

# **Model przypadków użycia - rola diagramów przypadków użycia Część 1**

**Wykładowca**

**Dr inż. Zofia Kruczkiewicz**

# Struktura wykładu

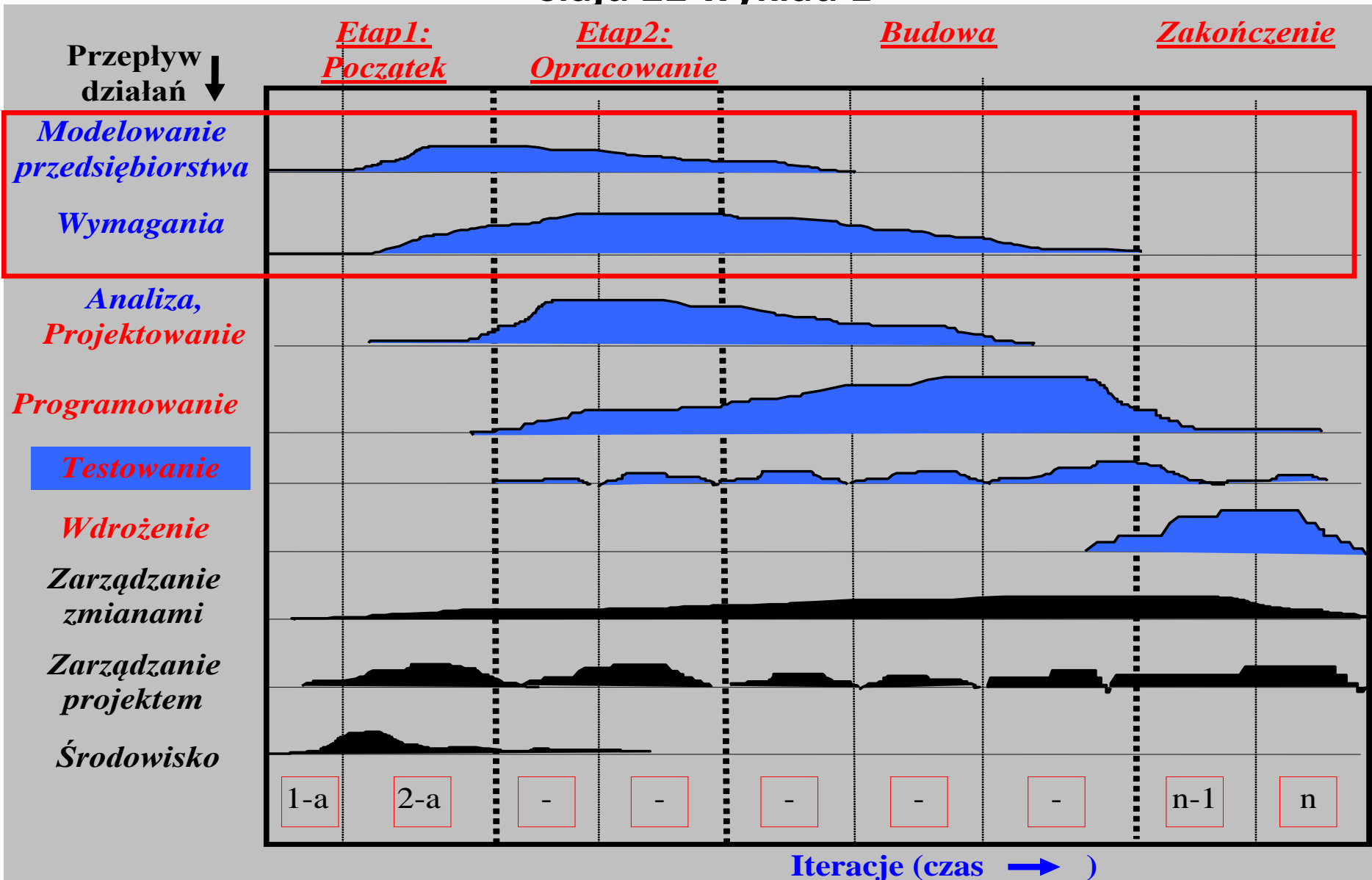
- I. **Wprowadzenie (wykład 1, wykład3)**
- II. **Elementy diagramów przypadków użycia (use-cases)**
- III. **Wytyczne tworzenia diagramów przypadków użycia (use-cases)**  
(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)
- III. **Model przypadków użycia – identyfikacja i specyfikacja wymagań (podsumowanie z wykładu 3)**
- IV. **Przykłady diagramów przypadków użycia (use-cases)**

# Tworzenie diagramów przypadków użycia

## I. Wprowadzenie (Wykład1, Wykład3)

# Proces - zunifikowany iteracyjno- przyrostowy proces tworzenia oprogramowania – **kiedy należy wykonać?** [3LU]

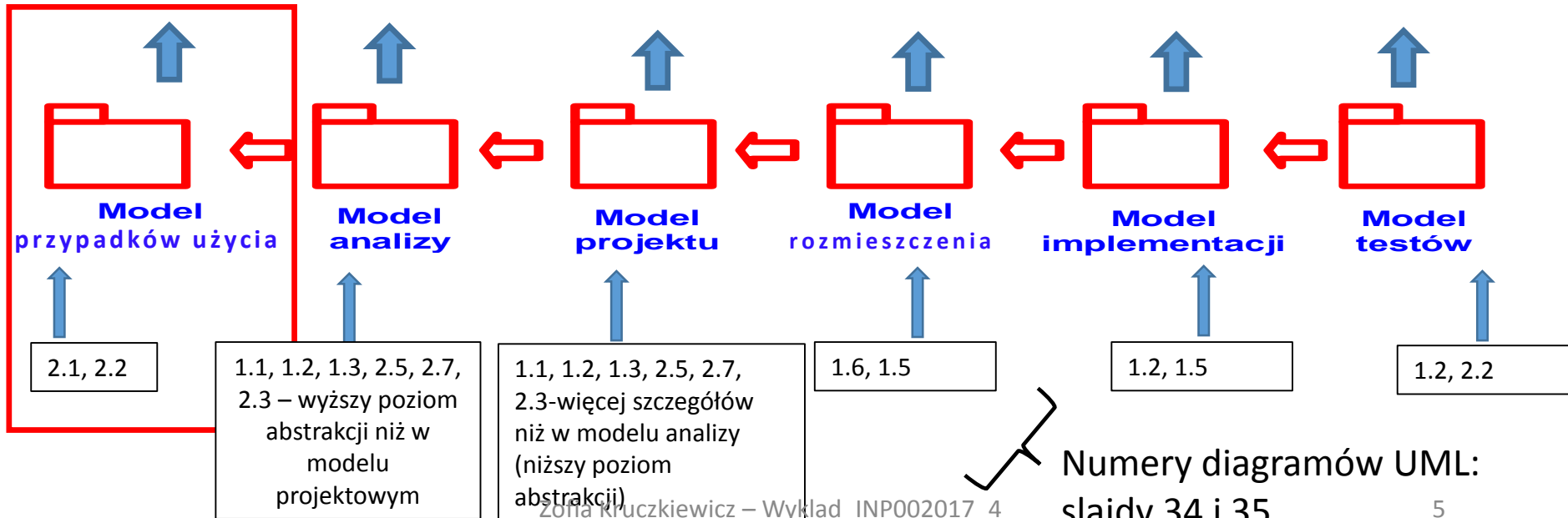
- slajd 22 wyklad 1



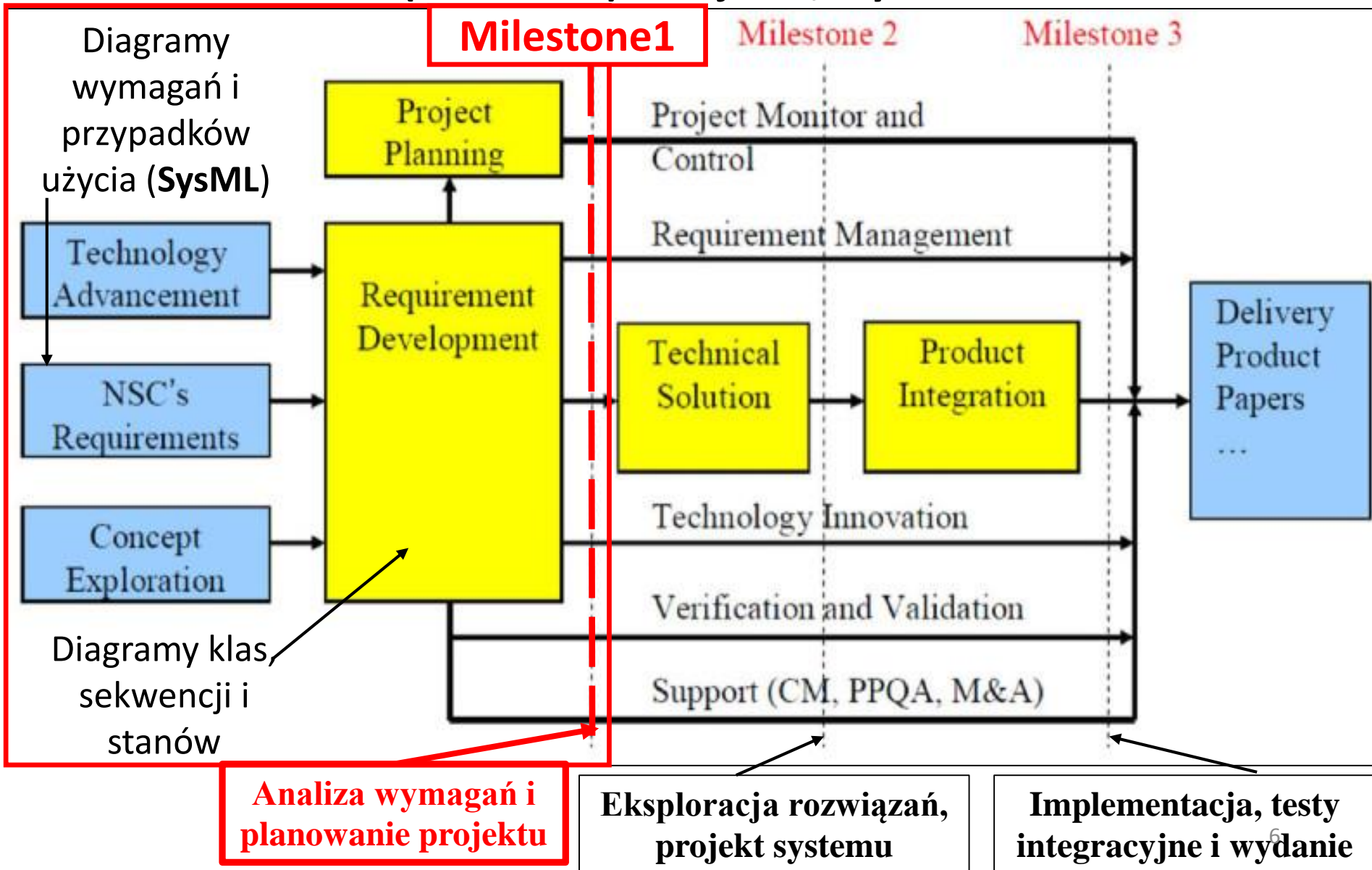
# Produkt - diagramy UML – modele, proces

## - slajd 45, wykład 1

Modelowanie struktury i dynamiki systemu	Implementacja systemu,	struktury i dynamiki generowanie kodu
Perspektywa koncepcji <i>co należy wykonać?</i>	Perspektywa specyfikacji <i>jak należy używać?</i>	Perspektywa implementacji <i>jak należy wykonać?</i>
<ul style="list-style-type: none"> <li>• model problemu np. przedsiębiorstwa</li> <li>• <u>wymagania</u></li> <li>• analiza (model konceptualny: diagram przypadków użycia, diagram klas, diagramy sekwencji,)</li> <li>• testy modelu</li> </ul>	<ul style="list-style-type: none"> <li>• <b>projektowanie</b> (model projektowy: architektura sprzętu i oprogramowania; dostęp użytkownika; przechowywanie danych)</li> <li>• <b>testy projektu</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>programowanie, wdrażanie</b> (specyfikacja programu : deklaracje, definicje; dodatkowe struktury danych: struktury „pojemnikowe”, pliki, bazy danych)</li> <li>• <b>testy oprogramowania</b></li> <li>• <b>wdrażanie</b></li> <li>• <b>testy wdrażania</b></li> </ul>



# Cykl życia tworzenia oprogramowania w dziedzinie medycyny nuklearnej: **Light-Weight Capability Maturity Model Integration (LW-CMMI)** – slajd 59, wykład 1



# Tworzenie diagramów przypadków użycia

## I. Wprowadzenie (Wykład1, Wykład3)

## II. Elementy diagramów przypadków użycia (use-cases)

# Diagramy UML 2 – część pierwsza

Na podstawie

## UML 2.0 Tutorial

<https://sparxsystems.com/resources/tutorials/uml2/use-case-diagram.html>

<https://sparxsystems.com/resources/tutorials/uml/use-case-model.html>



# Dwa rodzaje diagramów UML 2

## Diagramy UML modelowania strukturalnego

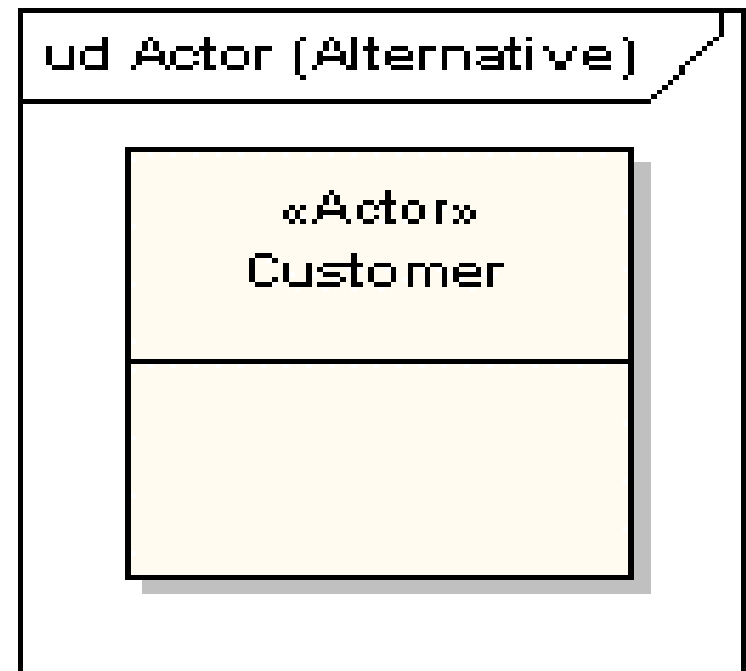
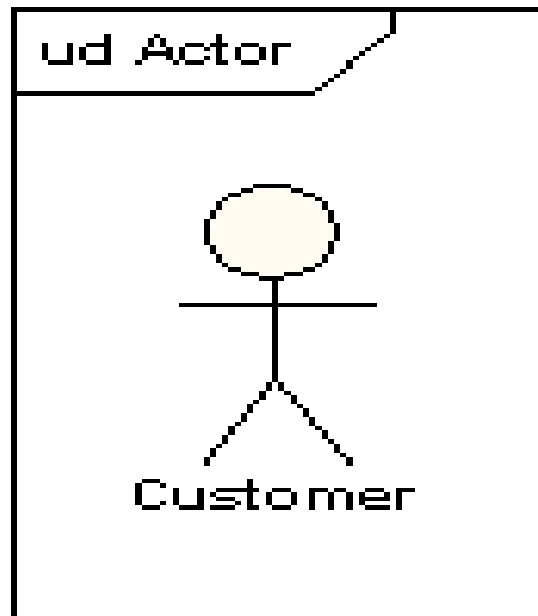
- Diagramy pakietów
- Diagramy klas
- Diagramy obiektów
- Diagramy mieszane
- Diagramy komponentów
- Diagramy wdrożenia

## Diagramy UML modelowania zachowania

- *Diagramy przypadków użycia*
- Diagramy aktywności
- Diagramy stanów
- Diagramy komunikacji
- Diagramy sekwencji
- Diagramy czasu
- Diagramy interakcji

# Diagramy przypadków użycia (*Use Case Diagram*)

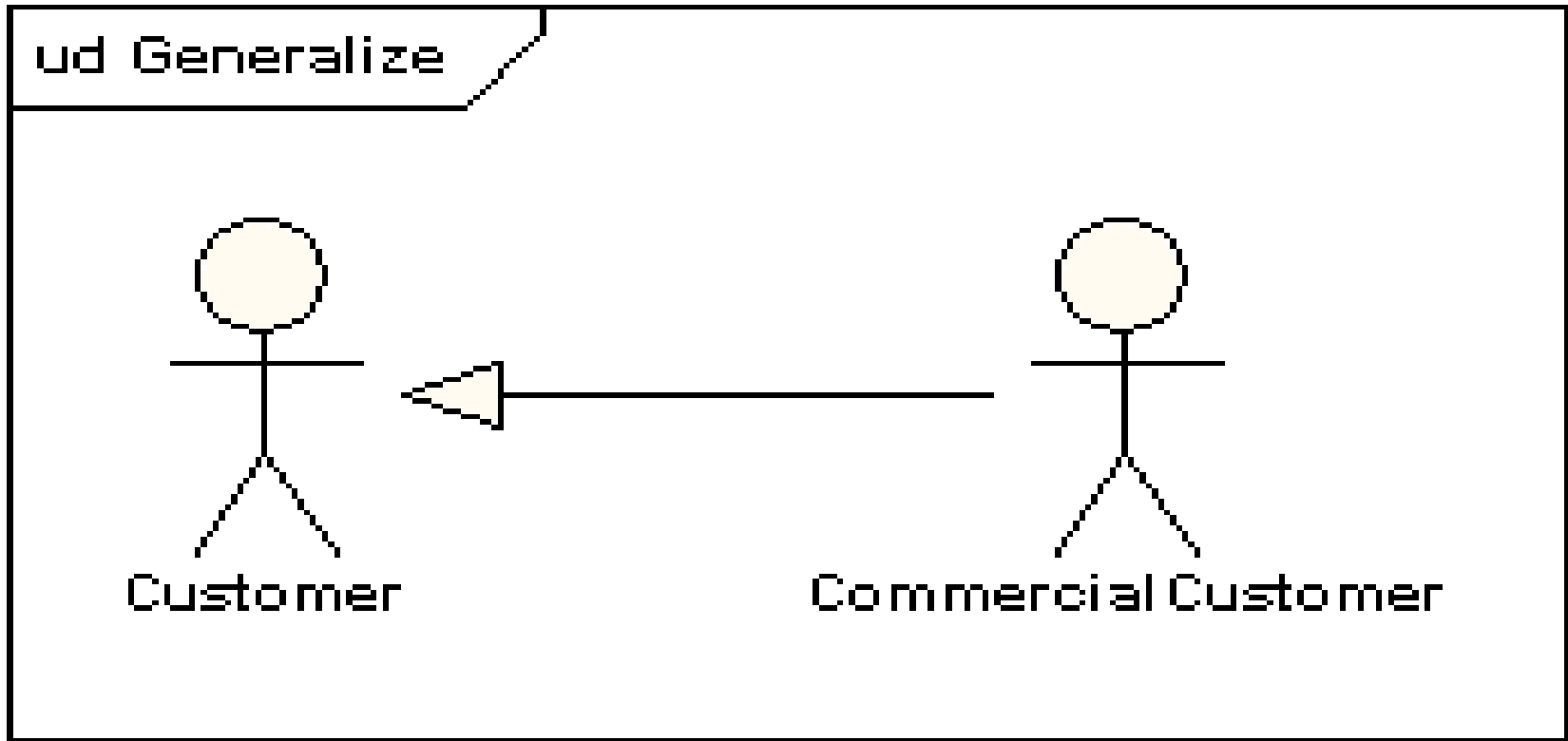
- **Diagramy przypadków użycia** opisują wymagania systemu
- **Przypadki użycia (Use cases)** oznaczają funkcje udostępniane użytkownikom lub innym zewnętrznym systemom (**actors**) przez projektowany system



## Actors

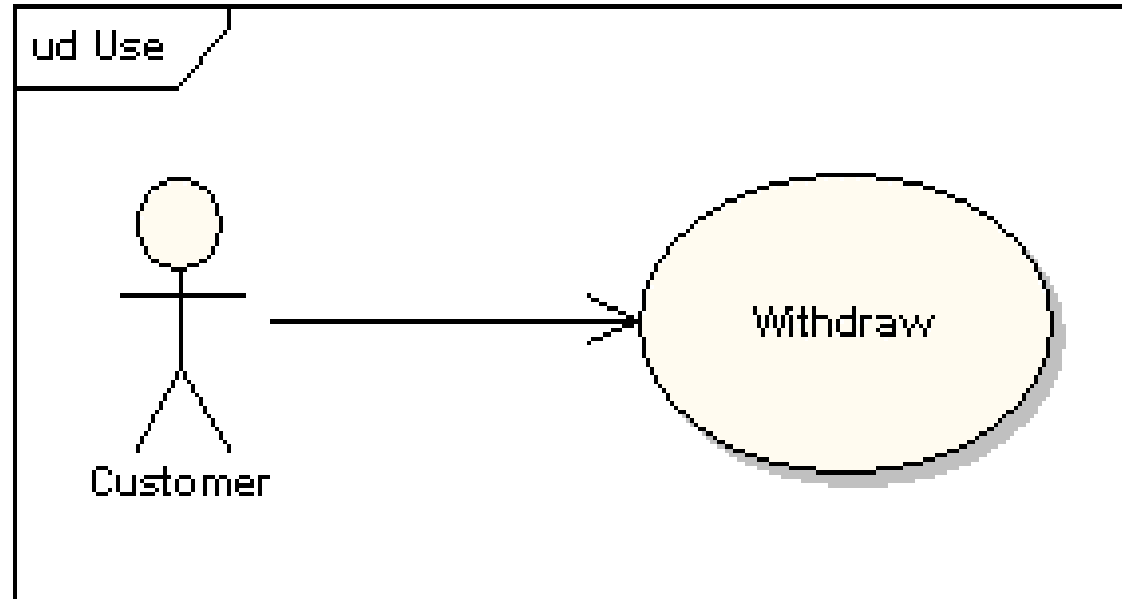
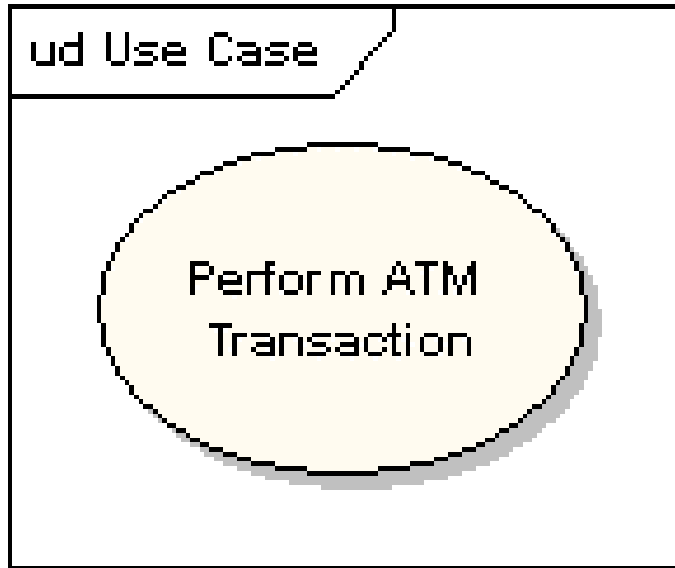
- zewnętrzni użytkownicy: ludzie, sprzęt, system
- rysowani jako figura lub klasa ze słowem kluczowym **«Actor»**

# Związek między aktorami typu *Generalization*



Oznacza to dziedziczenie funkcji (przypadków użycia) przez aktora np. *CommercialCustomer* od aktora *Customer* oraz korzystanie przez niego z nowych przypadków użycia.

# Przypadek użycia (*Use Case*)



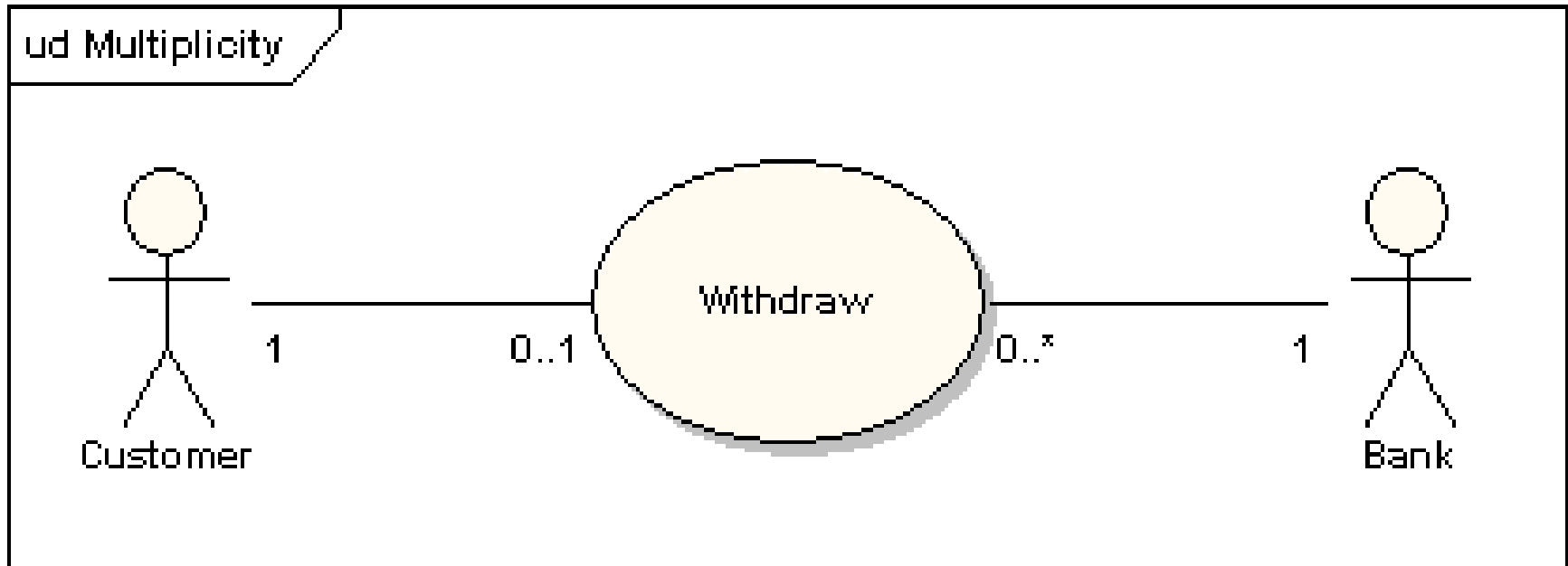
- Jednostka pracy
- Wysoki poziom zewnętrznej obserwacji systemu
- Notacja – elipsa
- Np. aktor *Customer* używa przypadku użycia *Withdraw* (pobiera pieniądze np. z konta)

# Definicja przypadku użycia (*Use Case*)

## Przypadek użycia zawiera:

1. Nazwę i opis
2. Cel, wymagania funkcjonalne spełniane dla użytkownika
3. Ograniczenia – warunki przed- po- przypadku użycia oraz nie zmieniające się na skutek wykonania przypadku użycia
4. Scenariusze – sekwencja zdarzeń między systemem i zewnętrznymi użytkownikami (opis tekstowy)
5. Diagramy scenariuszy – diagramy aktywności
6. Dodatkowe informacje – np. identyfikacja karty płatniczej przed dokonaniem wyciągu z konta

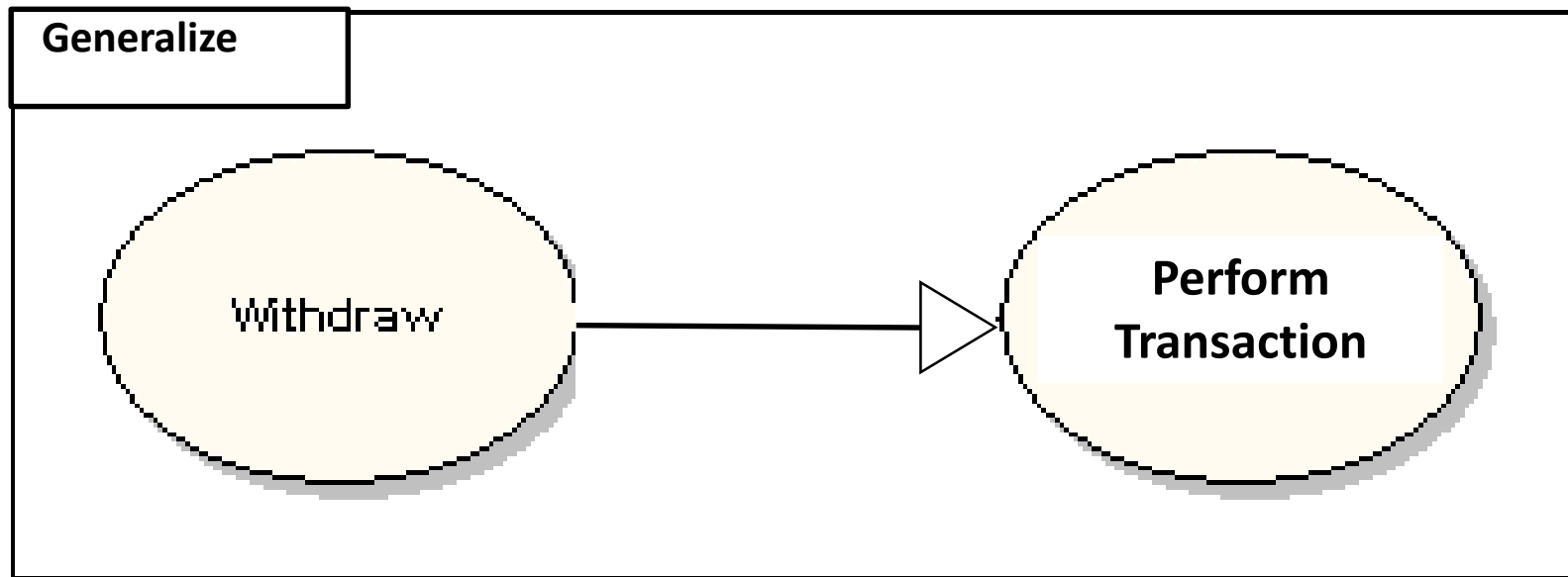
# Powiązania (*Association*) – liczność związku



- Liczność instancji na końcach połączenia

Np. aktor *Klient (Customer)* ma tylko jedną sesję wypłacania pieniędzy w danym momencie (*Withdraw*), natomiast *Bank* może mieć ich wiele w tym samym czasie

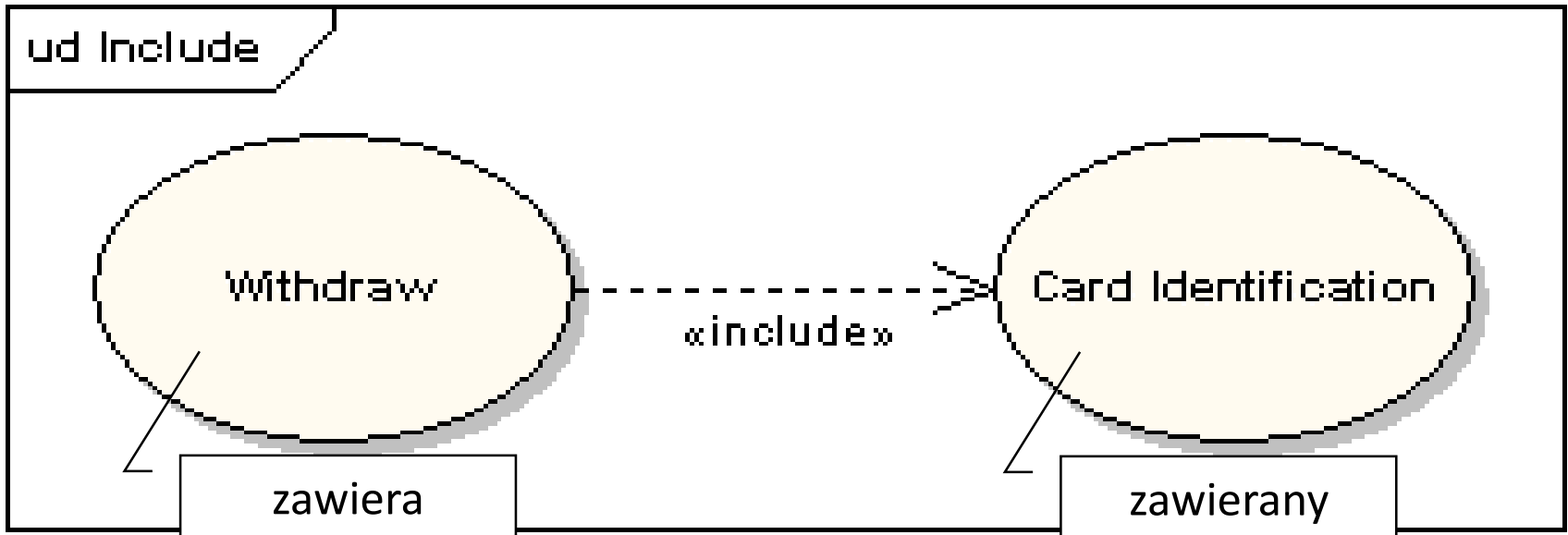
# Związek *Generalization*



**Przypadki użycia** mogą generalizować inne **przypadki użycia**.

Np. oznacza to dziedziczenie funkcjonalności przypadku użycia ***Perform Transaction*** przez przypadek użycia ***Withdraw***

# Związek *Zawieranie* - <<include>>

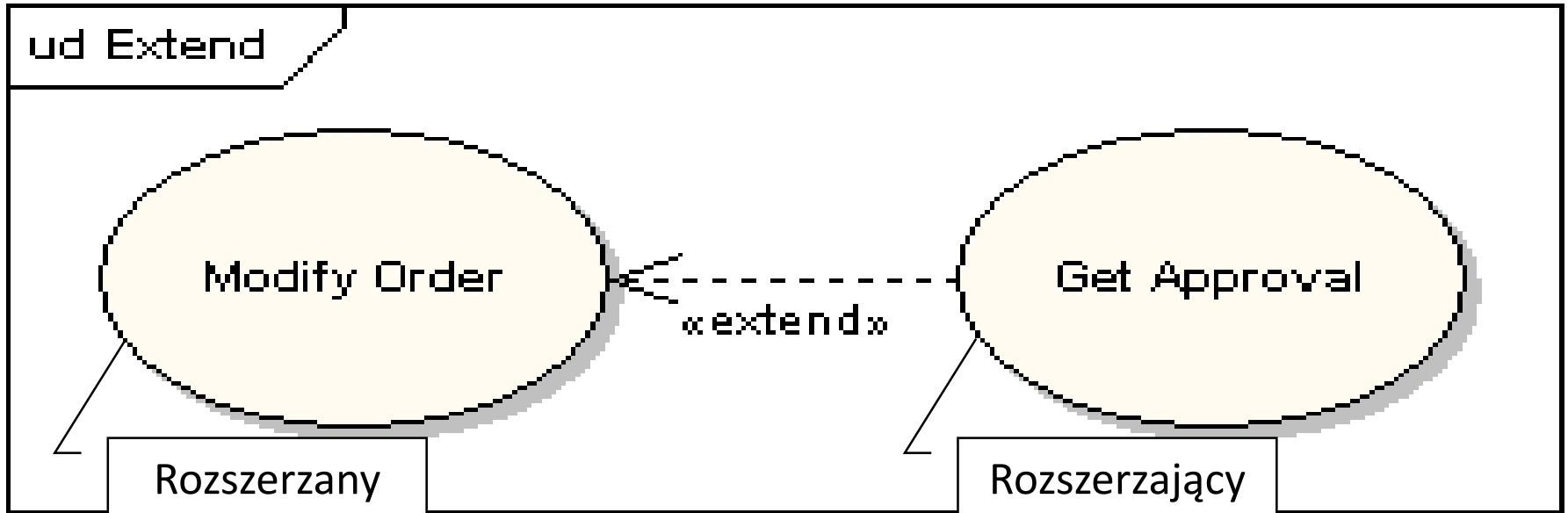


Przypadek użycia **zawiera** jeden lub wiele innych przypadków użycia eliminując powtarzanie funkcjonalności systemu dzięki tej **wieloużywalności**, czyli **zawieraniu**.

np. Pobranie z konta (**Withdraw**) **zawsze (obligatoryjnie)** musi wykonać identyfikację karty (**Card identification**)



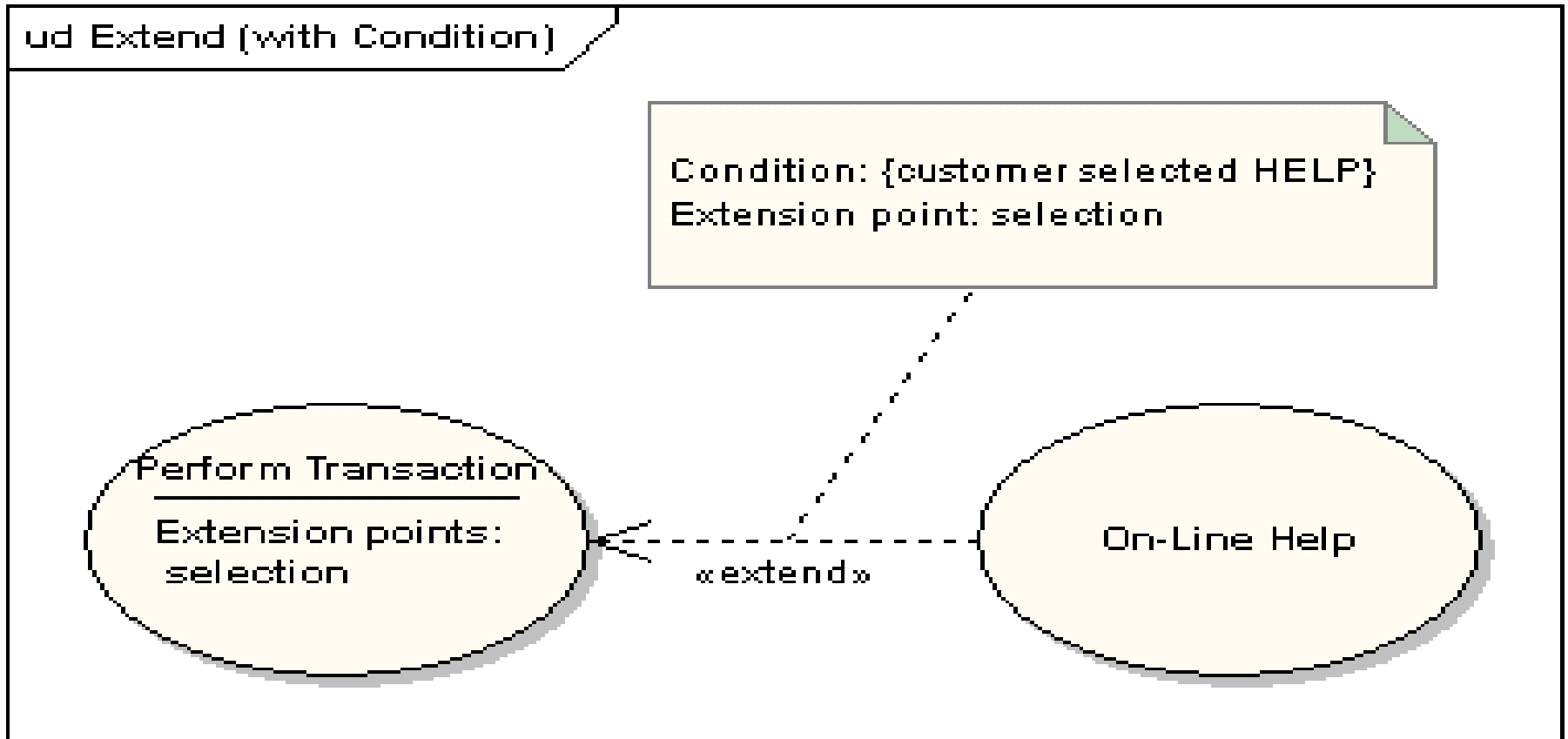
# Związek *Rozszerzanie* - <<extend>>



Jeden przypadek użycia **może być** użyty do rozszerzenia właściwości drugiego przypadku użycia

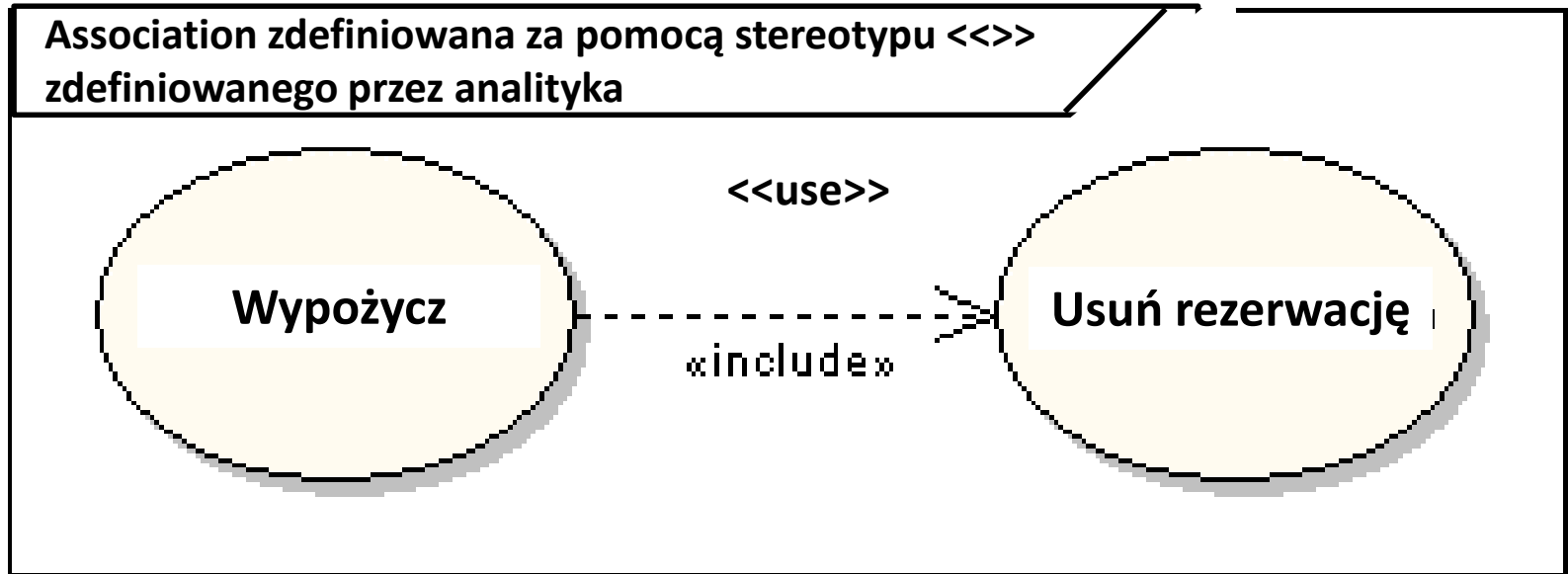
Np. przypadek użycia **Zezwolenie (Get Approval)** **opcjonalnie** rozszerza właściwości przypadku użycia **Modyfikuj zlecenie (Modify Order)**

# Punkty rozszerzające (Extension Points)



Np. Punkt, w którym przypadek użycia **Wykonanie transakcji** (*Perform Transaction*) jest warunkowo rozszerzany (Condition) przez rozszerzający przypadek użycia **Pomoc** (*On-Line Help*) zgodnie ze znaczeniem punktu rozszerzania (Extension Point) np. przez wybór (*selection*)

# Związek między przypadkami użycia typu **Association** definiowane przez analityka



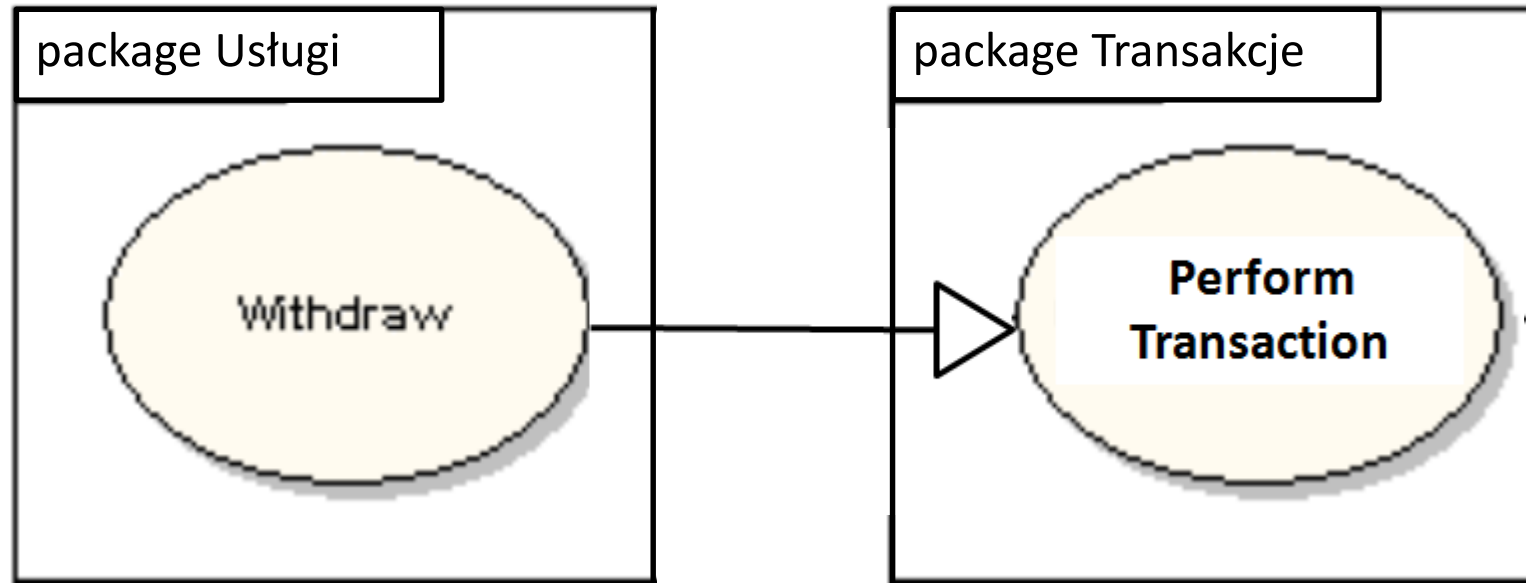
**Przypadki użycia** mogą być powiązane z innymi **przypadkami użycia** za pomocą asocjacji, której rodzaj może być zdefiniowany przez analityka za pomocą **stereotypu o podanej nazwie <<nazwa>>**.

Np. Stereotyp <<use>> może oznaczać, że przypadek użycia **Wypożycz** czasem **musi użyć** przypadek użycia **Usuń rezerwację**

- <<use>> znak stereotypu oznaczającego właściwości związku między przypadkami użycia

**Związek przypadku użycia <<use>> - skierowana strzałka**

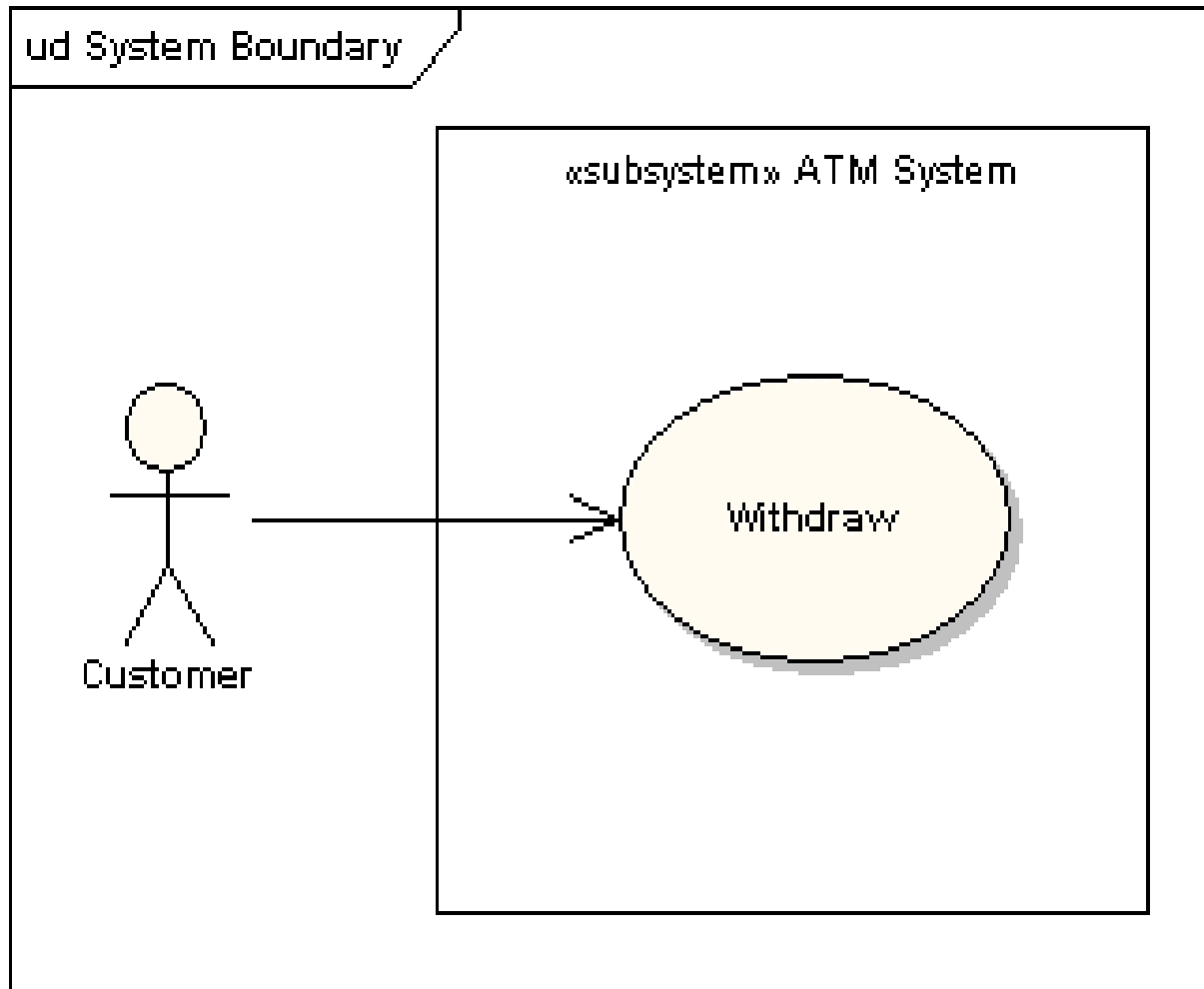
# Pakiet (package)



Pakiet służy do przechowywania przypadków użycia np. spełniających pewne funkcje w modelu – w celu uporządkowania diagramu.

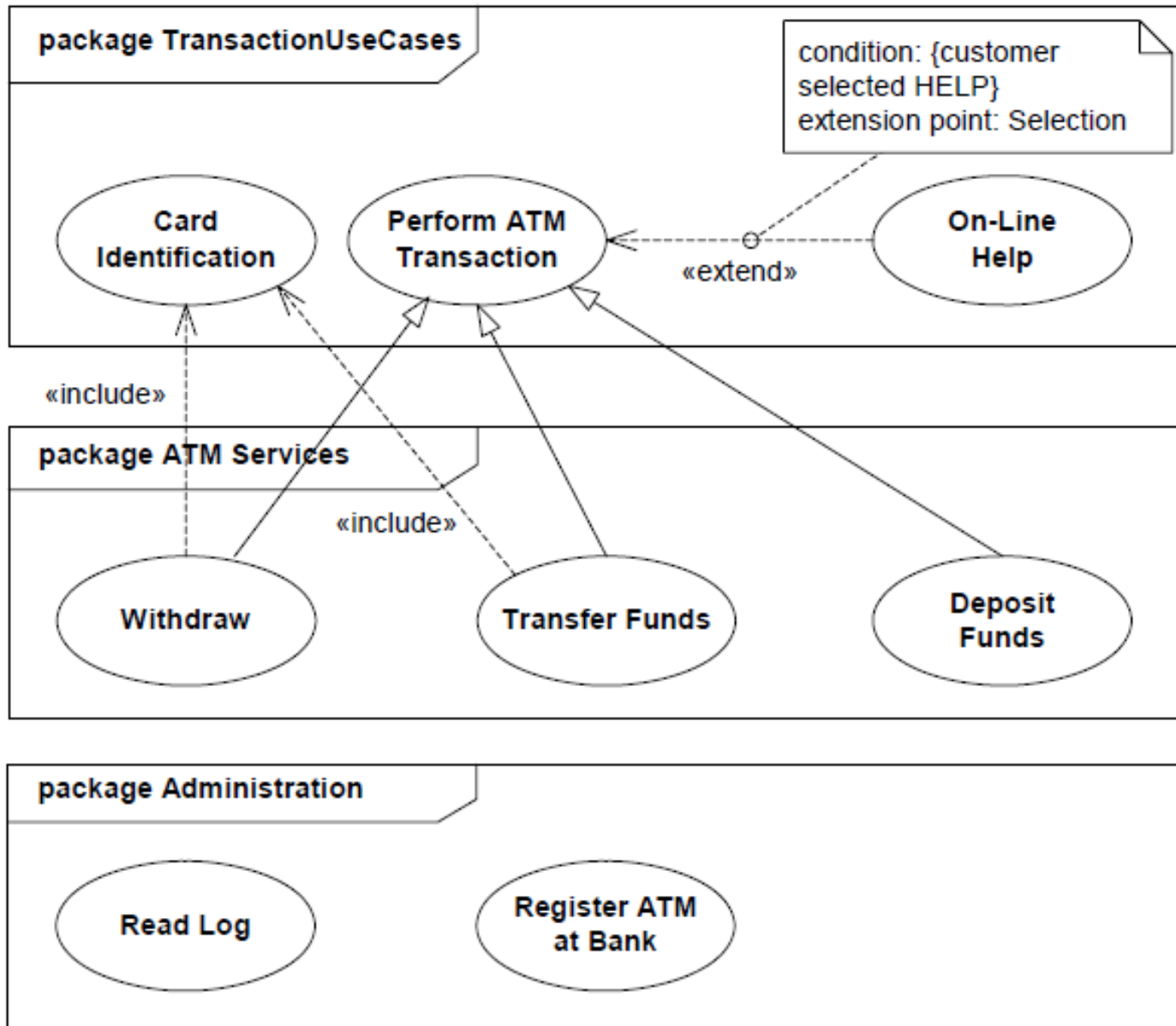
Np. Przypadek użycia *Withdraw* jest usługą w pakiecie *Usługi*, a przypadek użycia *Perform Transaction* jest umieszczony w pakiecie z transakcjami (*Transakcje*).

# Granice systemu (**System Boundary**)

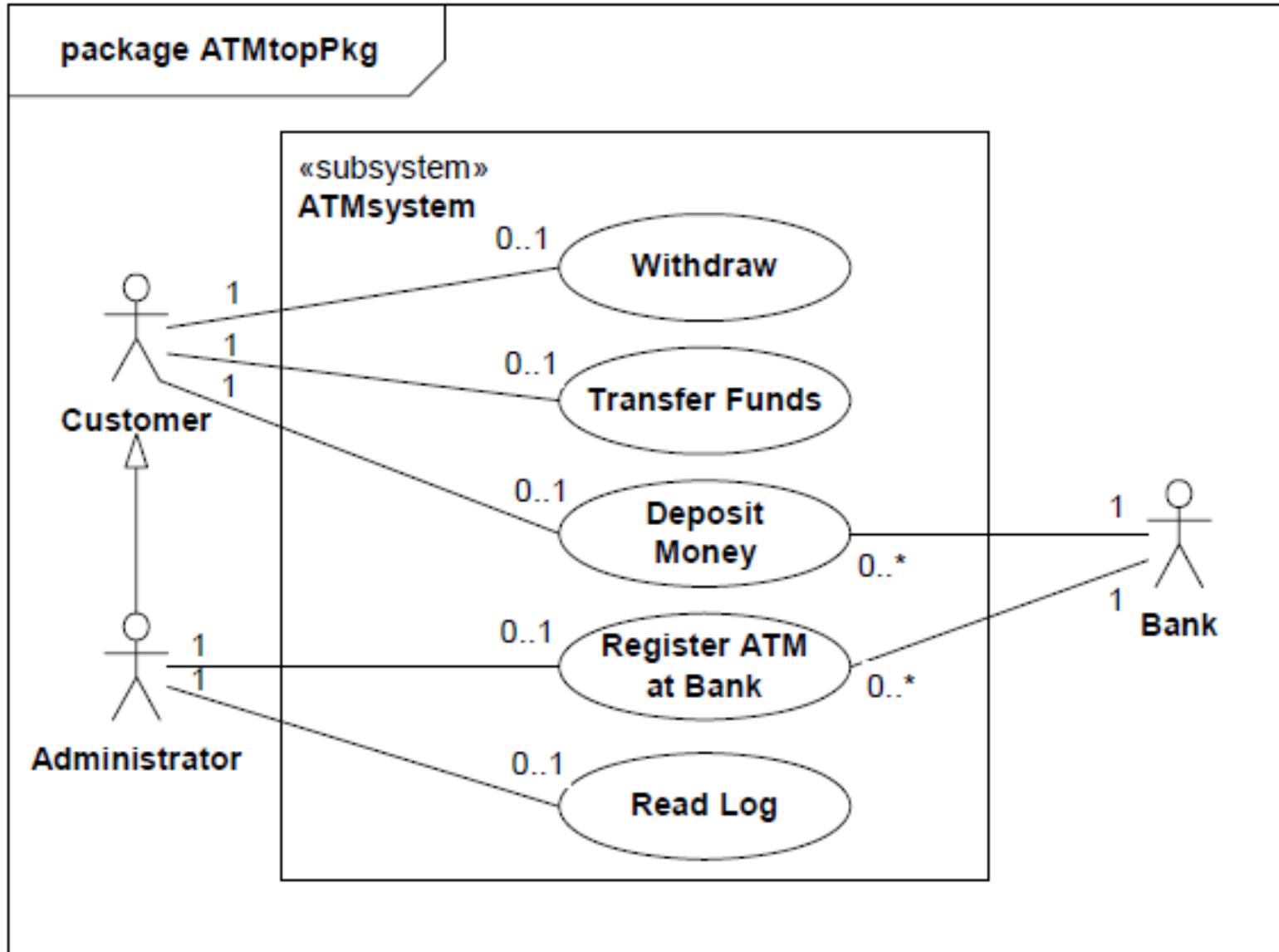


Aktorzy są na zewnątrz systemu np. **ATM System**, a przypadki użycia wewnątrz systemu.

# Przykład 1 diagramu przypadków użycia-formal-11-08-06, UML 2.4.1



## Przykład 2 diagramu przypadków użycia-formal-11-08-06, UML 2.4.1



# Tworzenie diagramów przypadków użycia

- I. Wprowadzenie (Wykład1, Wykład3)
- II. Elementy diagramów przypadków użycia (use-cases)
- III. **Wytyczne tworzenia diagramów przypadków użycia (use cases)**

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)



# Identyfikacja aktorów i przypadków użycia

1. Należy wyznaczyć granice systemu w ramach środowiska
- 2. Należy zdefiniować wymagania systemu na podstawie opisu biznesowego „świata rzeczywistego”** (wykład 3) i na ich podstawie należy podać :
  - użytkowników - aktorów systemu, zależności między aktorami typu dziedziczenie (**generalization**) lub powiązanie (**association**),
  - oczekiwane funkcje systemu (przypadki użycia),
  - powiązania między aktorami i przypadkami użycia oraz zależności między przypadkami użycia
3. Należy opisać każdy przypadek użycia np. wg szablonu podanego na slajdzie 13 (przykład na slajdach 38-40, 48-51)

# Wytyczne przy modelowaniu granic systemu (p.1, slajd 25)

1. Należy zidentyfikować aktorów działających wokół systemu. Oznacza to wyznaczenie grup użytkowników korzystających w określonym celu z projektowanego systemu (zarządzanie, pielęgnacja, usługi)
2. Należy uporządkować aktorów wg zależności typu powiązanie np. **Klient korzysta z usług Sprzedawcy** lub dziedziczenia: **Komercyjny Klient dziedziczy przypadki użycia od Klienta** Należy powiązać aktorów z przypadkami użycia za pomocą powiązań nadając im zidentyfikowane znaczenie za pomocą **stereotypu** wg podanych definicji

# Wytyczne przy modelowaniu wymagań stawianych systemowi (p.2, slajd 25)

1. Należy określić otoczenie systemu, czyli zidentyfikować aktorów
2. Dla każdego aktora należy podać działania, jakie każdy aktor oczekuje od systemu (slajdy 32, 38, 42)
3. Działania należy zapisać jako przypadki użycia
4. Należy wyłączyć powtarzające się ciągi działań i zastąpić je jednym nowym połączonym relacją **<<include>>** lub/i **<<extend>>** lub/i **<<use>>** (**przykład stereotypu zdefiniowanego przez analityka**) lub/i zwykłym powiązaniem typu **association** bez stereotypu lub/i za pomocą **Generalization** (**analiza wspólności i zmienności**) – slajdy 36, 37, 41.
5. Należy uwzględnić tych aktorów, przypadki użycia oraz zidentyfikowane powiązania między nimi
6. Można dodać do każdego aktora i przypadku użycia notatkę opisującą wymagania niefunkcjonalne (np. sprzęt, język, korzystanie z Internetu)

# Wytyczne przy modelowaniu przypadków użycia (p.3, slajd 25)

- 1. Należy opisać główny i nadzwyczajne ciągi zdarzeń** każdego przypadku użycia (slajd 33 – jeden ciąg zdarzeń, slajd 34 – drugi ciąg zdarzeń) podając: czynności i dane używane podczas działania przypadku użycia
- 2. Należy zdefiniować testy systemu** (slajd 35) w odniesieniu do wybranego aktora i powiązanego za nim jednego lub grupy przypadków użycia podając stan początkowy i końcowy oznaczający powodzenie testu przypadku użycia (np. *Wstawianie nowej książki jest możliwe tylko wtedy, gdy istnieje już jej tytuł w katalogu oraz posiada unikatowy numer. Po wstawieniu tej książki nie może być dwóch książek o tym samym numerze*)

# Tworzenie diagramów przypadków użycia

- I. Wprowadzenie (Wykład1, Wykład3)
- II. Elementy diagramów przypadków użycia (use-cases)
- III. Wytyczne tworzenia diagramów przypadków użycia (use-cases)

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

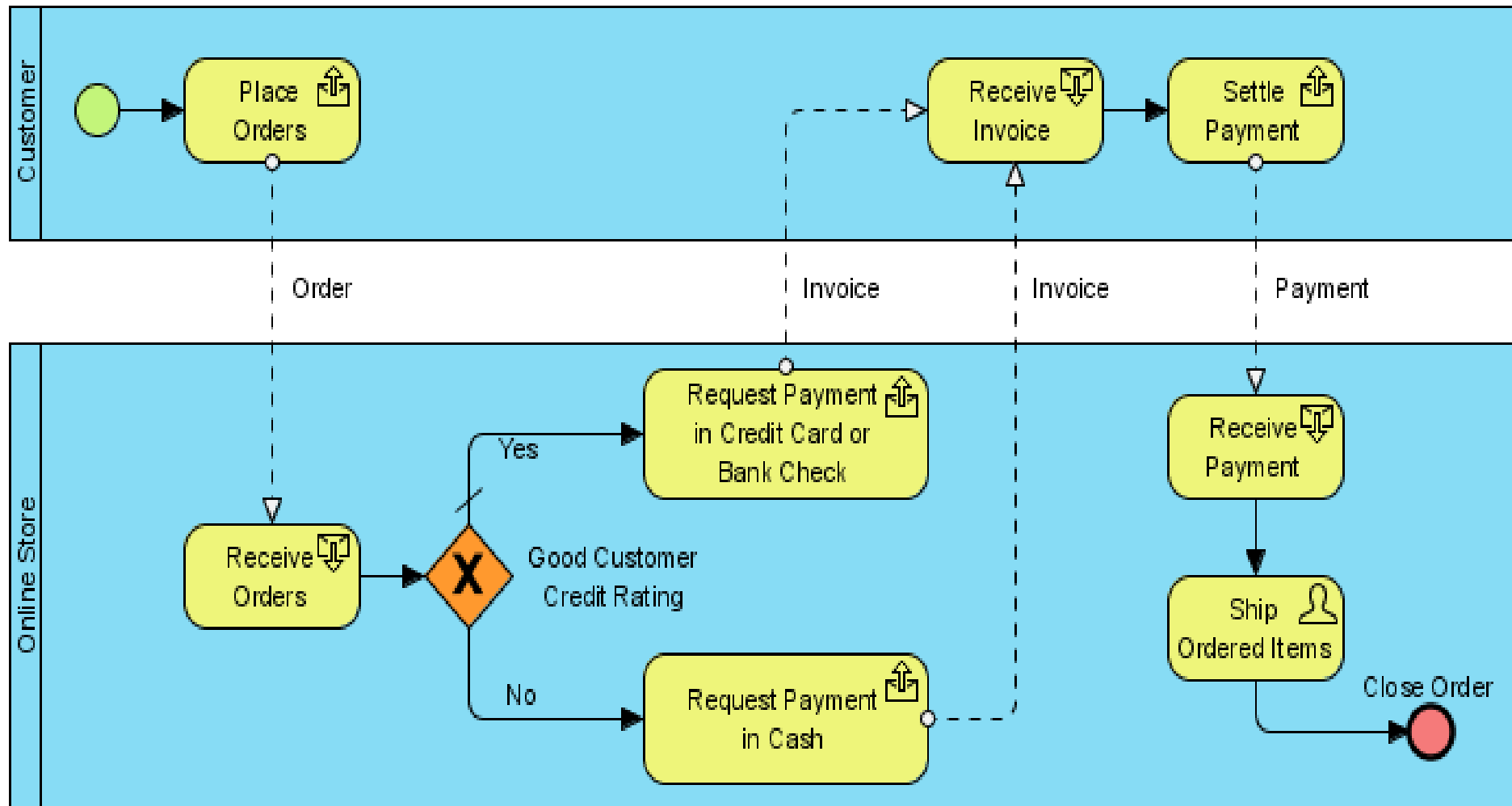
## **IV. Model przypadków użycia – identyfikacja i specyfikacja wymagań (podsumowanie z wykładu 3)**

# Proces identyfikacji i specyfikacji wymagań - opis „świata rzeczywistego”

Produkty wejściowe - czynności	Produkty wyjściowe	Opis produktu wyjściowego
lista kandydujących wymagań	lista znamionowa	status, szacowany koszt, priorytety, poziom ryzyka implementacji itp.
zrozumienie kontekstu systemu	<p><b>Model dziedziny*</b> (<i>domain model</i>) - najważniejsze obiekty systemu: „rzeczy” lub zdarzenia podawane przez ekspertów</p>	<p><b>diagram najważniejszych klas dziedziny</b> (<i>domain classes</i>) z niewielką ilością operacji-metod (około 10-50 w notacji UML), reszta przewidywanych klas w glosariuszu (<i>glossary</i>);</p>
	<p><b>Model biznesowy procesów*</b> (<i>business model</i>) - wewnętrzny model procesu biznesowego organizacji, wyszczególniany przez klientów systemu (<i>customers</i>)</p>	<p><b>„business use case” :</b></p> <p><b>a) Opis przypadków użycia („uses cases”) i aktorów („actors”)</b> odpowiadających procesowi biznesowemu oraz klientom procesu biznesowego</p> <p><b>b) biznesowy model obiektowy</b> (<i>business object model</i>) składający się z wykonawców (<i>workers</i>), encji biznesowych (<i>business entities</i>), jednostek pracy (<i>work units</i>) realizujących „use case”</p>

# Przykład Model biznesowy „świata rzeczywistego”

Przykład **modelu biznesowego** dotyczącego realizacji opłat z dokumentacji środowiska *Business Process Visual Architect* (BP-VA) do wizualnego modelowania diagramów procesów stosując BPMN (*Business Process Modeling and Notation*).

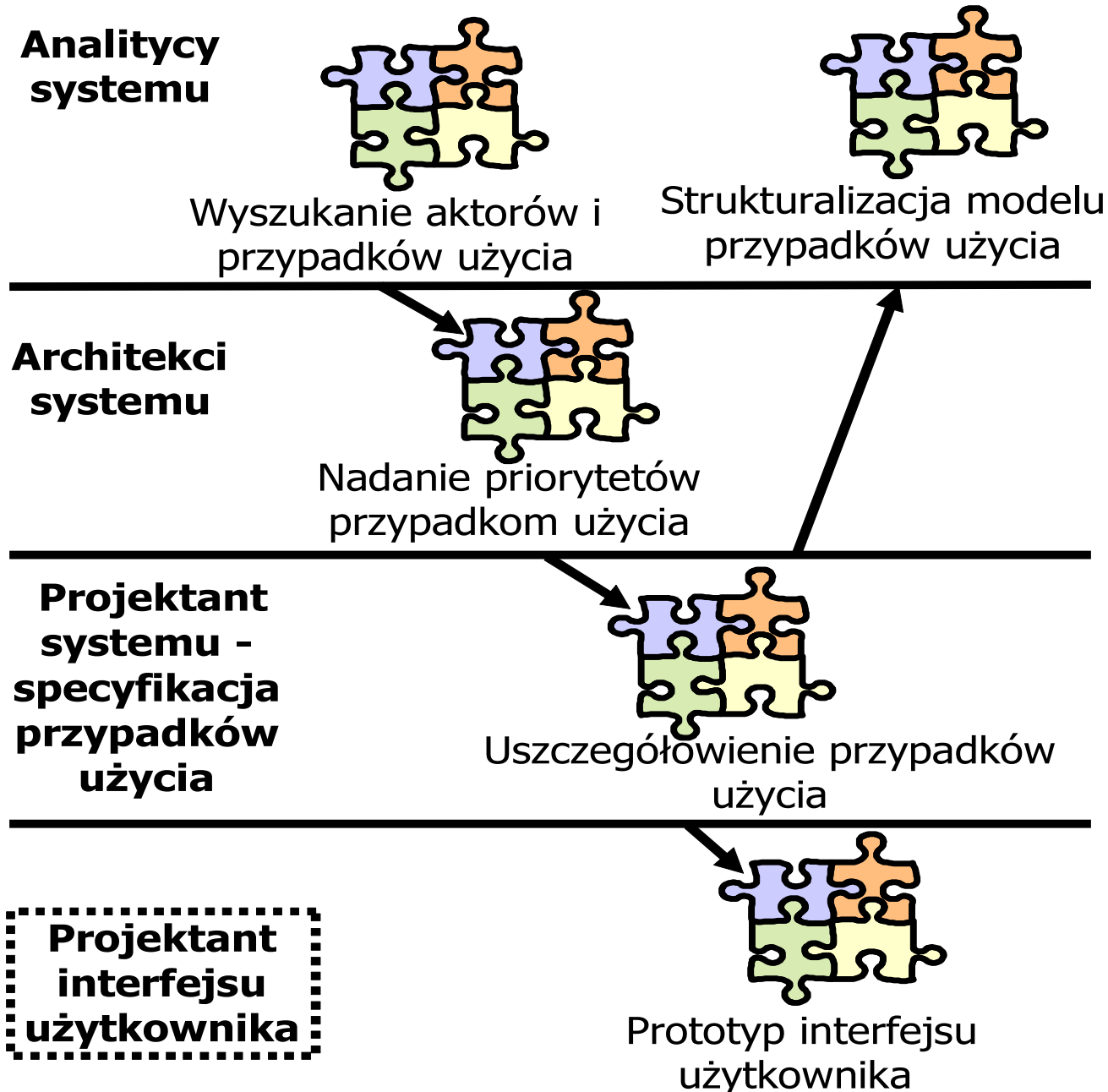


## Proces identyfikacji i specyfikacji wymagań – **diagram przypadków użycia**

<b>Diagram wymagań funkcjonalnych</b>	<b>Model przypadków użycia (identyfikacja przypadków użycia z modelu biznesowego)</b>	<p>Proces reprezentowania wymagań jako przypadków użycia oraz innych produktów specyfikowany za pomocą UML:</p> <ol style="list-style-type: none"><li><b>1. opis tekstowy</b> realizujących zachowanie systemu przy działaniu poszczególnych przypadków użycia lub aktorów i przypadków użycia - czyli opis sekwencji akcji odpowiednich do modyfikacji, przeglądu, projektowania i testowania,</li><li><b>2. model przypadków użycia zawierający aktorów i przypadki użycia oraz powiązania</b> (np. dziedziczenia) między nimi oraz: <b>dotatkowo diagram</b> czynności modelujący scenariusz przypadku użycia – <b>wykład 1, slajd 36</b></li><li><b>3. opis architektury</b> przypadków użycia</li><li><b>4. glosariusz</b> - definicje ważnych pojęć wyprowadzanych z modelu dziedziny lub modelu biznesowego,</li><li><b>5. prototyp interfejsu użytkownika</b> - interakcje między aktorami - ludźmi i oprogramowaniem</li></ol>
<b>Diagram wymagań niefunkcjonalnych</b>	<b>uzupełniające wymagania lub indywidualne wymagania</b>	<ol style="list-style-type: none"><li><b>1. specjalne wymagania</b> zawierające niefunkcjonalne wymagania w postaci opisu tekstowego</li><li><b>2. ograniczenia środowiska i implementacji</b> (np. typ komputera, typ plików, rodzaj systemu operacyjnego, typ oprogramowania Internetu), zależności, konserwacja, zdolność do poszerzania,</li></ol>



# Projektowanie GUI – wynika z definicji przypadków użycia



# Tworzenie diagramów przypadków użycia

- I. Wprowadzenie (Wykład1, Wykład3)
- II. Elementy diagramów przypadków użycia (use-cases)
- III. Wytyczne tworzenia diagramów przypadków użycia (use-cases)

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

- IV. Model przypadków użycia – identyfikacja i specyfikacja wymagań
- V. Przykłady diagramów przypadków użycia (use-cases)

# **Przykład 1: „Wypożyczalnia książek” - opis biznesowy „świata rzeczywistego” w języku klienta**

## **1. Opis zasobów ludzkich**

Pracownik wypożyczalni może dodawać do katalogu tytułów nowe tytuły, usuwać lub zmieniać. Każdy tytuł jest reprezentowany przez następujące dane: tytuł, autor, wydawnictwo, ISBN oraz informacje o liczbie egzemplarzy. Każdy egzemplarz książki jest opisany odrębną informacją zawierającą numer egzemplarza oraz jest powiązany z informacją o tytule. Numery egzemplarzy mogą się powtarzać dla różnych tytułów. Pracownik biblioteki może dodawać lub usuwać nowe egzemplarze o tytułach przechowywanych w wypożyczalni. Pracownik wypożyczalni wypożycza podane książki i czasopisma osobom zarejestrowanym, o ile je posiada. Wypożyczający może zarezerwować książkę lub czasopismo, które nie jest dostępne w danej chwili za pośrednictwem pracownika wypożyczalni. W momencie, kiedy zamówione rzeczy są dostępne - albo po zwrocie lub dzięki zakupowi, można je wypożyczyć i usunąć rezerwację. Rezerwację można usunąć niezależnie. Pracownik wypożyczalni może utworzyć, zmienić i usunąć dane o wypożyczających, wypożyczeniach i rezerwacjach

## **2. Przepisy, strategia firmy**

Pracownik odpowiada materialnie za niezgodność danych ze stanem wypożyczalni. Wypożyczalnia powinna być przyjazna dla klienta biznesowego

## **3. Dane techniczne**

Zakłada się, że klientów, jednocześnie przeglądających dane wypożyczalni może być ponad 1000 oraz wypożyczalnia może zawierać kilkadziesiąt tysięcy tytułów oraz przynajmniej dwukrotnie więcej egzemplarzy. Zaleca się stosowanie technologii Java.

# **Wymagania stawiane tworzonej aplikacji typu „Wypożyczalnia książek” - na podstawie opisu „świata rzeczywistego” wykonanego przez eksperta problemu reprezentującego stronę zamawiającą czyli użytkownika i klienta, przekazana wykonawcy systemu**

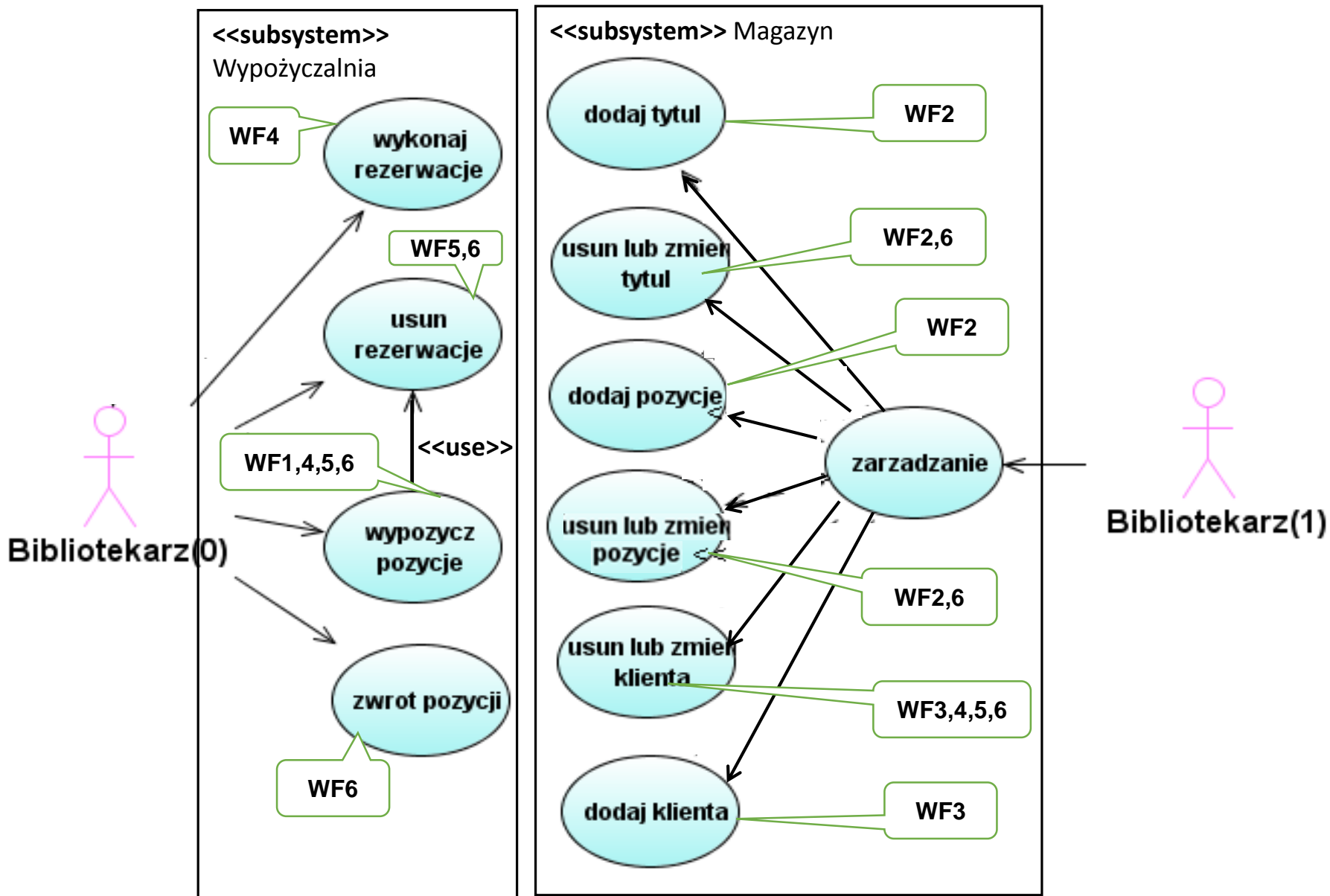
## **Wymagania funkcjonalne**

1. System powinien wspierać wypożyczalnię książek
2. Bibliotekarz wstawia nowe tytuły i książki o danym tytule w kilku egzemplarzach. Usuwa tytuły, książki i czasopisma. Może zmodyfikować dane o tytułach.
3. Bibliotekarz wstawia, usuwa lub zmienia dane osobowe wypożyczających.
4. Wypożyczający, którego dane osobowe są przechowywane w bibliotece, może zarezerwować książkę, która nie jest dostępna w danej chwili, **za pośrednictwem bibliotekarza**
5. Wypożyczający, którego dane osobowe są przechowywane w bibliotece - **za pośrednictwem bibliotekarza** może wypożyczyć książkę zarezerwowaną wcześniej lub bez rezerwacji w momencie, kiedy książka jest dostępna. W przypadku wcześniejszej rezerwacji wypożyczonej książki, należy ją usunąć. Rezerwację można usunąć niezależnie.
6. Bibliotekarz może łatwo utworzyć, zmienić i usunąć informację o wypożyczeniach i rezerwacjach

## **Wymagania нефunkcjonalne**

1. System powinien pracować w popularnych systemach (LINUX, Windows) i powinien mieć nowoczesny graficzny interfejs użytkownika
2. System powinien się rozwijać np. wprowadzenie możliwości zawiadamiania rezerwującego książkę o jej dostępności lub dłużnika o przekroczeniu terminu wypożyczenia przez Internet

# Biblioteka (wg Hans-Erik Erikson, Magnus Penker, UML Toolkit, John Wiley&SonS INC, 1998)



AKTOR	OPIS	PRZYPADKI UŻYCIA
Bibliotekarz	<p><i>Bibliotekarz wypożycza, rezerwuje książki i przyjmuje zwroty książek oraz usuwanie rezerwacji. Jest on odpowiedzialny za utrzymywanie zasobów biblioteki (wstawianie i usuwanie: tytułów książek, egzemplarzy książek oraz danych wypożyczających)</i></p>	<ul style="list-style-type: none"> <li>• zarządzanie</li> <li>• wykonaj rezerwacje</li> <li>• usun rezerwacje</li> <li>• wypożycz pozycje</li> <li>• zwrot pozycji</li> </ul>

## **PU wypożycz pozycje**

### **OPIS**

**CEL: obsługa bibliotekarza**

**WS (warunki wstępne):** inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

**WK (warunki końcowe):** pojawienie się nowej danej o wypożyczeniu w aplikacji, usunięcie rezerwacji wypożyczanej pozycji w przypadku, gdy była wcześniej zarezerwowana lub komunikat o przyczynach braku wypożyczenia

### **PRZEBIEG 1 (podstawowy):**

1. Przekazanie danych wypożyczanego tytułu oraz wypożyczającego przez aktora
2. Identyfikacja wypożyczającego – jeśli jest, przejdź do punktu 3 lub zakończ automatycznie wypożyczanie
3. Identyfikacja rezerwacji dla tytułu wypożyczanej książki – jeśli klient posiada taką rezerwację, przejdź do punktu 4 lub automatycznie zakończ wypożyczanie
4. Identyfikacja dostępności egzemplarza (pozycji) – jeśli jest przynajmniej jeden egzemplarz wolny, przejdź do punktu 5 lub zakończ automatycznie wypożyczanie
5. Rejestracja nowego wypożyczenia zawierającego informację o wypożyczającym i wypożyczonym egzemplarzu i wywołanie **PU usun rezerwacje**

## **PRZEBIEG 2 (alternatywny przy braku rezerwacji):**

1. Przekazanie danych wypożyczanego tytułu oraz wypożyczającego przez aktora
2. Identyfikacja tytułu - jeśli tytuł jest w systemie, przejdź do punktu 3 lub automatycznie zakończ wypożyczanie
3. Identyfikacja dostępności egzemplarza (pozycji) – jeśli jest przynajmniej jeden egzemplarz wolny, przejdź do punktu 4 lub zakończ automatycznie wypożyczanie
4. Identyfikacja wypożyczającego – jeśli jest, przejdź do punktu 5 lub zakończ automatycznie wypożyczanie
5. Rejestracja nowego wypożyczenia zawierającego informację o wypożyczającym i wypożyczonym egzemplarzu.



# Test PU: wypożycz pozycje

## Dane wejściowe:

Dane klienta wypożyczającego książkę, tytuł wypożyczanej książki

## Dane wyjściowe:

- Powstanie nowego wypożyczenia, jeśli dane wejściowe były w systemie oraz była wolna książka o podanym tytule
- Usunięcie rezerwacji wypożyczonej książki, jeśli była w systemie dla danego klienta

## **Przykład 2: System sporządzania rachunków**

### **- opis biznesowy „świata rzeczywistego” w języku klienta**

#### **1. Opis zasobów ludzkich**

Pracownik firmy handlowej może dodawać do katalogu produktów nowe produkty. Każdy produkt jest reprezentowany przez następujące dane: nazwa, cena netto lub nazwa, cena netto, podatek od ceny netto lub nazwa, cena netto i promocja od ceny netto lub nazwa, cena netto, podatek od ceny netto i promocja od ceny netto. Klient może założyć rachunek nadając mu unikalny numer oraz może dodawać nowe zakupy. Każda pozycja zakupu powinna zawierać inny produkt. W przypadku dokonywania zakupu tego samego produktu, który znajduje się w innej pozycji zakupu, należy jedynie zwiększyć ilość tego produktu w tym istniejącym zakupie. Klient może sprawdzić, jaką wartość ma rachunek oraz wartości zakupionych produktów z poszczególnych grup podatkowych. Pracownik firmy handlowej może dodatkowo dodawać nowe typy produktów oraz modyfikować i usuwać dane o produktach i promocjach.

#### **2. Przepisy, strategia firmy**

Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem produktów firmy handlowej. System sporządzania rachunków powinien być przyjazny dla klienta biznesowego.

#### **3. Dane techniczne**

Klient może przeglądać katalog sprzedaży za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu (tak obsługuje już wdrożone programy). Zakłada się, że klientów, jednocześnie przeglądających dane katalogu sprzedaży oraz dokonujących zakupu produktów może być ponad 1000 oraz firma handlowa może oferować kilkadziesiąt tysięcy typów produktów. Firma handlowa składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

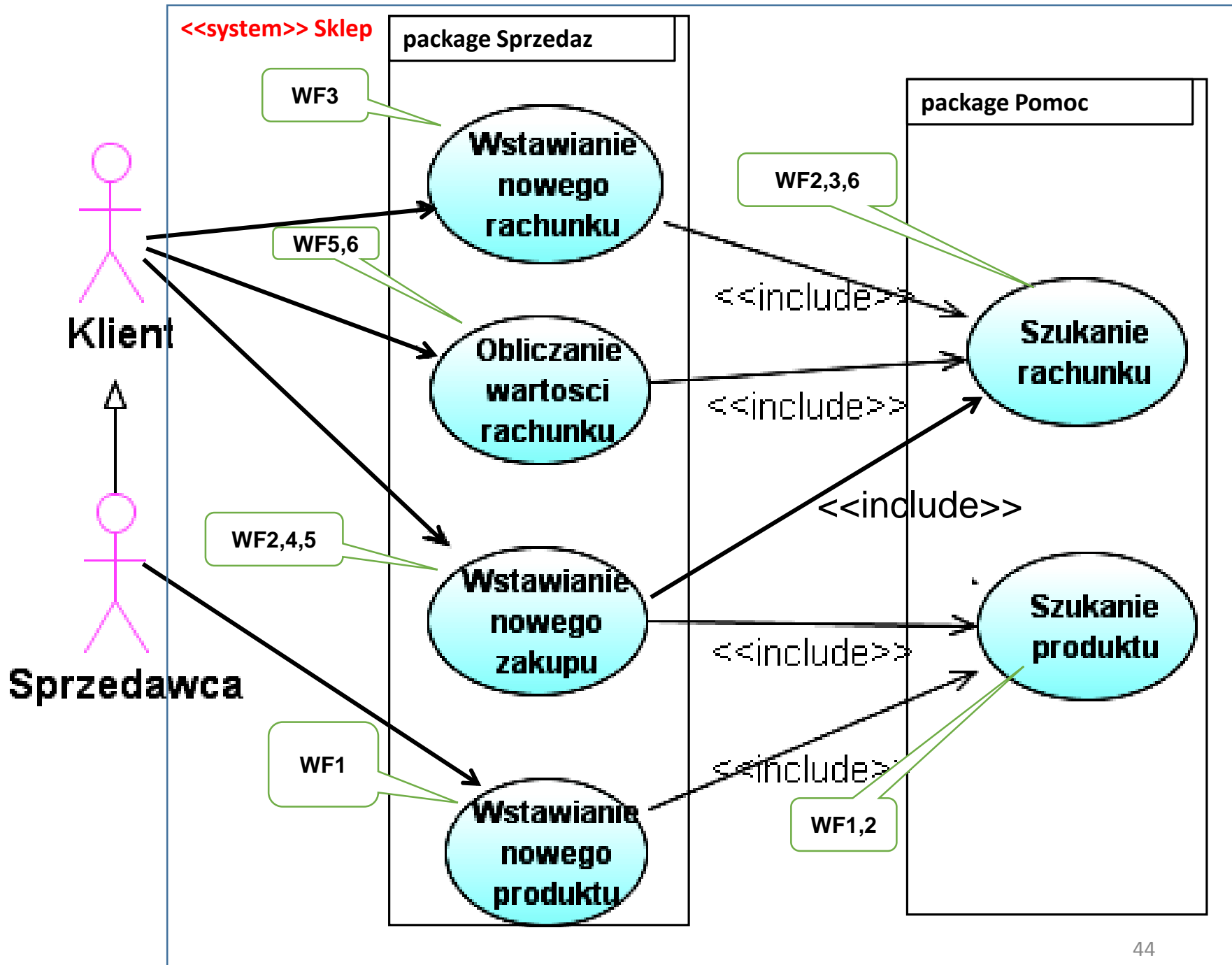
**Wymagania stawiane tworzonej aplikacji - na podstawie opisu „świata rzeczywistego” wykonanego przez eksperta problemu reprezentującego stronę zamawiającą czyli użytkownika i klienta, przekazana wykonawcy systemu**

### **Lista wymagań funkcjonalnych**

1. System zawiera katalog produktów , do którego można wstawiać produkty
2. Można zakupić **cztery typy produktów** różniące się sposobem obliczania ceny detalicznej: bez promocji i bez podatku, z promocją i bez podatku, z podatkiem bez promocji, z podatkiem i z promocją,
3. Można wprowadzić wiele rachunków
4. Pozycje rachunku muszą zawierać produkty różne w sensie nazwy, ceny, podatku i promocji
5. Każda pozycja rachunku powinna podać swoją wartość brutto oraz dane produktu oraz ilość zakupionego produktu.
6. Na rachunku powinna znajdować się wartość łączna wszystkich zakupów oraz wartości zakupów należących do wybranych kategorii

### **Lista wymagań niefunkcjonalnych**

1. Wstawianie produktów może odbywać się tylko przez uprawnione osoby
2. Wstawianie nowych rachunków oraz wstawianie nowych zakupów jest dokonywane przez klientów
3. Zakupy mogą być dokonane przez Internet przez aplikację uruchamianą przez przeglądarkę lub bez jej pośrednictwa



AKTOR	OPIS	PRZYPADKI UŻYCIA
Klient	<i>Klient może dokonywać zakupów wybranych produktów przez Internet korzystając z przeglądarki lub z aplikacji</i>	<ul style="list-style-type: none"> <li>• <b>PU Wstawianie nowego rachunku</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b></li> <li>• <b>PU Obliczanie wartości rachunku</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b></li> <li>• <b>PU Wstawianie nowego zakupu</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b> oraz powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie produktu</b></li> </ul>
Sprzedawca	<i>Sprzedawca może dodatkowo dodawać nowe produkty</i>	<ul style="list-style-type: none"> <li>• <b>PU Wstawianie nowego rachunku</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b></li> <li>• <b>PU Obliczanie wartości rachunku</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b></li> <li>• <b>PU Wstawianie nowego zakupu</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie rachunku</b> oraz powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie produktu</b></li> <li>• <b>PU Wstawianie nowego produktu</b> powiązane przez &lt;&lt;include&gt;&gt; z <b>PU Szukanie produktu</b></li> </ul>

# Wnioski - model przypadków użycia

- Opis w języku klienta
- Zewnętrzna postać systemu
- Strukturyzacja za pomocą przypadku użycia czyli struktura postaci zewnętrznej systemu
- Używany głównie jako kontrakt między klientem i wykonawcami, określający co system powinien robić i czego nie powinien robić
- Może zawierać redundancję, sprzeczności
- Przedstawia funkcjonalność systemu, dołączając architekturę ważnej funkcjonalności
- Definiuje przypadki użycia analizowane w modelu analizy

# Dodatkowe materiały

# 1. Opisy przypadków użycia do przykładu 2

## PU Szukanie produktu

OPIS

CEL: Poszukiwanie produktu

WS (warunki wstępne): może być wywołany z **PU Wstawianie nowego produktu**, **PU Wstawianie nowego zakupu**

WK (warunki końcowe): podanie produktu o podanych atrybutach obowiązkowych: nazwa i cena oraz jeśli jest to wymagane: z podatkiem i promocją lub komunikat o braku produktu

PRZEBIEG:

1. Szukanie produktu przebiega według atrybutów: nazwy i ceny (obowiązkowo) oraz podatku i promocji (jeśli jest to wymagane) zgodnie z danymi podanymi do przypadku użycia
2. Jeśli istnieje produkt o podanych atrybutach, zwracany jest produkt, w przeciwnym wypadku zwracana jest informacja o braku produktu.

## PU Wstawianie nowego produktu

OPIS

CEL: Wstawienie nowego produktu

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie produktu o podanych atrybutach obowiązkowych: nazwa i cena oraz jeśli jest to wymagane: z podatkiem i promocją, jeśli nie było takiego produktu

PRZEBIEG:

1. Należy podać atrybuty produktu: nazwę, cenę jako obowiązkowe dane oraz podatek i cenę detaliczną, jeśli jest to wymagane
2. Należy wywołać **PU Szukanie produktu**. Należy sprawdzić, czy produkt o podanych atrybutach już istnieje. Jeśli tak, należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy produkt.



## **PU Szukanie rachunku**

OPIS

CEL: Poszukiwanie rachunku

WS (warunki wstępne): może być wywołany z **PU Wstawianie nowego rachunku**, **PU Wstawianie nowego zakupu**, **PU Obliczanie wartości rachunku**

WK (warunki końcowe): podanie rachunku o podanym numerze lub komunikat o braku rachunku

PRZEBIEG:

1. Szukanie rachunku przebiega według numeru podanego do przypadku użycia
2. Jeśli istnieje rachunek o podanym numerze, zwracany jest rachunek, w przeciwnym wypadku zwracana jest informacja o braku rachunku.

## **PU Wstawianie nowego rachunku**

OPIS

CEL: Wstawienie nowego rachunku

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie rachunku o podanym numerze, jeśli jest to unikatowy numer

PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy numer rachunku się powtarza.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, można wstawić nowy rachunek i zakończyć PU, w przeciwnym wypadku należy zakończyć PU bez wstawiania nowego rachunku.

## PU Obliczanie wartosci rachunku

### OPIS

CEL: Obliczanie wartosci rachunku wg podanego podatku

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): podanie wartości całego rachunku o podanym numerze i parametrze wejściowym równym -2 lub wartości zakupionych towarów wg podanej kategorii podatku lub komunikat o braku rachunku

### PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku oraz wartość podatku lub wartość -2
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy rachunek o podanym numerze istnieje.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można obliczyć wartości wybranego rachunku i należy zakończyć PU, w przeciwnym wypadku należy obliczyć wartość rachunku
4. Należy uruchomić petle, w której sumowane są wartości zakupu obliczane jako iloczyn ceny jednostkowej zakupionego produktu i ilości zakupu. Jeśli zachodzi potrzeba sumowania wartości zakupu zależna od wysokości podatku, należy podać wartość podatku i sumować jedynie zakupy o podanym podatku, w przeciwnym wypadku sumowane są wszystkie zakupy (gdy zamiast podatku zostanie przekazana wartość -2).

## **PU Wstawianie nowego zakupu**

OPIS

CEL: Wstawianie nowego zakupu

WS (warunki wstępne): inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

WK (warunki końcowe): dodanie nowego zakupu o podanych atrybutach lub zwiększenie ilości zakupionego produktu, jeśli już taki produkt zakupiono lub komunikat o braku rachunku

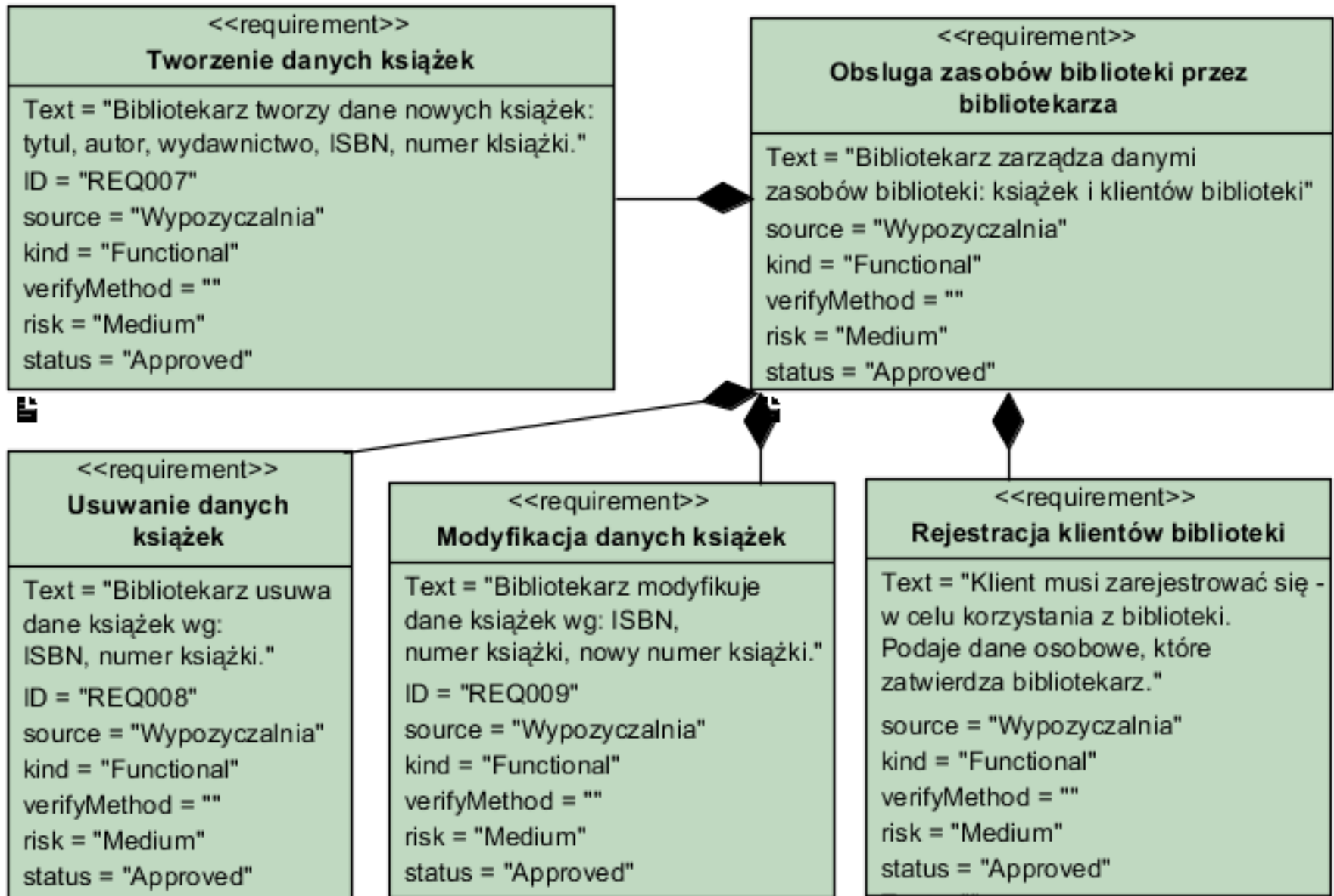
PRZEBIEG:

1. Należy podać numer rachunku, który powinien być niepowtarzalny, ponieważ służy do identyfikacji rachunku
2. Należy wywołać **PU Szukanie rachunku** w celu sprawdzenia, czy istnieje rachunek o podanym numerze.
3. Jeśli zwrócony wynik oznacza brak rachunku o podanym numerze, nie można wstawić nowego zakupu do rachunku i należy zakończyć PU, w przeciwnym wypadku należy wstawić nowy zakup
4. Należy wybrać produkt oraz ilość zakupionego produktu.
5. Należy wywołać **PU Szukanie produktu**. Jeśli wybrany produkt nie istnieje, należy zakończyć PU. W przeciwnym przypadku należy wstawić nowy zakup do rachunku, przeglądając, czy istnieje już zakup z takim samym produktem. Jeśli istnieje, nie tworzy się nowego zakupu, tylko powiększa się ilość zakupu istniejącego o ilość nowego zakupu, w przeciwnym przypadku wstawia się nowy zakup.

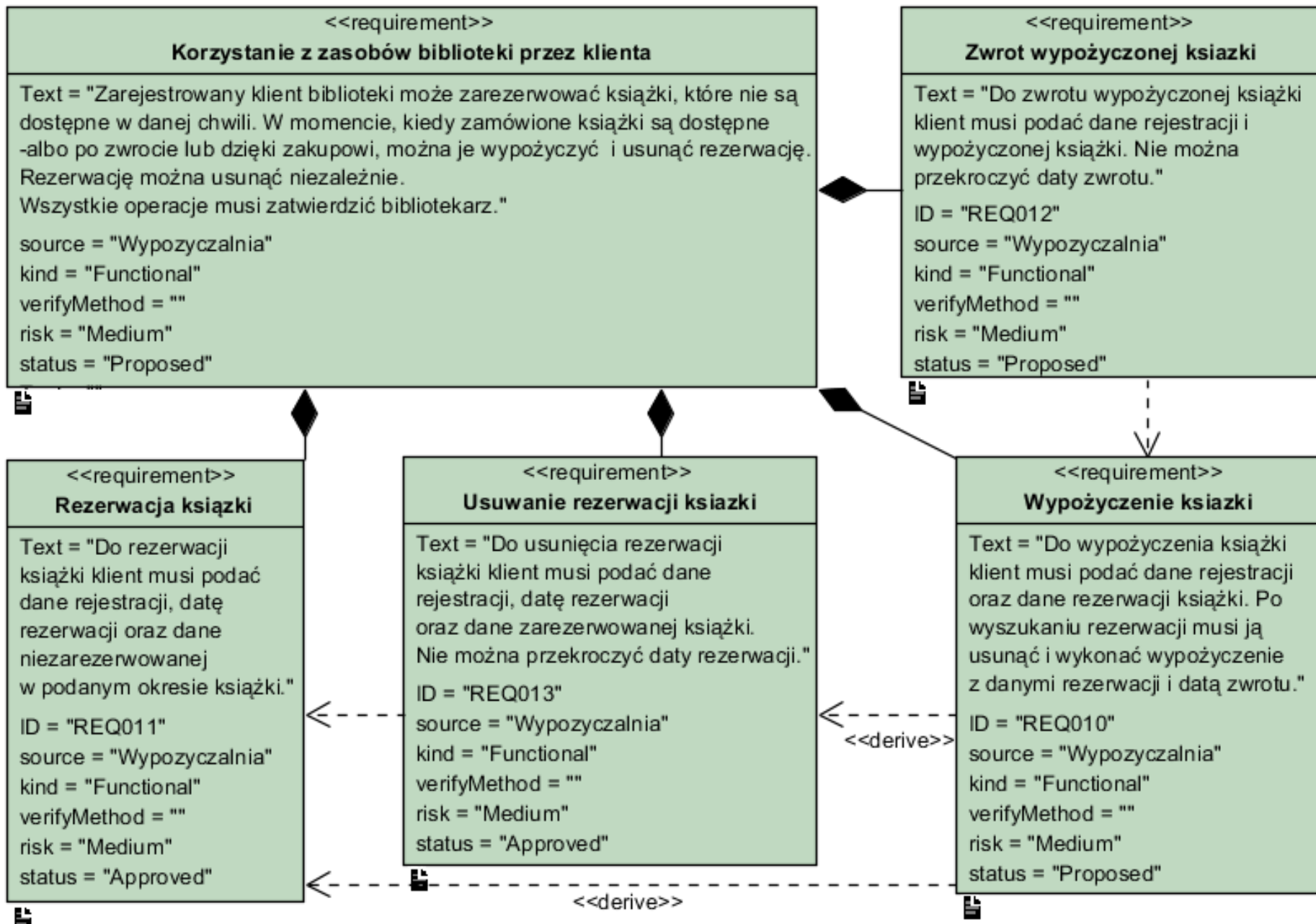
# Przykład 3 Diagram wymagań funkcjonalnych – program Biblioteka

(wykład 3, instrukcja do lab3:

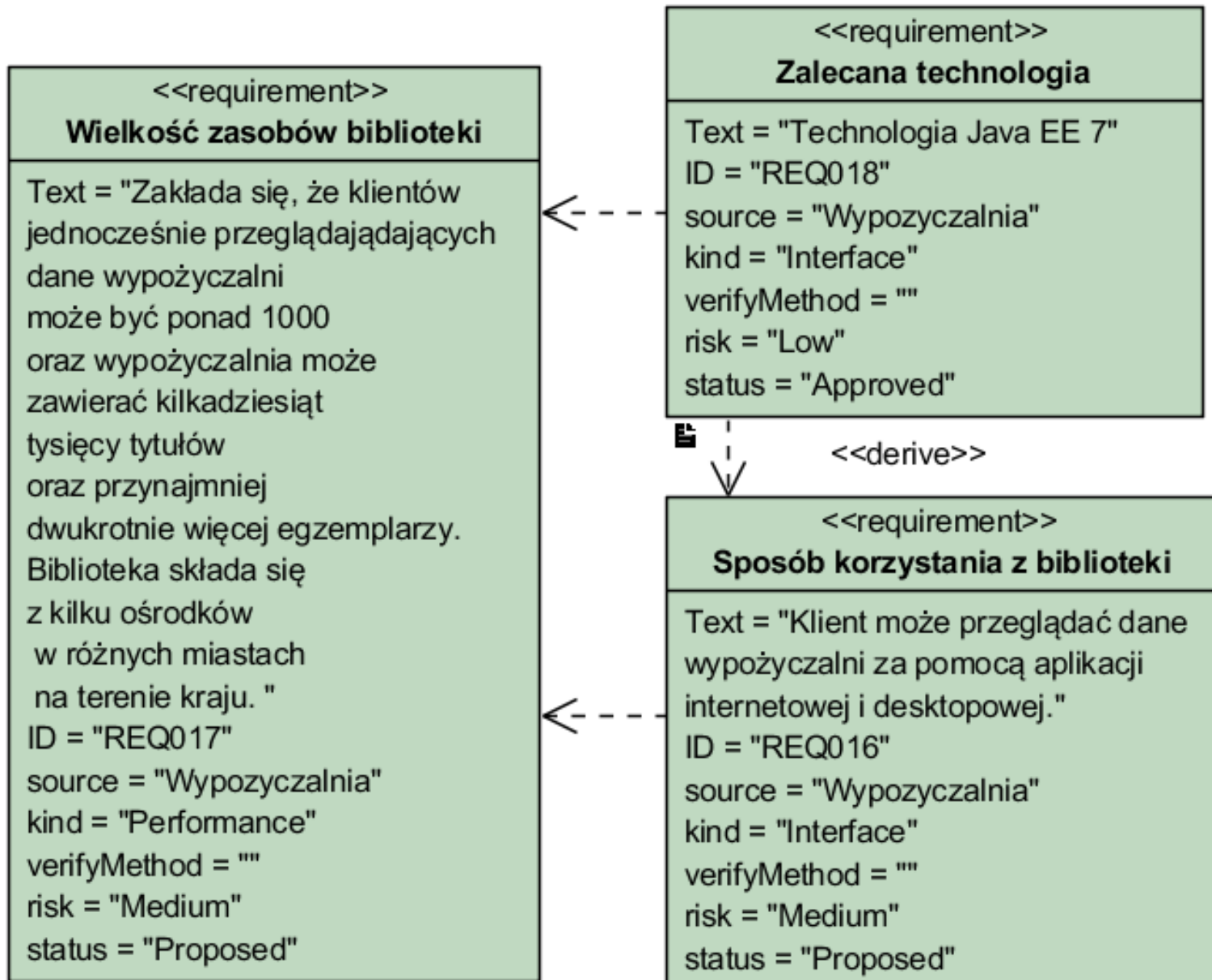
[http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/INP002017/Lab\\_INP002017\\_4.pdf](http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/INP002017/Lab_INP002017_4.pdf))



## Przykład 3 (cd) Diagram wymagań funkcjonalnych – program Biblioteka

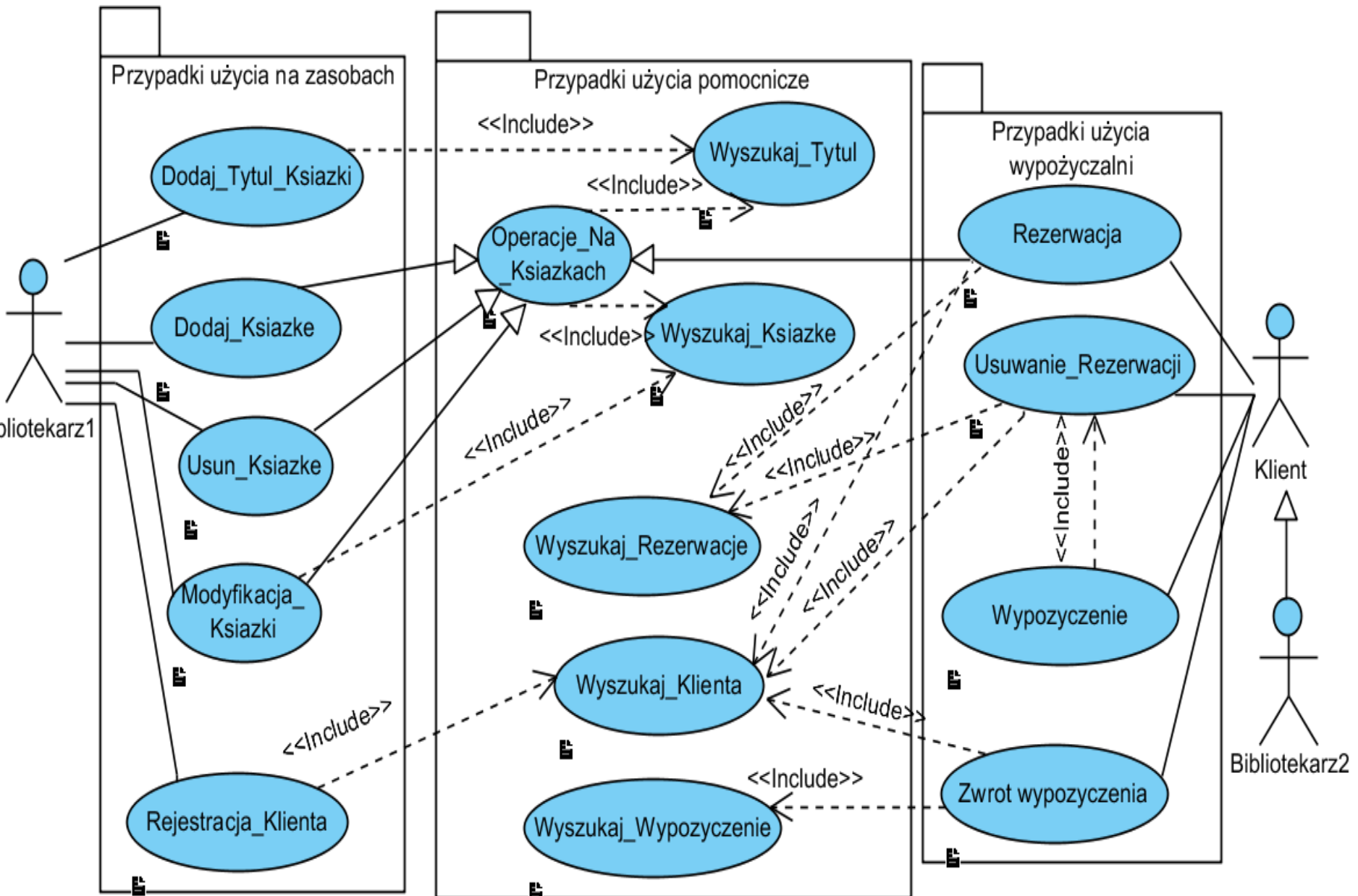


## Przykład 3 (cd) Diagram wymagań niefunkcjonalnych – program Biblioteka





# Przykład 3 (cd) – diagram przypadków użycia



# Dodaj\_Tytul\_Ksiazki



Info Use Case Notes Flow of Events Details Requirements Diagrams Test Plan References

Rank:

ID:

Status:

Justification:

Primary Actors:

Supporting Actors:

Task Pool:

Description:

**B** **F** **+**

Scenariusz:

1. Należy podać dane tytułu: imię i nazwisko autora, tytuł książki, wydawnictwo, ISBN i ewentualnie nazwisko aktora czytającego książkę, jeśli jest nagrana
2. Należy sprawdzić, czy dane wprowadzanego tytułu są unikalne za pomocą wywołania **PU Wyszukaj\_Tytul**, przekazując ISBN tytułu i ewentualnie aktora
3. Jeśli tytuł o podanym ISBN i ewentualnie aktorze istnieje, należy zakończyć przypadek użycia, w przeciwnym razie należy zapisać dane.

Abstract  Leaf  Root



New Open Save Close Print Export Import Properties Community Circle

Projekt\_lab2

- UML Diagrams
  - Use Case Diagrams
    - Use Case Diagram
  - Class Diagram (
    - Library1\_classes
    - Library1\_classe
  - Sequence Diagrams
    - sub\_business\_t
    - sub\_business\_t
    - sub\_business\_t
    - sub\_business\_t
    - sub\_business\_t
    - sub\_business\_t
    - sub\_business\_t

9+ [Icons]

# Dodaj\_Tytul\_Ksiazki



Level: User

Complexity: Medium

Use Case Status: Initial

Implementation Status: Started

Preconditions: Tworzenie danych ksiazek

Post-conditions:

Author:

Assumptions: