

**Politechnika  
Warszawska**

## Wymagania funkcjonalne

dr hab. Michał Śmiałek, prof. uczelni



**Wydział  
Elektryczny**  
POLITECHNIKA WARSZAWSKA

1DI1615: Inżynieria wymagań oprogramowania



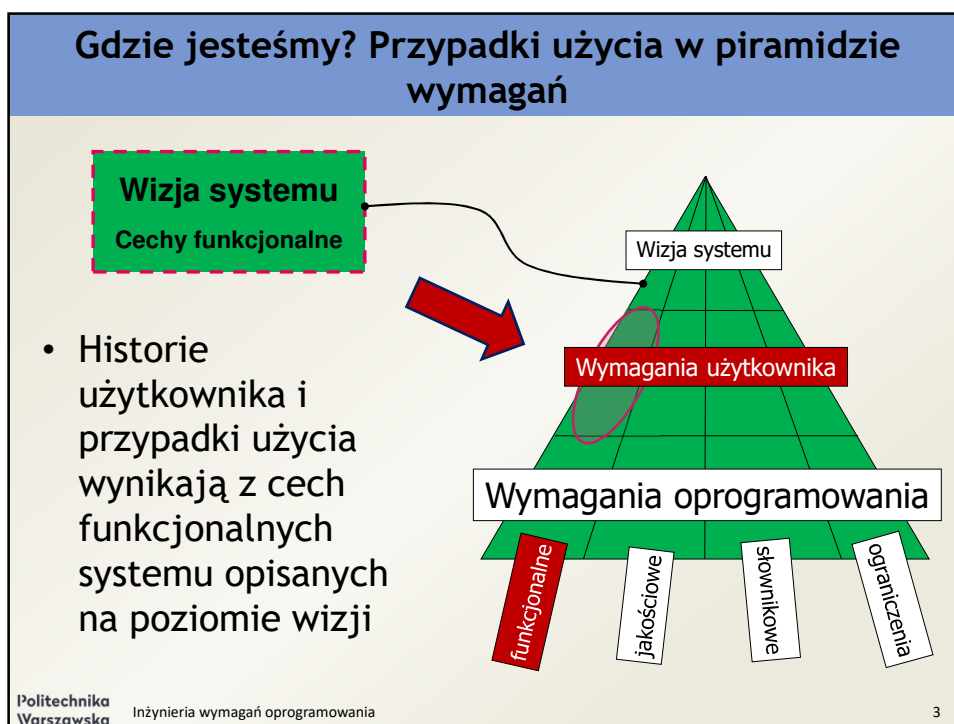
1

### Treść wykładu

- Identyfikacja jednostek funkcjonalnych
- Historie użytkownika
- Przypadki użycia i aktorzy
- Relacje między przypadkami użycia
- Opisy jednostek funkcjonalnych



2



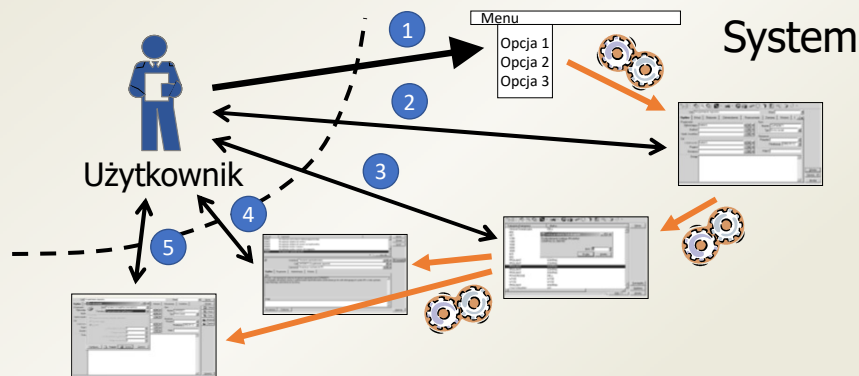
3



4

## Jak podzielić funkcjonalność systemu na „kawałki”?

- Musimy zidentyfikować małe, zamknięte fragmenty funkcjonalności. Co może stanowić taką jednostkę funkcjonalności?



## Czym są jednostki funkcjonalne?

- Zamknięta całość - realizacja kompletnego, choć niewielkiego celu biznesowego
- Umożliwienie określenia zakresu funkcjonalnego
- Typowe rodzaje jednostek funkcjonalnych
  - Historie użytkownika
  - Przypadki użycia
- Łatwość opisu i dyskusji z klientem
- Widok systemu z punktu widzenia użytkowników

### Cechy jednostek funkcjonalnych

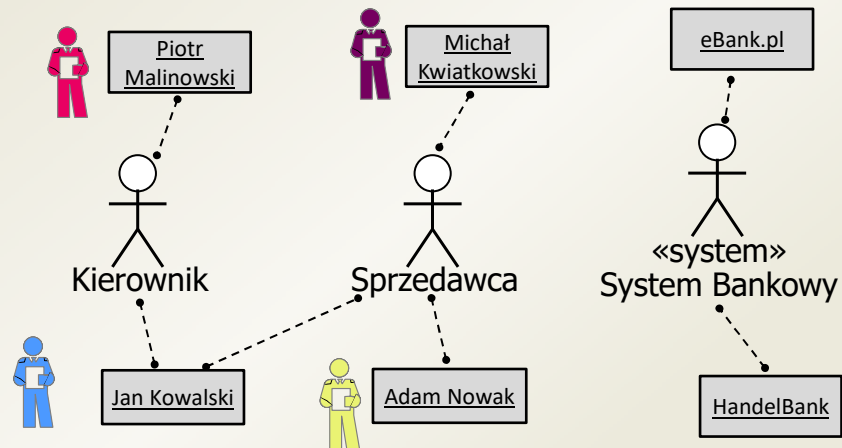
- Niezależność - realizacja funkcjonalności niezależnie od innych
- Negocjowalność - możliwość ustalania szczegółów między klientem a wykonawcą
- Cennaść dla klienta - korzyść, jaką klient odniesie;
- Szacowalność - racjonalne oszacowanie rozmiarów oraz kosztu wykonania
- Mały rozmiar - realizacja w stosunkowo krótkim czasie (kilka dni do kilku, np. 2-3 tygodni)
- Testowalność - możliwość udowodnienia spełnienia wymagania poprzez wykonanie testów

### Pierwszy krok - identyfikacja użytkowników

- Aktor
  - Klasa obiektów **SPOZA** modelowanego systemu
  - Obiekty mogą być ludźmi lub zewnętrznymi systemami
  - Komunikuje się z systemem za pośrednictwem interfejsu (interfejsów), np. GUI lub API
  - Rola grana przez obiekty w stosunku do systemu: dla aktora istnieje zestaw dostępnych przypadków użycia
- Rola użytkowników
  - Stosowana w metodykach zwinnych
  - Podobne znaczenie co aktor

## Aktorzy i obiekty

- Aktor może mieć wiele instancji (obiektów)
- Dany obiekt (osoba, system) może wchodzić w rolę kilku aktorów

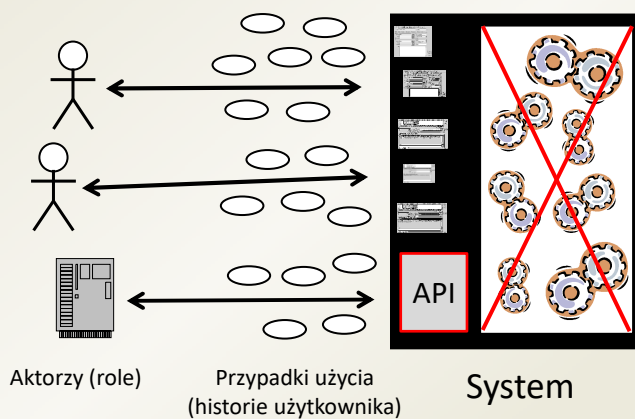


## Metody identyfikacji aktorów

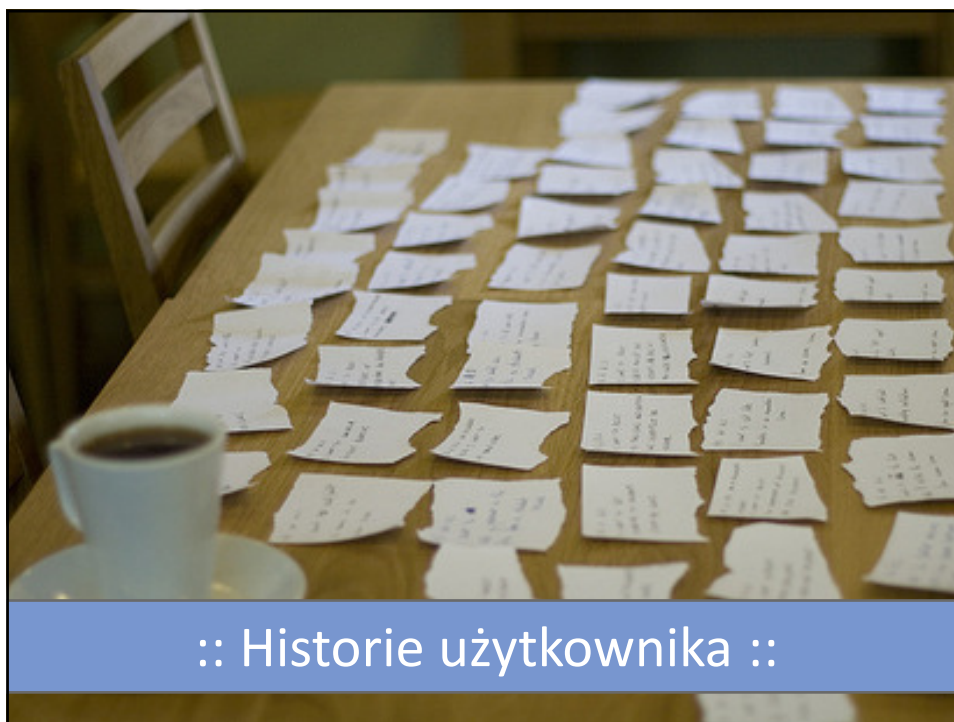
- Zastanów się nad cechami aktora
  - przyczyna korzystania z systemu
  - doświadczenie w ramach danej dziedziny problemu
  - biegłość w posługiwaniu się aplikacjami komputerowymi
  - możliwość nabrania biegłości w posługiwaniu się budowanym systemem
  - częstość korzystania z systemu
  - relacja do rzeczywistych stanowisk pracy czy potocznie przyjętych określeń zakresu odpowiedzialności
- Stwórz słownik aktorów
  - Każdy aktor posiada krótki opis uwzględniający jego cechy

## Metody identyfikacji jednostek funkcjonalnych

- Proces twórczy (techniki pracy grupowej itp.)
- Traktujemy system jak czarną skrzynkę



11



12



## Co to są historie użytkownika?

- Spopularyzowane przez metodykę XP w latach 90-tych
  - Gra planowania
- Pojedyncze potrzeby klienta
  - Prosty, krótki opis cechy systemu z punktu widzenia użytkownika/klienta systemu
  - NIE powinny dotyczyć potrzeb deweloperów (technologia)
- Bardzo prosta notacja
  - Dają się zapisać na jednej stronie karty katalogowej
  - Opis zazwyczaj stanowi jedno zdanie
  - Pobudzają dyskusję
- Więcej szczegółów:
  - Mike Cohn - User Stories Applied

### Wyświetlenie kursów

Jako student chcę wyświetlić opisy kursów, aby móc zorientować się jaki będzie mój zakres studiów.

## Struktura historii użytkownika

- Odpowiedź na trzy pytania:
  - Kto? - identyfikacja roli użytkownika
  - Co? - cel, który użytkownik chce osiągnąć
  - Dlaczego? - jaki efekt spowoduje osiągnięcie celu
- Wzorce pisania historii użytkownika
  - **Jako <rodzaj użytkownika> chcę <jakiś cel>, aby <jakiś powód>**
  - Jako klient chcę złożyć zamówienie na samochód, aby rozpocząć proces jego zakupu
  - **<rodzaj użytkownika> chce <jakiś cel>, aby <jakiś powód>**
  - Klient chce złożyć zamówienie na samochód
- Pojedynczy, elementarny cel osiągany na rzecz użytkownika

## Przykłady

Jako **kierownik** chcę wprowadzić nowy samochód, aby klienci go mogli zobaczyć i kupić.

Jako **klient** chcę przejrzeć listę dostępnych samochodów, aby móc zdecydować się na zakup.

Jako **sprzedawca** chcę wyszukać klienta, aby uzyskać o nim informacje sprzedażowe.

Jako **klient** chcę sprawdzić status zamówienia, aby móc zaplanować odbiór samochodu.

Jako **sprzedawca** chcę móc obsłużyć płatność przez klienta, aby szybko sfinalizować zamówienie

Jako **serwisant** chcę zarejestrować brak części, aby móc dokończyć naprawę samochodu.

## Przykłady (jeszcze więcej)

Jako nauczyciel chcę wprowadzić nowy kurs, aby studenci go mogli zobaczyć.

Jako student chcę wyświetlić opisy kursów, aby móc zorientować się jaki będzie mój zakres studiów.

Jako nauczyciel chcę usunąć swój kurs, aby zrezygnować z jego prowadzenia.

Jako student, chcę zapisać się na kurs, aby móc podnieść swoje umiejętności.

Jako administrator chcę usunąć dowolny kurs, aby nie były wyświetlane kursy już nieaktualne.

Jako student, gdy chcę zapisać się na powtarzany kurs, muszę dokonać opłaty, aby zmobilizować się do nauki.






17

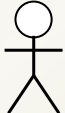
### Aktorzy - współpracownicy systemu

- Oznaczenie aktora
  - Ikona człowieka
  - Stereotyp nadaje dodatkowe znaczenie
  - Dobra praktyka: aktorów będących systemami oznaczamy odpowiednią ikoną



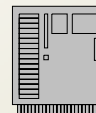
ikona podstawowa

Sprzedawca



ikona ze stereotypem

«system»  
System Bankowy



System Bankowy

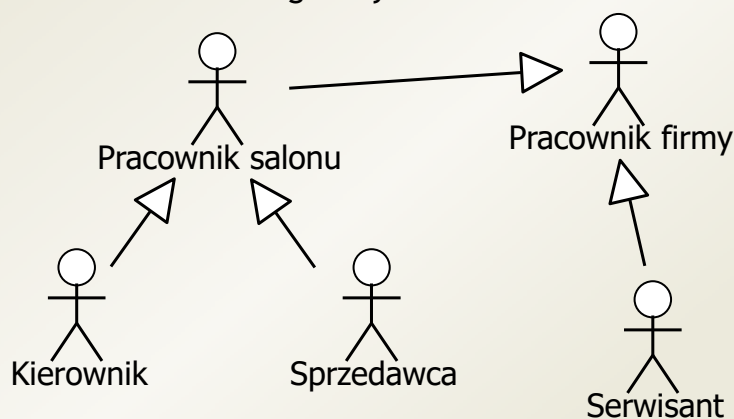
stereotyp wyrażony zmienioną ikoną

Politechnika Warszawska
Inżynieria wymagań oprogramowania
18

18

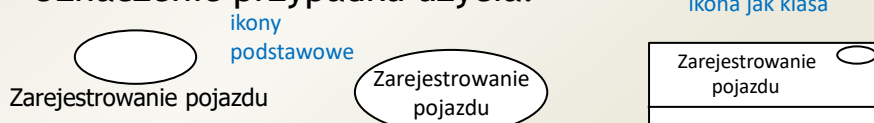
## Generalizacje dla aktorów

- Aktorzy mogą wchodzić w relacje generalizacji
  - Aktor ogólny - wspólny zakres odpowiedzialności dla kilku aktorów szczegółowych



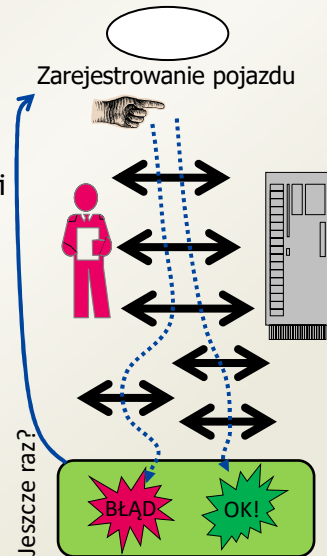
## Przypadki użycia systemu

- Definicja przypadku użycia:
  - Klasa zachowań modelowanego systemu (*podmiotu*) prowadzących do osiągnięcia obserwowalnego **REZULTATU** dla jakiegoś aktora (lub aktorów)
  - Zazwyczaj obejmuje różne warianty zachowania systemu
  - Nie odnosi się do wewnętrznej struktury systemu
  - Zawiera tylko opis obserwowalnych interakcji między systemem i środowiskiem (aktorami)
  - Może prowadzić do zmiany stanu systemu
- Oznaczenie przypadku użycia:



## Przypadek użycia - kompletny opis zachowania

- Każdy przypadek użycia ma cel
  - Cel zazwyczaj wynika z nazwy
  - Zachowanie prowadzi do celu
- Cel realizowany jest poprzez wymianę komunikatów
  - Komunikaty aktor-system, odpowiedzi systemu
- Zachowania alternatywne
  - Osiągnięcie celu inną drogą lub porażka
- Po zakończeniu, system musi być w stanie stabilnym
  - Można uruchomić ten sam lub inny p.u.
- Uruchomienie przez aktora
  - Najczęściej pierwszy komunikat jest od aktora do systemu



Politechnika  
Warszawska

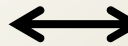
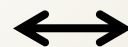
Inżynieria wymagań oprogramowania

21

21

## Przypadek użycia - o czym pamiętać?

1. Rozpoczyna się od interakcji aktora z systemem (zazwyczaj)
2. Zawiera sekwencję interakcji aktora z systemem
  - NIE pojedyncza funkcja!
3. Dąży do osiągnięcia wymiernego celu (rezultatu biznesowego)
  - Wartość dla aktora (np. pozyskanie danych, wprowadzenie danych, zmiana stanu)
  - Alternatywy mogą się kończyć niepowodzeniem na drodze do celu



Politechnika  
Warszawska

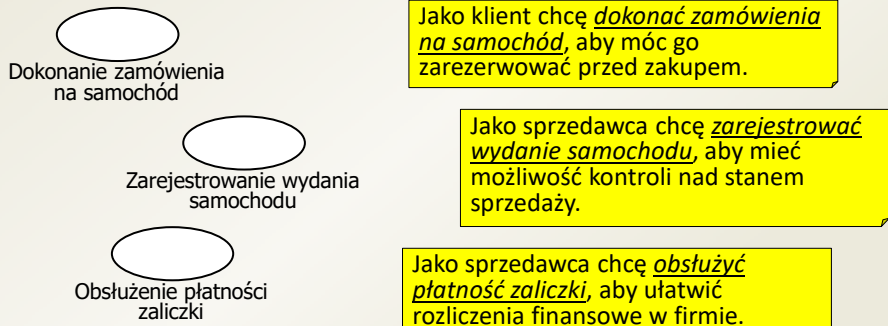
Inżynieria wymagań oprogramowania

22

22

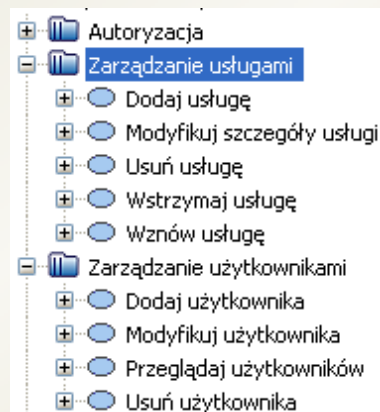
## Przypadki użycia i historie użytkownika

- Historie użytkownika i przypadki użycia stanowią równoważny opis wymagań funkcjonalnych
  - Koncentracja na korzyści dla konkretnego aktora/użytkownika
  - Przed implementacją konieczny dokładniejszy opis (scenariusze, scenopisy)



## Pakiety przypadków użycia jako eposy

- Pakiety przypadków użycia porządkują po kilka-kilkanaście p.u. w logiczne grupy
- Pakiet odpowiada historii użytkownika na poziomie eposu
- Eposy obejmują wiele „zwykłych” historii użytkownika
- Dekompozycja problemu ułatwia radzenie sobie ze złożonością specyfikacji
- Sposób organizacji wymagań w pakietach może w przyszłości ułatwić zaprojektowanie systemu

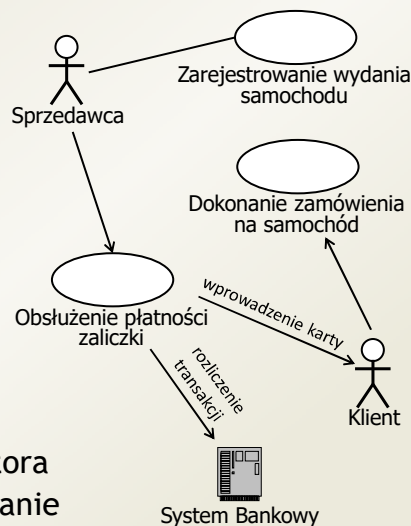


Jako kierownik chcę zarządzać usługami, aby móc dostarczać usługi klientom.

Jako administrator chcę zarządzać użytkownikami, aby udzielać dostępu do systemu.

## Asocjacje aktorów i przypadków użycia

- Uczestnictwo aktora w przypadku użycia
  - Prosta asocjacja
  - Krotności (zazwyczaj 1-1)
- Aktorzy główni
  - Asocjacja skierowana do przypadku użycia
  - Aktor rozpoczyna i oczekuje rezultatu
- Aktorzy poboczni
  - Asocjacja skierowana do aktora
  - Aktor jest proszony o wykonanie akcji w trakcie



Politechnika  
Warszawska

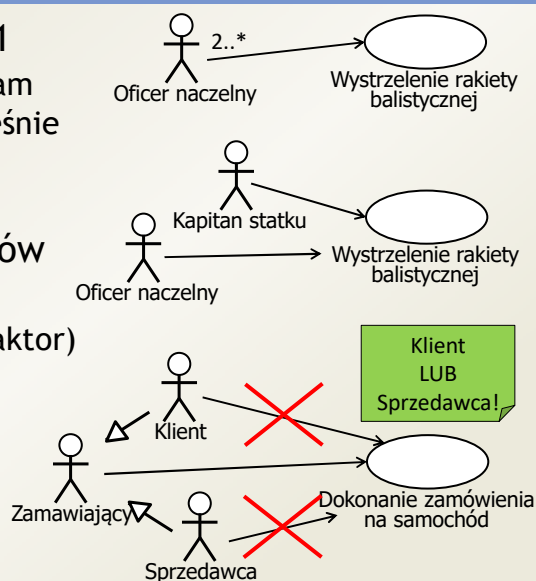
Inżynieria wymagań oprogramowania

25

25

## Aktorzy i przypadki użycia

- Przykład krotności >1
    - np. kilka osób (ten sam aktor) musi jednocześnie rozpocząć przypadek użycia
  - Przykład wielu aktorów głównych
    - np. kilka osób (inny aktor) musi jednocześnie rozpocząć przypadek użycia
- **Uwaga na błędy!**



Politechnika  
Warszawska

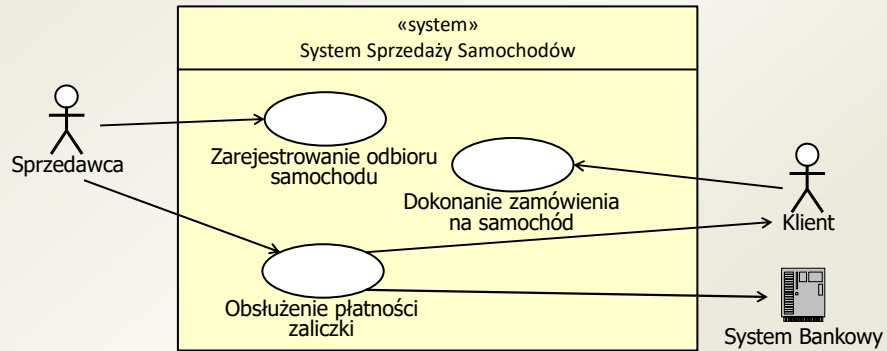
Inżynieria wymagań oprogramowania

26

26

## Podmiot dla przypadków użycia

- Podmiotem jest jakiś system (podsystem)
  - Przypadki użycia realizowane przez system
  - Ikona przypadku użycia zawarta w ikonie systemu
  - Wyraźne zaznaczenie bariery system-środowisko



Politechnika  
Warszawska Inżynieria wymagań oprogramowania

27

27



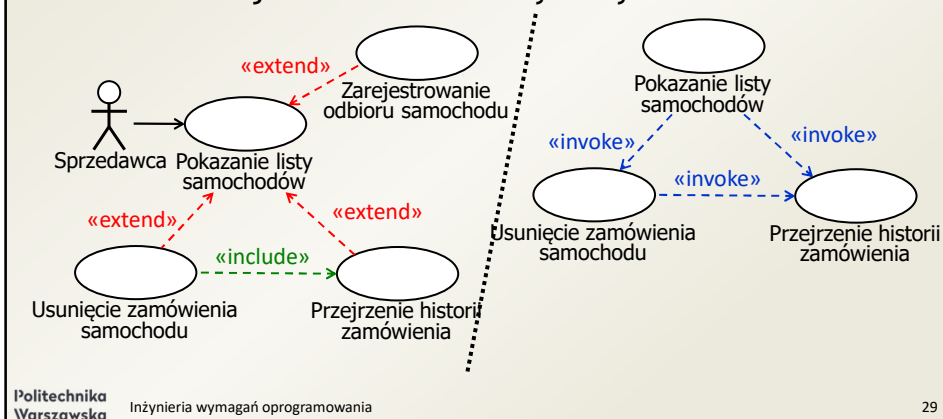
:: Relacje przypadków użycia ::

28



## Relacje między przypadkami użycia

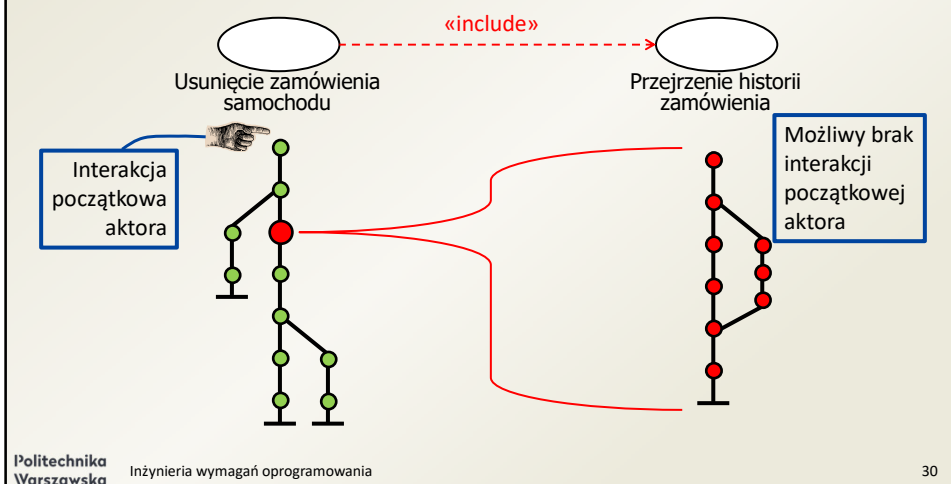
- Standardowe relacje UML
  - Include i Extend - semantyka wywołania/przeplatania
- Preferowane rozszerzenie UML
  - Invoke - ujednolicona semantyka wywołania



29

## Relacja Include (włączenie)

- Zachowanie włączanego p.u. wstawiane bezwarunkowo w zachowanie włączającego

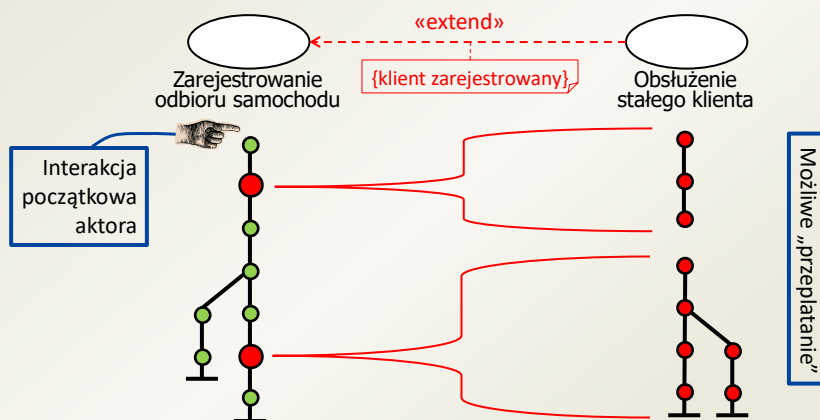


30



## Relacja Extend (rozszerzenie)

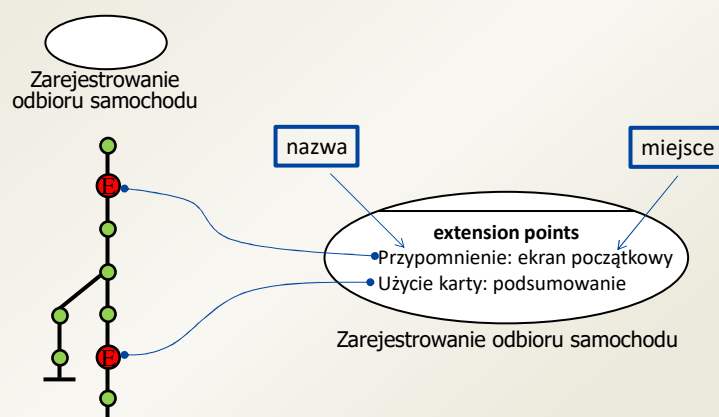
- Zachowanie rozszerzającego p.u. warunkowo wplecione w zachowanie p.u. rozszerzanego



31

## Punkty rozszerzenia

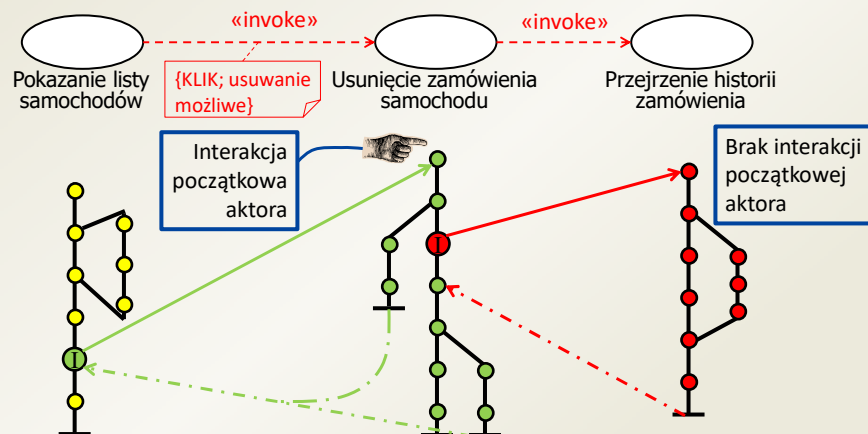
- Informacja na temat możliwych punktów wplecenia zachowania rozszerzającego



32

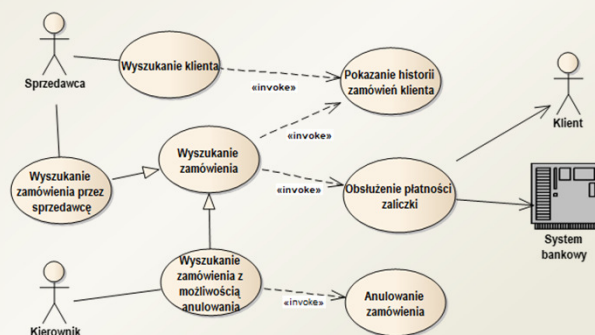
## Relacja Invoke (wywołanie)

- Zachowanie wywoływane warunkowo lub bezwarunkowo (poprzez tzw. zdania „invoke”)



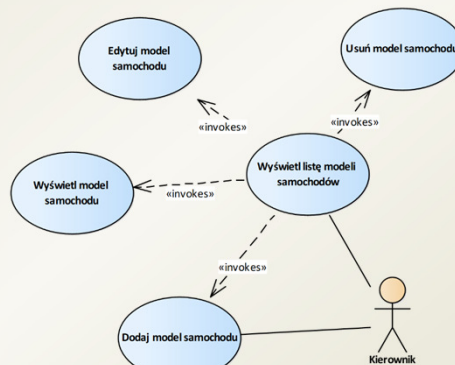
## Relacje generalizacji

- Przypadki użycia specjalizujące dziedziczą cechy przypadku użycia ogólnego
  - Uwaga na prawidłowe stosowanie relacji generalizacji
  - Specjalizacja oznacza możliwość uruchamiania „potomków” z poziomu „przodka”



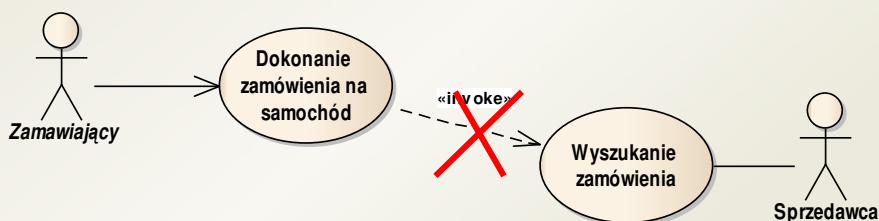
## Wzorce przypadków użycia

- Powtarzalne konfiguracje przypadków użycia
- Wzorzec zawiera nieduży fragment modelu przypadków użycia z odpowiednimi relacjami
- Przykład - wzorzec CRUD



## Błędy w modelowaniu relacji

- Relacje między przypadkami użycia NIE oznaczają następstwa czasowego!
  - Kolejność wykonania przypadków użycia można pokazać używając diagramu czynności lub modelu procesu w języku BPMN





:: Opisy jednostek funkcjonalnych ::

37

### Kompletny opis historii użytkownika

- Karta
  - jedno zdanie, mieszczące się na typowej karcie katalogowej, zapisane zgodnie z wzorem typu „Jako ... chcę ... aby ...”
- Konwersacja
  - dodatkowy opis historii użytkownika, który jest wynikiem dyskusji nad szczegółami wpływającymi na jej implementację; zapisana w swobodnym języku naturalnym
- Konfirmacja (potwierdzenie)
  - zestaw kilku testów akceptacyjnych, które miałyby potwierdzić prawidłowość implementacji historii użytkownika; opisane w sposób skrótowy - mieszczą się na drugiej stronie karty katalogowej

38

### Przykłady konwersacji

- Historia użytkownika „Jako serwisant chcę zarejestrować brak części, aby móc dokończyć naprawę samochodu”
  - Serwisant powinien móc wybrać parametry części z dostępnej listy.
  - Po wpisaniu częściowego numeru części system powinien odpowiedzieć cały symbol na podstawie katalogu części.
  - Zgłoszenie braku części powinno być porównane ze stanem magazynu części.
- Forma „historyki”
  - Jako serwisant chcę wybrać parametry części z dostępnej listy, aby nie musieć pamiętać wszystkich parametrów.

### Przykłady confirmacji (testów)

- Testy na tym poziomie piszemy w sposób uproszczony
  - Lista parametrów wyświetlona serwisantowi powinna zawierać wszystkie parametry wprowadzone podczas konfiguracji parametrów.
  - Gdy serwisant wprowadzi kilka liter symbolu, powinien otrzymać odpowiedzi zgodne z aktualnym stanem katalogu części.
  - Jeśli serwisant zgłosi brak części, a dana część jest w magazynie, powinno zostać pokazane ostrzeżenie.

## Opis przypadku użycia

- Opis podobny do konwersacji
- Warunek rozpoczęcia (1), warunki zakończenia (kilka)



{ **Warunek rozpoczęcia:** Serwisant jest zalogowany do systemu ORAZ Serwisant ma uprawnienia do zgłaszania braków }

Serwisant na formularzu może wprowadzić [zgłoszenie braku części]. Podczas wprowadzania numeru [części], serwisant powinien uzyskać odpowiedź systemu na podstawie [katalogu części]. Przy zapisywaniu zgłoszenia, powinno ono być porównane ze stanem [magazynu części].

{ **Warunki zakończenia:** (sukces) => [Zgłoszenie braku części] dodane do [listy zgłoszeń braków]; (porażka) => [Lista zgłoszeń braków] nie zmieniona }

## Kompletna fiszka opisu wymagania funkcjonalnego

- Atrybuty + opis
- Fiszka generowana z narzędzia CASE lub utrzymywana jako część dokumentu

<b>ID:</b> PU232	<b>Nazwa:</b> Edycja danych samochodu w magazynie		
<b>Wersja:</b> 0.4	<b>Waga klienta:</b> ważne	<b>Waga wykonawcy:</b> średnie	
<b>ZATWIERDZONE</b>	<b>Trudność:</b> 3	<b>Wydanie:</b> 12.09	<b>Odpowiedzialny:</b> Krzysiek
<p>{ <b>Warunek rozpoczęcia:</b> Magazynier jest zalogowany ORAZ Włączona możliwość edycji magazynu }</p> <p>Po wybraniu [samochodu], magazynier może wprowadzić nowe dane [samochodu]. Zmienione dane są walidowane przez system. Zmianie podlegają jedynie [dane dodatkowe samochodu].</p> <p>{ <b>Warunki zakończenia:</b> (sukces) =&gt; dane [samochodu] zaktualizowane w [magazynie samochodów]; (porażka) =&gt; dane [samochodu] nie zmienione }</p>			