

Diagramy przypadków użycia

Wykład2

Zofia Kruczkiewicz

Tworzenie diagramów przypadków użycia

- 1. Elementy diagramów przypadków użycia (use-cases)**
- 2. Wytyczne tworzenia diagramów przypadków użycia (use-cases)**
(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)
- 3. Model przypadków użycia – identyfikacja i specyfikacja wymagań**
- 4. Przykłady diagramów przypadków użycia (use-cases)**
- 5. Podsumowanie**
- 6. Dodatek**

Tworzenie diagramów przypadków użycia

1. 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>

Dwa rodzaje diagramów UML 2

Diagramy UML modelowania strukturalnego

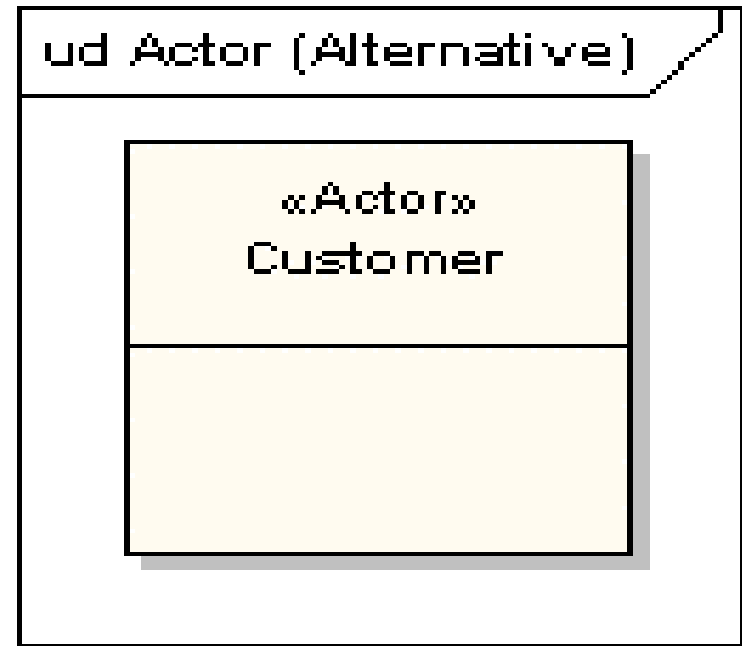
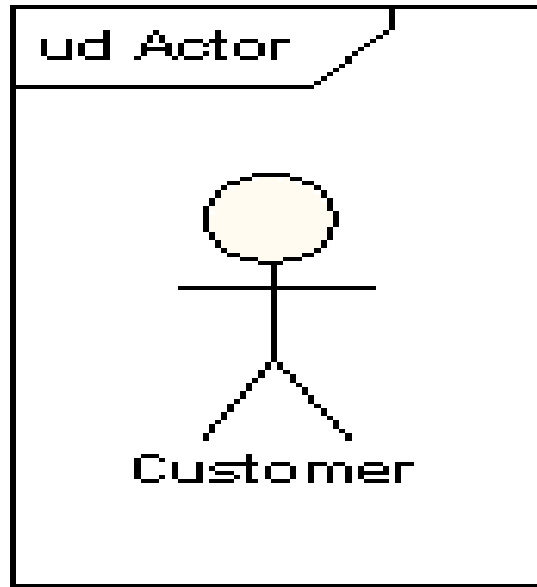
- **Diagramy pakietów**
- **Diagramy klas**
- **Diagramy obiektów**
- **Diagramy mieszane**
- **Diagramy komponentów**
- **Diagramy wdrożenia**

Diagramy UML modelowania zachowania

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

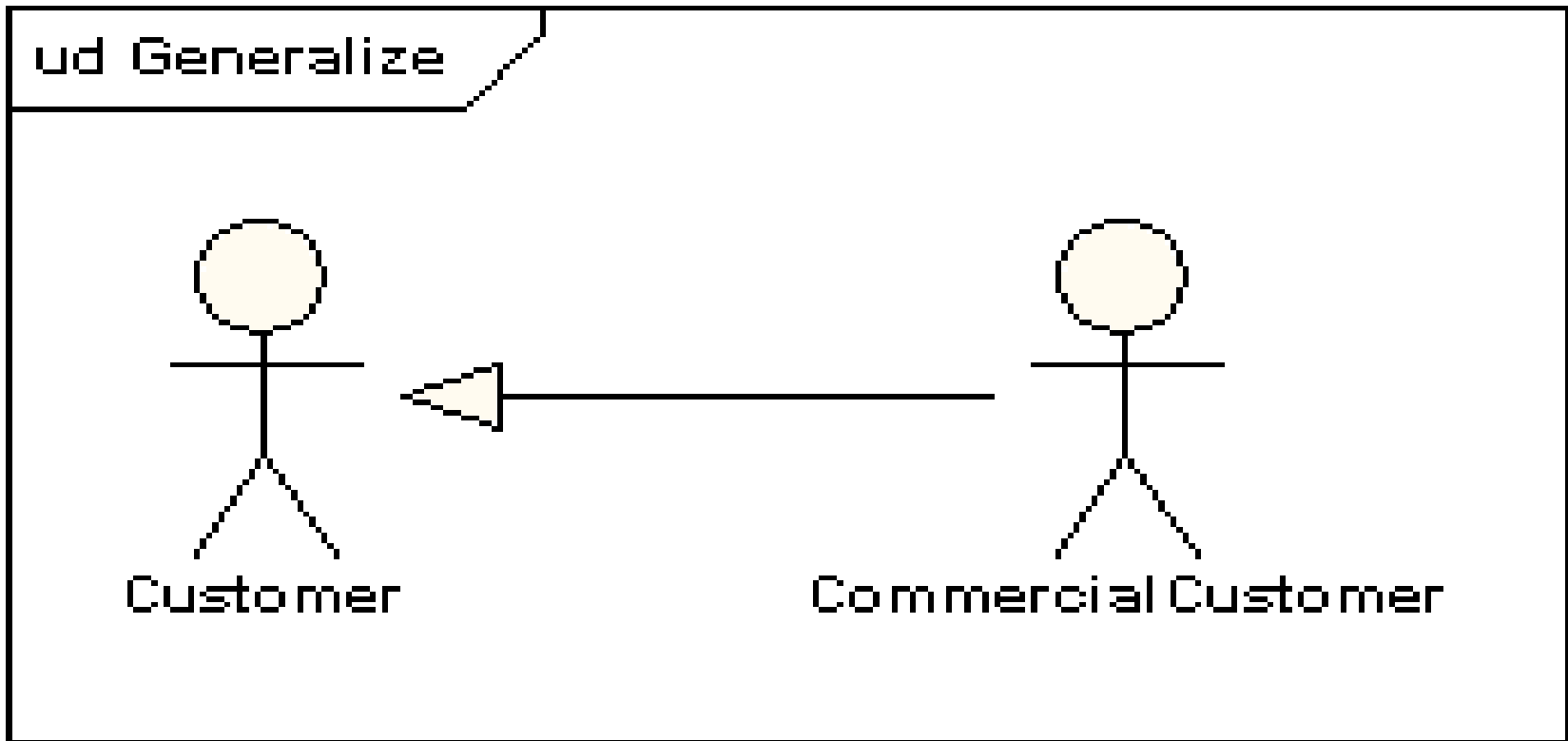
Diagramy przypadków użycia (Use Case Diagram)

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



Actors

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

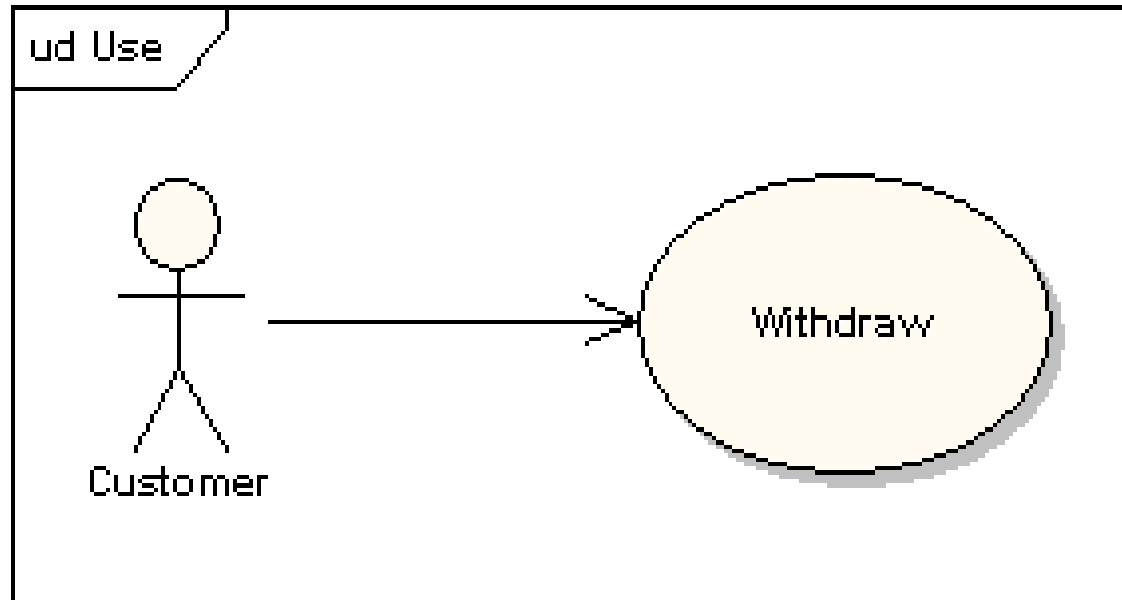
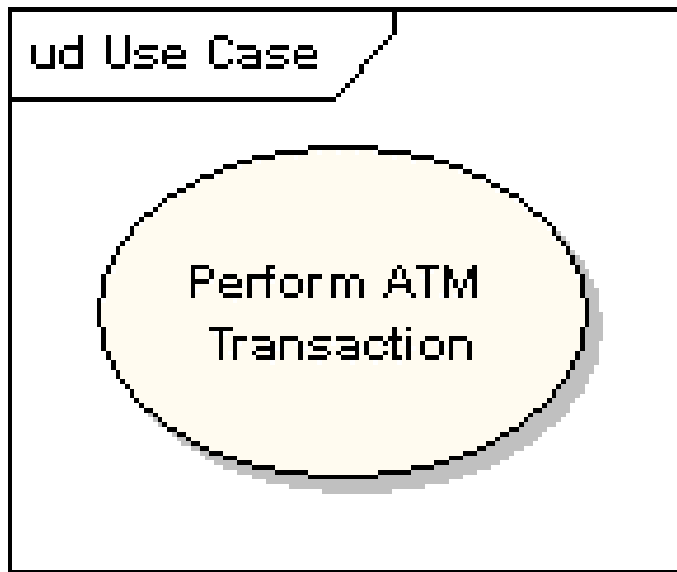


Związek między aktorami typu *Generalization*

Oznacza to dziedziczenie funkcji (przypadków użycia) przez aktora np.

Aktor *CommercialCustomer*

- posiada funkcje dziedziczone od aktora *Customer*
- może korzystać z innych przypadków użycia



Przypadek użycia (*Use Case*)

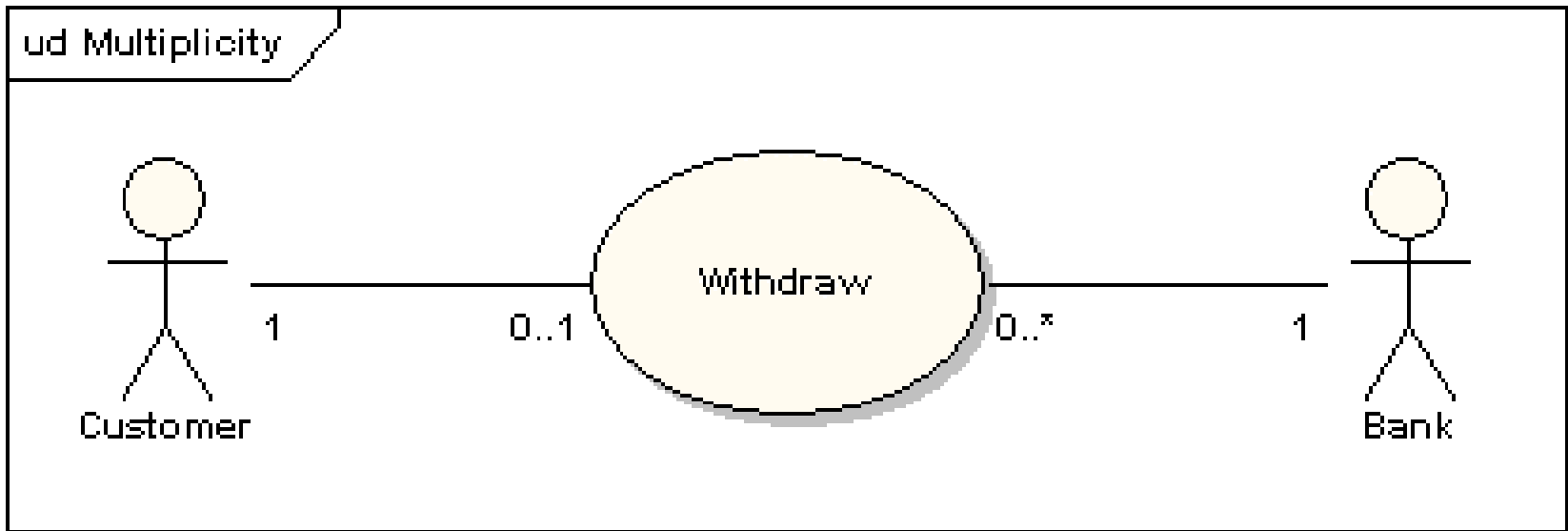
- Jednostka pracy
- Wysoki poziom zewnętrznej obserwacji systemu
- Notacja – elipsa
- <<>> znak stereotypu oznaczającego właściwości związku

Związek użycia przypadku użycia *Use* - skierowana strzałka

- Np. aktor *Customer* używa przypadku użycia *Withdraw* (pobiera pieniądze np. z konta)

Przypadek użycia powinien być zdefiniowany w następujący sposób:

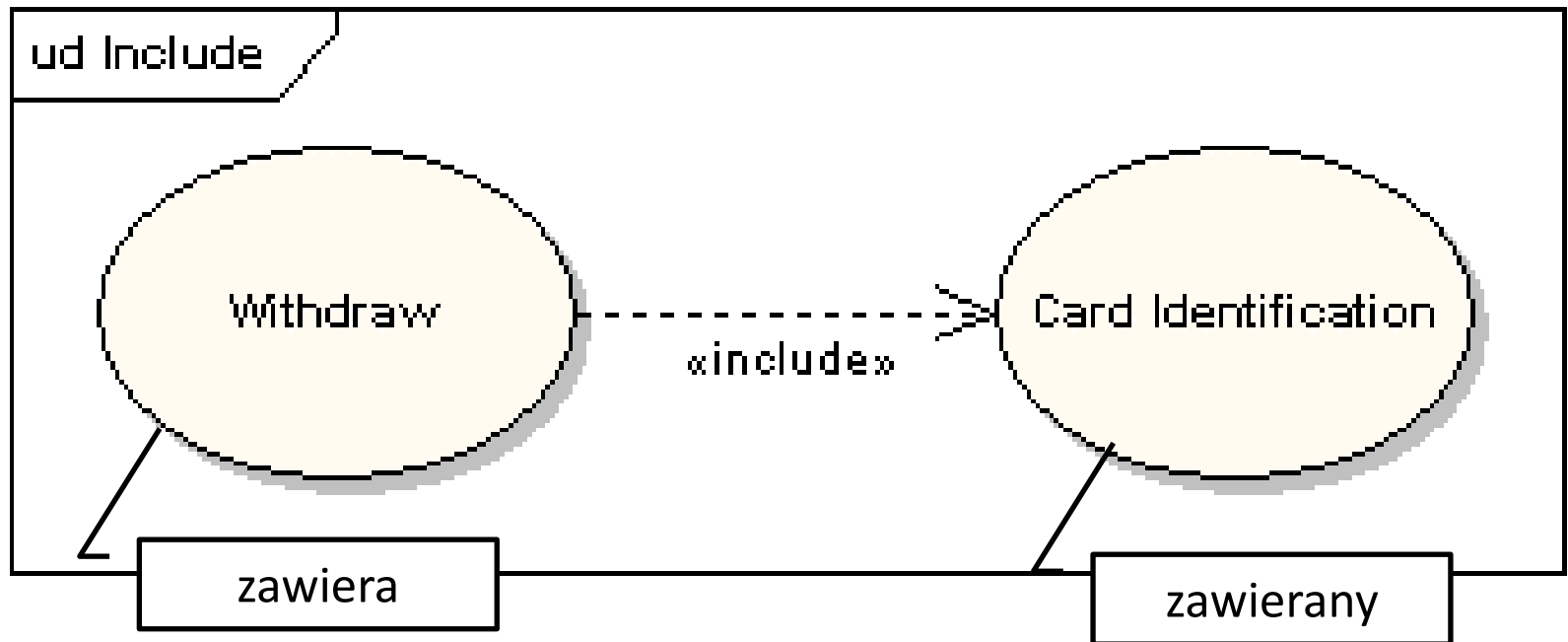
- **Nazwa** (frazą czasownika) **i opis** (krótki nieformalny opis tekstowy)
- **Cel, wymagania** funkcjonalne spełniane dla użytkownika
- **Ograniczenia** – warunki „przed-” i „po-” wykonaniu przypadku użycia oraz niezmienniające się na skutek wykonania przypadku użycia
- **Scenariusze** – sekwencja zdarzeń między systemem i zewnętrznymi użytkownikami (opis tekstowy)
- **Diagramy scenariuszy** – diagramy aktywności
- **Dodatkowe informacje** – np. identyfikacja karty płatniczej przed dokonaniem wyciągu z konta



Powiązania (*Association*) – liczność związku (*Multiplicity*)

Liczność instancji przypadku użycia na końcach połączenia między aktorem i przypadkiem użycia.

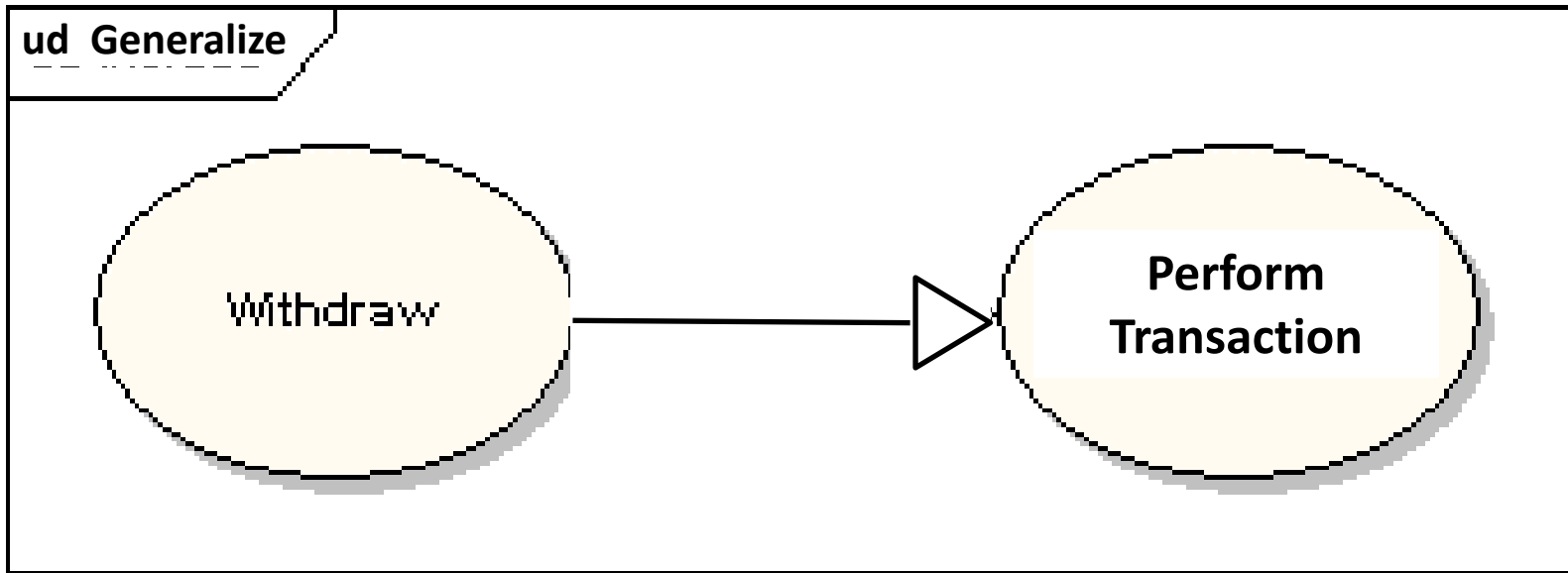
Np. aktor *Klient (Customer)* ma tylko jedną sesję wypłacania pieniędzy w danym momencie (*Withdraw*), natomiast *Bank* może mieć ich wiele w tym samym czasie



Związek Zawieranie <<include>>

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

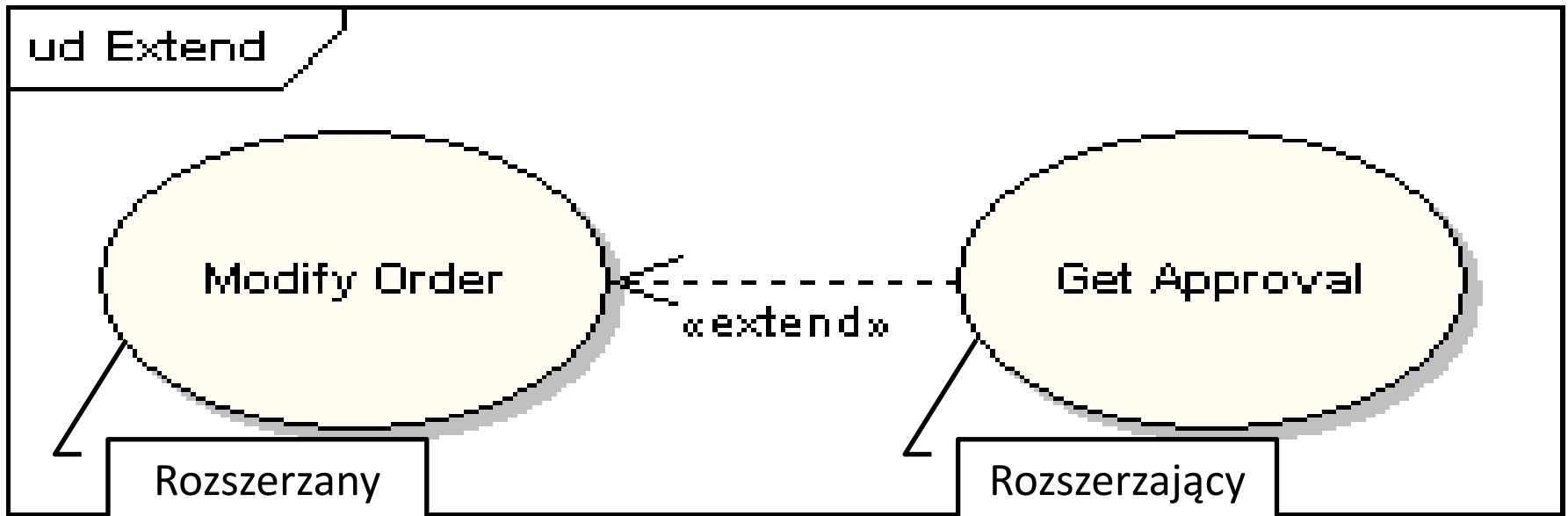
np. Pobranie z konta (*Withdraw*) **zawsze (obligatoryjnie)** musi wykonać identyfikację karty (*Card identification*). Również inne przypadki użycia mogą zawierać PU *Card identification*.



Związek między przypadkami użycia typu *Generalization*

Przypadki użycia mogą generalizować inne przypadki użycia.

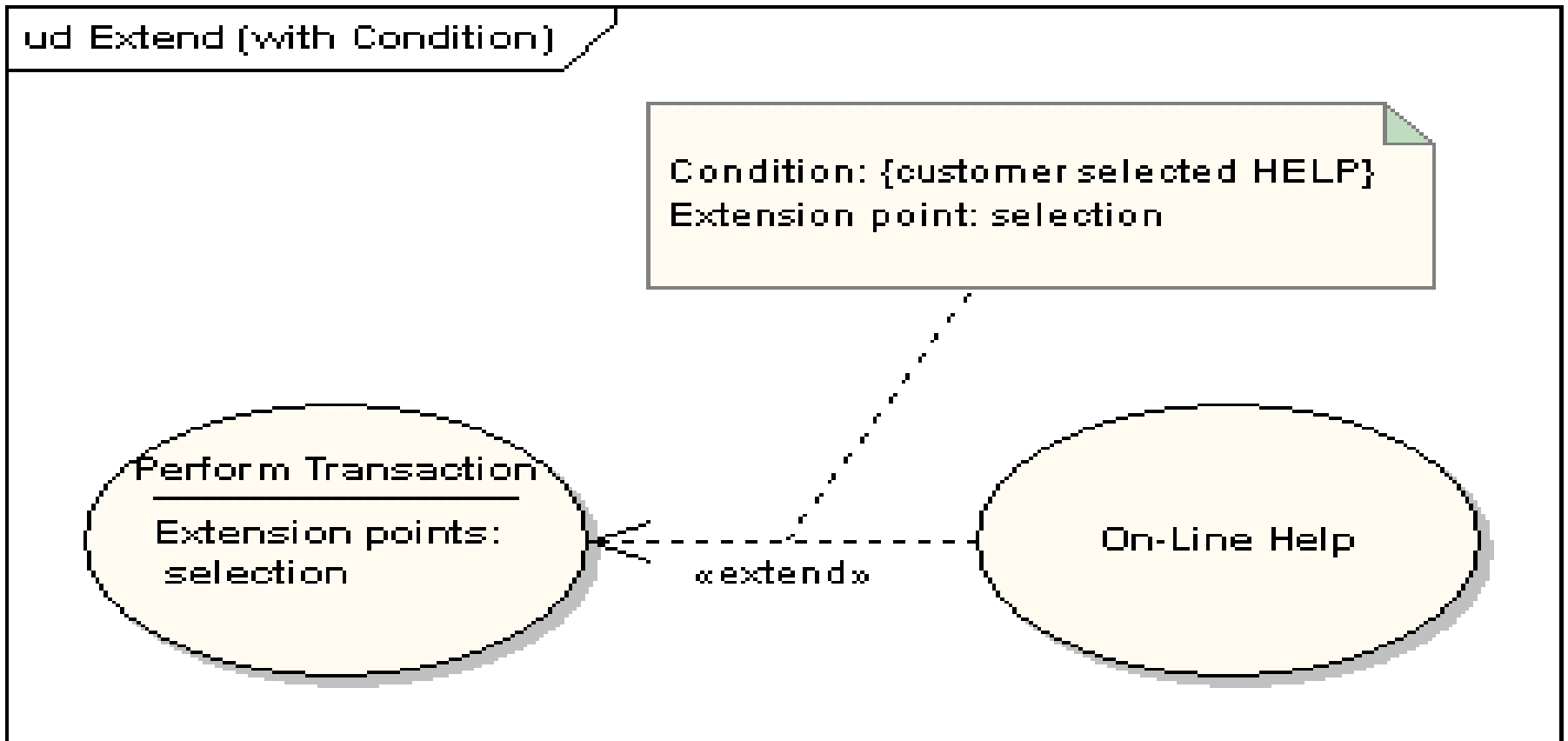
Np. Oznacza to dziedziczenie funkcjonalności przypadku użycia *Perform Transaction* przez przypadek użycia *Withdraw*



Związek *Rozszerzanie* <<extend>>

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

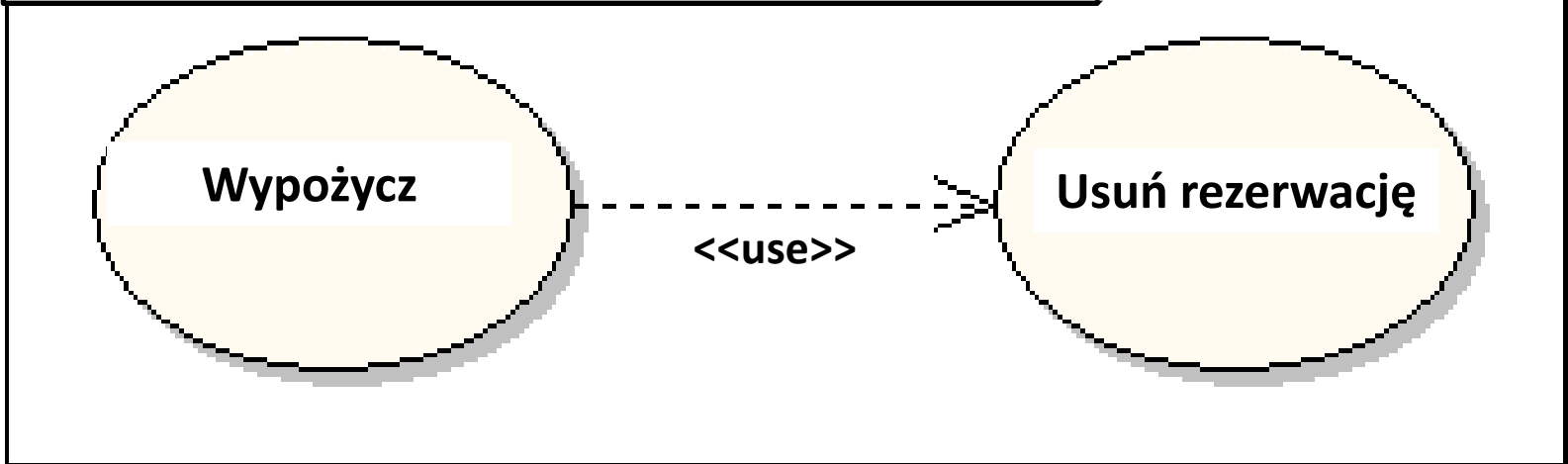
Np. Przypadek użycia *Zezwolenie (Get Approval)* **opcjonalnie** rozszerza właściwości przypadku użycia *Modyfikuj zlecenie (Modify Order)* - może wystąpić przed uruchomieniem procesu „Modify Order”



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)*

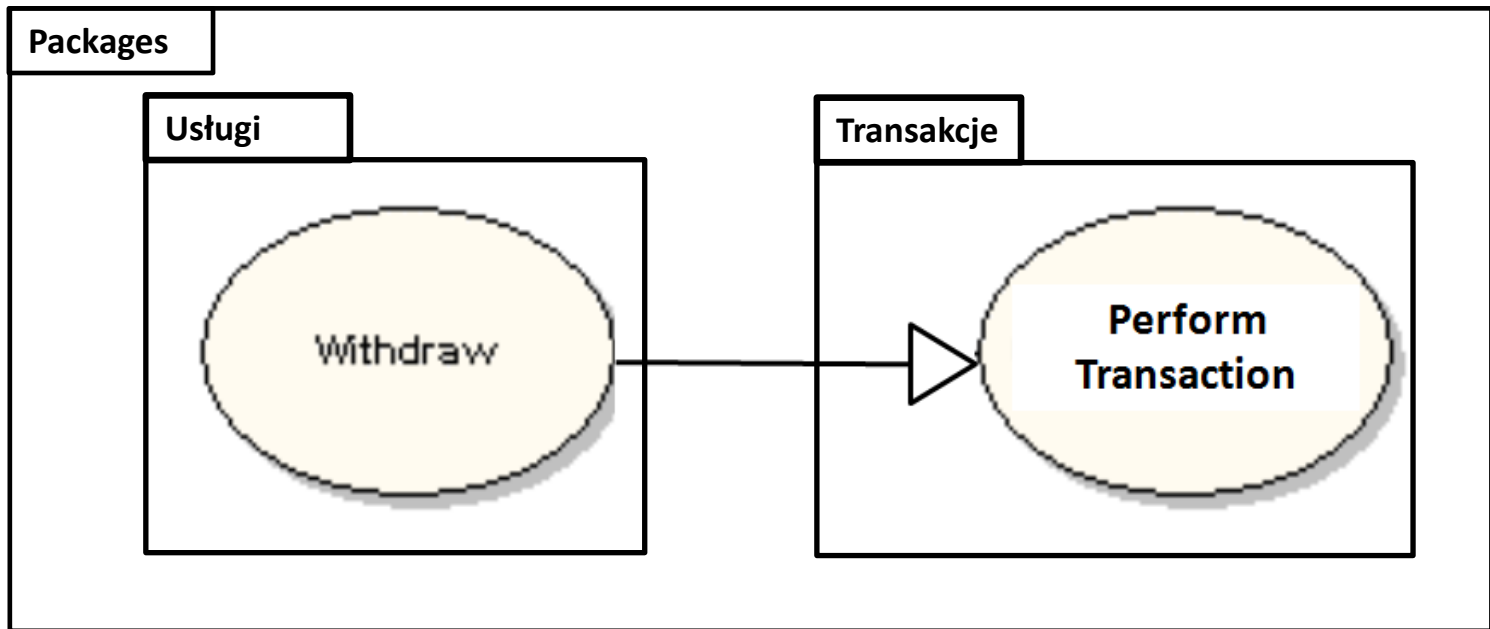
Association zdefiniowana za pomocą stereotypu <<use>>
zdefiniowanego przez analityka



Związek między przypadkami użycia typu *Association*

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>>.

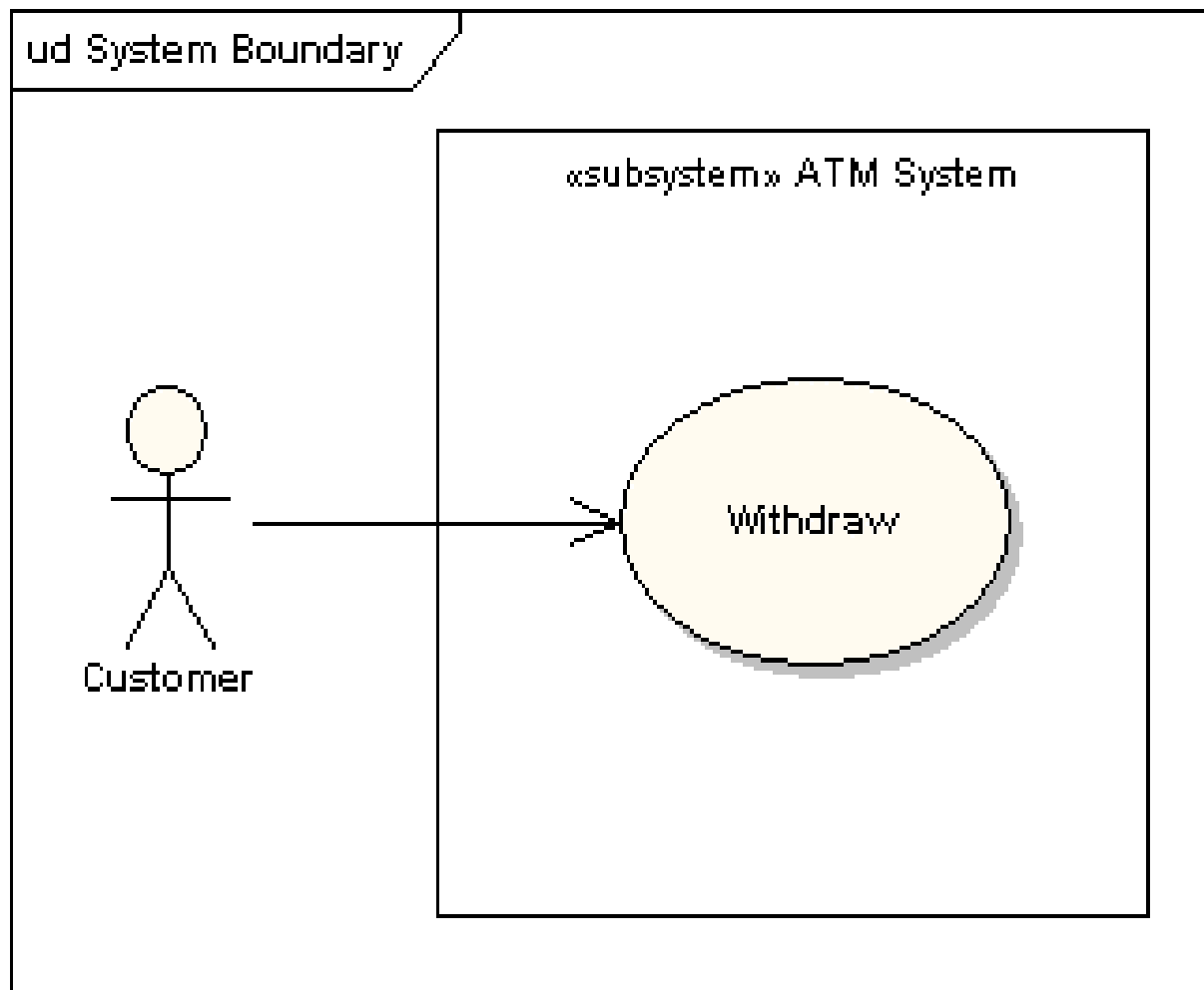
Np. Stereotyp <<use>> może oznaczać, że przypadek użycia *Wypożycz* czasem musi użyć przypadek użycia *Usuń rezerwację*, aby zrealizować proces *Wypożycz* – tylko wtedy, gdy ta rezerwacja została wykonana wcześniej.



Pakiet (package)

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

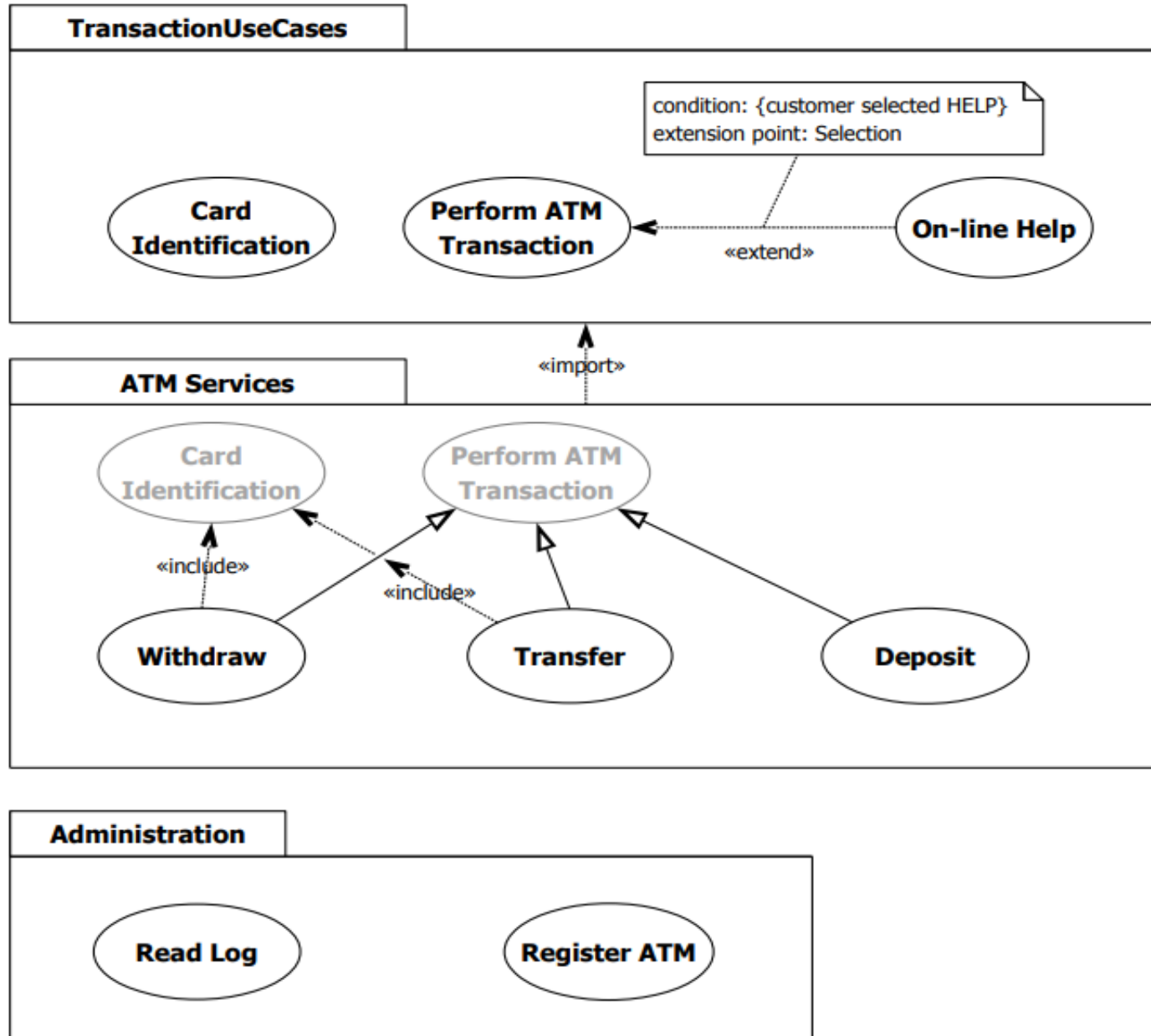
Np. Przypadek użycia *Withdraw* jest usługą w pakiecie *Usługi*, a przypadek użycia *Perform Transaction* jest umieszczony w pakiecie z transakcjami (*Transakcje*).



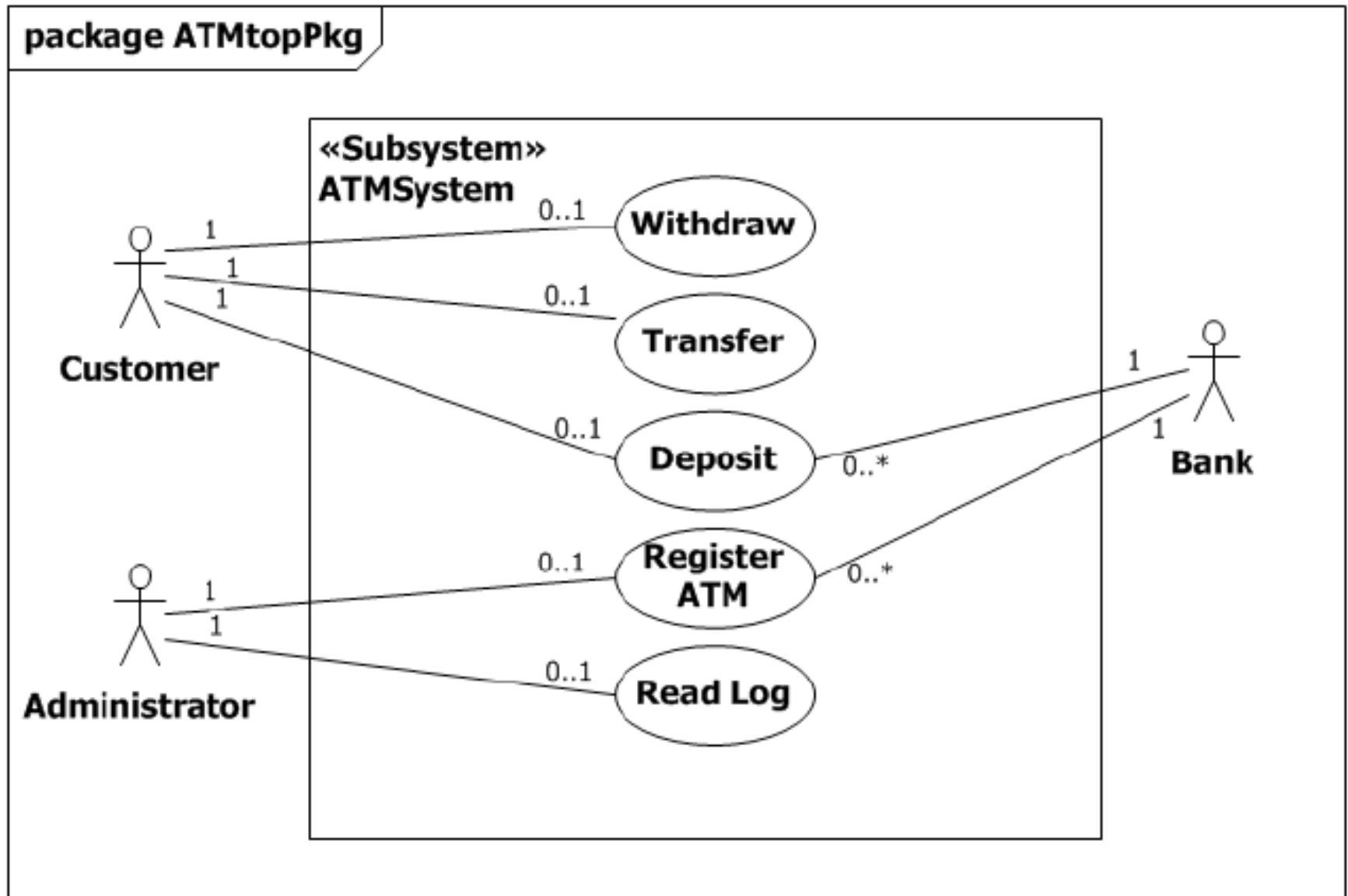
Granice systemu (System Boundary)

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

Przykład 1 diagramu przypadków użycia – <https://www.omg.org/spec/UML> Unified Modeling Language, v2.5.1 (omg.org)



Przykład 2 diagramu przypadków użycia- <https://www.omg.org/spec/UML>
[Unified Modeling Language, v2.5.1 \(omg.org\)](https://www.omg.org/spec/UML)



Tworzenie diagramów przypadków użycia

1. **Elementy diagramów przypadków użycia (use-cases)**
2. **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 oprogramowania na podstawie opisu biznesowego „świata rzeczywistego”** (slajdy 38-41, 54-57, 71-72) 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 9 (przykłady na slajdach 48-51, 62-69, 75-77)

Wytyczne przy modelowaniu granic systemu (p.1, slajd 21)

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

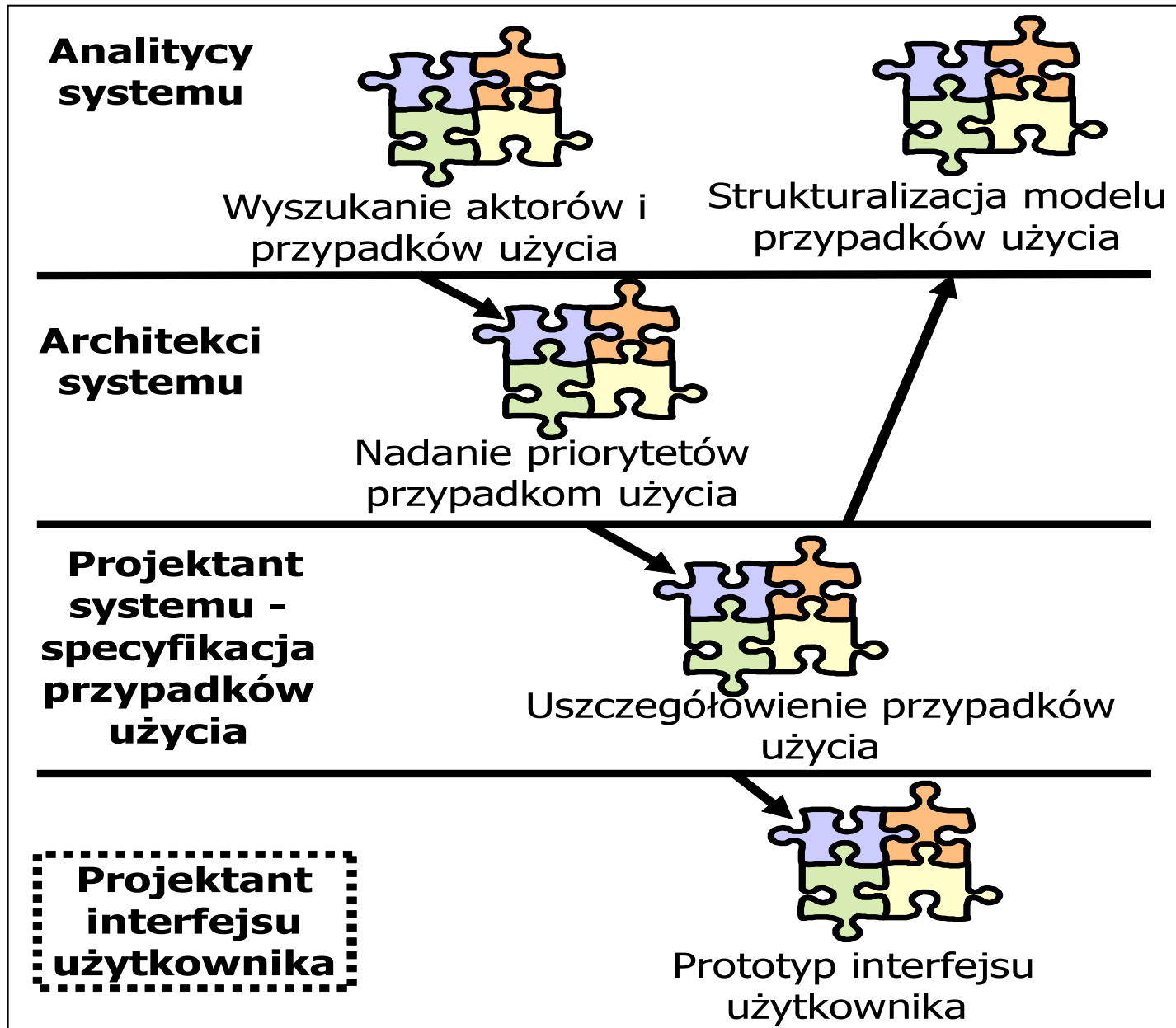
Wytyczne przy modelowaniu wymagań stawianych systemowi oprogramowania (p.2, slajd 21)

- Należy określić otoczenie systemu, czyli zidentyfikować aktorów
- Dla każdego aktora należy podać działania, jakie każdy aktor oczekuje od systemu (slajdy 44, 73)
- Działania należy zapisać jako przypadki użycia
- 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 relacji **Generalization** (analiza wspólności i zmienności) – slajdy 45-46, 36, 74, 61.
- Należy uwzględnić tych aktorów, przypadki użycia oraz zidentyfikowane powiązania między nimi
- Można dodać do każdego aktora i przypadku użycia notatkę opisującą wymagania нефункционалне (np. sprzęt, język, korzystanie z Internetu)

Wytyczne przy modelowaniu przypadków użycia (p.3, slajd 21)

- **Należy opisać główny i nadzwyczajne ciągi zdarzeń** każdego przypadku użycia (slajd 75 – pierwszy ciąg zdarzeń, slajd 76 – drugi ciąg zdarzeń) podając: czynności i dane używane podczas działania przypadku użycia
- **Należy zdefiniować testy systemu** (slajd 77) w odniesieniu do wybranego aktora i powiązanego z 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*)

Rozdzielenie tworzenia interfejsu użytkownika i pozostałych warstw oprogramowania (wykład 1, wykład 6)



Tworzenie diagramów przypadków użycia

1. **Elementy diagramów przypadków użycia (use-cases)**
2. **Wytyczne tworzenia diagramów przypadków użycia (use-cases)**

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

3. **Model przypadków użycia – identyfikacja i specyfikacja wymagań**

Proces identyfikacji wymagań – z modelu biznesowego „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 (wykład 3)	<i>Model dziedziny)* (domain model)</i> -najważniejsze obiekty systemu: „rzeczy” lub zdarzenia podawane przez ekspertów	diagram najważniejszych klas dziedziny (domain classes) z niewielką ilością operacji-metod (około 10-50 w notacji UML), reszta przewidywanych klas w glosariuszu (<i>glossary</i>);
	<i>Model biznesowy)* (business model)</i> - wewnętrzny model procesu biznesowego organizacji, wyszczególniany przez stronę zamawiającą systemu (<i>customers</i>)	„ business use case ” : a) Opis przypadków użycia („uses cases”) i aktorów („actors”) odpowiadających procesowi biznesowemu oraz klientom procesu biznesowego b) biznesowy model obiektowy (business object model) składający się z wykonawców (<i>workers</i>), encji biznesowych (<i>business entities</i>), jednostek pracy (<i>work units</i>) z „use case”

Proces specyfikacji wymagań stawianych oprogramowaniu

funkcjonalne wymagania	Model przypadków użycia (identyfikacja przypadków użycia z modelu biznesowego)	<p>Proces reprezentowania wymagań jako przypadków użycia UML oraz innych produktów:</p> <ol style="list-style-type: none">1. opis tekstowy realizujących zachowanie systemu przy działaniu poszczególnych przypadków użycia - czyli opis sekwencji akcji odpowiednich do modyfikacji, przeglądu, projektowania i testowania,2. model przypadków użycia zawierający aktorów i przypadki użycia oraz powiązania (np. dziedziczenia) między nimi oraz: dotatkowo diagram czynności modelujący scenariusz przypadku użycia – wykład 33. opis architektury przypadków użycia4. glosariusz - definicje ważnych pojęć wyprowadzanych z modelu dziedziny lub modelu biznesowego,5. prototyp interfejsu użytkownika - interakcje między aktorami - ludźmi i oprogramowaniem
niefunkcjonalne wymagania	uzupełniające wymagania lub indywidualne wymagania	<ol style="list-style-type: none">1. specjalne wymagania zawierające niefunkcjonalne wymagania w postaci opisu tekstowego2. ograniczenia środowiska i implementacji (np. typ komputera, typ plików, rodzaj systemu operacyjnego, typ oprogramowania Internetu), zależności, konserwacja, zdolność do poszerzania,

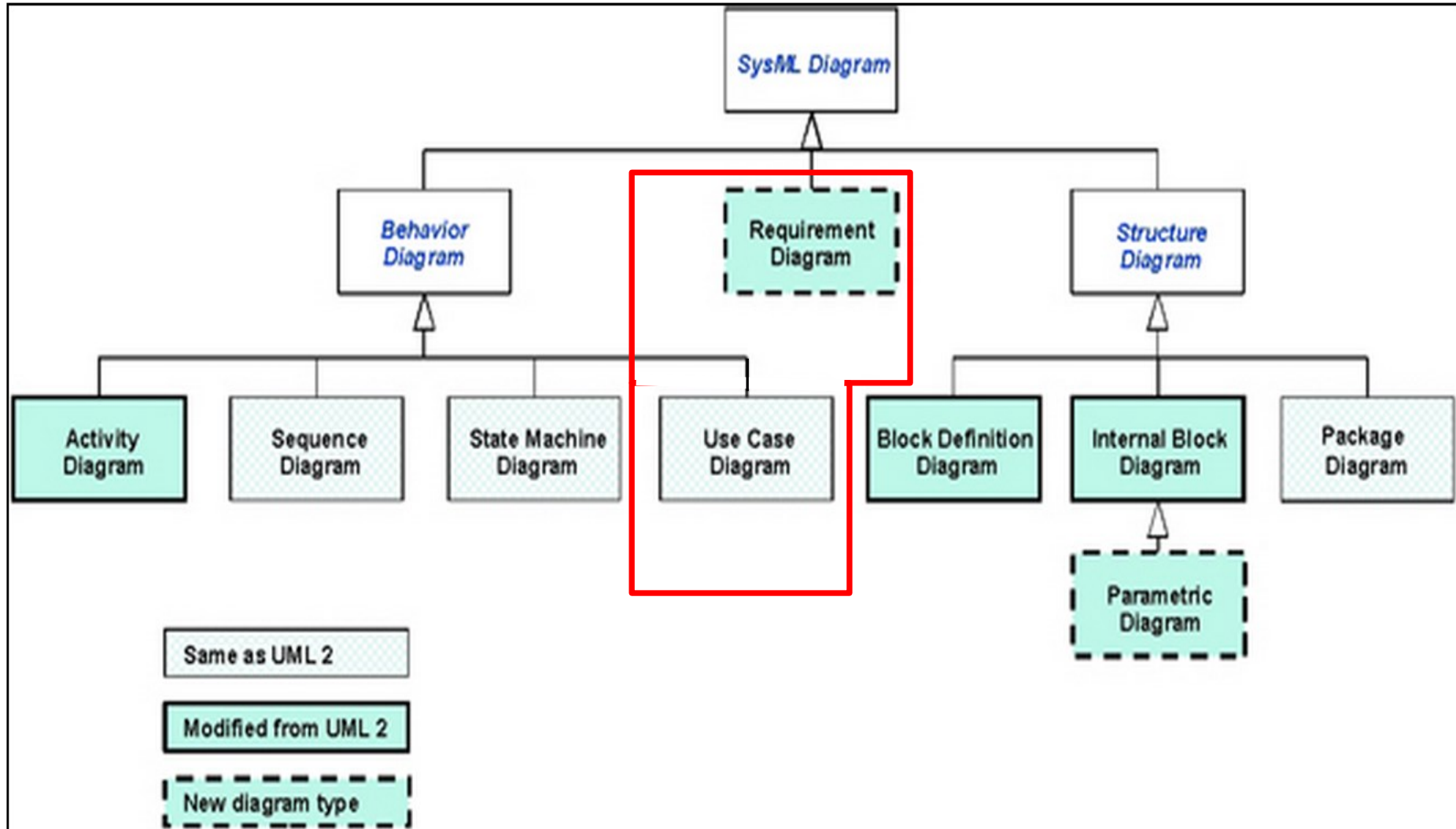
Rola wymagań funkcjonalnych i нефункциональных programu zidentyfikowanych na podstawie modelu biznesowego „świata rzeczywistego”

Wymagania przedstawiają czynności projektowanego oprogramowania

1. Co program powinien robić, aby zautomatyzować procesy wykonywane przez tzw. zasoby ludzkie? – reprezentowane przez **wymagania funkcjonalne**.
2. Jakie rozwiązania technologiczne zastosować, aby program spełniał oczekiwania strony zamawiającej w zakresie bezpieczeństwa (security), niezawodności (safety), skalowalności i wydajności - reprezentowane przez **wymagania нефункциональные**.

Definiowanie wymagań

Zastosowanie diagramu wymagań SysML (wykład 1)



**Przykład 1 identyfikacji przypadków użycia
na diagramie przypadków użycia na
podstawie zdefiniowanych wymagań
wg przykładu z:**

**OMG Systems Modeling Language
(OMG SysML™) *Version 1.7 Beta1***

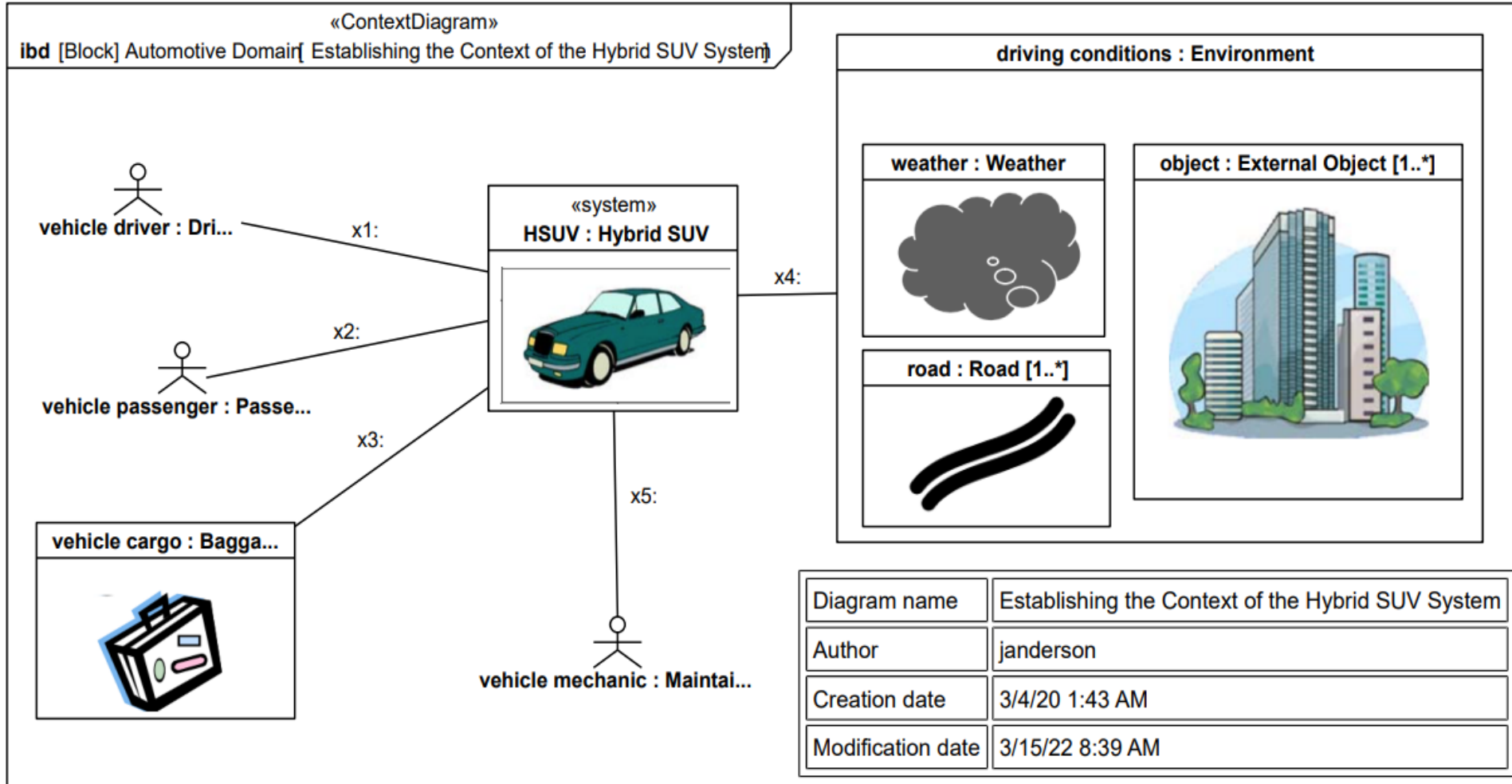
***dotyczącego obsługi samochodu typu SUV (Sport Utility
Vehicle) z napędem hybrydowym***

<https://www.omg.org/spec/SysML/1.7/Beta1/PDF>

Proces identyfikacji i specyfikacji przypadków użycia

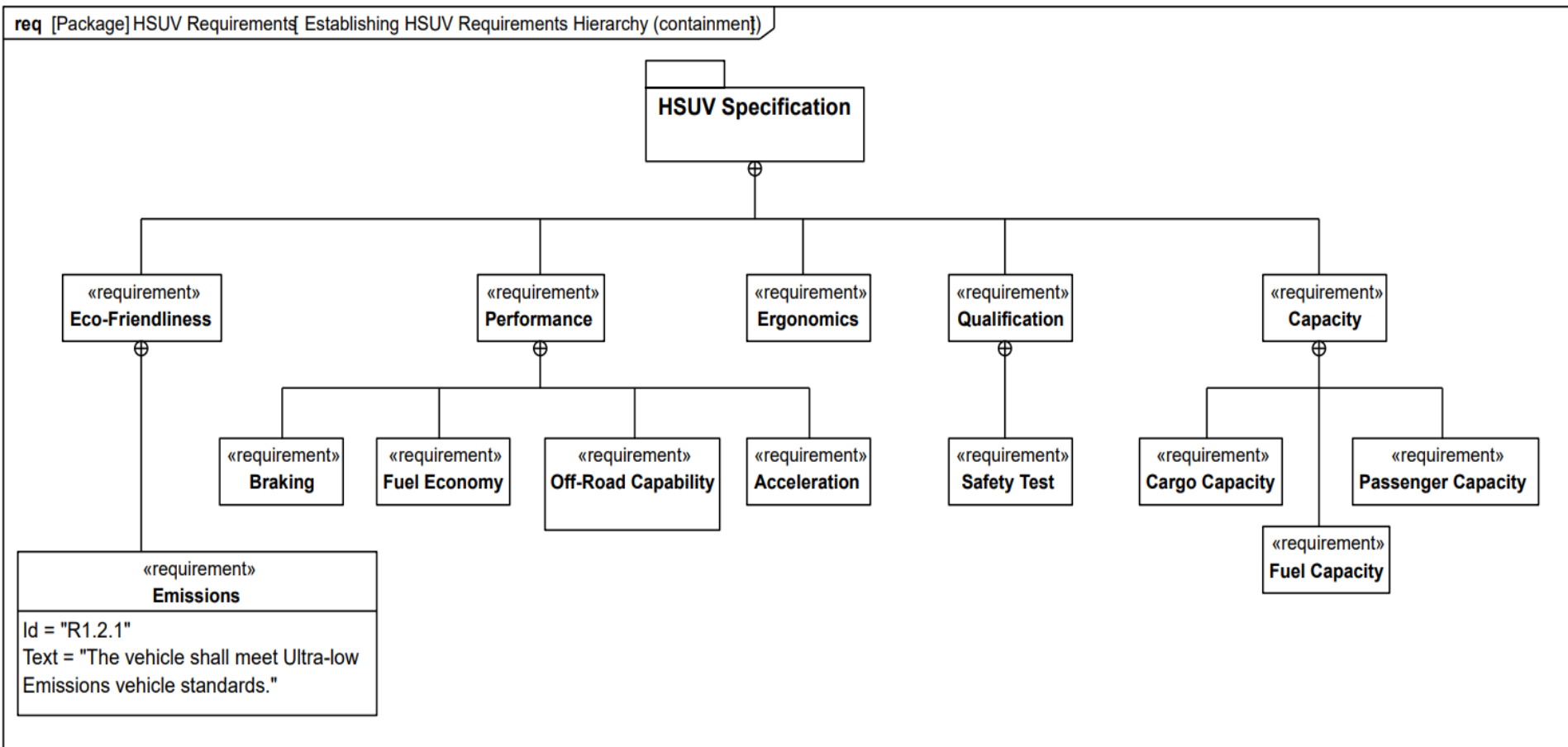
- I. Model biznesowy „świata rzeczywistego” w postaci *diagramu kontekstu* utworzonego przez eksperta problemu reprezentującego stronę zamawiającą czyli użytkownika i klienta, przekazana wykonawcy systemu
- II. **Diagram wymagań oprogramowania** prezentujący hierarchię wymagań oraz wymagania pochodne utworzony przez wykonawców na podstawie modelu biznesowego „świata rzeczywistego”
- III. Model przypadków użycia aplikacji oparty na diagramie przypadków użycia ” utworzony przez wykonawców na podstawie **diagramu wymagań tworzonego oprogramowania**

I. Definiowanie elementów modelu biznesowego „świata rzeczywistego” za pomocą diagramu kontekstu- [SysML, v1.7 \(omg.org\)](http://SysML, v1.7 (omg.org))



II. Diagram wymagań oprogramowania

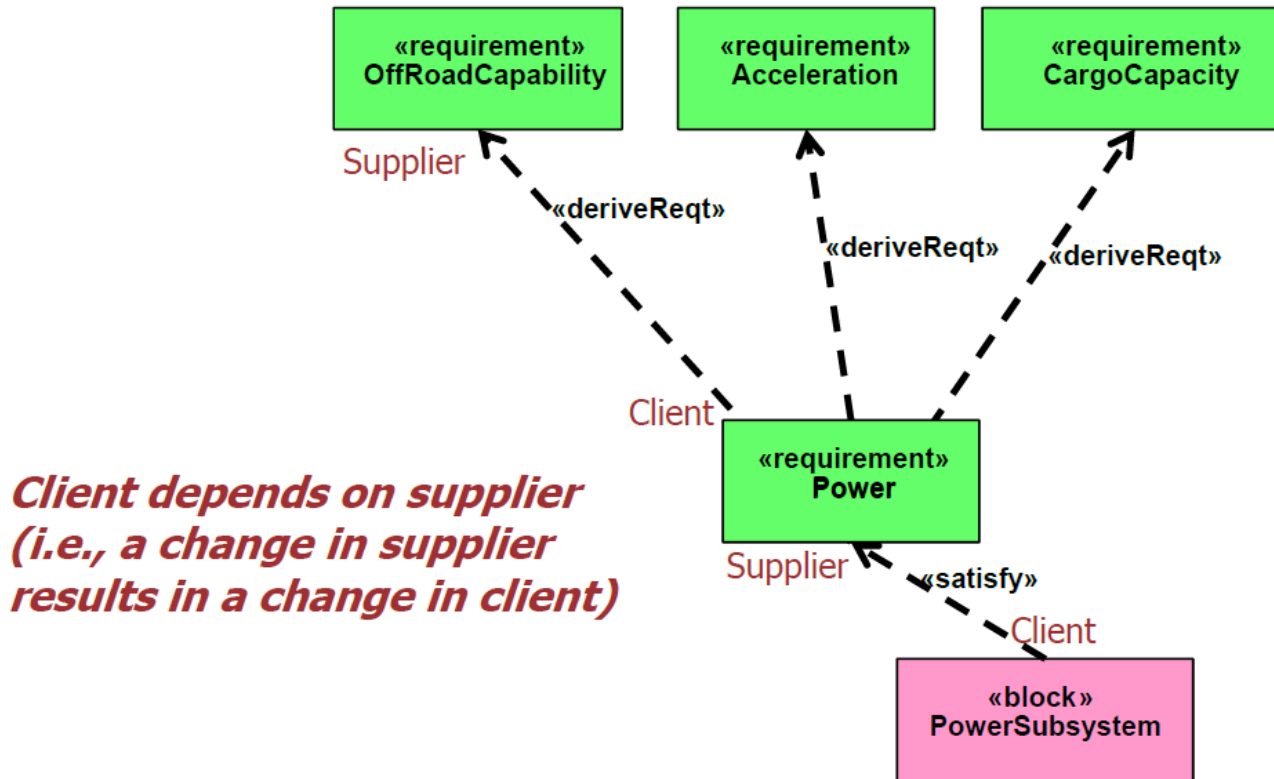
a) hierarchia wymagań



Znaczenie wybranych relacji : <<deriveReq>> i <<satisfy>> między wymaganiami z diagramu wymagań



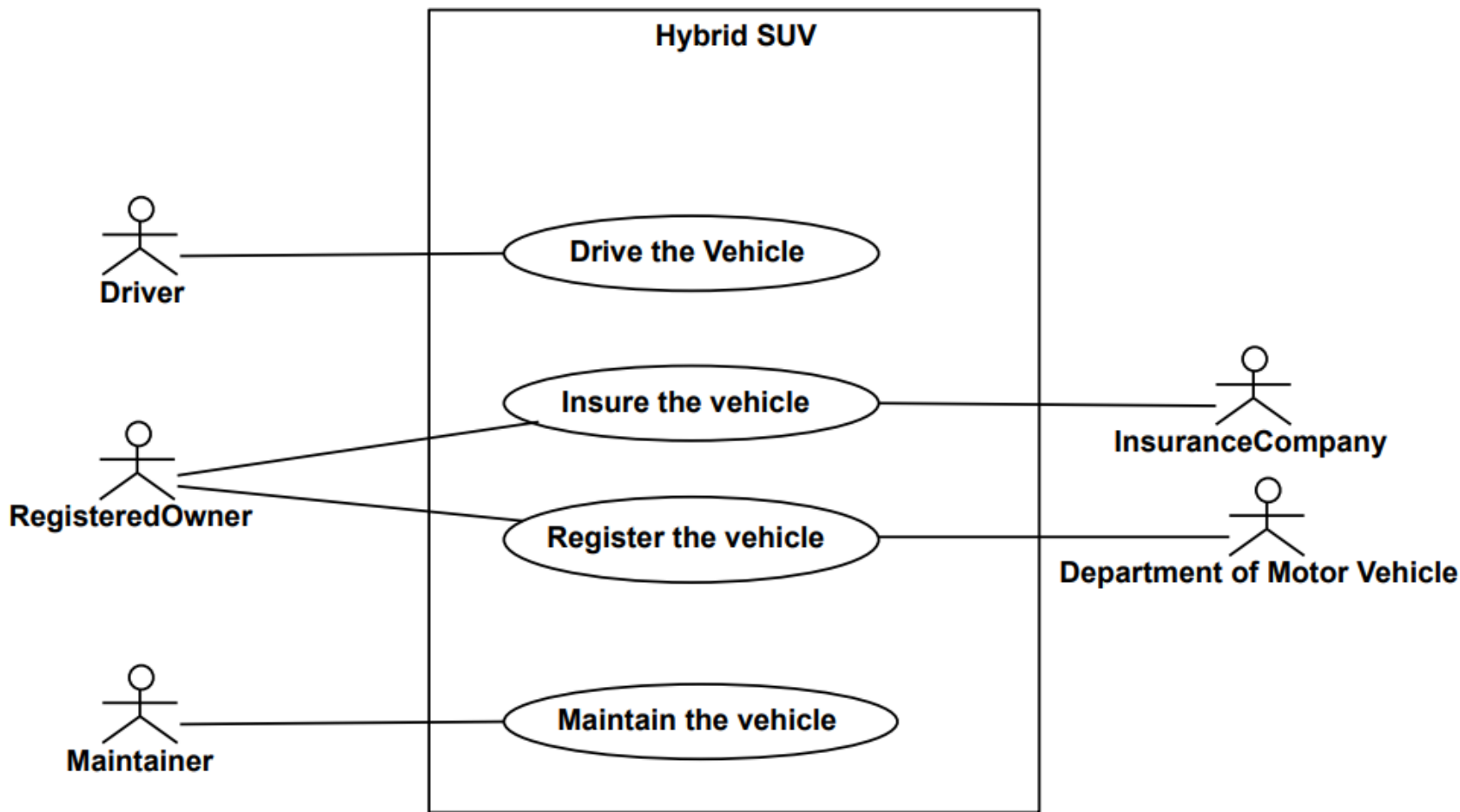
Example of Derive/Satisfy Requirement Dependencies



Arrow Direction Opposite Typical Requirements Flow-Down

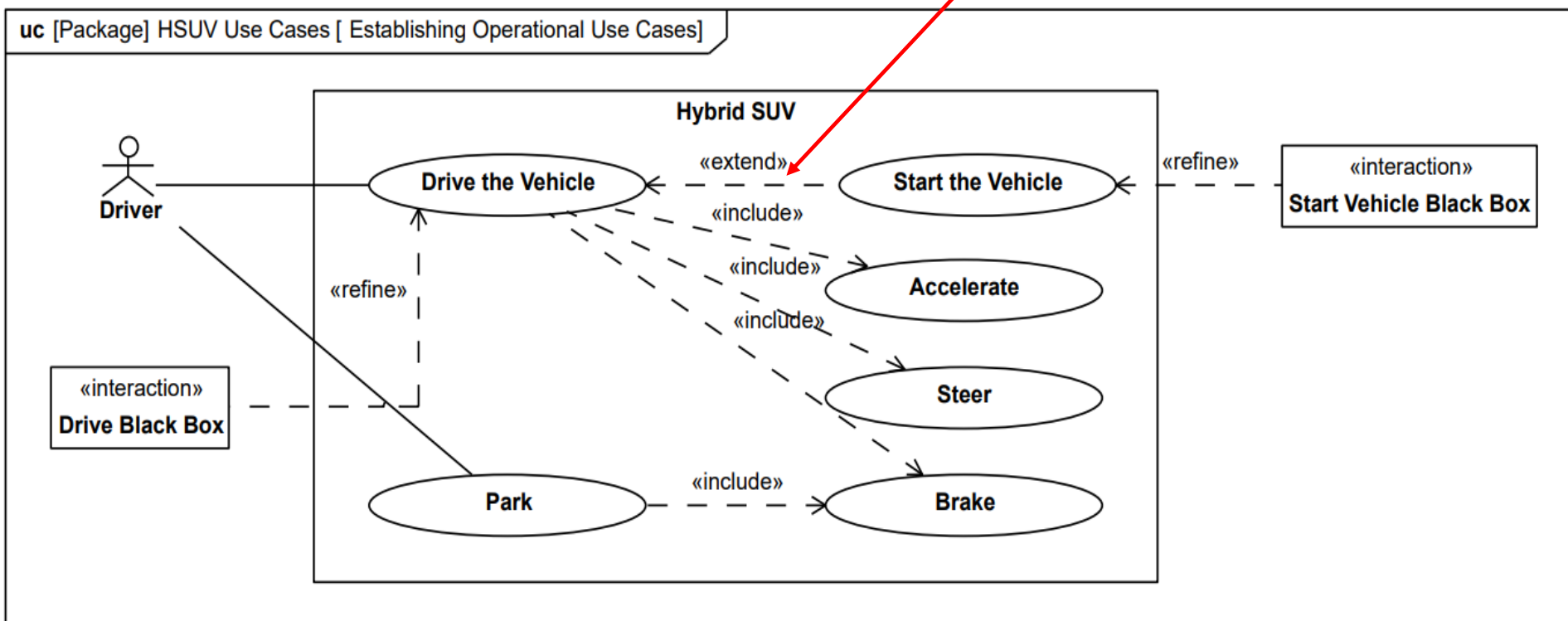
III. Diagram przypadków użycia zawierający specyfikację wymagań z diagramu wymagań oprogramowania

uc [Package] HSUV Use Cases[Establishing Top Level Use Cases]



III. Diagram przypadków użycia zawierający specyfikację wymagań z diagramu wymagań oprogramowania (cd)

Opcjonalna czynność „Start the vehicle” – może wystąpić przed wykonaniem procesu „Drive the vehicle”



“Accelerate”, “Steer” i “Brake” są częścią procesu „Drive the vehicle”

**Przykład 2 identyfikacji i specyfikacji
przypadków użycia na diagramie
przypadków użycia
System sporządzania rachunków**

Proces identyfikacji i specyfikacji przypadków użycia

- I. Opis biznesowy „świata rzeczywistego” w języku klienta wykonanego przez eksperta problemu reprezentującego stronę zamawiającą czyli użytkownika i klienta, przekazana wykonawcy systemu
- II. Sformułowanie wymagań funkcjonalnych i niefunkcjonalnych aplikacji oraz utworzenie diagramu wymagań przez wykonawców na podstawie opisu „świata rzeczywistego”
- III. Model przypadków użycia aplikacji oparty na diagramie przypadków użycia utworzony przez wykonawców

I. Opis biznesowy „świata rzeczywistego” w języku klienta

1. Opis zasobów ludzkich

Co robią pracownicy?

2. Przepisy i strategia firmy

Co ogranicza działalność firmy?

3. Dane techniczne

Dane ilościowe:

ilu pracowników,
ile danych,
jak często,

Dane o lokalizacji firmy

Dane o klientach firmy

Dane o używanym sprzęcie i oprogramowaniu

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 dodawać nowe typy produktów. Pracownik ten może dodatkowo 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.

II. Wymagania stawiane tworzonej aplikacji

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** o unikalnym numerze
4. Pozycje rachunku muszą zawierać produkty różne w sensie nazwy, ceny, podatku i promocji – **pozycje rachunku to wynik zakupu produktu.**
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 – na żądanie należy **podać wartość rachunku.**

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

Diagram wymagań funkcjonalnych programu

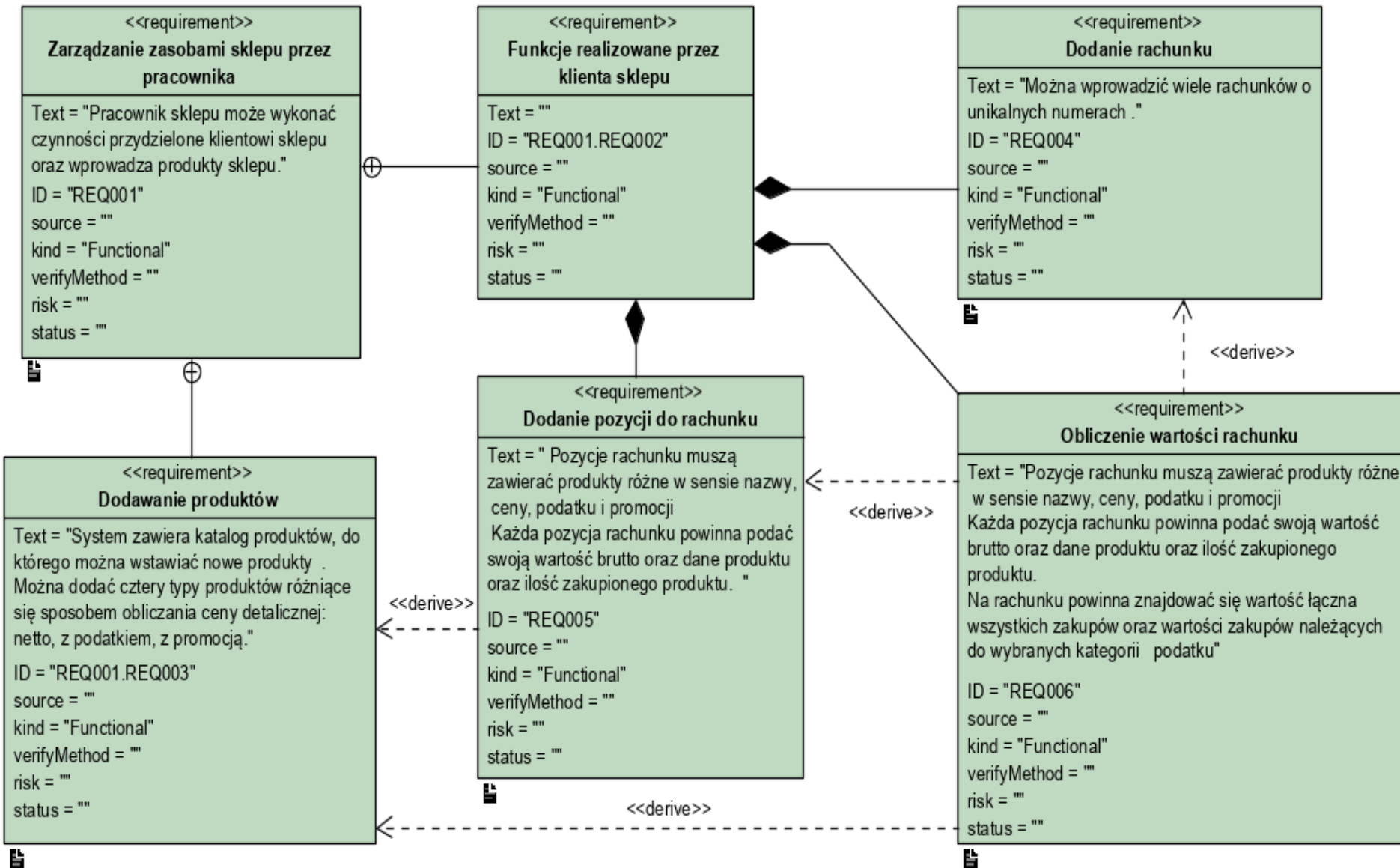
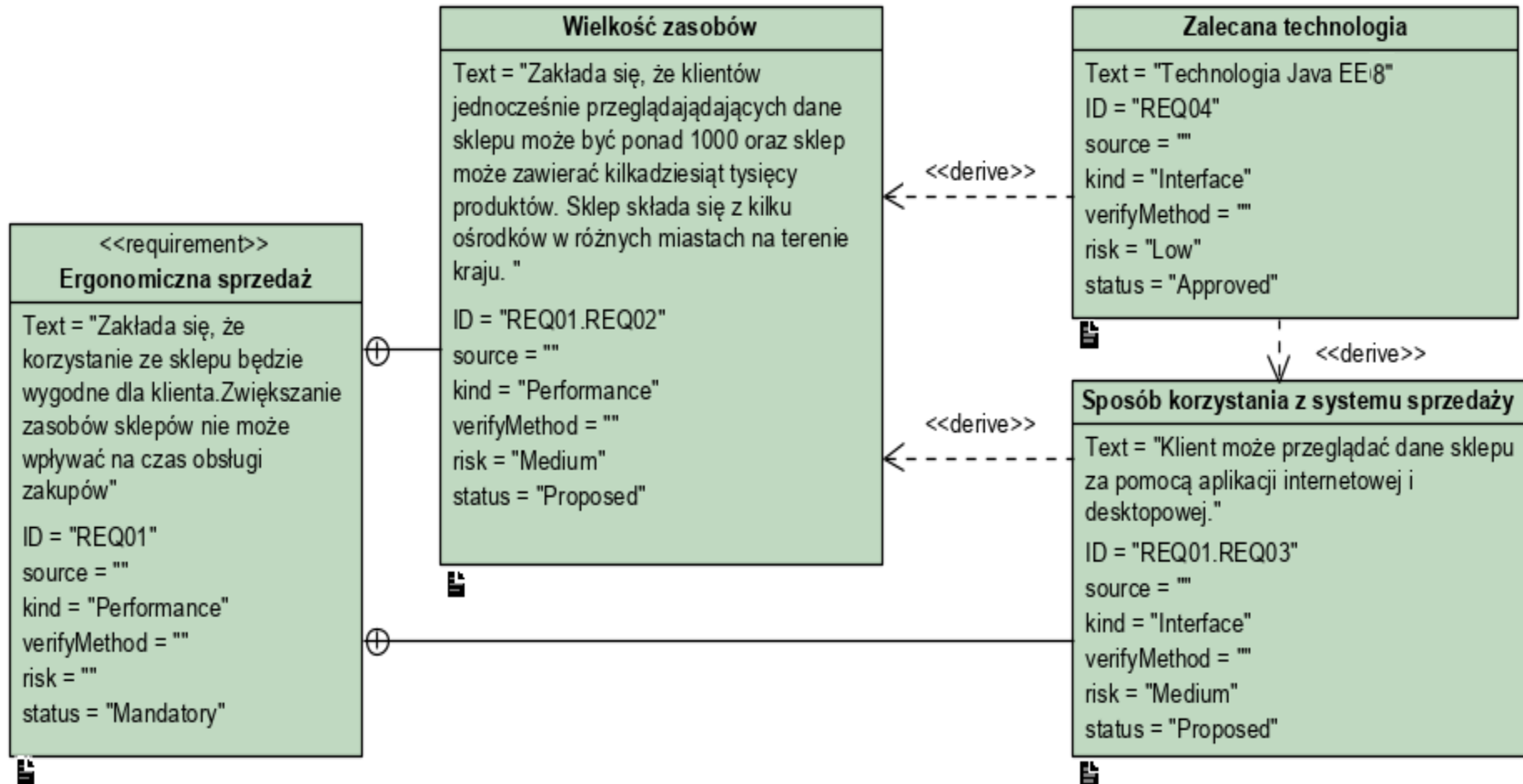


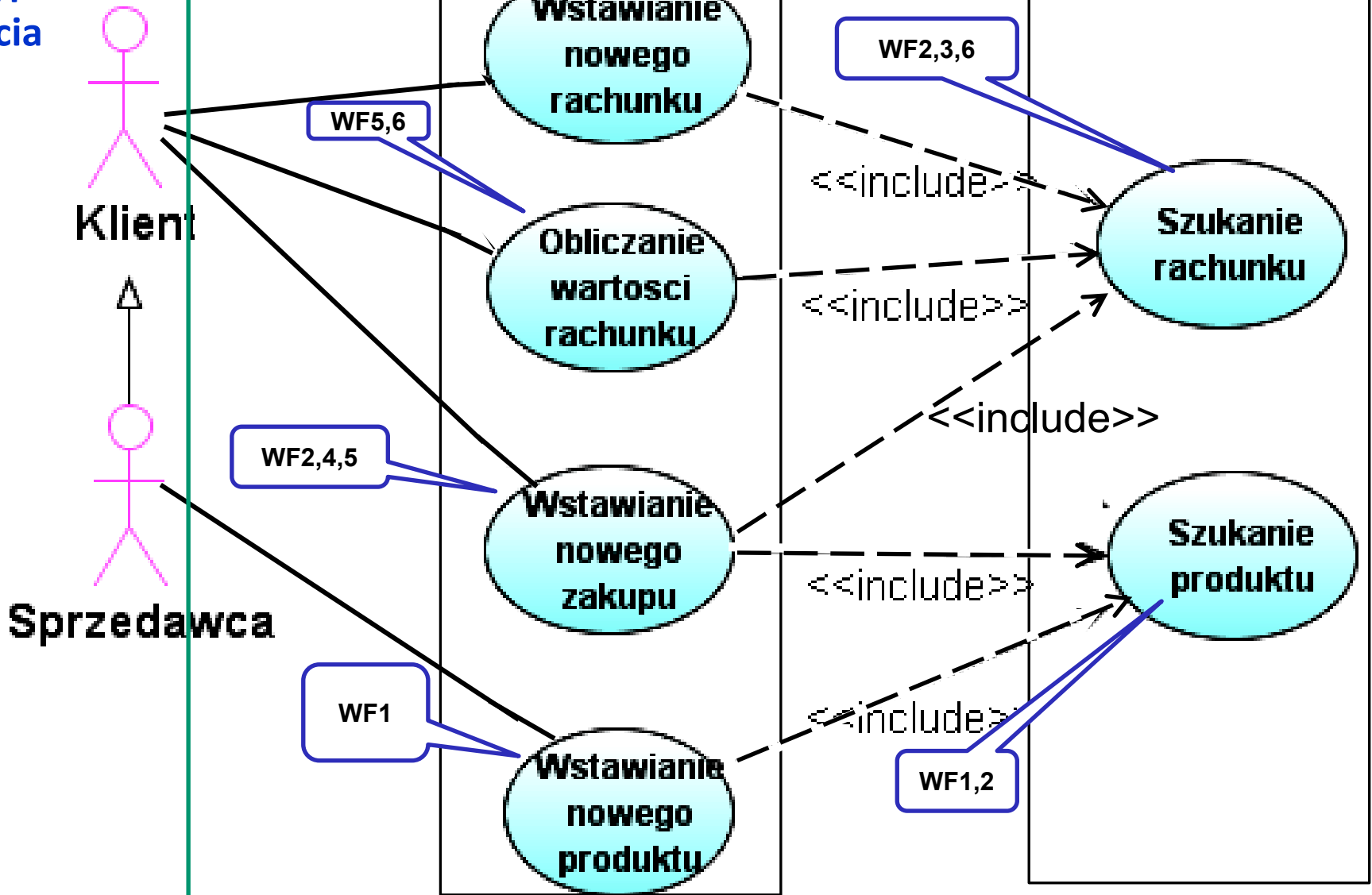
Diagram wymagań niefunkcjonalnych programu



III. Model przypadków użycia aplikacji oparty na diagramie przypadków użycia. Przykład definicji aktorów



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">• PU Wstawianie nowego rachunku powiązane przez <<include>> z PU Szukanie rachunku• PU Obliczanie wartości rachunku powiązane przez <<include>> z PU Szukanie rachunku• PU Wstawianie nowego zakupu powiązane przez <<include>> z PU Szukanie rachunku oraz <u>powiązane przez <<include>> z PU Szukanie produktu</u>
Sprzedawca	<i>Sprzedawca może dodatkowo dodawać nowe produkty</i>	<ul style="list-style-type: none">• PU Wstawianie nowego rachunku powiązane przez <<include>> z PU Szukanie rachunku• PU Obliczanie wartości rachunku powiązane przez <<include>> z PU Szukanie rachunku• <u>PU Wstawianie nowego zakupu powiązane przez <<include>> z PU Szukanie rachunku oraz powiązane przez <<include>> z PU Szukanie produktu</u>• PU Wstawianie nowego produktu powiązane przez <<include>> z PU Szukanie produktu

Rezultat –
diagram
przypadków
użycia



Powiązanie wybranych wymagań z jednym z przypadków użycia z diagramu przypadków użycia. Scenariusze przypadków użycia: slajdy 48-51

Wstawianie nowego produktu > Wstawianie nowego produktu Details

9+  

Wstawianie nowego produktu

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



Level: User

Complexity: Medium

Use Case Status: Complete

Implementation Status: Started

Preconditions:

 [Zarządzanie zasobami sklepu przez pracownika](#)  [Dodawanie produktów](#)
inicjalizacja przez uruchomienie programu (np. otwarcie strony WWW, start aplikacji)

Post-conditions:

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

Author: Zofia Kruczkiewicz

Assumptions:

Opisy przypadków użycia

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 promocję, 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 oraz dane produktu i jego ilość
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 wypadku wstawia się nowy zakup.**

Tworzenie diagramów przypadków użycia

1. **Elementy diagramów przypadków użycia (use-cases)**
2. **Wytyczne tworzenia diagramów przypadków użycia (use-cases)**

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

3. **Model przypadków użycia – identyfikacja i specyfikacja wymagań**
4. **Przykłady diagramów przypadków użycia (use-cases)**

**Przykład 3 identyfikacji i specyfikacji
przypadków użycia na diagramie
przypadków użycia
Biblioteka**

Proces identyfikacji i specyfikacji przypadków użycia (slajd 38)

- I. Opis biznesowy „świata rzeczywistego” w języku klienta
- II. Sformułowanie wymagań funkcjonalnych i niefunkcjonalnych aplikacji oraz wykonanie diagramu wymagań.
- III. Model przypadków użycia aplikacji oparty na diagramie przypadków użycia

I. Opis biznesowy „świata rzeczywistego”

1. Opis zasobów ludzkich

Bibliotekarz może dodawać do katalogu tytułów nowe tytuły. **Każdy tytuł jest reprezentowany przez dane:** tytuł, autor, wydawnictwo, ISBN oraz informacje o liczbie egzemplarzy i miejscu ich przechowywania i występuje w bibliotece jako pojedyncza informacja dla każdego tytułu. **Pewna grupa tytułów opisuje książki nagrane na wybranym nośniku**, dlatego dodatkowo tytuł zawiera **dane nagrania np. nazwisko aktora**. **Każdy egzemplarz, niezależnie, czy jest książką czy kasetą, jest opisany odrębną informacją zawierającą numer egzemplarza**, który może się powtarzać dla różnych tytułów. Bibliotekarz może dodawać nowe tytuły i egzemplarze oraz je przeszukiwać, natomiast klient może jedynie przeszukiwać tytuły i sprawdzać egzemplarze wybranych tytułów.

W celu wypożyczenia książki **klient musi ją zarezerwować podając dane rejestracji, dane książki oraz datę rezerwacji**. Klient musi **wypożyczyć** zarezerwowaną książkę w terminie jej rezerwacji podając dane rejestracji i rezerwacji, wyszukać rezerwację i następnie ją usunąć. Musi wykonać dane wypożyczenia zawierające: dane rejestracji, dane zarezerwowanej książki oraz datę zwrotu. **Rezerwacje można usunąć bez konieczności jej wypożyczenia – w terminie rezerwacji**.

W celu **zwrotu książki** należy podać dane rejestracji oraz dane wypożyczonej książki. Zwrot musi nastąpić w okresie wypożyczenia podanym w danych wypożyczenia.

Opis biznesowy „świata rzeczywistego” (cd)

2. Przepisy

Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem wypożyczalni.

3. Dane techniczne

Klient może przeglądać dane wypożyczalni za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu. Zakłada się, że klientów jednocześnie przeglądających dane wypożyczalni może być ponad 1000 oraz wypożyczalnia może zawierać kilkadziesiąt tysięcy tytułów oraz przynajmniej dwukrotnie więcej egzemplarzy. Biblioteka składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

II. Sformułowanie wymagań funkcjonalnych i нефункциональных

Lista wymagań funkcjonalnych programu

1. System zawiera katalog zarejestrowanych klientów w wyniku **rejestracji klientów**.
2. System zawiera katalog tytułów w wyniku **dodawania tytułów książek**
3. System zawiera dwa typy egzemplarzy do wypożyczenia: książki i kasety z nagraniami dźwiękowymi książek w wyniku **dodawania tych egzemplarzy**.
4. Każdy egzemplarz zawiera tytuł, nazwisko autora, ISBN, wydawnictwo, jeśli jest to książka oraz dodatkowo nazwisko aktora, jeżeli jest to nagranie dźwiękowe.
Może wystąpić wiele egzemplarzy książek oraz kaset z tymi samymi tytułami. Każdy egzemplarz, zarówno książka i kaset, posiadają numer niepowtarzający się w ramach pozostałych identycznych danych (ISBN lub ISBN i nazwisko aktora).
5. W celu znalezienia tytułu należy podać ISBN lub ISBN i nazwisko aktora dla nagranej książki
6. W celu wybrania właściwego egzemplarza należy podać ISBN, jeśli jest to książka oraz dodatkowo nazwisko aktora, jeśli jest to kasetka oraz numer egzemplarza.
7. Zarówno egzemplarze typu książka lub kasetka, mogą być przeznaczane do wypożyczenia na okres umowny oraz na okres ściśle określony w wyniku **rezerwacji książki**.
8. Książki można **wypożyczać** jedynie po wykonaniu rezerwacji i następnie **usunięcia rezerwacji**.
9. **Zwrot wypożyczonej książki** musi nastąpić w wyznaczonym terminie.

Lista wymagań нефункциональных programu

1. Wstawianie danych o tytułach i egzemplarzach może odbywać się przez uprawnione osoby
2. Wyszukiwanie informacji powinno odbywać się samodzielnie przez klienta
3. Operacje zarządzania i wyszukiwania informacji mogą być dokonane przez Internet przez aplikację uruchamianą przez przeglądarkę lub bez jej pośrednictwa

Diagram wymagań funkcjonalnych programu

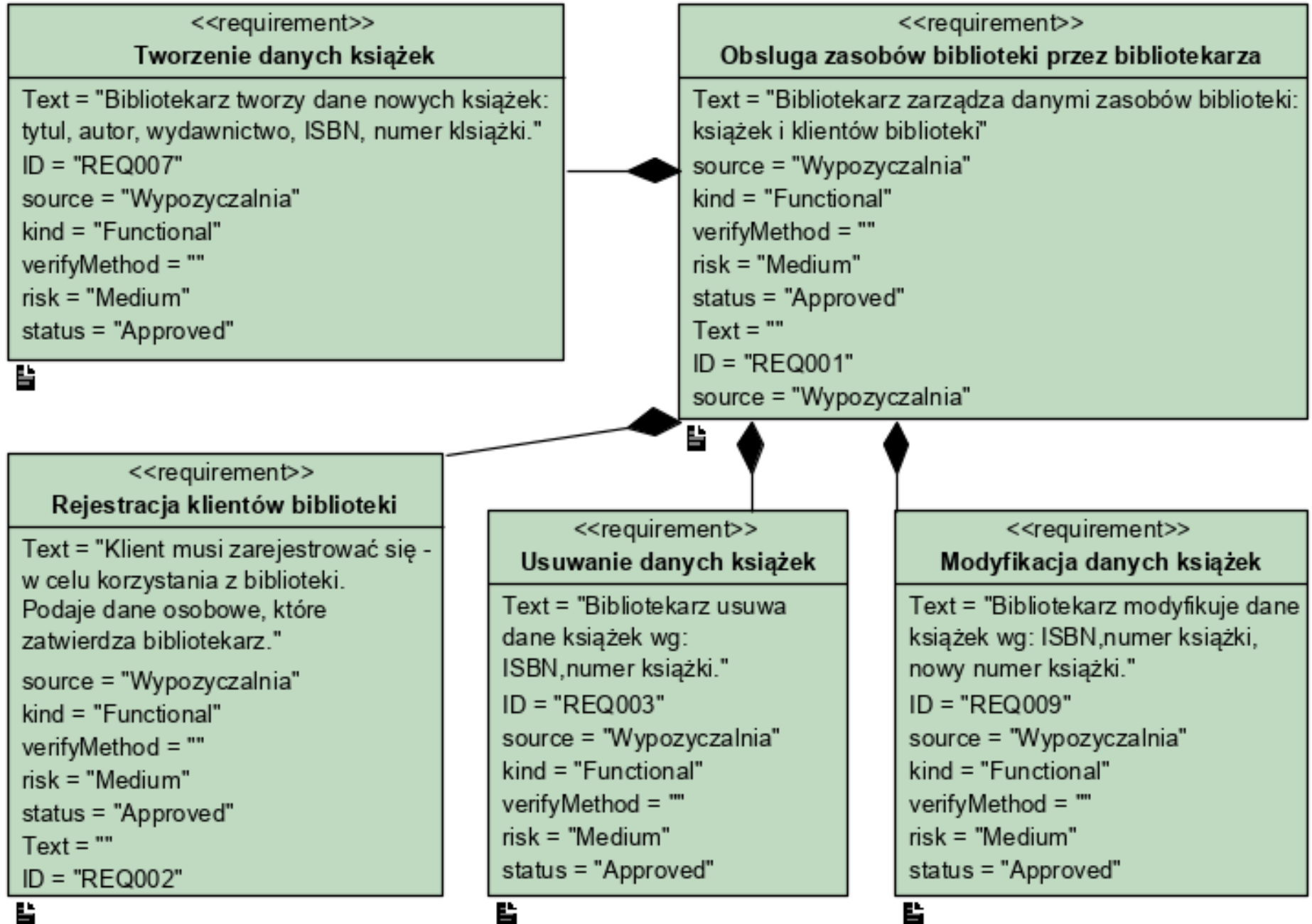


Diagram wymagań funkcjonalnych programu (cd)

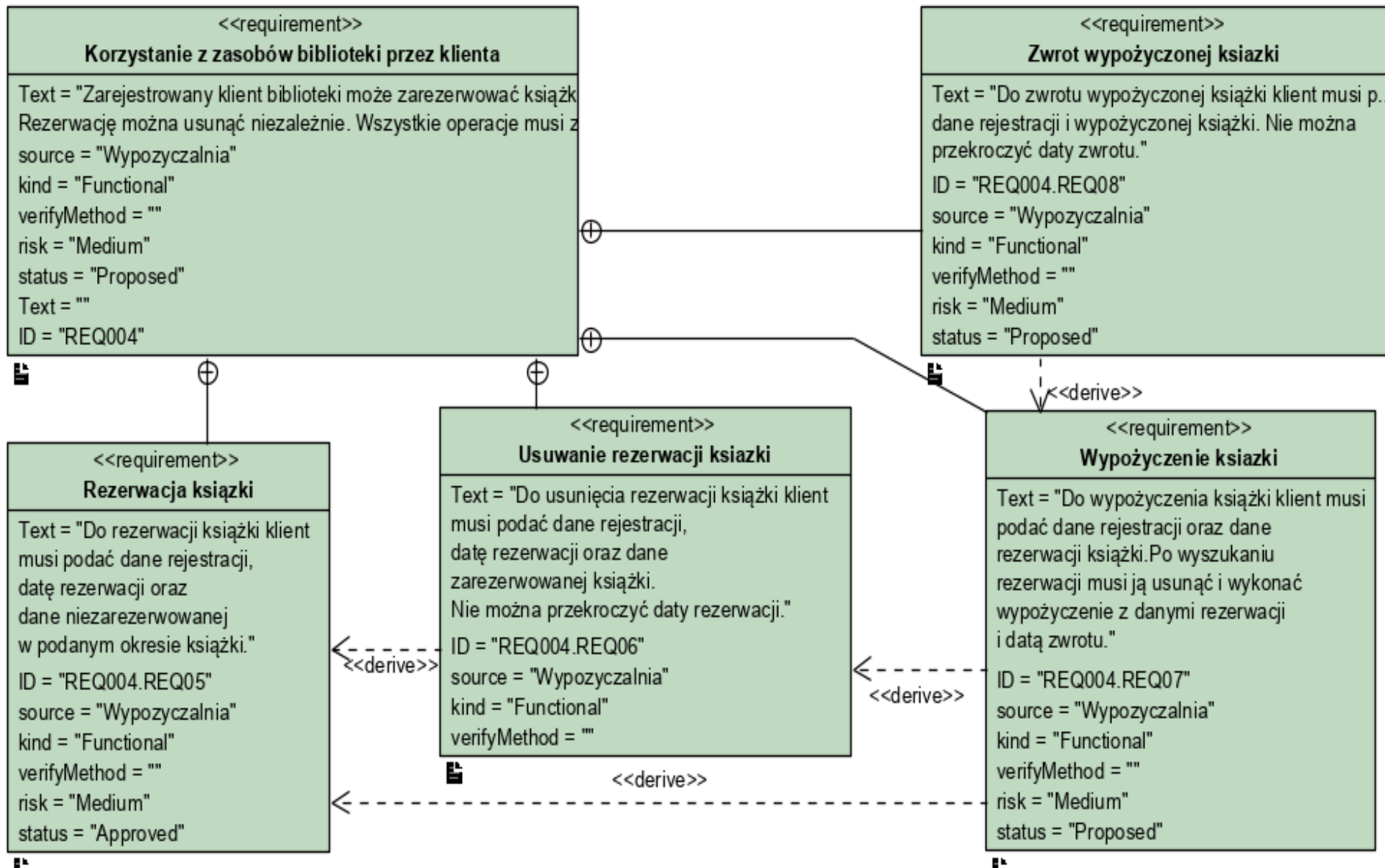
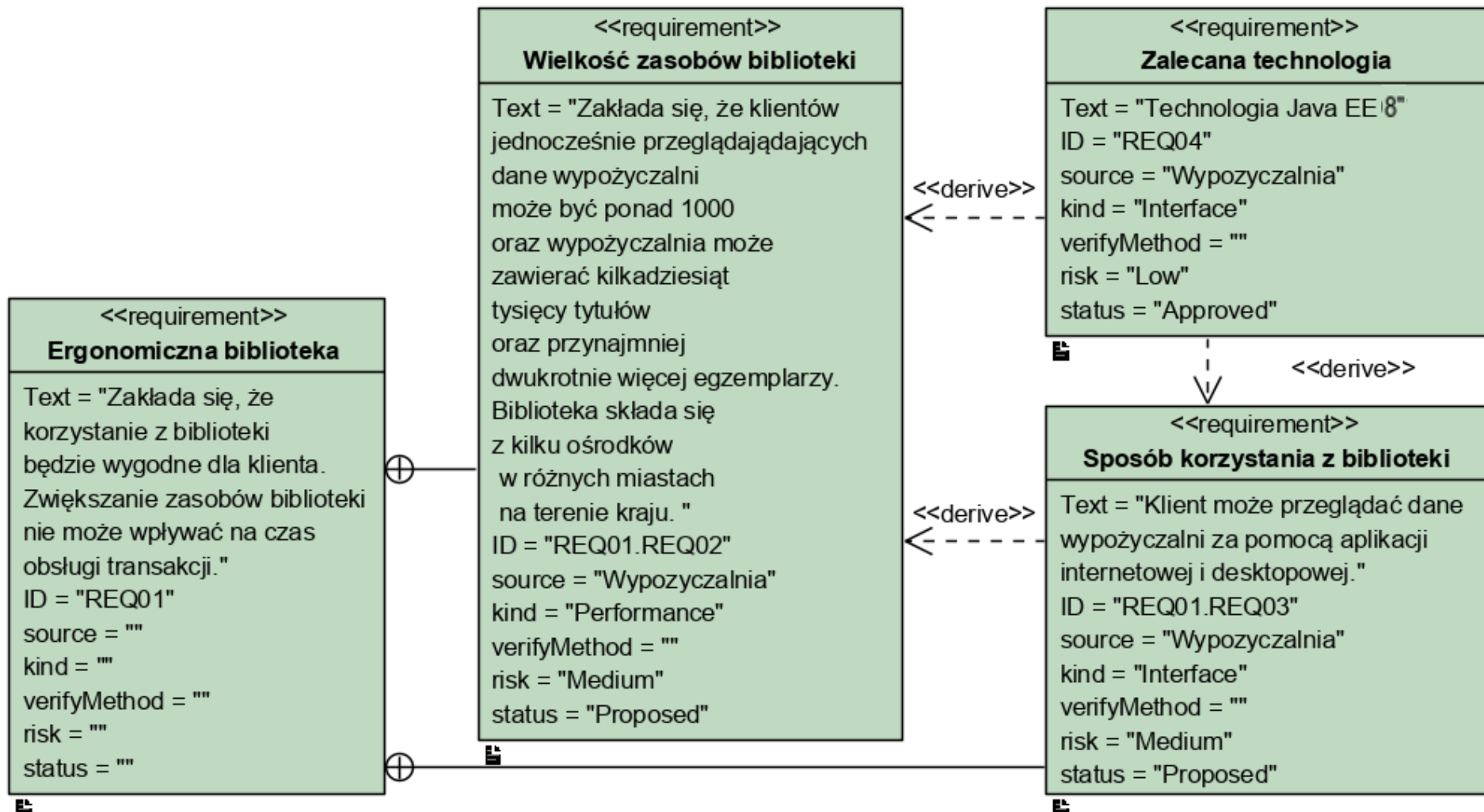




Diagram wymagań niefunkcjonalnych programu (cd)




Operacje_Na_Ksiazkach



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

Rank: Medium  

ID: UC12

Status: Identify  Next

Justification:

Primary Actors: 

Supporting Actors: 

Task Pool:

Description:



Scenariusz:

1. Należy podać atrybuty tytułu: ISBN jako obowiązkowa i ewentualnie aktora, jeśli książka jest nagrana oraz numwr książki
2. Należy wywołać **PU Wyszukaj Tytuł**. Należy sprawdzić, czy tytuł o podanym ISBN i ewentualnym aktorze już istnieje. Jeśli nie, należy zakończyć PU.
3. Należy utworzyć egzemplarz zawierający numer podany do wyszukiwania egzemplarza i należy przekazać go do **PU Wyszukaj Książkę**.

Abstract Leaf Root

Dash Project ITSM UeXceler Diagram View Team Tools Modeling Window Help

New Open Save Close Print Export Import Properties Community Circle

Diagram Navigator

- 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
 - sub_business_t
 - sub_business_t
 - sub_business_t

Operacje_Na_Ksiazkach Details

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

Level: User

Complexity: Medium

Use Case Status: Base

Implementation Status: Started

Preconditions:

- Tworzenie danych ksiazek
- Usuwanie danych ksiazek
- Rezerwacja ksiazki
- Modyfikacja danych ksiazek
- Dodaj tytul ksiazki
- Dodaj Ksiazke
- Usun Ksiazke
- Modyfikacja Ksiazki
- Rezerwacja

Post-conditions:

Author:

W polu **Precondition** podano jako warunki wstępne listę specyfikowanych wymagań oraz listę przypadków użycia, które za pomocą związku **Generalization** dziedziczą funkcjonalność przypadku użycia **Operacje_Na_Ksiazkach**.

Rezerwacja



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

Rank: ^ v

ID:

Status: Next

Justification:

Primary Actors: ...

Supporting Actors: ...

Task Pool:

Description:

1. Po wyszukaniu książki sprawdza się za pomocą [PU Wyszukaj Rezerwacje](#), czy dany egzemplarz książki nie jest zarezerwowany w podanym terminie. Jeśli tak, należy zakończyć PU
2. Należy wyszukać klienta wg numeru karty bibliotecznej za pomocą [PU Wyszukaj Klienta](#) Jeśli nie istnieje, należy zakończyć PU.
3. Należy wykonać rezerwację, umieszczając na niej dane wyszukiwanej książki, dane klienta oraz datę rezerwacji. Dane rezerwacji należy umieścić w zbiorze rezerwacji zarezerwowanej książki i w zbiorze rezerwacji znalezionego klienta.

Abstract Leaf Root

New Open Save Close Print Export Import Properties Community Circle

Diagram Navigator

- Projekt_lab2
 - UML Diagrams
 - Use Case Diagrams
 - Use Case Diagram
 - Class Diagram (
 - Libray1_classes
 - Library1_classe
 - Sequence Diagrams
 - sub_business_t
 - sub_business_t
 - sub_business_t
 - sub_business_t
 - sub_business_t
 - sub_business_t

Rezerwacja Details

Rezerwacja

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

Level: User

Complexity: Medium

Use Case Status: Initial

Implementation Status: Started

Preconditions: [Rezerwacja książki](#)

Post-conditions:

Author:

Assumptions:



Usuwanie_Rezerwacji



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

Rank: Medium

ID: UC14

Status: Identify Next

Justification:

Primary Actors: Klient

Supporting Actors:

Task Pool:

Description:

1. Należy podać numer karty klienta i numer rezerwacji
2. Należy wyszukać klienta za pomocą PU [Wyszukaj_Klienta](#). Jeśli nie został znaleziony, należy zakończyć PU z informacją o braku rezerwacji.
2. Należy wyszukać rezerwacje w zbiorze klienta za pomocą PU [Wyszukaj_Rezerwacje](#). Jeśli nie ma, należy zakończyć PU z informacją o braku rezerwacji.
3. Należy usunąć rezerwację ze zbioru klienta i książki należącej do rezerwacji i przekazać jej dane

Abstract Leaf Root

Projekt_lab2 * - Visual Paradigm Community Edition[Zofia Kruczkiewicz] (not for commercial use)

Dash Project ITSM UeXceler Diagram View Team Tools Modeling Window Help

New Open Save Close Print Export Import Properties Community Circle

Diagram Navigator Projekt_lab2 UML Diagrams Use Case Diagram Class Diagram (Library1_classes Library1_classe Sequence Diagram sub_business_t sub_business_t sub_business_t sub_business_t sub_business_t sub_business_t)

Usuwanie_Rezerwacji Properties

Usuwanie_Rezerwacji

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

Level: User
 Complexity: Medium
 Use Case Status: Initial
 Implementation Status: Started

Preconditions: [Usuwanie rezerwacji książki](#) [Wypożyczenie](#)

Post-conditions:

Author:



Assumptions:

W polu **Precondition** podano jako warunki wstępne specyfikowane wymaganie oraz przypadek użycia **Wypożyczenie**, który uruchamia zawierany w nim przypadek użycia **Usuwanie_Rezerwacji** powiązany za pomocą związku <<include>>. Oznacza to, że każdy zawierany przypadek użycia podaje listę przypadków użycia, które go zawierają.

Wypożyczenie



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

Rank:  

ID:

Status:

Justification:

Primary Actors: 

Supporting Actors: 

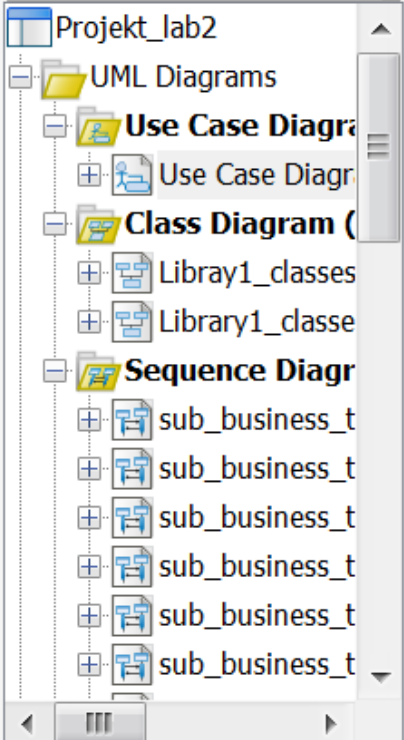
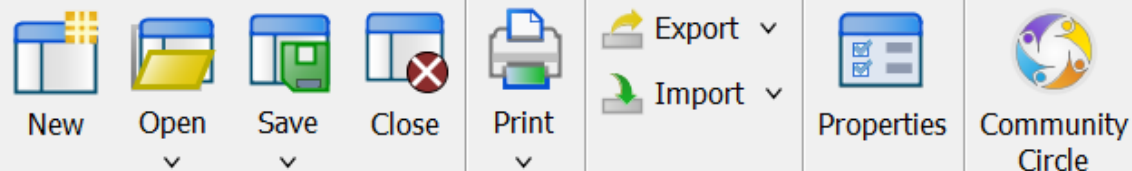
Task Pool:

Description:



- Należy przekazać dane rezerwacji i numer karty bibliotecznej klienta.
- Należy usunąć rezerwację za pomocą PU [Usuwanie Rezerwacji](#). Jeśli zwrócone dane zawierają dane rezerwacji, należy pobrać dane klienta, dane książki, okres wypożyczenia i wykonać wypożyczenie w podanym okresie rezerwacji. w przeciwnym wypadku należy zakończyć PU.
- Dane wypożyczenia umieścić w danych książki i zbiorze wypożyczeń klienta.

Abstract Leaf Root



Wypożyczenie Details



Wypożyczenie



Level:	User
Complexity:	Medium
Use Case Status:	Initial
Implementation Status:	Started
+ Preconditions:	Wypożyczenie książki
+ Post-conditions:	
Author:	
+ Assumptions:	



**Przykład 4 identyfikacji i specyfikacji
przypadków użycia na diagramie
przypadków użycia
Wypożyczalnia książek**

I. 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 opcjonalnie 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, jeśli jest**. 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.

II. Wymagania funkcjonalne i нефункционаłne aplikacji

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 wypożyczanej książki, należy ją usunąć. Rezerwację można usunąć niezależnie.
6. Bibliotekarz może łatwo utworzyć, zmienić i usunąć informację o wypożyczeniach i rezerwacjach

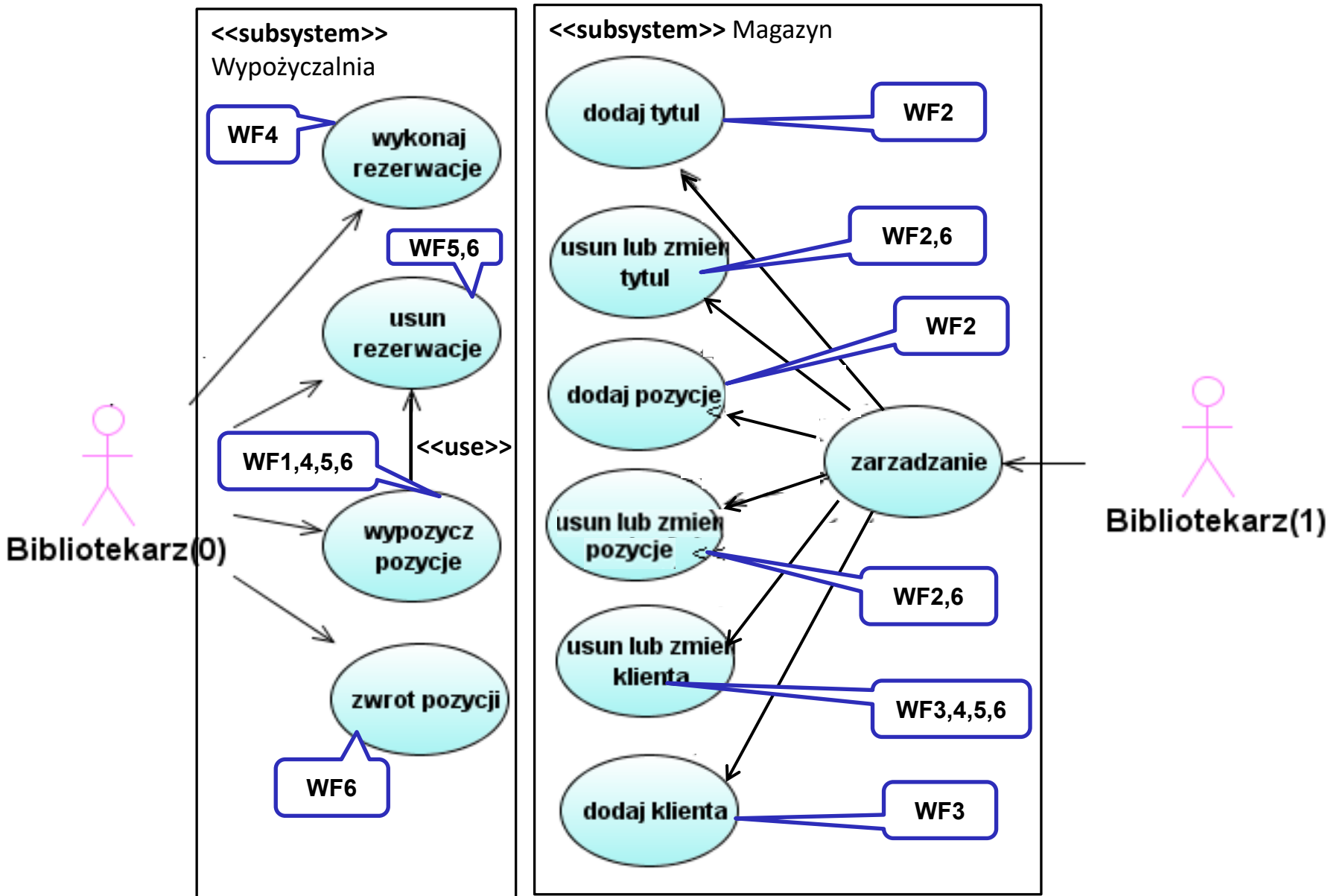
Wymagania нефункционаłne

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

III. Model przypadków użycia aplikacji oparty na diagramie przypadków użycia

AKTOR	OPIS	PRZYPADKI UŻYCIA
Bibliotekarz	<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>	<ul style="list-style-type: none">• zarządzanie• wykonaj rezerwacje• usun rezerwacje• wypożycz pozycje• zwrot pozycji

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



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 – rezultat zastosowania relacji <<use>>):

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 zawierającą 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

Tworzenie diagramów przypadków użycia

1. Elementy diagramów przypadków użycia (use-cases)
2. Wytyczne tworzenia diagramów przypadków użycia (use-cases)

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

3. Model przypadków użycia – identyfikacja i specyfikacja wymagań
4. Przykłady diagramów przypadków użycia (use-cases)
5. Podsumowanie

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 tworzonego systemu, dołączając architekturę ważnej funkcjonalności
- Definiuje przypadki użycia analizowane w modelu analizy

Tworzenie diagramów przypadków użycia

1. Elementy diagramów przypadków użycia (use-cases)
2. Wytyczne tworzenia diagramów przypadków użycia (use-cases)

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

3. Model przypadków użycia – identyfikacja i specyfikacja wymagań
4. Przykłady diagramów przypadków użycia (use-cases)
5. Dodatek

5. Dodatek – metodyka DevOps rozwijana w ramach Inżynierii Oprogramowania

2. Definicje inżynierii oprogramowania

- slajd 17 prezentowany podczas 1-go wykładu

- [Fritz Bauer]

Opracowanie sprawdzonych zasad inżynierii oraz ich zastosowanie w celu wytworzenia niedrogiego i niezawodnego oprogramowania, działającego efektywnie na rzeczywistych maszynach.

- [IEEE 1993]

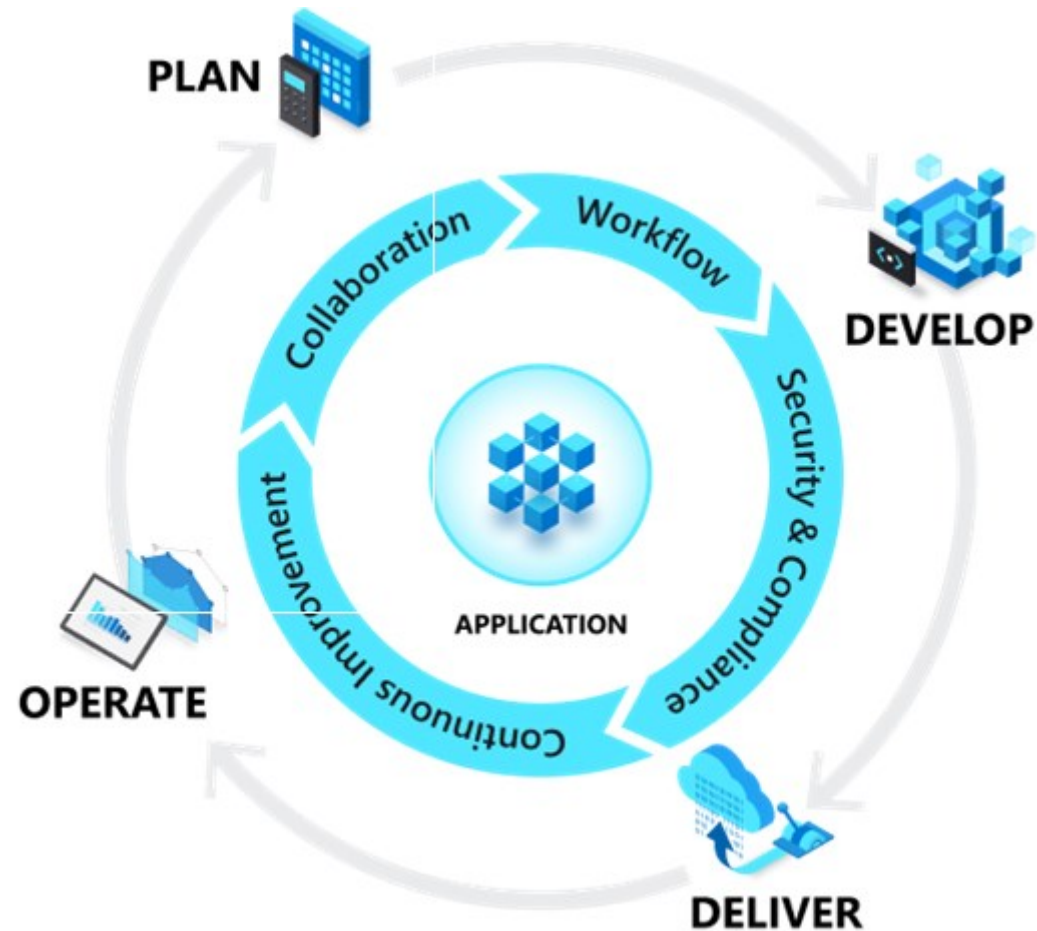
1. Zastosowanie systematycznego, zdyscyplinowanego, poddającego się ocenie ilościowej, podejścia do wytwarzania, stosowania i pielęgnacji oprogramowania, czyli wykorzystanie technik tradycyjnej inżynierii w informatyce.
2. Dziedzina wiedzy zajmująca się badaniem metod jak w p.1

Przykład jednej z metodyk - DevOps (Development Operations) jako cykl życia tworzenia aplikacji

<https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-is-devops/>

Inżynieria oprogramowania jako dziedzina wiedzy **Informatyki** oznacza **naukę** obejmującą między innymi definiowanie i badanie **metodyk** tworzenia oprogramowania.

Metodyka DevOps nawiązuje do praktyk przypisanych do 5 – poziomów dojrzałości wytwarzania oprogramowania zdefiniowanych w ramach **CMMI** (slajd 72 – wykład 1).



Dziękuję za uwagę