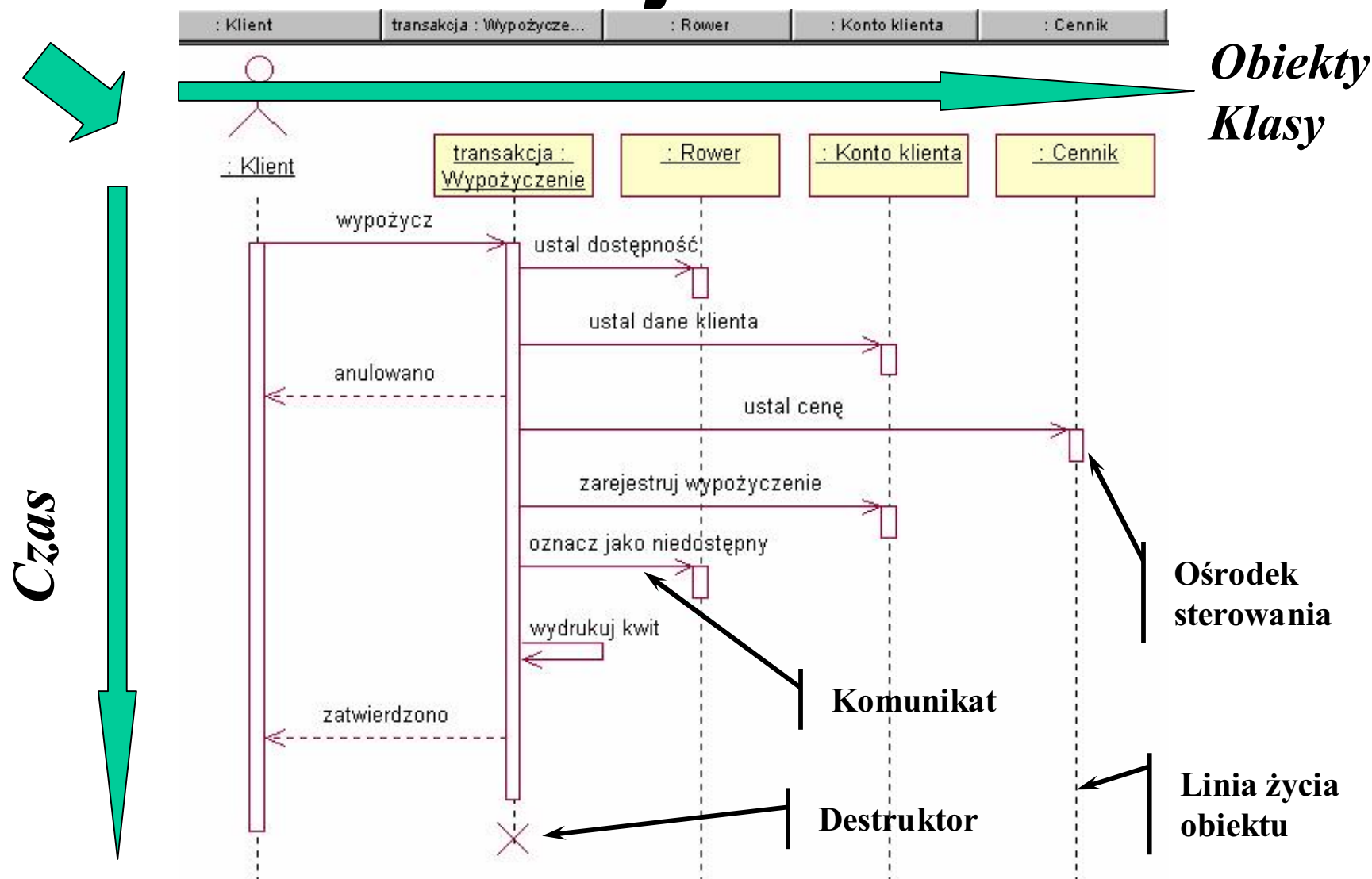
Diagram sekwencji

- Uwypukla kolejność komunikatów w czasie przesyłanych pomiędzy obiektami
- Komunikat to żądanie wykonania jednej z metod klasy (wywołanie jednej z metod klasy)
- Komunikat może być wysłany z otoczenia systemu lub z wnętrza innej metody. Wysłanie komunikatu nie kończy działania metody.
- Komunikat może przekazywać dane we do wywoływanej metody

Notacja diagramów sekwencji

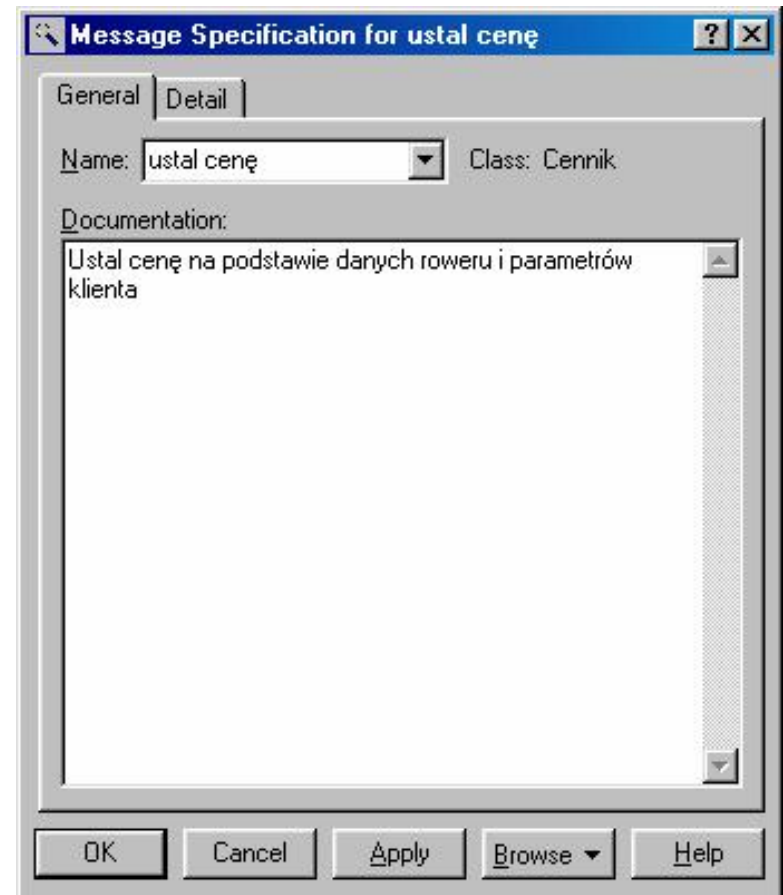
	Klasa	<i>Class</i>
	Obiekt	<i>Object</i>
	Komunikat obiekt-obiekt obiekt-sam zwrotny inne	<i>Message</i>
	Ośrodek sterowania	<i>Focus of Control</i>
	Destruktor	<i>Destruction Marker</i>

Notacja diagramów sekwencji - kolejność



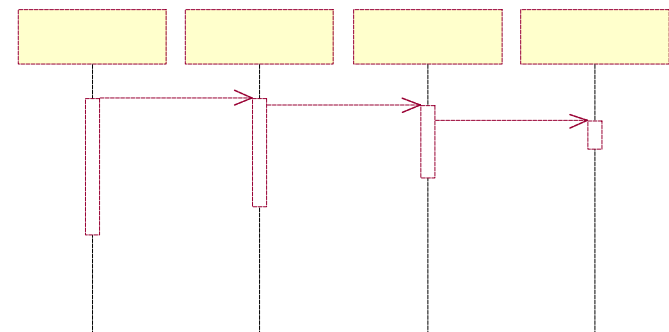
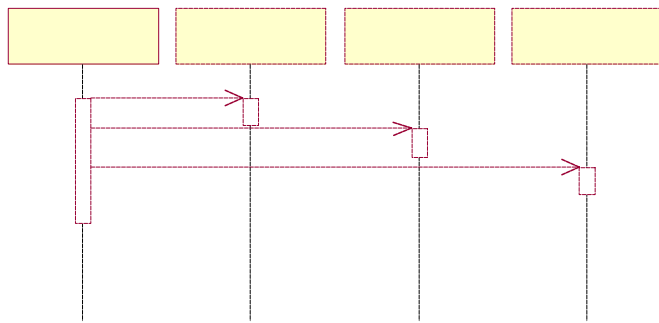
Dokumentacja komunikatu

- Należy dokumentować diagramy sekwencji poprzez dokumentowanie komunikatów (tzw. skrypty)



Sposoby wsp ółpracy obiektów

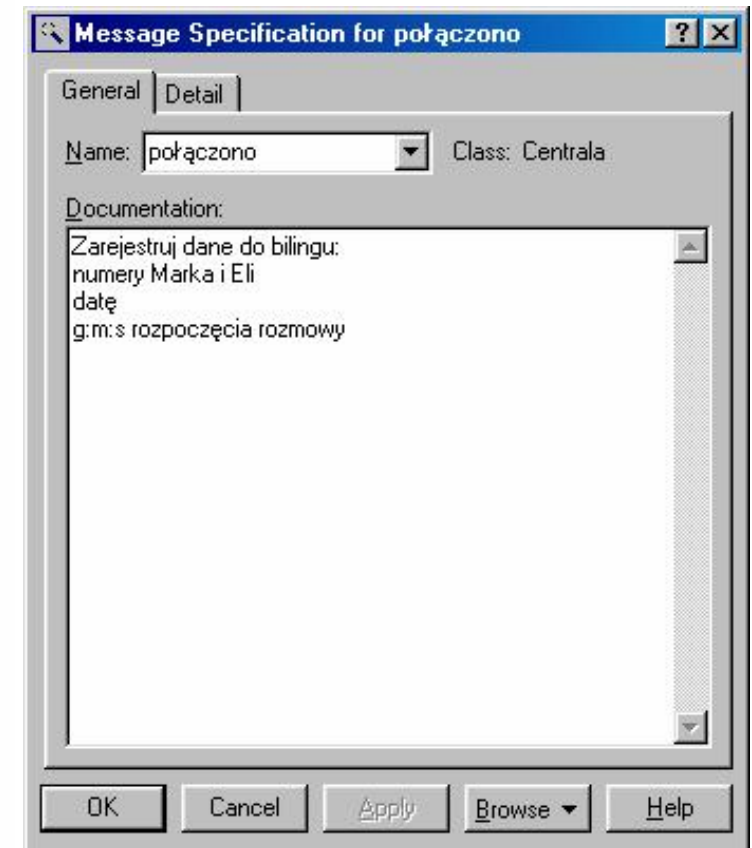
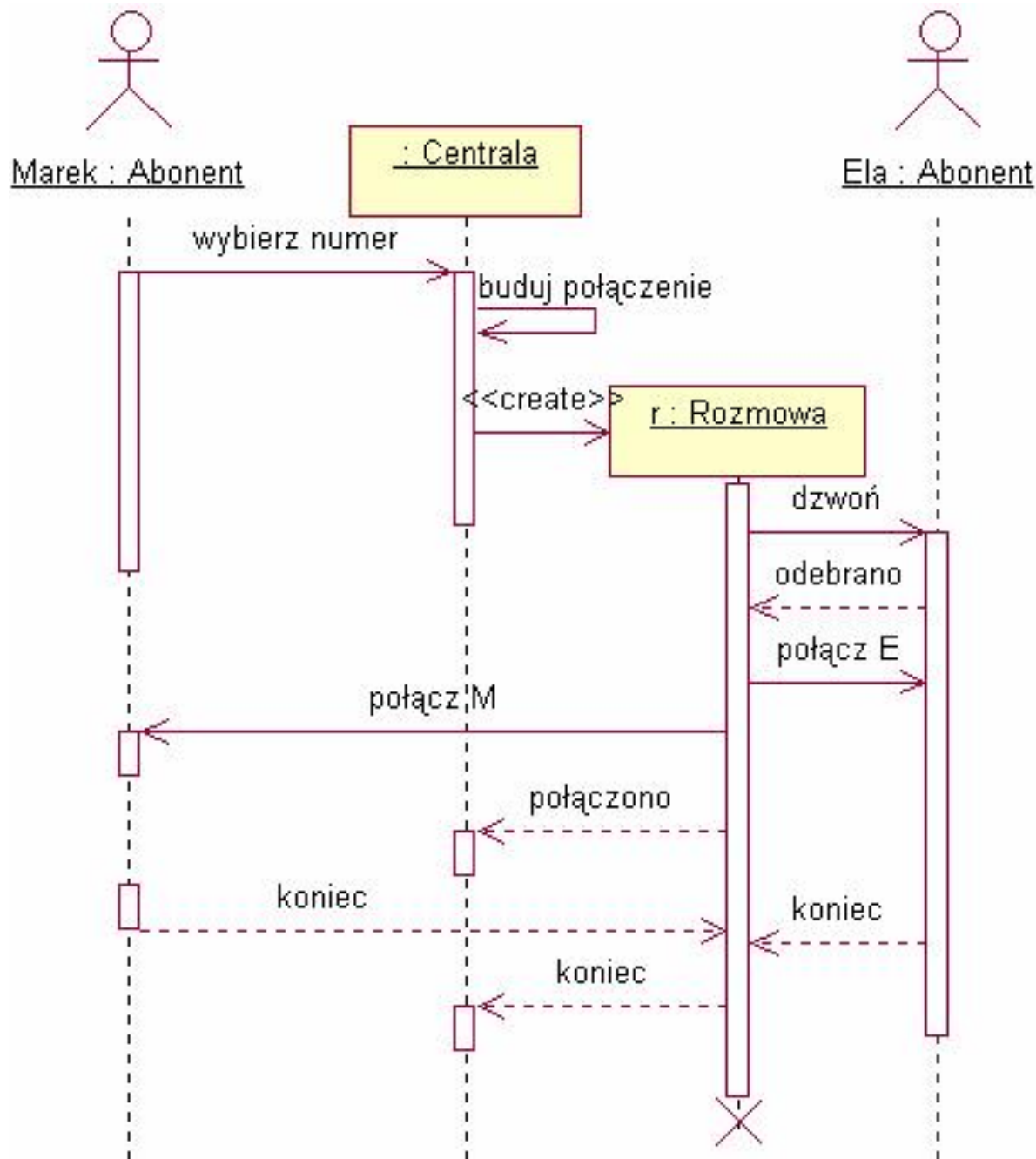
- Scentralizowany – jeden obiekt kontroluje cały przebieg przypadku użycia (wysyła komunikaty)
- Zdecentralizowany – obiekty komunikują się bezpośrednio



Ograniczenia diagramów sekwencji

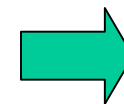
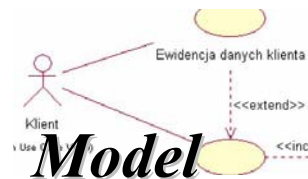
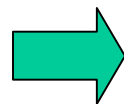
- Opisują potencjalny (ale nieobowiązkowy) scenariusz działań w trakcie realizacji funkcji systemu
- Brak na nim danych o:
 - warunku wysłania lub nie komunikatu
 - liczby wysłanych komunikatów
- Komunikat to nie:
 - dane
 - zdarzenie

Diagram sekwencji - przykład



Podsumowanie

- Zorientowane obiektowo podejście do specyfikacji wymagań użytkownika wypracowało specyficzne metody i techniki:
 - Scenariusze użycia SI
 - Model przypadków użycia
 - Diagram sekwencji
- Są one ukierunkowane na użytkownika, na jego sposób widzenia i pracy z SI, jego potrzeby, jego obszar pojęciowy
- Umożliwiają tworzenie formalnego modelu wymagań na podstawie opisu funkcjonalności SI oraz śledzenie go aż do poziomu kodu programu



Program