

# Diagram przypadków użycia - Use case diagram

## Przypadek użycia

- Jest przypadkiem, w którym dany system jest używany w celu spełniania jednego lub większej liczby wymagań użytkowników.
- Wychwytuje fragment funkcji udostępnianych przez system.
- Określają wymagania funkcjonalne systemu.

## Przypadki użycia

- Przypadki użycia dotyczą każdej innej części projektu systemu.
- Przypadki użycia nie określają wymagań нефunkcjonalnych systemu

## Wychwytywanie przypadków użycia

- Analiza opisu systemu z punktu widzenia przyszłego użytkownika.
  - Wymagania konieczne (bez nich system nie będzie funkcjonował).
  - Wymagania opcjonalne (pożądane, mogą być porzucone jako pierwsze w chwili napotkania nieprzewidzianych trudności w chwili napotkania nieprzewidzianych trudności w realizacji systemu).
- Których funkcji aktor oczekuje od systemu?
- Czy aktor musi czytać, modyfikować, tworzyć lub likwidować pewne informacje w systemie?
- Czy aktor powinien być powiadomiony o zdarzeniach w systemie lub sam informować system? Co te zdarzenia reprezentują w kategoriach funkcji?
- Czy codzienna praca aktora może być uproszczona lub bardziej efektywna?
- Jakich danych we/wy wymaga system? Skąd pochodzą i dokąd są kierowane?
- Jakie są wąskie gardła obecnego działania?

## Aktorzy

Aktorzy mogą być:

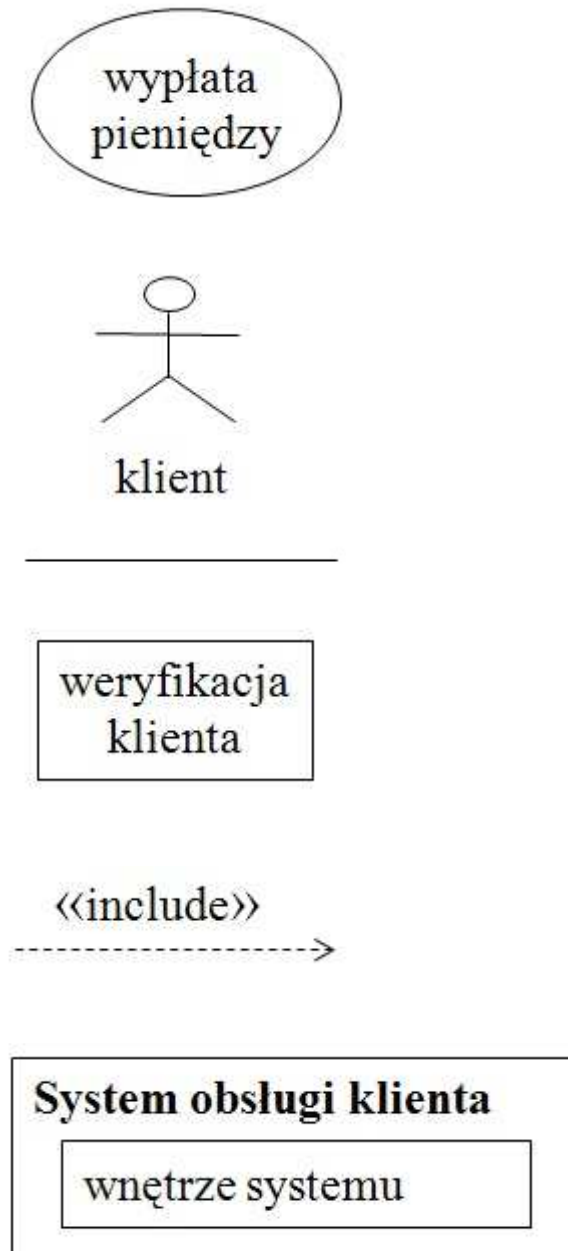
- Ludźmi wchodzącymi w interakcję,
- Systemami zewnętrznymi
- Częściami systemu, które mają wpływ na funkcjonowanie systemu, ale same przez ten system nie mogą być zmieniane (jak np. zegar systemowy).

## Rozpoznawanie aktorów

- Kto będzie używał podstawowych funkcji?
- Kto wymaga wspomagania w swej pracy i przy których codziennych zadaniach?
- Kto ma administrować systemem, konserwować i utrzymywać w działaniu?
- Jakimi urządzeniami zawiaduje system?

- Z którymi systemami system ma współpracować (np. wymieniać dane)?
- Kto jest zainteresowany rezultatami działania systemu?

## Notacja



**Przypadek użycia:** Powinien mieć unikalną nazwę, opisującą przypadek użycia z punktu widzenia jego zasadniczych celów. Czy lepiej jest stosować nazwę opisującą czynność ("wypłata pieniędzy") czy polecenie ("wyplac pieniadze") - zdania są podzielone.

**Aktor:** Powinien mieć unikalną nazwę.

**Interakcja:** Pokazuje interakcję pomiędzy przypadkiem użycia a aktorem.

**Blok ponownego użycia:** Pokazuje fragment systemu, który jest używany przez kilka przypadków użycia; może być oznaczony jako samodzielny przypadek użycia.

**Relacja typu << include >> lub << extend >>:**

Pokazuje związek zachodzący między dwoma przypadkami użycia lub przypadkiem użycia a blokiem ponownego użycia.

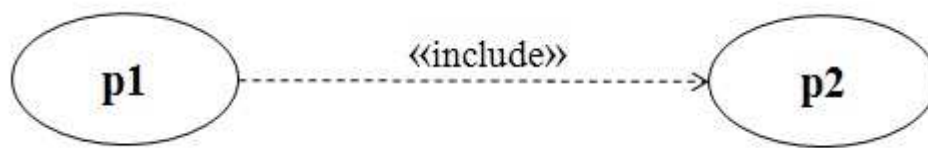
**Nazwa systemu wraz z otoczeniem systemu:** Pokazuje granicę pomiędzy systemem a jego otoczeniem.

## Zależności między przypadkami użycia

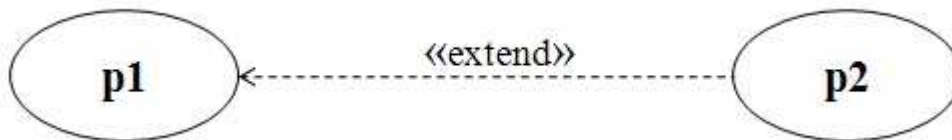
Przypadki użycia mogą być konstruowane w dowolnej kolejności, chyba że występują między nimi relacje typu:

<< include >> czy << extend >>.

p1 jest tu przypadkiem bazowym i zawsze występuje jako pierwsze w kolejności działania.



Przebieg podstawowy (sekwencyjny): p1 zawsze włącza (używa) p2.



Przebieg opcjonalny (alternatywny): p1 jest czasami rozszerzane o p2 (inaczej: p2 czasami rozszerza p1).

<< include >>: wskazuje na wspólny fragment wielu przypadków użycia (wyłączony "przed nawias"); wykorzystywane w przebiegach podstawowych (operacje zawsze wykonywane)

<< extend >>: strzałka prowadzi od przypadku użycia, który czasami rozszerza inny przypadek użycia - wykorzystywane w przebiegach opcjonalnych (operacje nie zawsze wykonywane)

## Stopień szczegółowości diagramów

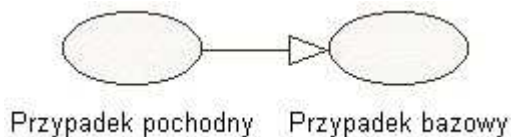
Model przypadków użycia dostarcza bardzo abstrakcyjnego spojrzenia na system - spojrzenia z pozycji aktorów, którzy go używają. Nie włącza zbyt wielu szczegółów, co pozwala wnioskować o funkcjonalności systemu na odpowiednio wysokim poziomie. Podstawowym (choć nie jedynym) zastosowaniem jest tu dialog z przyszłymi użytkownikami zmierzający do sformułowania poprawnych wymagań na system.

**Model zbyt szczegółowy - utrudnia analizę,  
zbyt ogólny - nie pozwala na wykrycie bloków ponownego użycia.**

## Dziedziczenie przypadków użycia

Dziedziczenie pozwala na współdzielenie większości zachowania ogólnego przypadku użycia.

Dziedziczenie przypadków użycia - jeden przypadek użycia dziedziczy zachowanie innego i zarazem dodaje lub modyfikuje część tego zachowania. Np. z przypadku użycia "zapisz na uniwersytet" może dziedziczyć inny: "zapisz krewnego profesora na uniwersytet". Oznacza to, że ogólna procedura jest podobna, ale np. obowiązują inne reguły jeśli zapisujący się jest krewnym profesora.



**Dziedziczenie:** Analogiczne do dziedziczenia klas.

## Wskazówki

- Dla każdego aktora, znajdź funkcje (zadania), które powinien wykonywać w związku z jego działalnością w zakresie zarówno dziedziny przedmiotowej, jak i wspomagania działalności systemu informacyjnego.
- Staraj się powiązać w jeden przypadek użycia zespół funkcji realizujących podobne cele. Unikaj rozbicia jednego przypadku użycia na zbyt wiele pod-przypadków.
- Nazwy dla przypadków użycia: powinny być krótkie, ale jednoznacznie określające charakter zadania lub funkcji. Nazwy powinny odzwierciedlać czynności z punktu widzenia aktorów, a nie systemu, np. "wpłacanie pieniędzy", a nie "przyjęcie pieniędzy od klienta".
- Opisz przypadki użycia przy pomocy zdań w języku naturalnym, używając terminów ze słownika.
- Uporządkuj aktorów i przypadki użycia w postaci diagramu.
- Niektóre z powstałych w ten sposób przypadków użycia mogą być mutacjami lub szczególnymi przypadkami innych przypadków użycia. Przeanalizuj powiązania aktorów z przypadkami użycia i ustal, które z nich są zbędne lub mogą być uogólnione.
- Wyodrębnij "przypadki bazowe", czyli te, które stanowią istotę zadań, są normalnym, standardowym użyciem. Pomiń czynności skrajne, wyjątkowe, uzupełniające lub opcjonalne.
- Nazwij te przypadki bazowe. Ustal powiązania przypadków bazowych z innymi przypadkami, poprzez ustalenie ich wzajemnej zależności: sekwencji czy alternatywy.
- Dodaj zachowania skrajne, wyjątkowe, uzupełniające lub opcjonalne. Ustal powiązanie przypadków bazowych z tego rodzaju zachowaniem. Może ono być także powiązane w pewną strukturę.
- Staraj się, aby bloki specyfikowane wewnątrz każdego przypadku użycia nie były zbyt ogólne lub zbyt szczegółowe. Zbyt szczegółowe bloki utrudniają analizę. Zbyt ogólne bloki zmniejszają możliwość wykrycia bloków ponownego użycia. Struktura nie może być zbyt duża i złożona.
- Staraj się wyizolować bloki ponownego użycia. Przeanalizuj podobieństwo nazw przypadków użycia, podobieństwo nazw i zachowania bloków ponownego użycia występujących w ich specyfikacji. Wydzielenie bloku ponownego użycia może być powiązane z określeniem bardziej ogólnej funkcji lub dodaniu nowej specjalizacji do istniejącej funkcji.