

# Projektowanie systemów informatycznych

**Roman Simiński**

[roman.siminski@us.edu.pl](mailto:roman.siminski@us.edu.pl)

[siminskionline.pl](http://siminskionline.pl)

## Diagramy przypadków użycia

# Diagramy przypadków użycia jako narzędzie modelowania wymagań

Nazwa diagramu UML:

Diagramy przypadków użycia — *use case diagrams*.

Cel stosowania:

- ▶ *Określenie wymagań stawianych systemowi* — czyli co system ma robić z punktu widzenia jego otoczenia.
- ▶ *Określenie granic systemu* — jego otoczenia oraz elementów, które wchodzą z nim w interakcję.

Diagramy przypadków użycia opisują, *co robi system z punktu widzenia zewnętrznego obserwatora*.

Eksponują to, *co robi system, a nie jak to robi*.

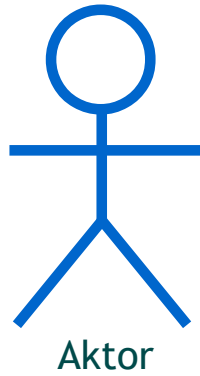
# Diagramy przypadków użycia jako narzędzie modelowania wymagań

## Główne zastosowania:

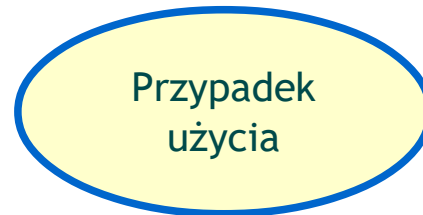
- ▶ *Określanie i doprecyzowanie funkcji systemu* — opisywane przypadki użycia zwykle generują nowe wymagania, a projekt przybiera coraz wyraźniejszy kształt.
- ▶ *Komunikacja z klientami* — prostota notacji i intuicyjność sprawiają, że diagramy przypadków użycia są dobrym sposobem porozumiewania się projektantów z przyszłymi użytkownikami systemu.
- ▶ *Generowanie przypadków testowych* — opis danego przypadku użycia może zasugerować sposoby testowania i konkretne dane testowe.
- ▶ *Zrozumienie różnych scenariuszy wykorzystania projektowanego systemu.*
- ▶ *Porozumiewanie się projektantów systemu i jego przyszłych użytkowników.*

# Komponenty diagramów przypadków użycia

▶ Aktor



▶ Przypadek użycia

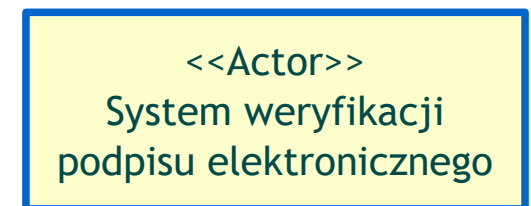
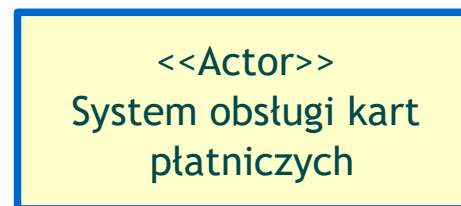


▶ Związek

---

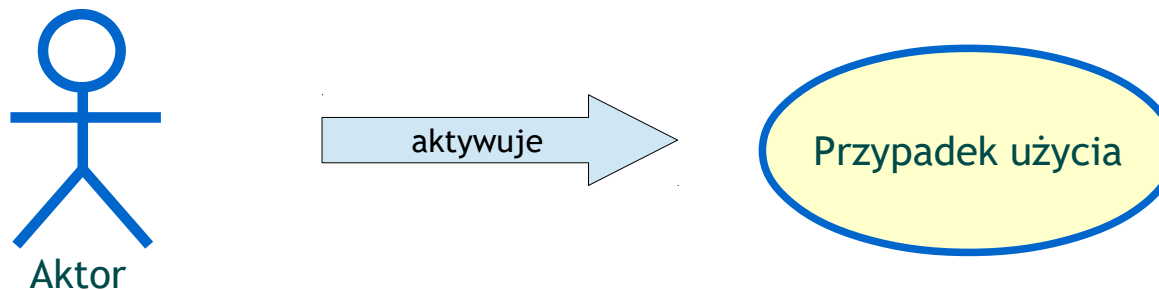
# Komponenty diagramów przypadków użycia – aktorzy

- ▶ *Aktorami mogą być ludzie, urządzenia, inne systemy informatyczne.*
- ▶ Pojęcie *aktora* odpowiada *roli* jaką on odgrywa w stosunku do systemu.
- ▶ Pojęcie *aktora* nie odpowiada zawsze np. *konkretnej osobie fizycznej*, bo ta może wcielać się w różne *role* (ta sama osoba fizyczna może się logować do systemu raz jako administrator, a innym razem jako zwykły użytkownik).
- ▶ Jeden aktor może reprezentować całą grupę fizycznych użytkowników systemu (aktor *Klient* reprezentuje wszystkich potencjalnych klientów systemu).
- ▶ Aktorzy są zwykle aktywni — inicjują przypadku użycia, choć mogą być również pasywni (np. aktor tylko zatwierdzający przelew bankowy).



# Komponenty diagramów przypadków użycia – przypadki użycia

- ▶ *Przypadek użycia* reprezentuje *sekwencję operacji* wykonywanych przez system, inicjowanych przez aktora.
- ▶ *Przypadek użycia* modeluje *oczekiwane zachowanie systemu* wobec danego aktora, nie precyzując *sposobu realizacji* tego zachowania.
- ▶ Uruchomienie danego *przypadku użycia* ma dostarczyć aktorowi *wymiernych wyników*.
- ▶ Przypadek użycia zwykle opisuje pewien większy, dłuższy, bardziej złożony proces a *nie elementarną akcję*.



# Przypadki użycia - przydatne informacje

- ▶ Nazwa przypadku użycia *zazwyczaj* zawiera *rzeczownik* określający *cel* uaktywnienia przypadku, poprzedzony *czasownikiem* opisującym rodzaj *aktywności*.

Porada lekarska

Sprzedaż towaru

Wystawienie faktury

Rezerwacja książki

Wypożyczenie  
książki

Wyszukanie  
książki

Wykonanie  
przelewu

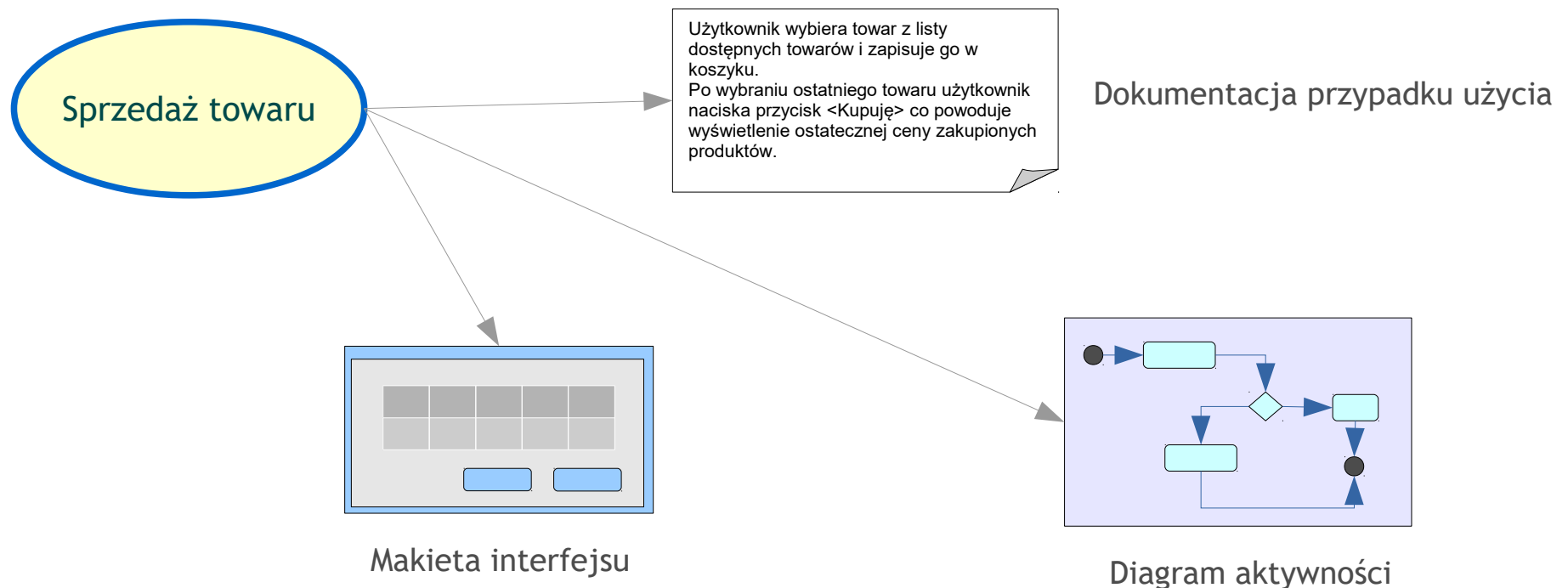
Sprawdzenie  
stanu konta

Przegląd  
transakcji

# Przypadek użycia to nie tylko elipsa na diagramie

Dla każdego przypadku użycia:

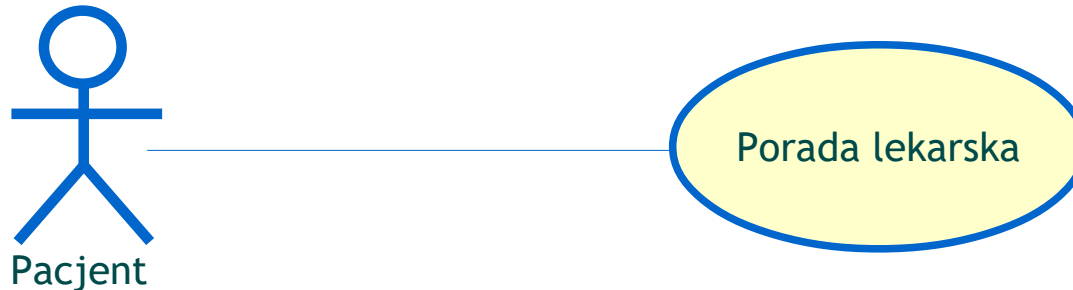
- ▶ Definiuje się *dokumentację przypadku użycia* – *use case document* (szczegóły później).
- ▶ Można zaproponować *rysunek* lub *makieta interfejsu użytkownika*, wtedy łatwiej tworzy się opis danego przypadku.
- ▶ Można zaproponować *diagram aktywności* – *activity diagram*.





## Komponenty diagramów przypadków użycia – związki

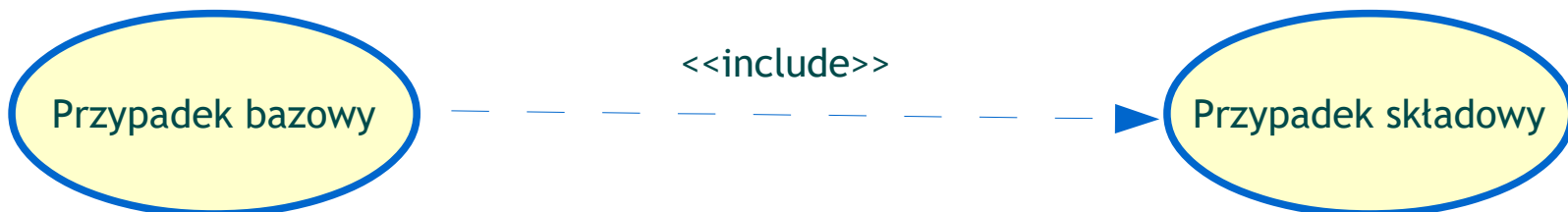
- ▶ Związek pomiędzy *aktorem a przypadkiem użycia* opisuje udział aktora w danym przypadku użycia.
- ▶ Pomędzy przypadkami użycia mogą występować związki – zostaną przedstawione za chwilę.



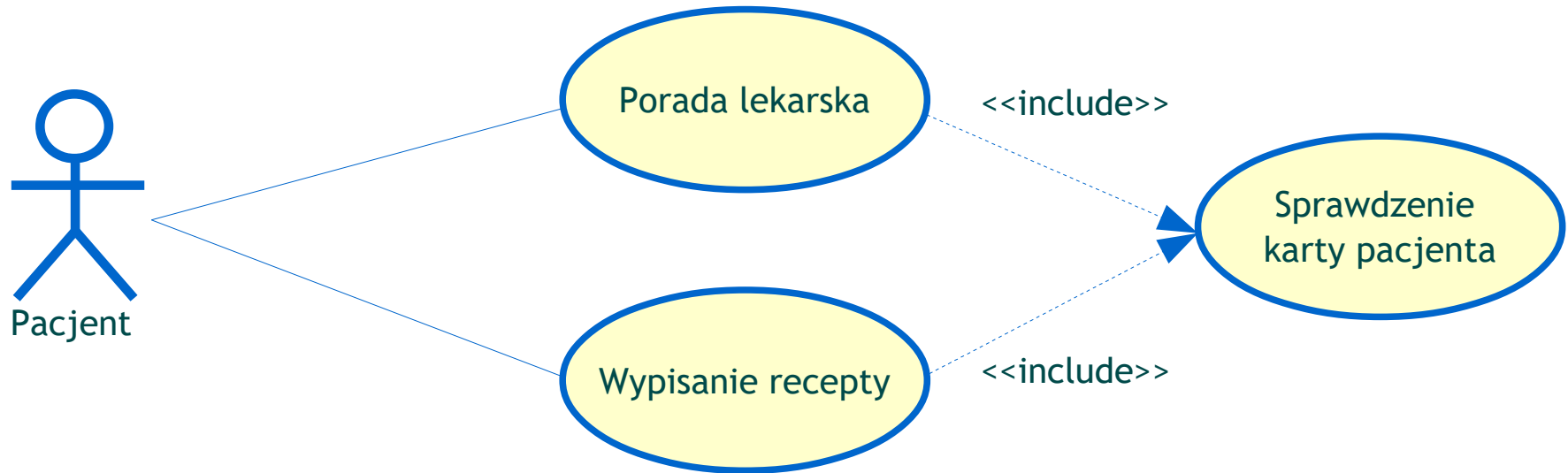
Aktor *Pacjent* inicjuje przypadek użycia *Porada lekarska*. Jest to pewien *nieelementarny* proces, wynikiem którego jest umówienie wizyty lekarskiej oraz przygotowanie kartoteki pacjenta. Ten przypadek użycia dostarcza aktorowi *konkretnych* efektów działania systemu – porada lekarska może się odbyć.

# Związki pomiędzy przypadkami użycia – zawieranie

- ▶ Czasem wiele przypadków użycia posiada pewną wspólną sekwencję operacji (wspólne zachowanie) – można ją wyróżnić w postaci odrębnego przypadku użycia.
- ▶ Wspólny, odrębny przypadek użycia włącza się do *przypadku bazowego* wykorzystując *związek zawierania*, opisany stereotypem `<<include>>`.
- ▶ Przypadek bazowy występuje jako pierwszy i zawsze używa przypadku składowego.
- ▶ Składowy przypadek użycia reprezentuje podzadanie, zadania jakim jest przypadek bazowy.



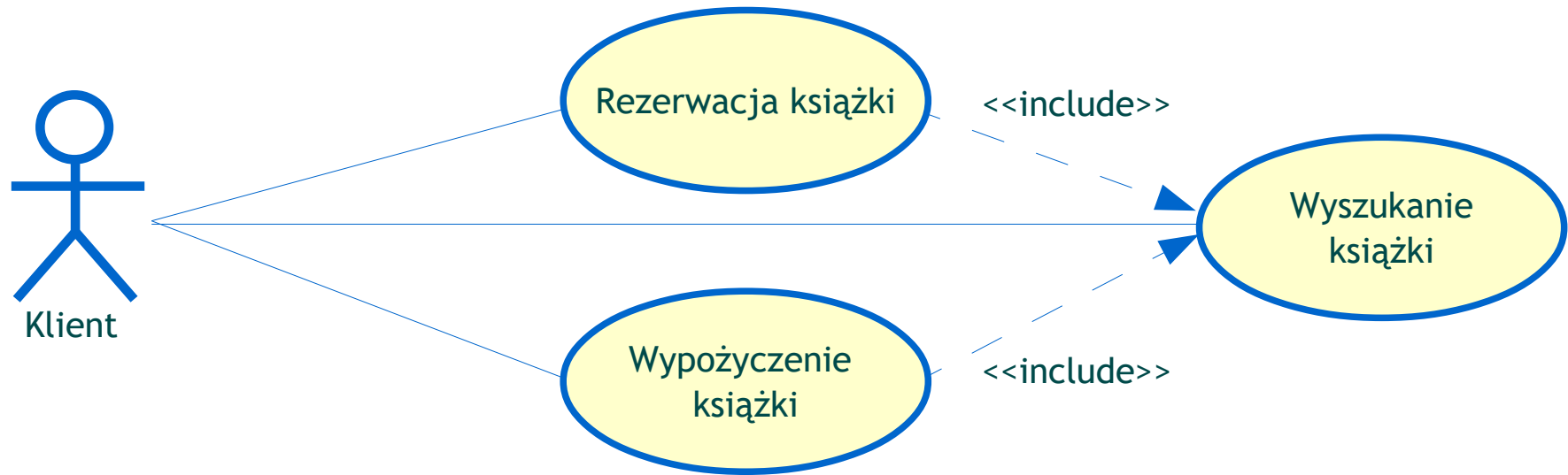
## Zawieranie, przykład *Przychodnia*



Zarejestrowany pacjent pewnej prywatnej przychodni lekarskiej korzysta z porad lekarskich, wymagających umówienia terminu i przygotowania dokumentów wymagających sprawdzenia karty pacjenta.

Czasem jednak pacjent może prosić jedynie o wypisanie recepty na leki do stałego przyjmowania, to nie wymaga umówienia wizyty lekarskiej, jednak konieczne jest przygotowanie dokumentów i sprawdzenie karty pacjenta.

## Zawieranie, przykład *Biblioteka*

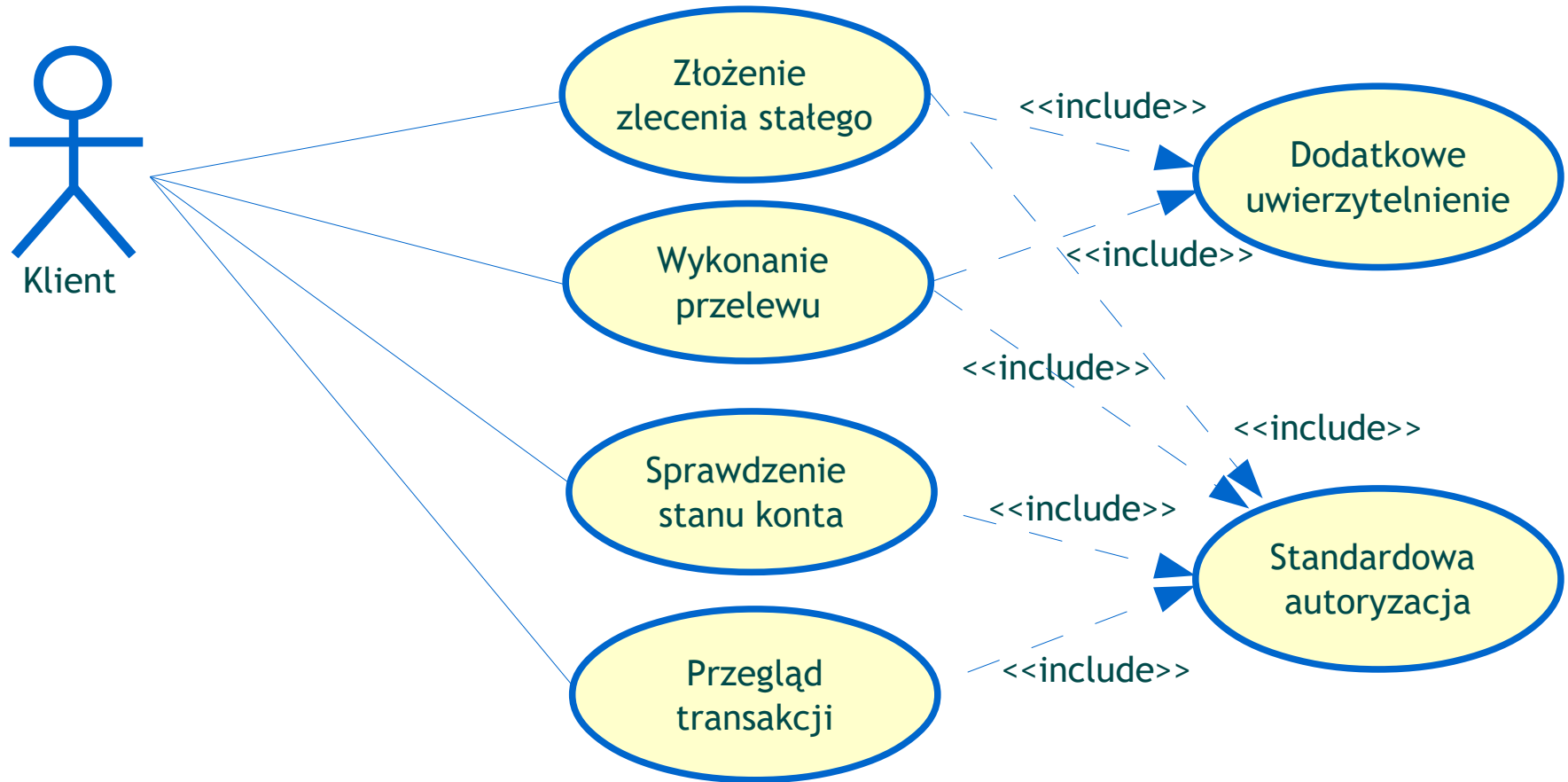


Klient pewnej biblioteki może przeszukiwać jej zasoby katalogowe w poszukiwaniu konkretnej książki.

Klient pewnej biblioteki może rezerwować konkretną książkę przed jej fizycznym wypożyczeniem, co wymaga zwykle jej wyszukania.

Klient pewnej biblioteki może wypożyczyć konkretną książkę, co wymaga zwykle jej wyszukania.

# Zawieranie, przykład *Konto*

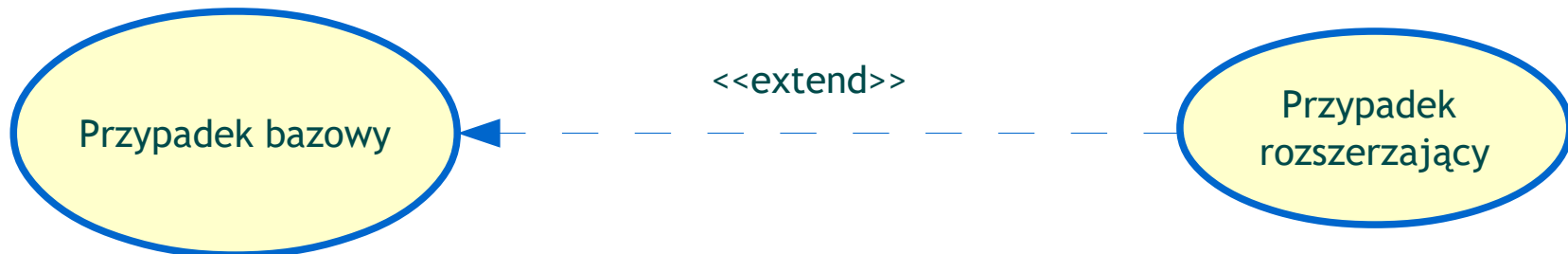


Zalogowany klient pewnego systemu bankowości elektronicznej, może wykonywać przelewy i składać zlecenia stałe. Wymaga to jednak zawsze dodatkowego uwierzytelnienia klienta. Przegląd transakcji i sprawdzenie stanu konta wymaga zawsze standardowej autoryzacji klienta.

# Związki pomiędzy przypadkami użycia – rozszerzanie

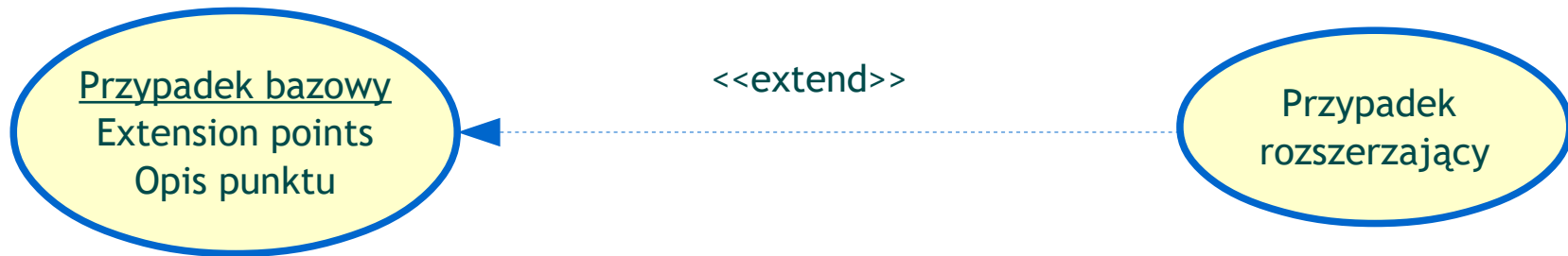
- ▶ Czasem pewien przypadek bazowy opcjonalnie realizuje jakaś czynność opcjonalną. Przypadek bazowy może działać samodzielnie i nie wykonywać funkcji opcjonalnej.
- ▶ Jednak po dojściu do pewnego punktu rozszerzającego, może zostać uruchomiony przypadek rozszerzający. Po zakończeniu przypadku rozszerzającego, wznowiane jest wykonanie przypadku bazowego.
- ▶ Rozszerzający przypadek użycia opisany jest stereotypem `<<extend>>`

Notacja wykorzystująca komentarz z warunkiem rozszerzenia:

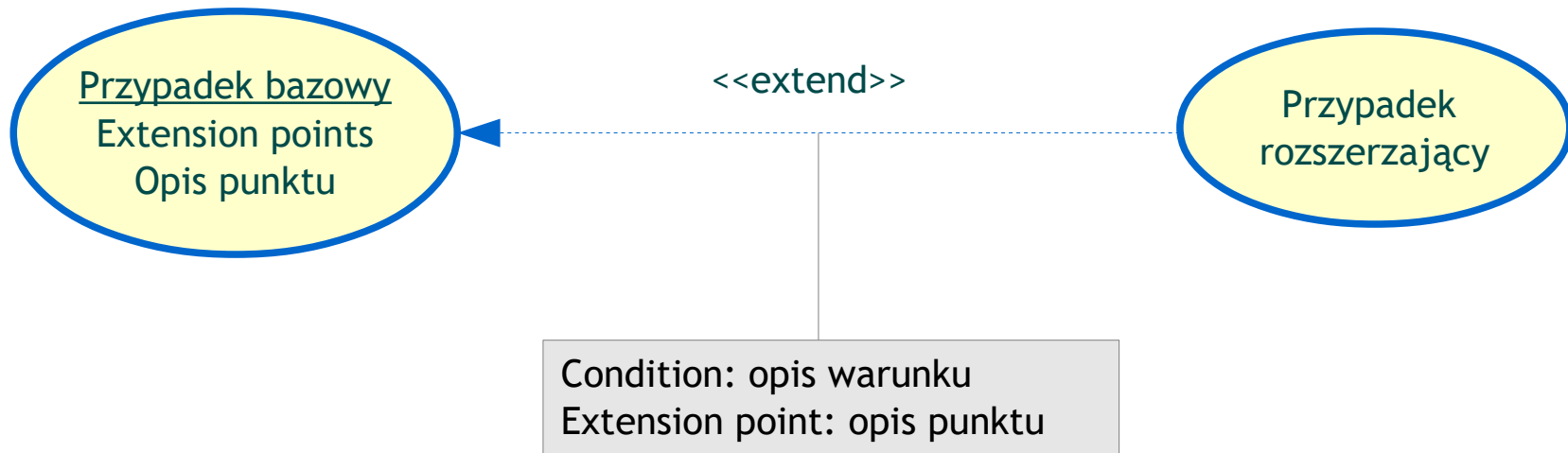


# Związki pomiędzy przypadkami użycia – rozszerzanie

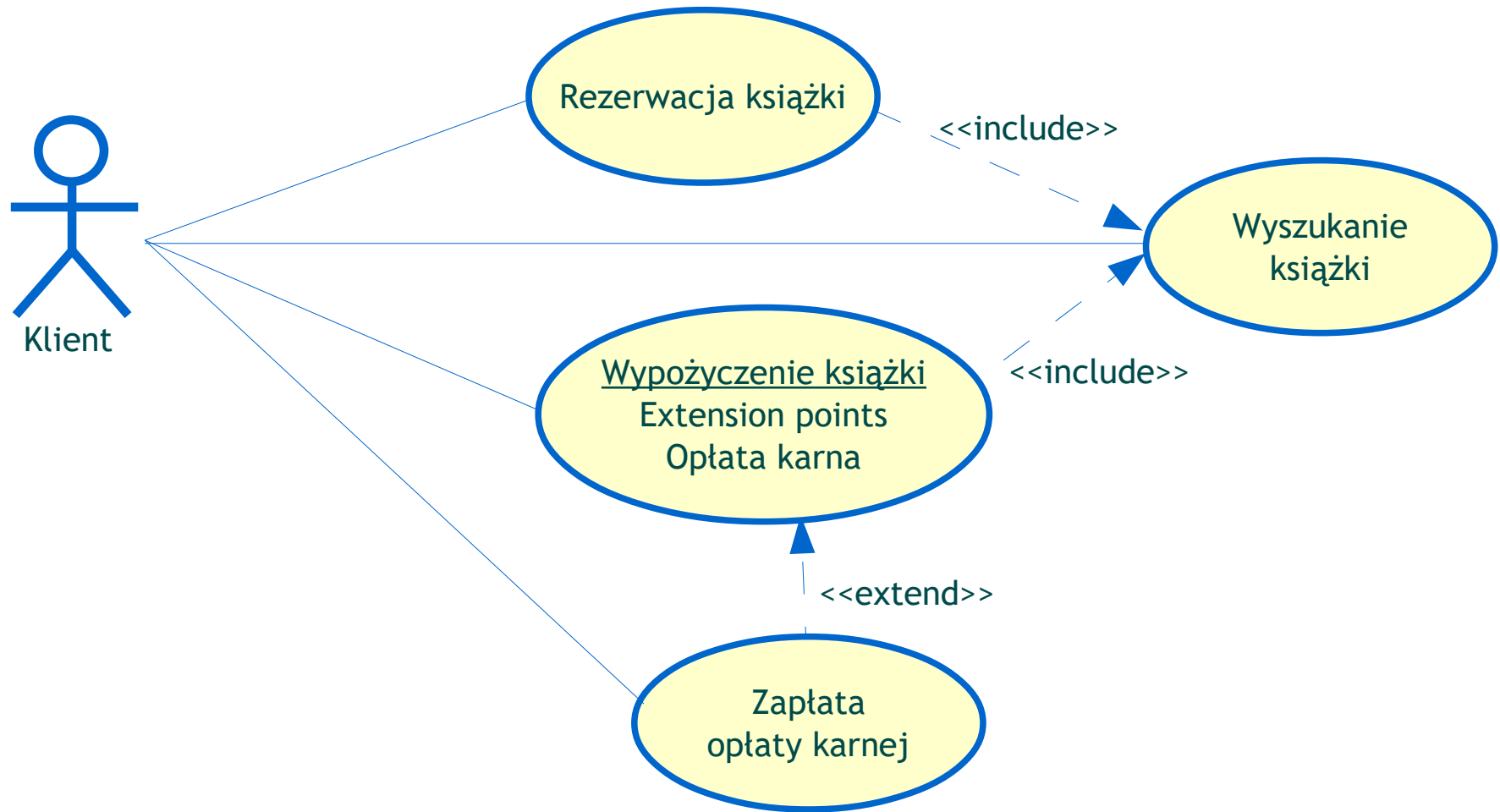
Typowa notacja dla rozszerzania:



Notacja wykorzystująca komentarz z warunkiem rozszerzenia:



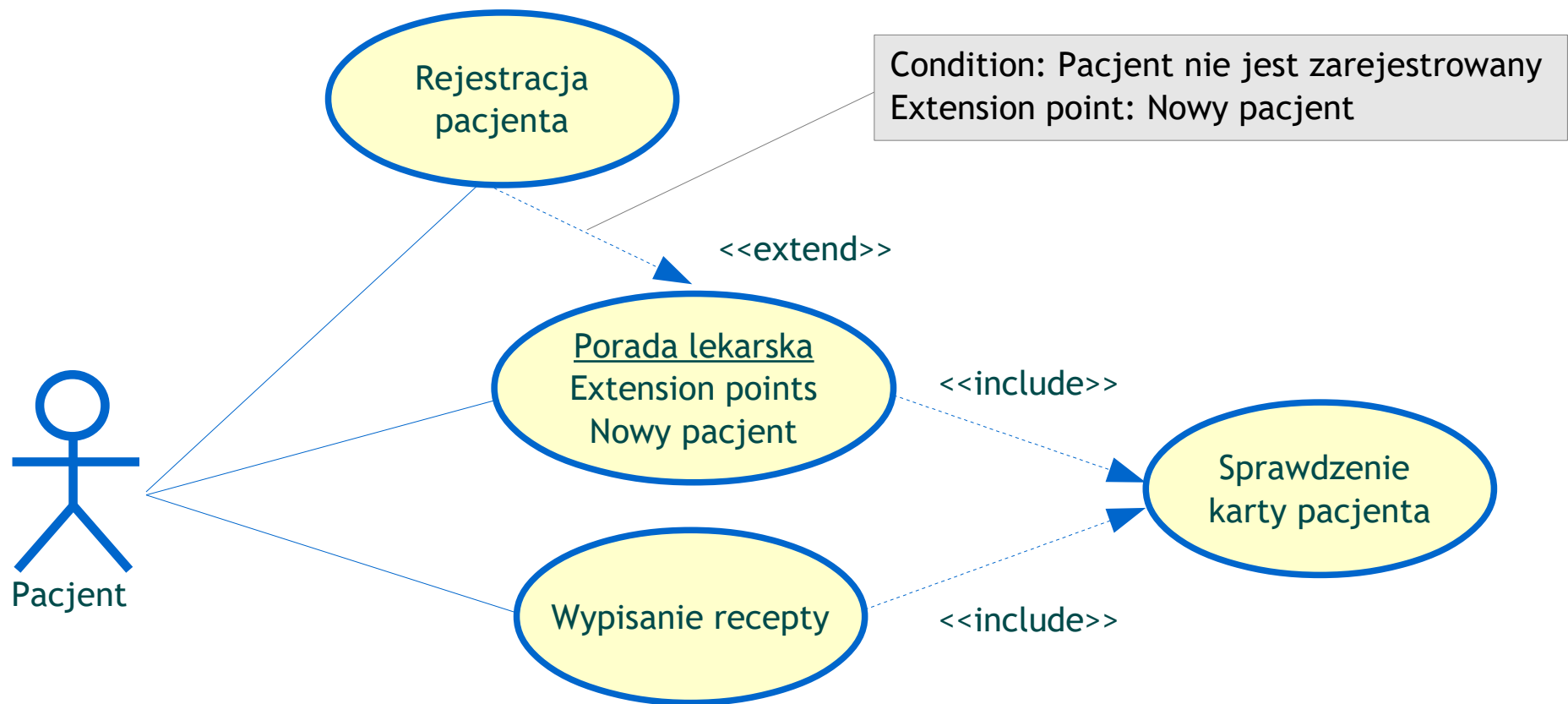
# Rozszerzanie, przykład *Biblioteka*



Klient pewnej biblioteki może wypożyczyć konkretną książkę. Jednak na tym etapie sprawdza się, czy nie zalega z opłatą karną za nieterminowe oddawanie książek, jeżeli tak, użytkownik musi opłacić wszystkie zaległości.



# Rozszerzanie, przykład *Przychodnia*

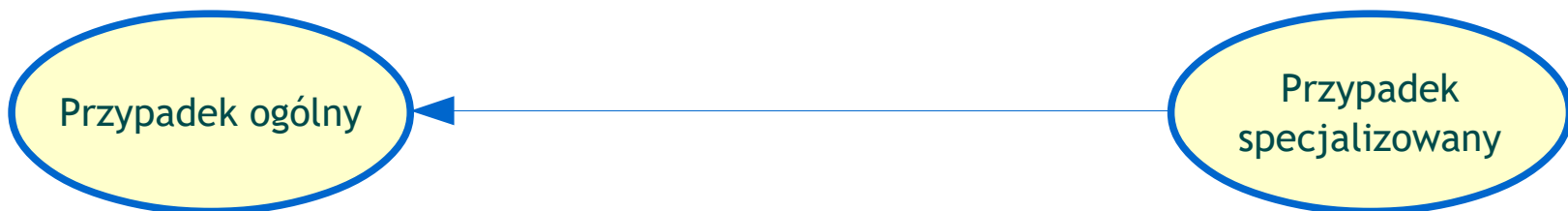


Do pewnej prywatnej poradni lekarskiej może się zgłosić pacjent, który nigdy wcześniej nie korzystał z jej usług. Skorzystanie z porady lekarskiej musi być wtedy poprzedzone rejestracją pacjenta w systemie przychodni.

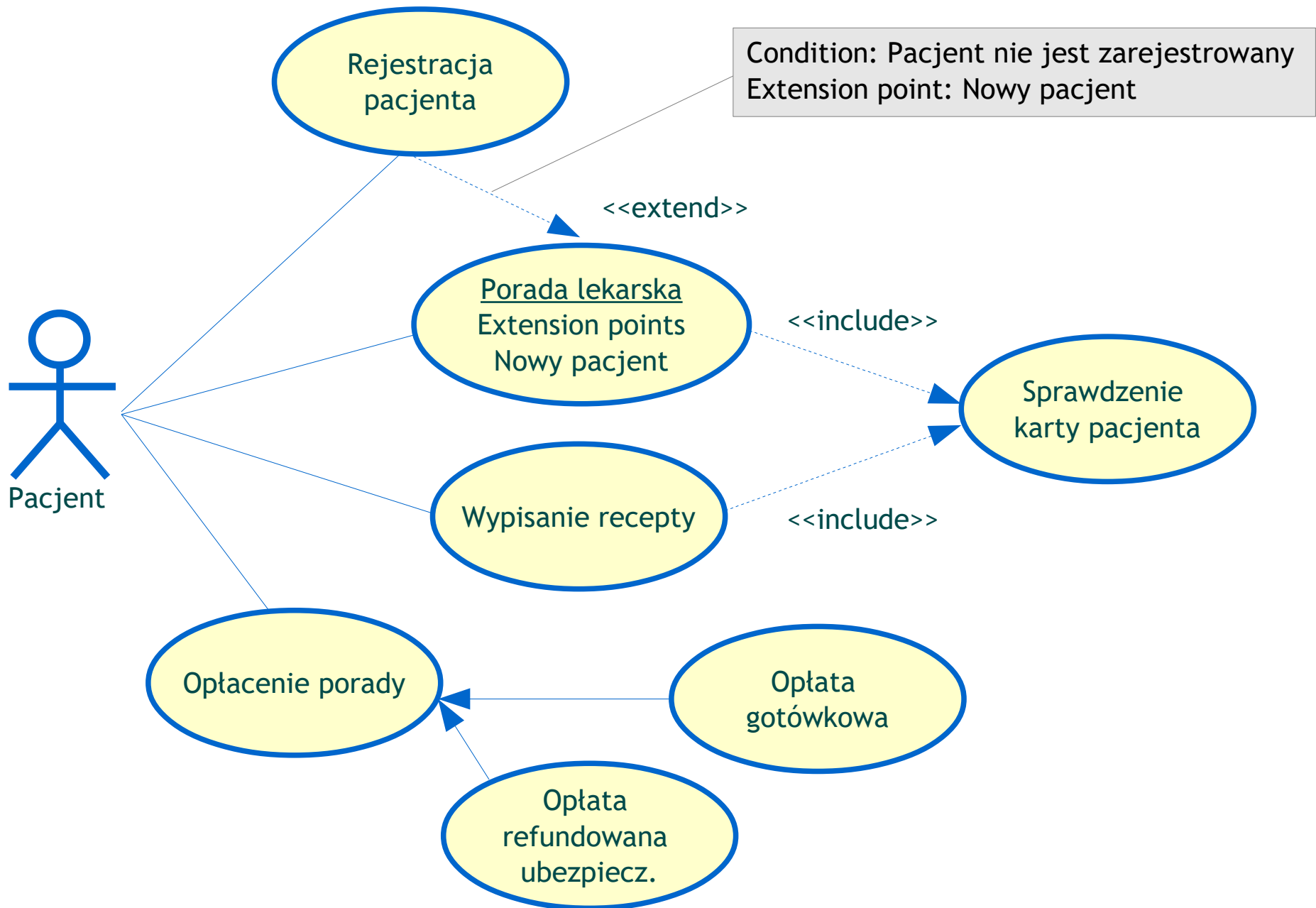
Pacjent może się zarejestrować w systemie przychodni *bez korzystania z porady lekarskiej*.

# Związki pomiędzy przypadkami użycia – uogólnienie

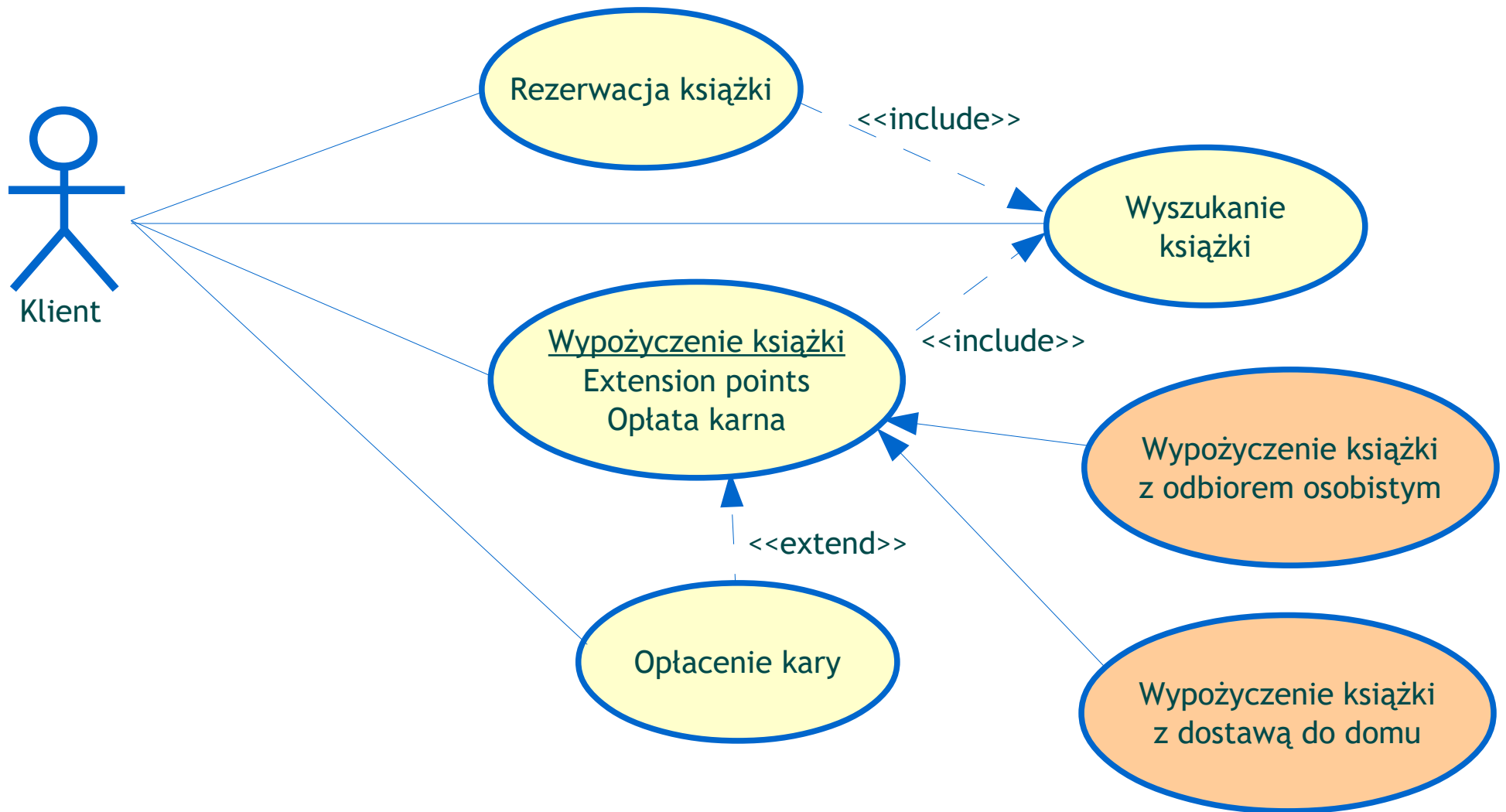
- ▶ Czasem pewien można zidentyfikować przypadki określające ten sam rodzaj przetwarzania realizowanego przez system. Przypadki są podobne, lecz nie jednakowe.
- ▶ W stosunku do przypadków można zastosować związek *generalizacja-specjalizacja*, i zastosować podejście analogiczne do klas.
- ▶ Przypadek będący specjalizacją pewnego przypadku uogólnionego, dziedziczy po nim całe zachowanie.
- ▶ Potomek może dodać nowe elementy do odziedziczonego zachowania lub wręcz całkowicie zmienić odziedziczone zachowanie.



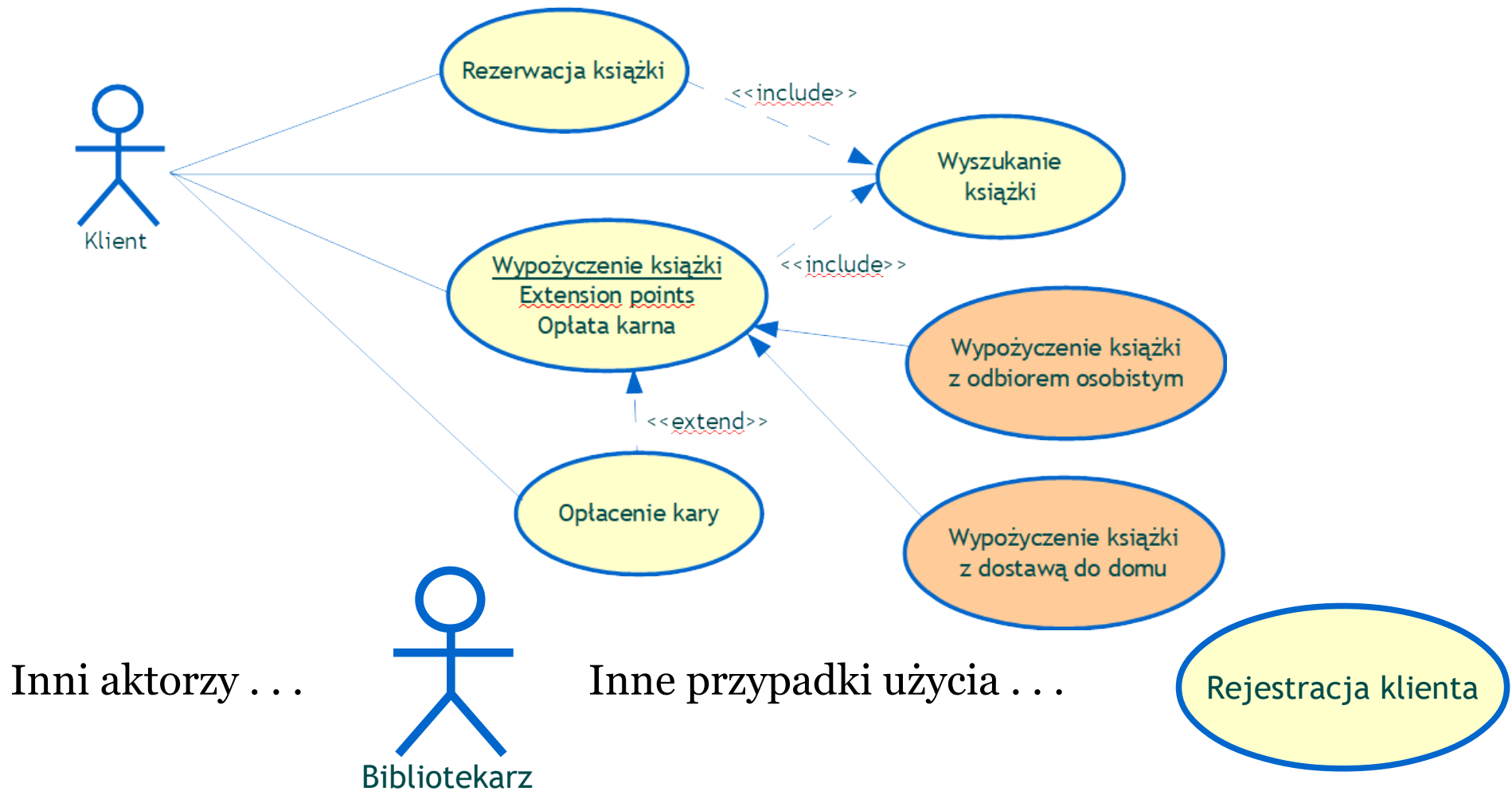
# Uogólnienie, przykład *Przychodnia*



# Uogólnienie, przykład *Biblioteka*



# Zadanie na ćwiczenia: opracuje SRS i dopracuj diagram dla *Biblioteka*



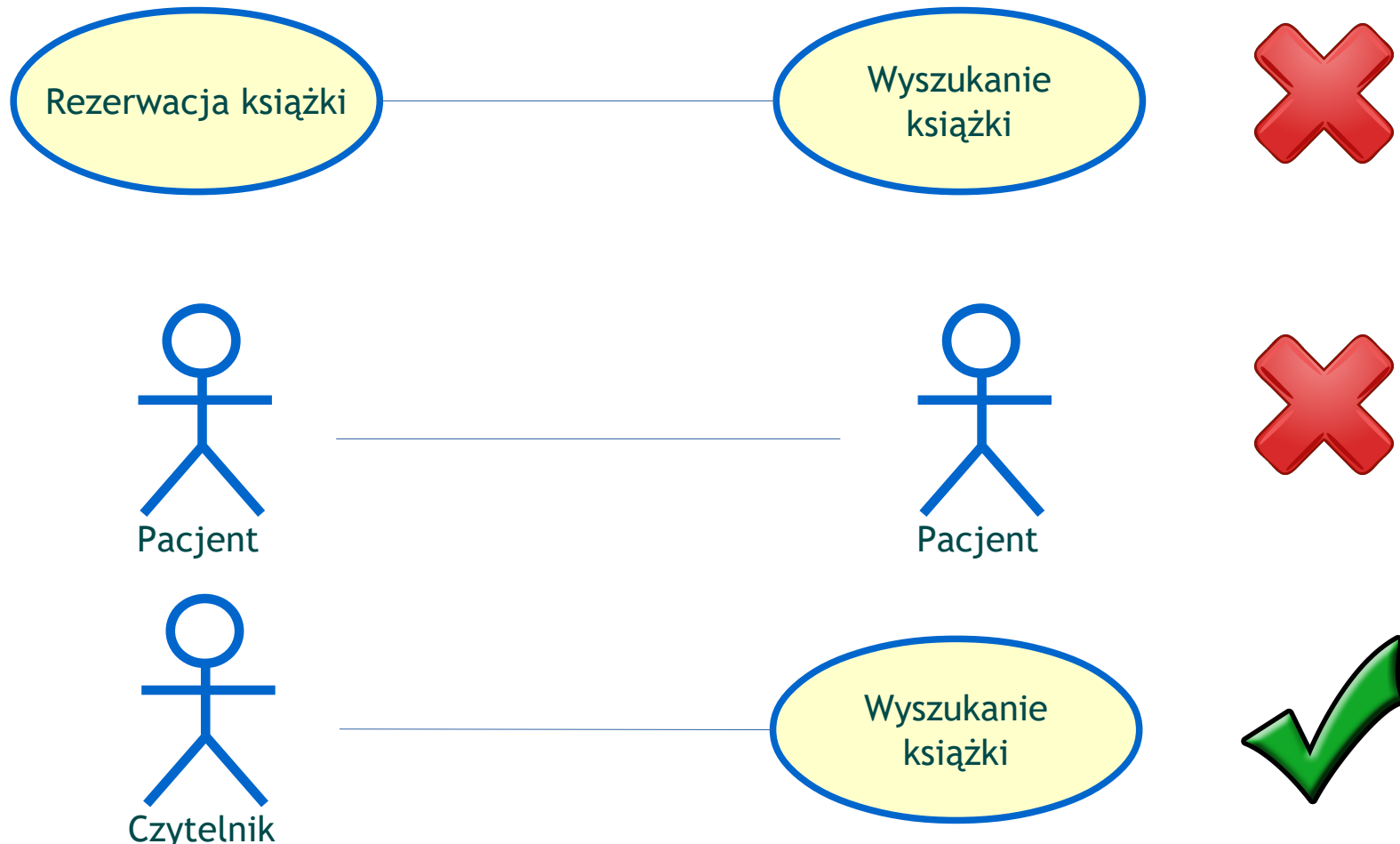
Do tworzenia diagramów Use Case można wykorzystać darmowy system **ArgoUML** – dostępne via Internet

# Suplement

**Wyjaśnienie wybranych wątpliwości  
dotyczących diagramów Use Case**

# Uwagi dodatkowe, związki tylko aktor - przypadek użycia

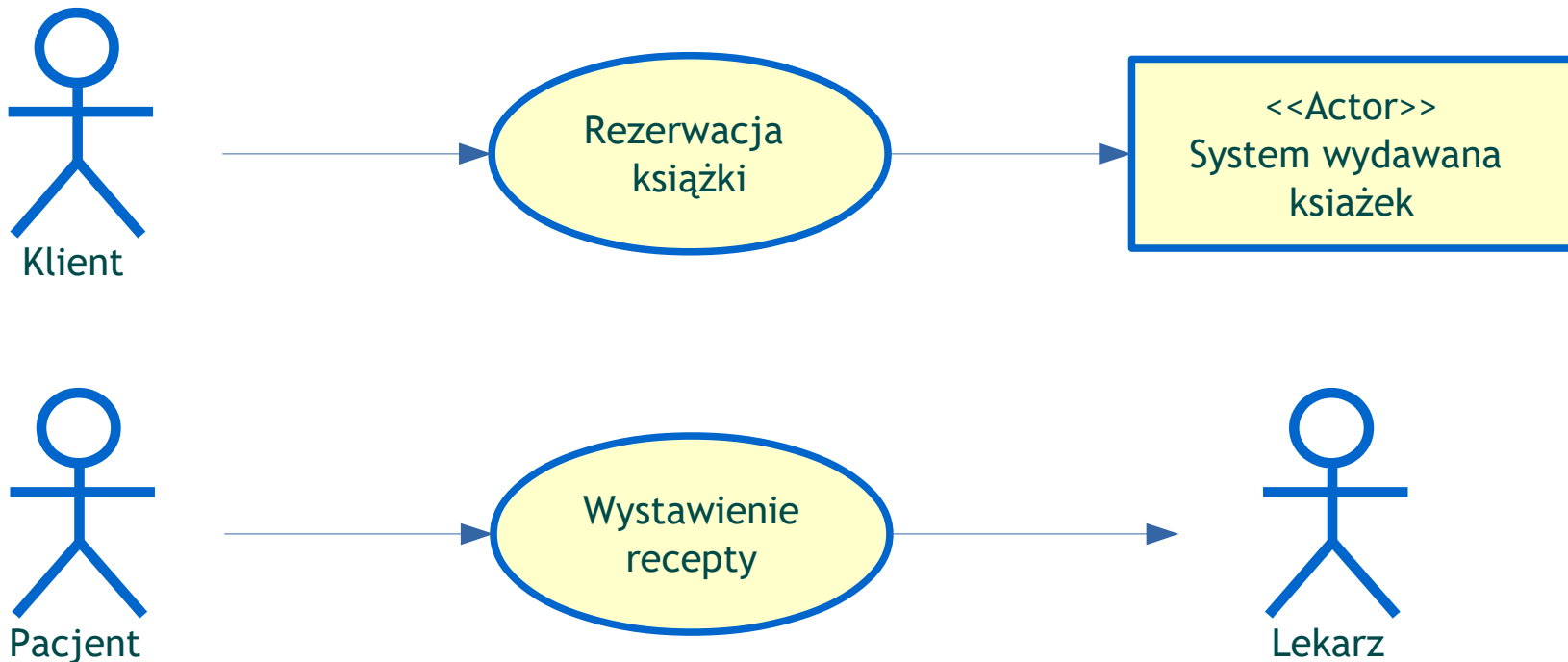
- ▶ Linia oznaczająca „aktywację” (nieprzerywana, bez strzałek) przypadku użycia może występować *tylko* pomiędzy *aktorem* a przypadkiem użycia.



Uwaga, związki *include*, *extend* i *dziedziczenie* to osobna sprawa.

## Powiązania - co z tymi strzałkami

- ▶ Standardowo powiązania pomiędzy aktorami a przypadkami użycia to *nieprzerywana linia bez strzałek*.
- ▶ Powszechną praktyką jest jednak wykorzystanie strzałek, określających czy aktor jest aktywny czy pasywny.

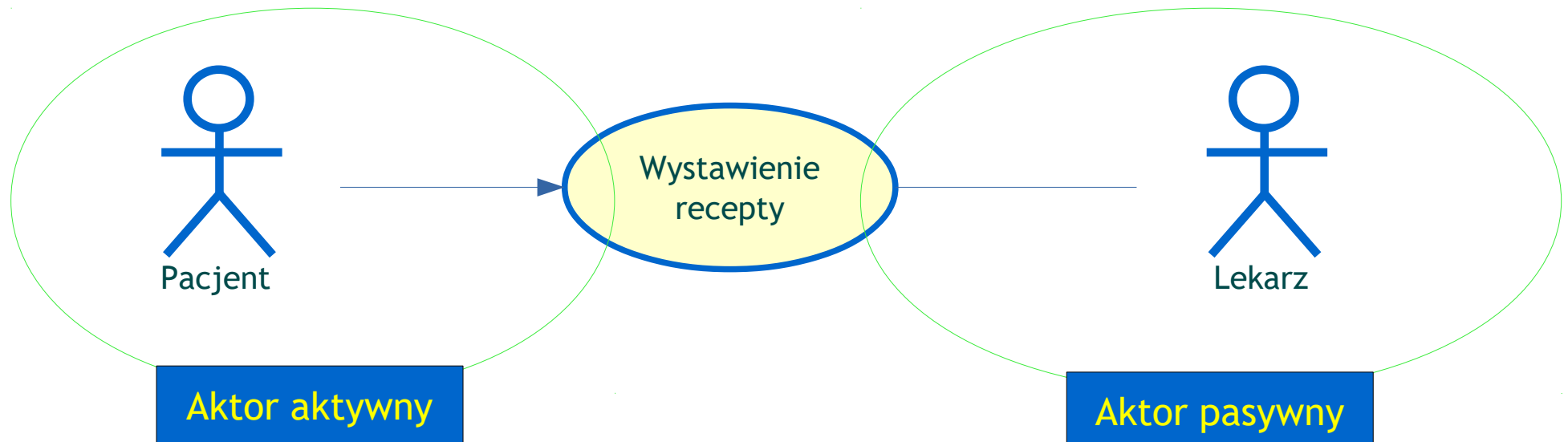


- ▶ Aktor pasywny – w sensie biznesowym – nie inicjuje bezpośrednio interakcji z systemem, ale jego istnienie jest konieczne dla funkcjonowania systemu.

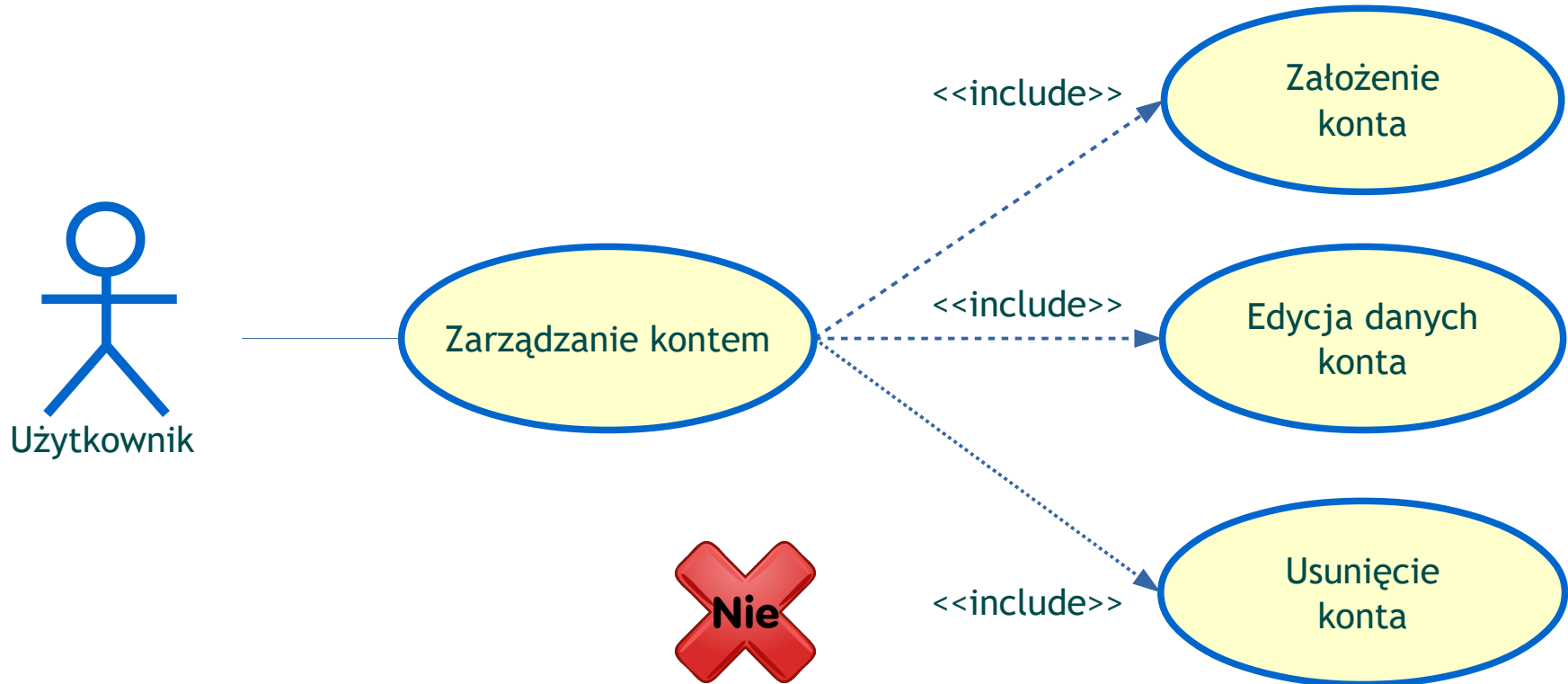


## Powiązania - co z tymi strzałkami, cd...

- ▶ Uwaga – kierunek strzałki *nie określa* kierunku przepływu danych.
- ▶ Dane mogą przepływać w obie strony.
- ▶ Strzałka określa jedynie inicjowanie aktywności.
- ▶ Ponieważ UML standardowo nie definiuje związków ze strzałkami, ich *interpretacja może się zmieniać* w zależności od *konkretnej metodyki* czy *konkretnego narzędzia*.
- ▶ Można spotkać podejścia w którym związek bez strzałek określa *aktora pasywnego*.
- ▶ Pojęcie aktorów pasywnych i aktywnych nie jest również określone w UML.

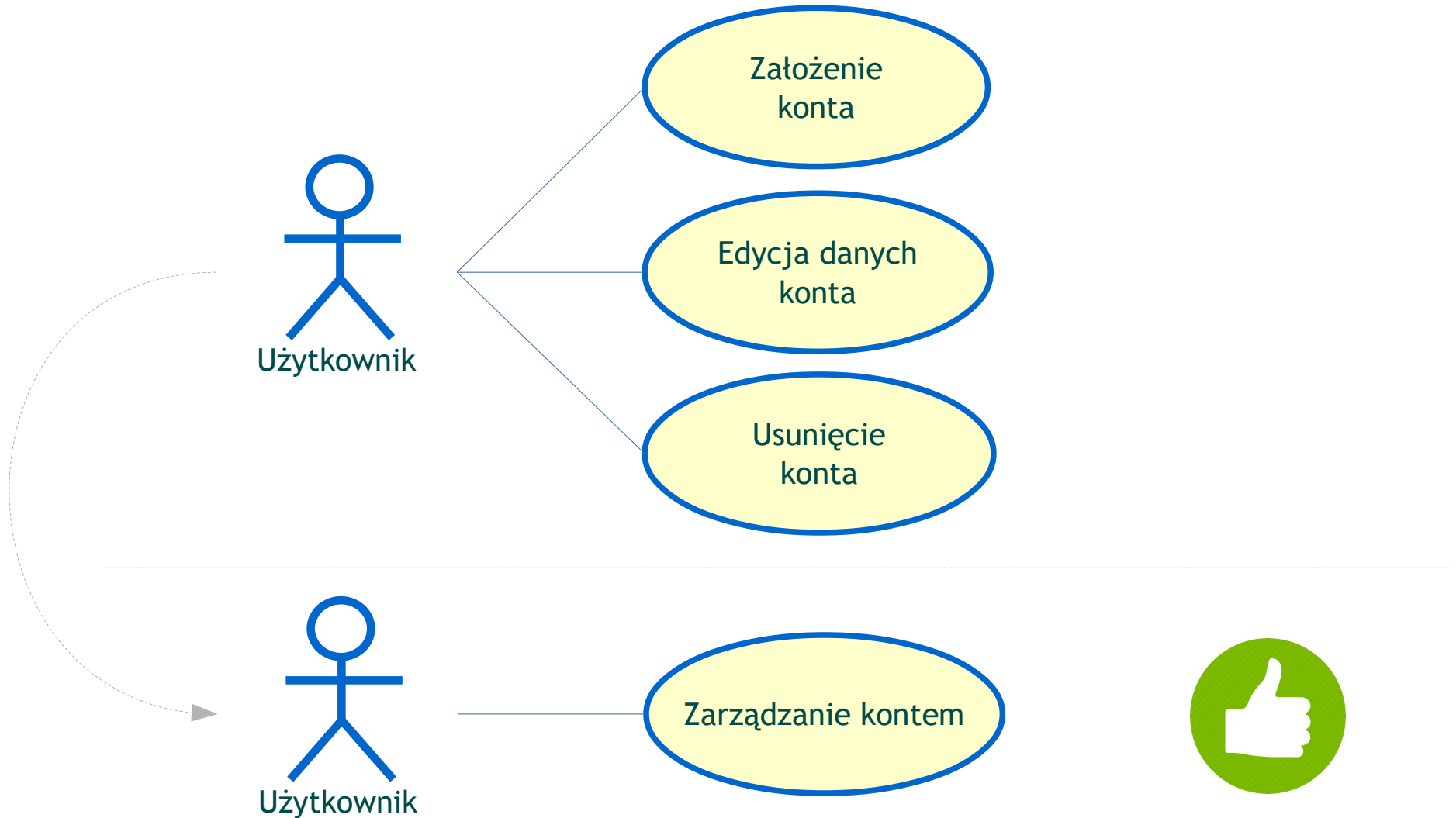


# Nie dekomponujemy, Use Case nie są dla programisty



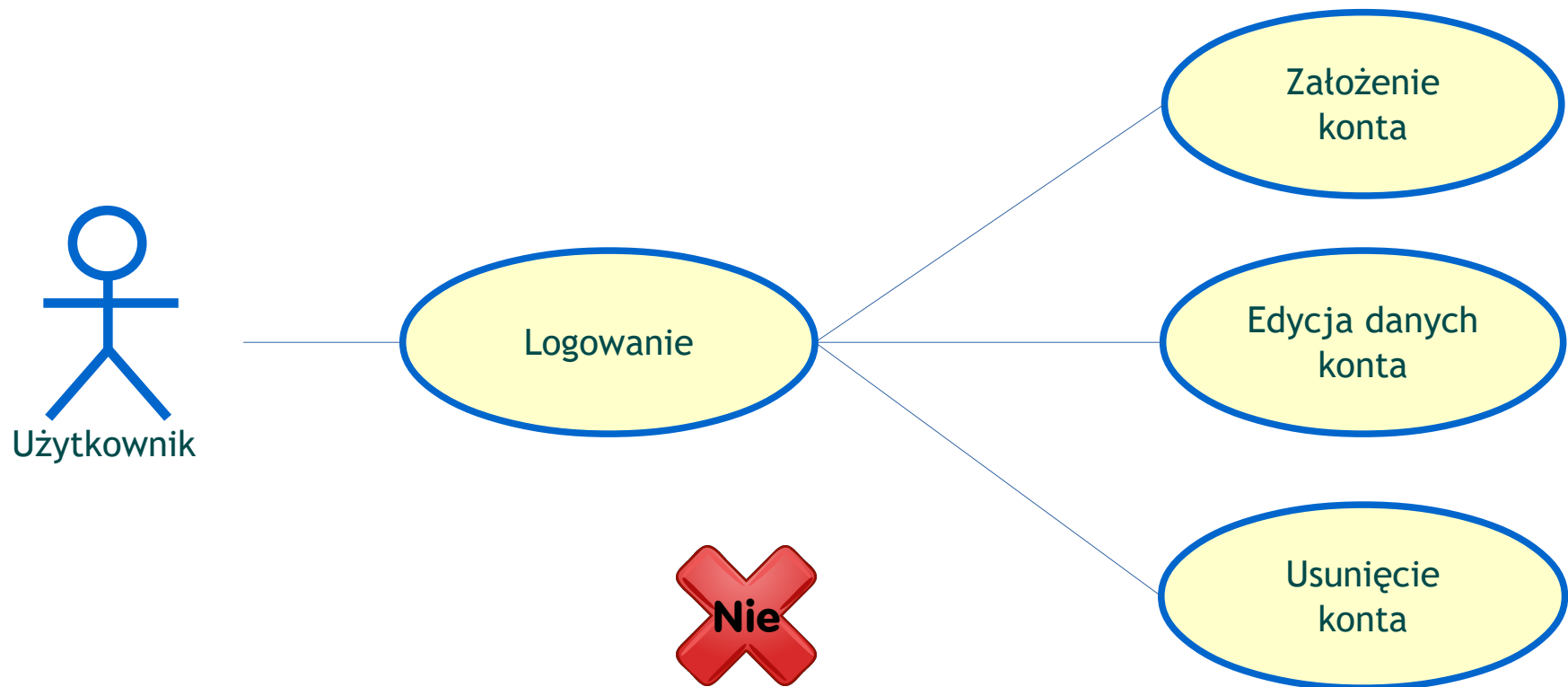
# Uwagi dodatkowe - duża liczba przypadków użycia

- ▶ Jeżeli na pojedynczym diagramie zaczyna być „ciasno” ze względu na dużą liczbę przypadków użycia, należy definiować ogólniejsze przypadki.



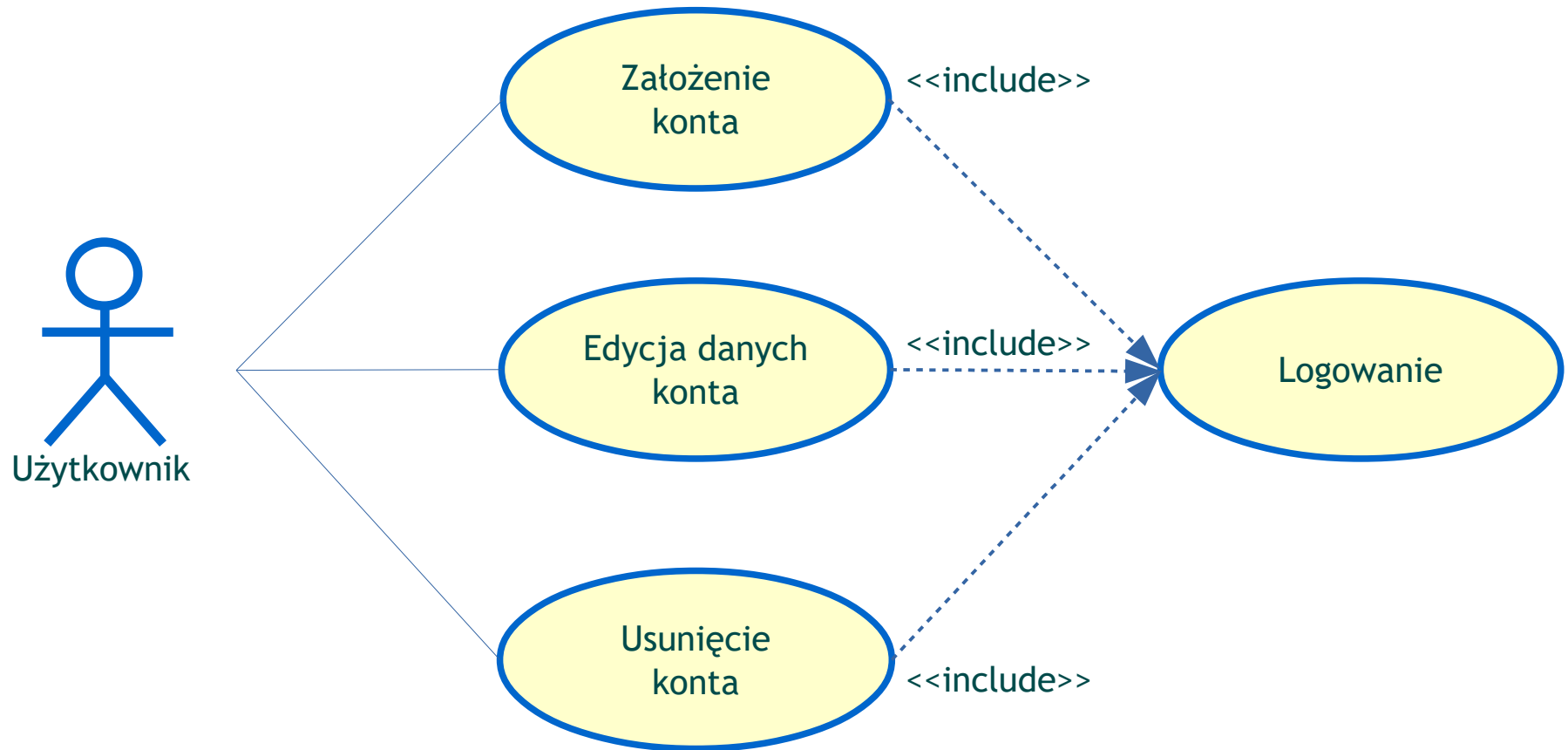
# Nie określamy zależności czasowych, Use Case nie są dla programisty

Fragment Use Case dla sklepu internetowego



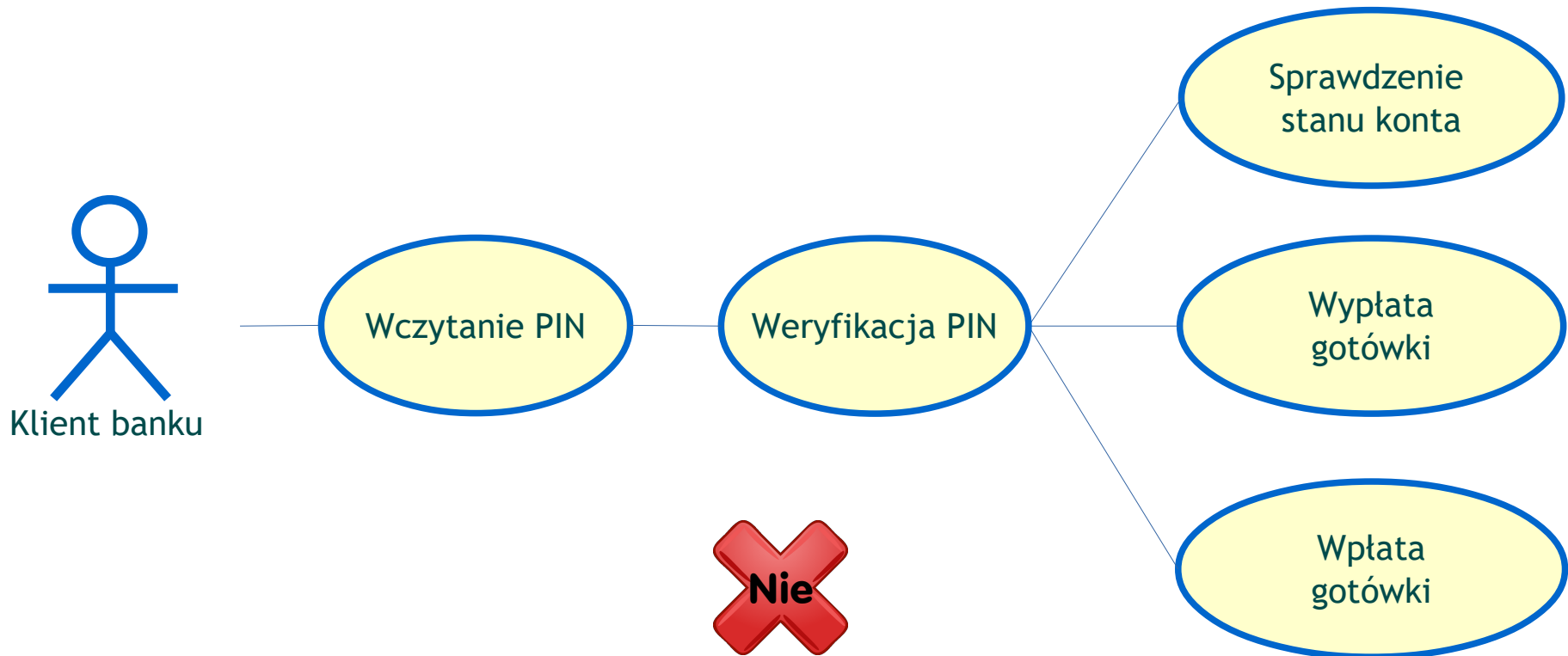
# Nie określamy zależności czasowych, Use Case nie są dla programisty

Fragment Use Case dla sklepu internetowego



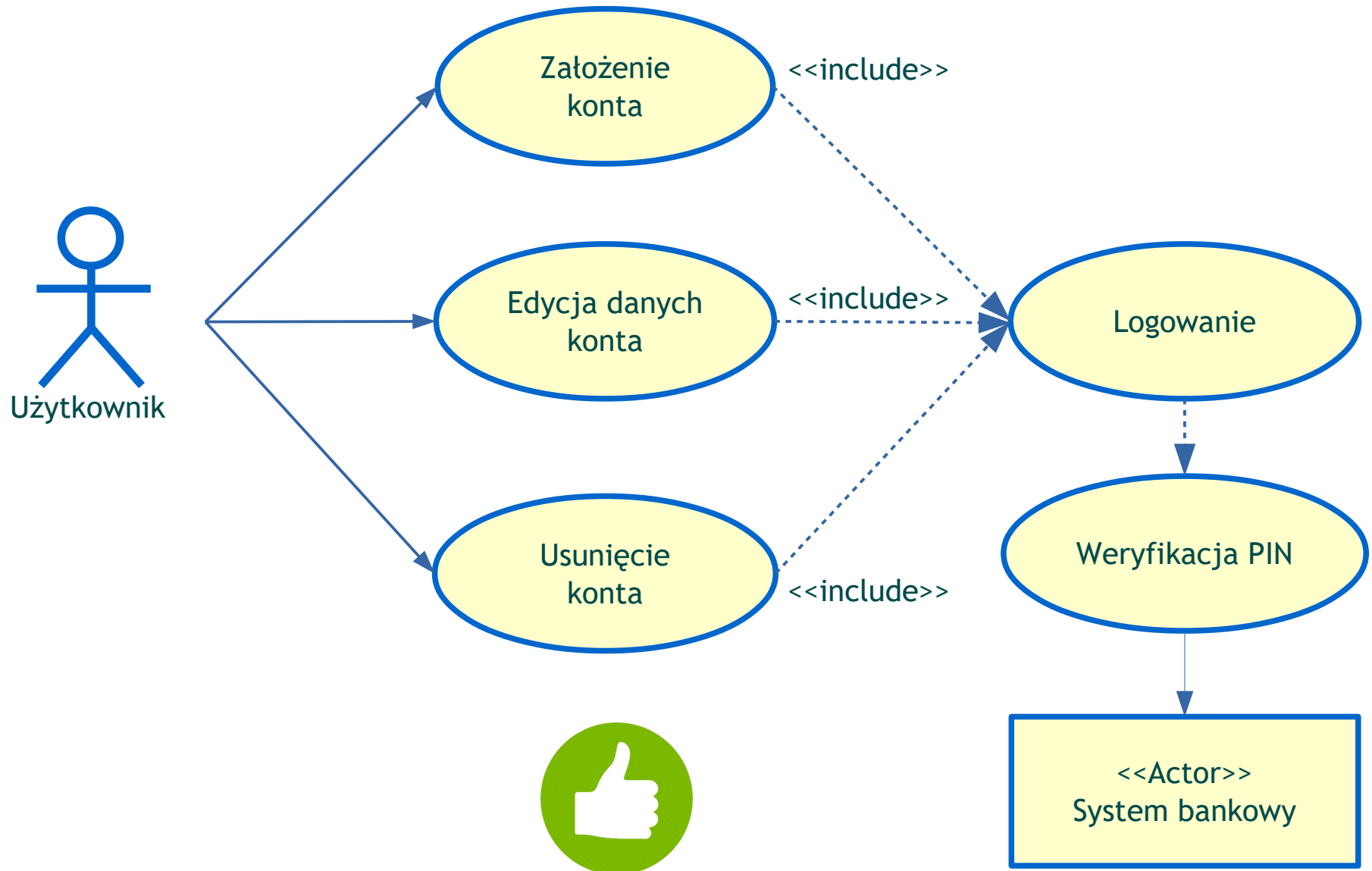
# Nie określamy zależności czasowych, Use Case nie są dla programisty

Fragment Use Case dla bankomatu



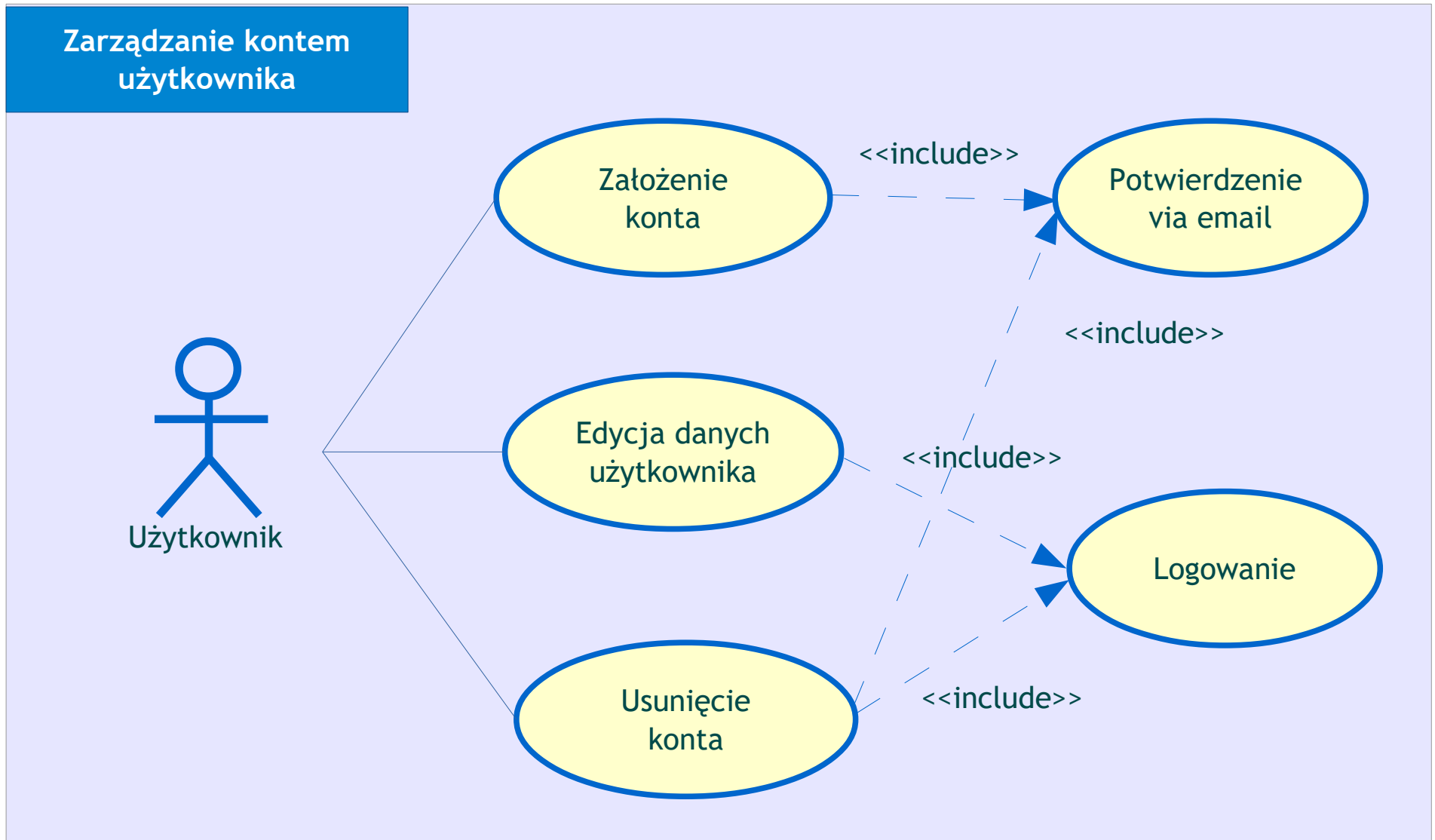
# Nie określamy zależności czasowych, Use Case nie są dla programisty

Fragment Use Case dla sklepu internetowego



# Uwagi dodatkowe - duża liczba przypadków użycia

- ▶ Dla złożonego przypadku użycia można rozrysować osobny, szczegółowy diagram.





## Uwagi dodatkowe - funkcje systemu aktywowane „czasem”

- ▶ Aktor może być rozumiany jako *źródło zdarzeń* powodujących *akcję* wykonywaną przez system.
- ▶ Jeżeli jakieś akcje mają być wykonywane automatycznie okresowo, można wprowadzić aktora reprezentującego zdarzenie czasowe.

