Inżynieria wymagań – Wykład 2

Agenda

- Przypadki użycia (use cases).
- Metoda punktów przypadków użycia szacowania pracochłonności.

Zachowanie systemu

- Zachowanie systemu jest opisem tego jak system działa i reaguje - jest to widoczny z zewnątrz przejaw aktywności systemu.
- Zachowanie systemu jest opisywane przez model przypadków użycia.
- Model przypadków użycia opisuje:
  - system,
  - jego środowisko,
  - związki pomiędzy systemem a jego środowiskiem.

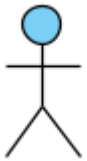
Diagram przypadków użycia

- Model przypadków użycia opisuje wymagania funkcjonalne względem systemu z wykorzystaniem przypadków użycia.

- Przypadek użycia odpowiada pojedynczej funkcjonalności systemu.

- Diagram przypadków użycia użycia (*Use Case Diagram*) jest graficznym przedstawieniem funkcjonalności systemu wraz z otoczeniem tej funkcjonalności, tak jak je widzi użytkownik.

Aktor a przypadek użycia



Kupiec

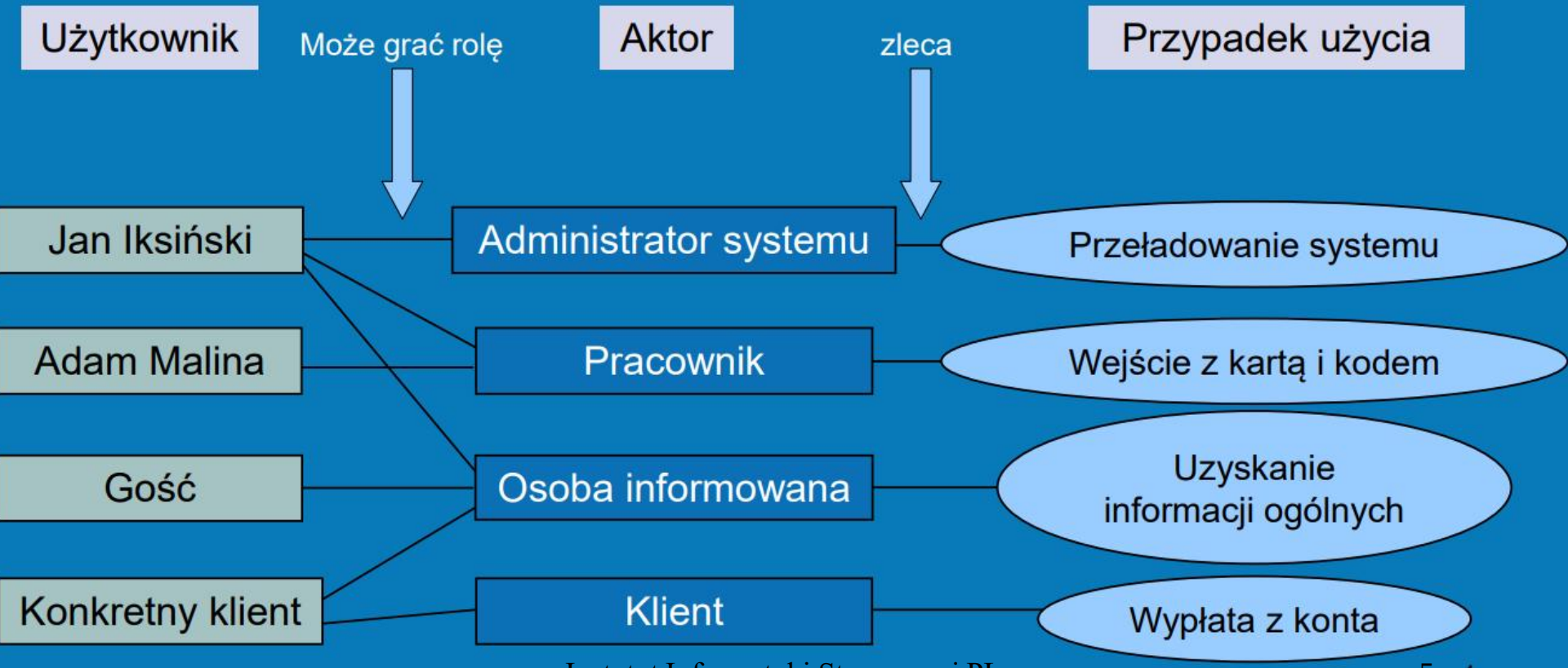
przypadek użycia

Reprezentuje rolę, którą może grać w systemie jakiś jego **użytkownik** (np. **kierownik, urzędnik, klient**) lub **system zewnętrzny** wchodzący w interakcję z modelowanym systemem. Nie musi być fizycznym obiektem.

Reprezentuje sekwencję operacji inicjowaną przez aktora, niezbędnych do wykonania **zadania zleconego** (zainicjowanego) **przez aktora**, np. potwierdzenie pisma, złożenie zamówienia, itp.

Aktor - konkretny byt, czy rola?

Aktor modeluje grupę osób pełniących pewną rolę, a nie konkretną osobę.



Identyfikacja aktorów

Aby poprawnie określić aktorów należy odpowiedzieć na pytania:

- Jaka grupa użytkowników potrzebuje wspomagania ze strony systemu (np. osoba wysyłająca korespondencję)?
- Jacy użytkownicy są konieczni do tego, aby system działał i wykonywał swoje funkcje (np. administrator systemu)?
- Z jakich elementów zewnętrznych (innych systemów, sensorów, czujników, itp.) musi korzystać system, aby realizować swoje funkcje.

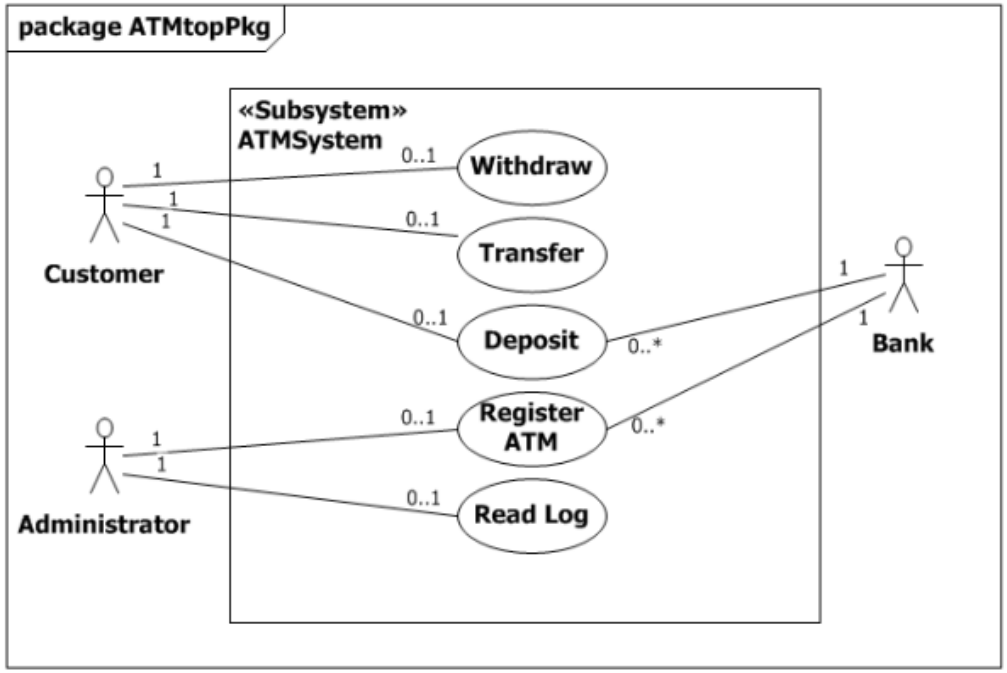
Po wyszukaniu aktorów, należy ustalić:

- Nazwę dla każdego aktora/roli,
- Zakresy znaczeniowe dla poszczególnych nazw aktorów oraz relacje pomiędzy zakresami (np. sekretarka, pracownik administracji, pracownik, dowolna osoba).
- Niekiedy warto ustalić hierarchię dziedziczenia dostępu do funkcji systemu dla aktorów.

Znalezienie przypadków użycia

- Dlaczego aktor chce używać systemu?
- Czy aktor będzie tworzył, przechowywał, zmieniał, usuwał lub czytał dane w systemie? Jeżeli tak – dlaczego?
- Czy aktor będzie potrzebował informować system o zewnętrznych zdarzeniach lub zmianach?
- Czy aktor będzie potrzebował być informowanym o określonych zdarzeniach w systemie?

Przykład diagramu przypadku użycia



- https://www.omg.org/spec/UML/2.5.1/PDF
- Pokazanie tylko istotnych dla aktora działań

Identyfikacja przypadków użycia – porady (1)

- 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 podprzypadków.
- 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 pojęć.

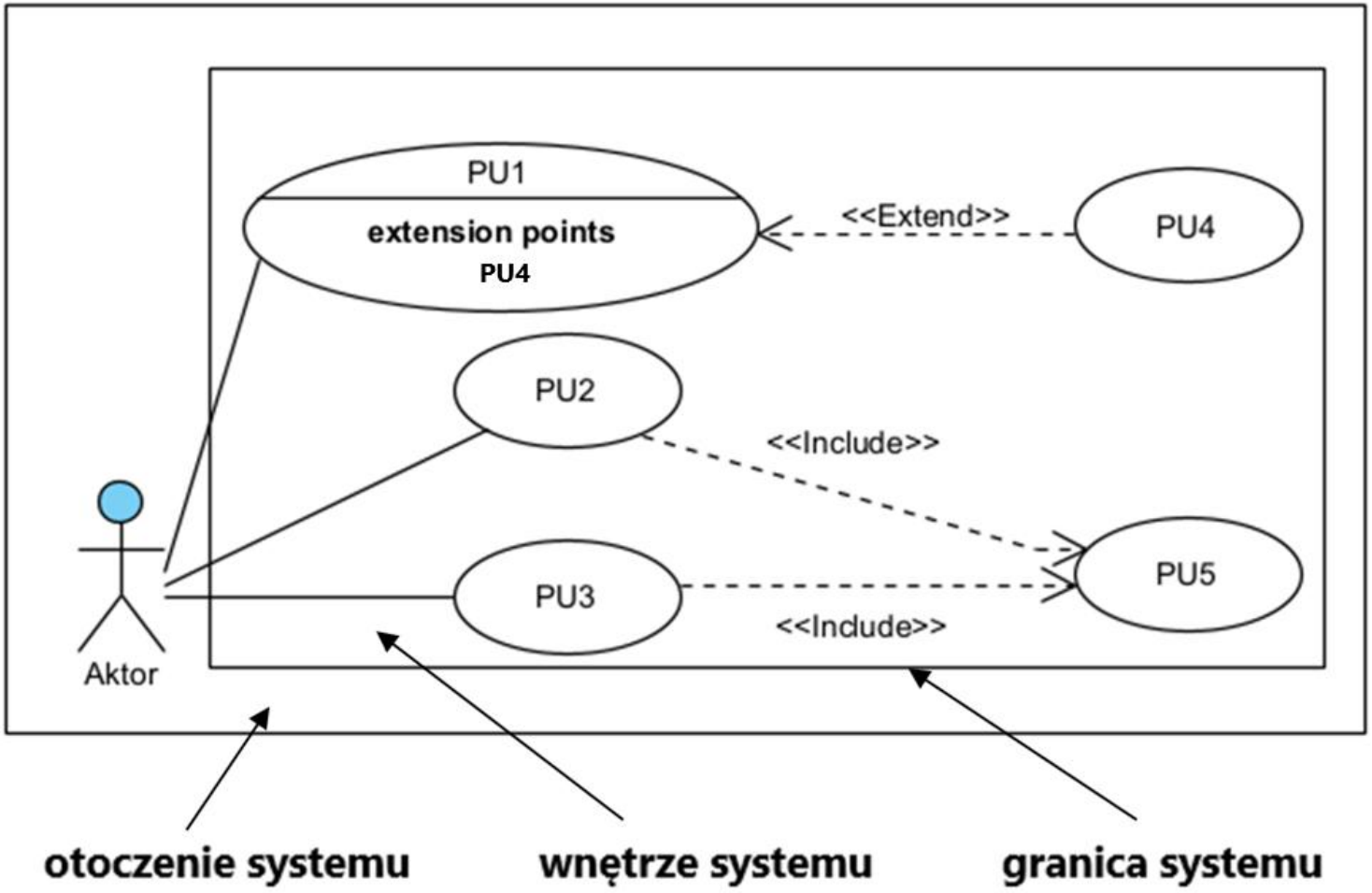
Identyfikacja przypadków użycia porady (2)

- Uporządkuj aktorów i przypadki użycia w postaci diagramu przypadków użycia.
- 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.

Przypadki użycia realizują cele biznesowe

- Zgodnie z definicją UML logowanie nie jest przypadkiem użycia, ponieważ nie dostarcza wartości aktorowi.
- Można użyć jednak logowania jako przypadku użycia, ponieważ zawiera ono i wiąże bardziej złożone zachowania, o ile jest to uzasadnione do pokazania tych zależności.
- Nie ma przyzwolenia na czynności zawierające typowe przypadki CRUD (jak: „utwórz nowy wpis”, „usuń wpis”), bo nie dostarczają wartości aktorom - stosuje się ogólne zapisy, np. „zapisz się na kurs”.

Aktor jest poza systemem

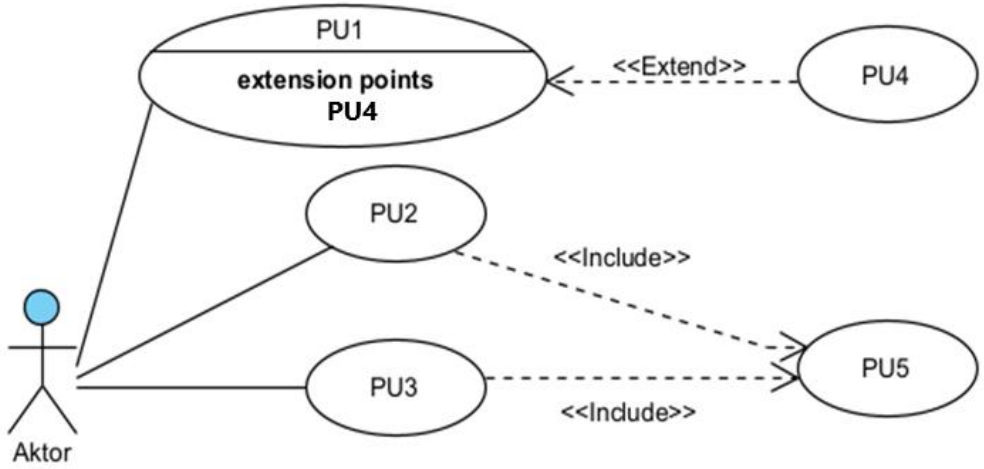


Przypadki konkretne i abstrakcyjne (1)

- Kiedy wyciągamy zachowanie poprzez include lub extend, te nowe przypadki najczęściej nie są wykonywane samodzielnie.
- Dlatego określa się je jako abstrakcyjne, ponieważ istnieją tylko jako część innego przypadku, który określamy jako konkretny.
- Ponieważ istnieje możliwość gdy jeden konkretny przypadek wywołuje przez <<extend>> inny konkretny przypadek, to dla uniknięcia nieporozumień wszystkie przypadki rozszerzajace i włączane muszą być abstrakcyjne.

Przypadki konkretne i abstrakcyjne (2)

- Przypadki PU1, PU2 i PU3 uważamy za konkretne.
- Przypadki PU4 i PU5 uważamy za abstrakcyjne.
- Model musi być zrozumiały po ukryciu przypadków abstrakcyjnych.

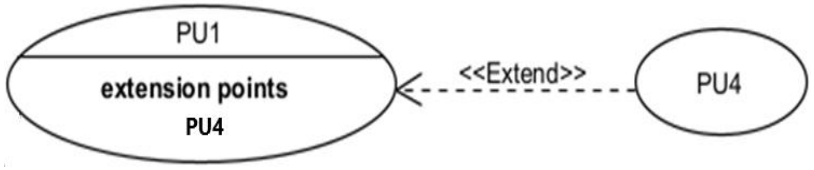


Związki include i extend

- Wydobycie z przypadku użycia zachowania, które będzie zaprezentowane w oddzielnym przypadku jest strukturyzacją.
- Przykłady przyczyn strukturyzacji to zachowanie: wspólne, opcjonalne, wyjątkowe lub takie, które ma zostać rozwinięte w późniejszych iteracjach.
- **Nadużywanie strukturyzacji prowadzi do dekompozycji funkcjonalnej, co utrudnia testowanie oraz implementację.**

Kiedy używamy extend?

- Extend łączy od przypadku rozszerzającego do przypadku bazowego.
- Wstawia zachowanie przypadku rozszerzającego do przypadku bazowego tylko gdy mamy spełniony warunek rozszerzenia.



Dokonaj płatności

1. Ciąg podstawowy

...

N-1 Klient kończy zakupy

N. Klient wybiera opcję dokonania płatności

N+1. …

...

2. Ciągi alternatywne

A1 Zmiana sposobu płatności

...

3. Punkty rozszerzalności

Punkt rozszerzalności „Specyfika płatności" mieści się w kroku N ciągu podstawowego i w kroku A1.7 ciągu alternatywnego A1 „Zmiana sposobu płatności"

Dokonaj płatności kartą

Przypadek rozszerza przypadek „Dokonaj płatności" w punkcie rozszerzalności „Specyfika płatności"

1. Ciąg podstawowy

1. **Jeżeli klient wybierze płatność kartą**, system prosi o podanie numeru karty
2. Klient wpisuje i zatwierdza numer karty
3. System łączy się z …

Dynamiczne planowanie przypadku użycia z extend

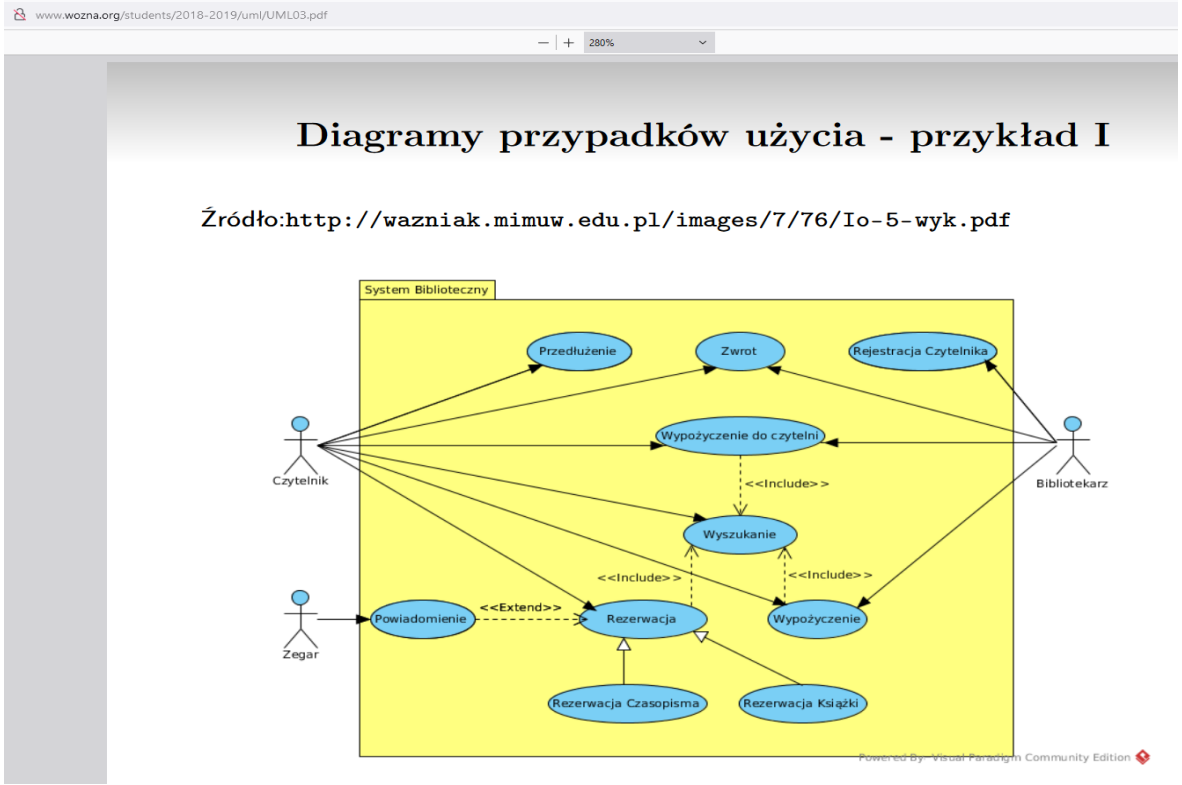
- Nazwane punkty rozszerzalności można umieszczać wszędzie, gdzie oczekujemy, że wymagania ulegną zmianie.
- Przypadek użycia nie musi wiedzieć, w jaki sposób zostanie rozszerzony (i czy będzie) – decydują o tym warunki w przypadkach rozszerzających.
- Do punktu rozszerzalności może być „doczepiona” dowolna liczba przypadków rozszerzających.

Zastosowanie związku extend

- Można zauważyć, że istnieje sytuacja gdy mamy dwie wersje specyfikacji – wersja A posiada swoje wymagania w jednym przypadku rozszerzającym, a wersja B w drugim. Wymagania wspólne znajdują się w przypadku bazowym (wspólnym).
- W danym punkcie rozszerzalności pojawia się kilka możliwych rozwojów wypadków, każdy określony indywidualnymi warunkami – można w ten sposób pokazywać gdzie nastąpiły zmiany w wymaganiach.

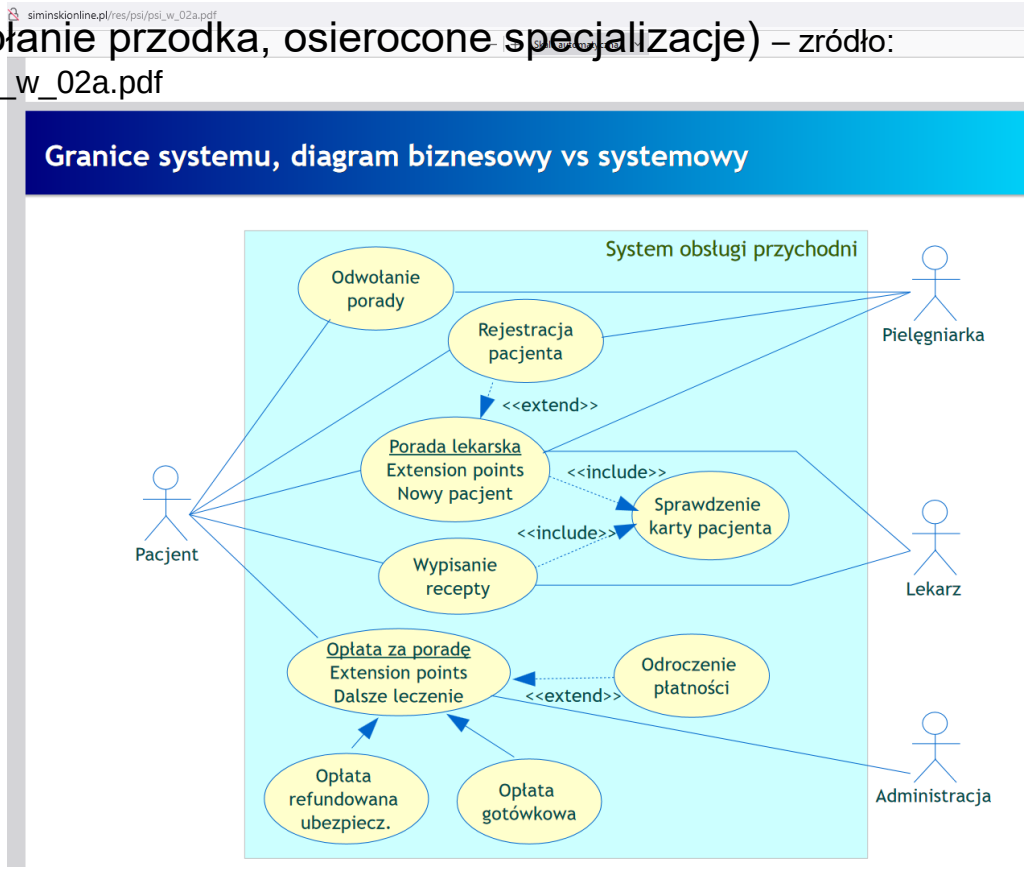
Stare diagramy przypadków użycia są nieaktualne i swoiście błędne (1)

Przestarzałe interakcje ze strzałkami (nie stosuje się obecnie), generalizacja (wołanie przodka, a są osierocone specjalizacje) - źródło: www.wozna.org/students/2018-2019/uml/uml03.pdf



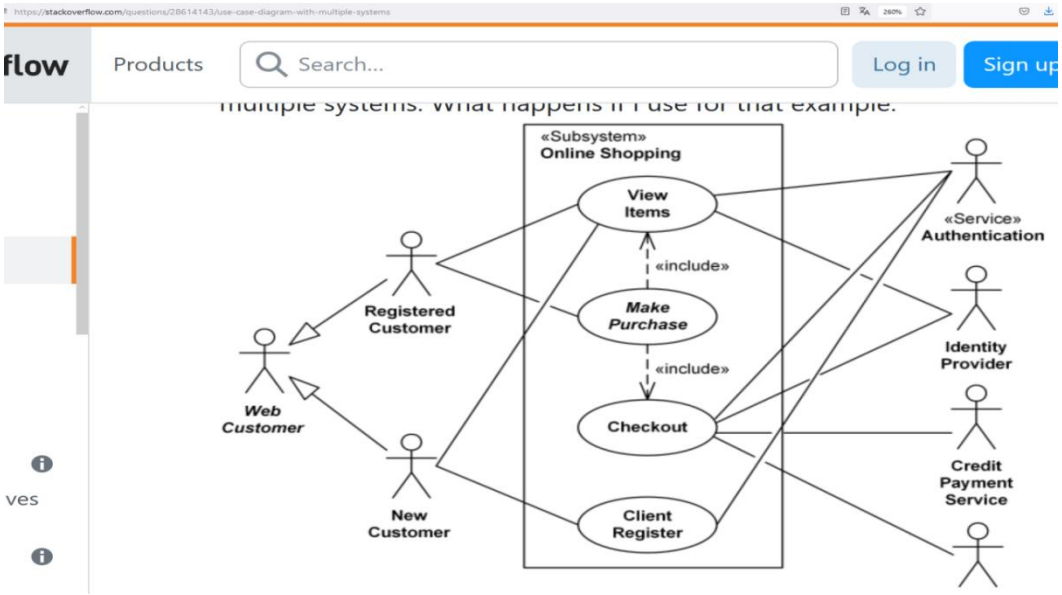
Stare diagramy przypadków użycia są nieaktualne i swoiście błędne (2)

Generalizacje (wołanie przodka, osierocone specjalizacje) – zródło: siminskionline.pl/psi/psi_w_02a.pdf



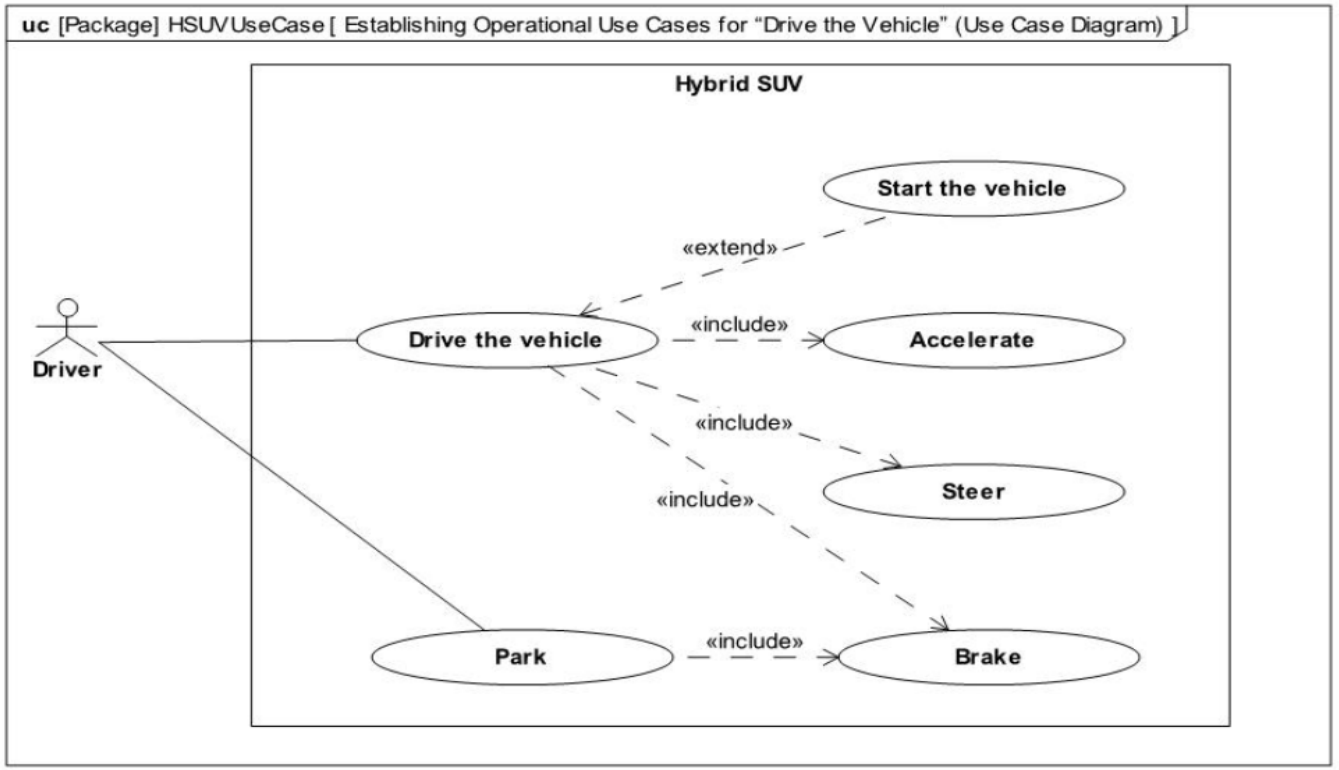
Stare diagramy przypadków użycia są nieaktualne i swoiście błędne (3)

Złe użycie zawierania <<include>> (nie ma tu dwóch z których można wyłonić „includowanego”), błędnie połączono aktorów z „includowanymi” przypadkami użycia (tego się nie robi), interakcje między aktorami nic nie wnoszą – źródło: stackoverflow.com/questions/28614143/use-case-diagram-with-multiple-systems



Stare diagramy przypadków użycia są nieaktualne i swoiście błędne (4)

Złe użycie zawierania <<include>> dla Accelerate i Steer– źródło: https://www.omg.org/spec/SysML/1.6/PDF



Właściwy poziom szczegółów

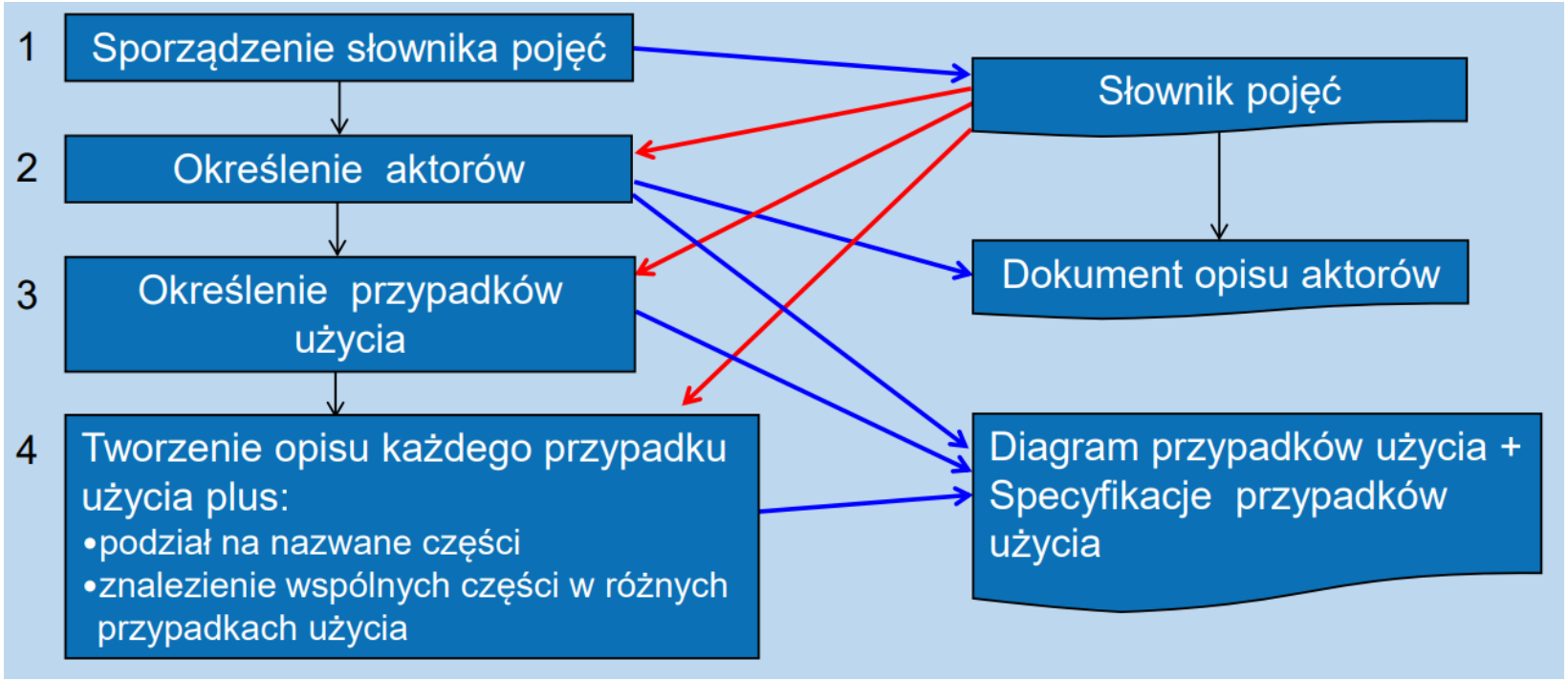
- Należy skupić się w opisie przypadków użycia na opisie zachowania, a nie dotykać zjawisk przetwarzania wewnętrznego w systemie.
- Pominąć należy założenia architektoniczne w opisie PU.
- Poza opisem PU jest także obszar szczegółów projektu interfejsu użytkownika.

Nazwa przypadku użycia

Nazwa powinna być:

- unikatowa, intuicyjna i samowyjaśniająca,
- jasno i jednoznacznie definiująca obserwowalny wynik uzyskiwany z przypadku użycia,
- tworzona z pespektywy aktora wyzwalającego przypadek użycia,
- opisywać zachowanie, którego przypadek dotyczy,
- zaczynać się od czasownika lub prostej kombinacji czasownika z rzeczownikiem.

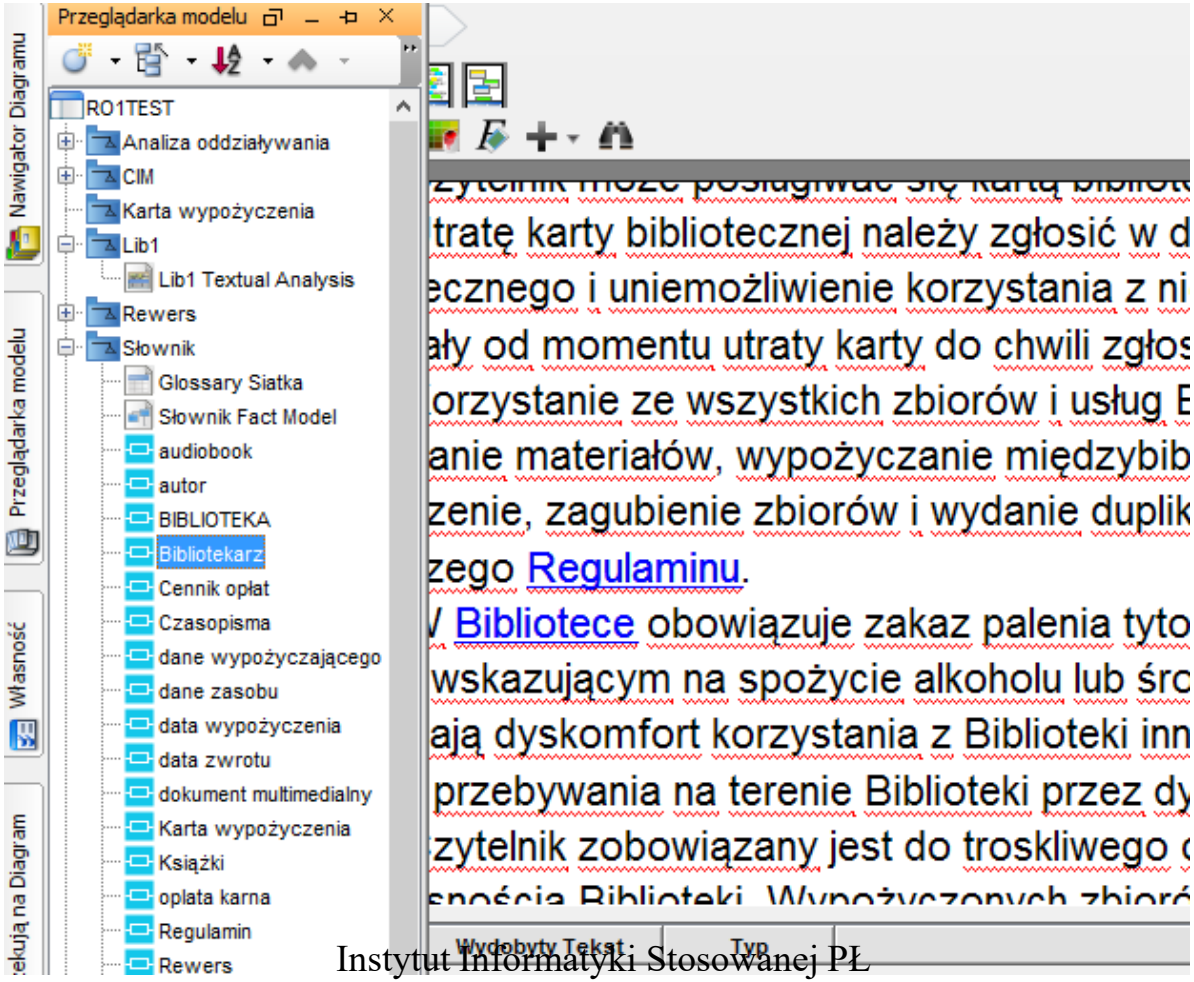
Miejsce przypadku użycia w modelu wymagań



Sporządzanie słownika pojęć

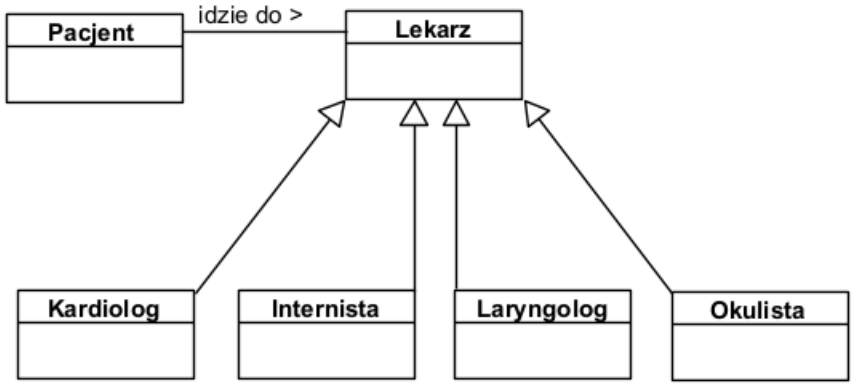
- Słownik dotyczy dziedziny problemowej.
- Tworzenie go polega na wyłowieniu wszystkich terminów z wymagań użytkownika.
- Terminy mogą odnosić się do aktorów, przypadków użycia, obiektów, operacji, zdarzeń, itp.
- Terminy w słowniku powinny być zdefiniowane w sposób precyzyjny i jednoznaczny.
- Posługiwanie się terminami ze słownika powinny być regułą przy opisie każdego kolejnego problemu, sytuacji czy modelu.

Słownik pojęć

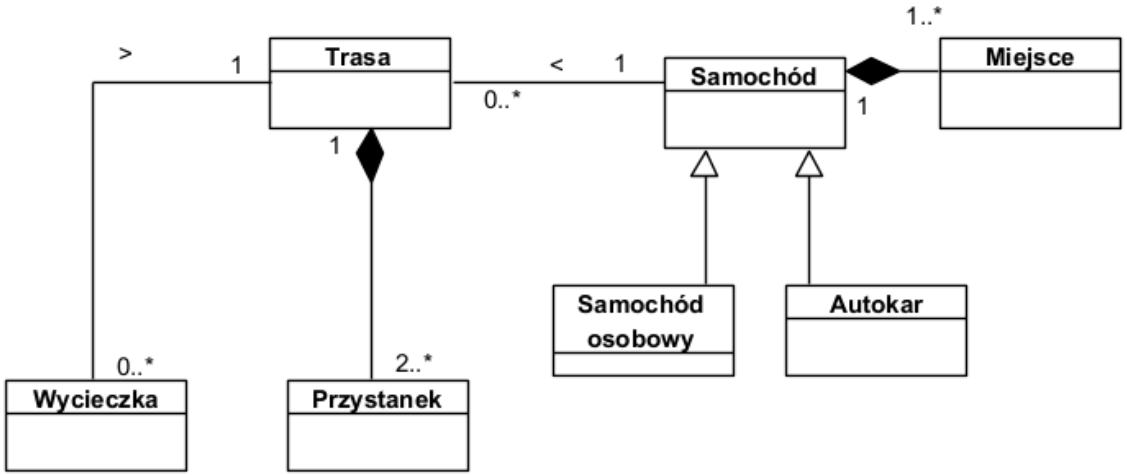


Wyrażenia ze słownika tworzą model pojęciowy

Model pojęciowy konsultacji medycznej.



Model pojęciowy wycieczki po trasie.



Opis tekstowy przypadku użycia

Przypadek użycia powinien być definiowany przez opis atrybutowy wg przyjętego szablonu, np:

- identyfikator (np. UC13) + nazwa (fraza-czasownik),
- źródło pochodzenia PU, odpowiedzialna osoba, autor,
- opis (krótki nieformalny opis tekstowy),
- wyzwalacz – zdarzenie uruchamiające PU z punktu widzenia użytkownika (aktora),
- warunki początkowe i końcowe „przed-” i „po-” wykonaniu przypadku użycia,
- scenariusze – sekwencja zdarzeń między systemem i aktorami (opis tekstowy),
- inne, m.in. priorytet, krytyczność.

Scenariusze – ciąg podstawowy, ciągi alternatywne

- Sam diagram przypadków użycia nie umożliwia pełnej specyfikacji wymagań funkcjonalnych.
- Do każdego przypadku użycia znajdującego się na diagramie potrzebny jest opis w postaci odpowiedniego scenariusza.
- Dopiero diagram przypadków użycia wraz ze zbiorem scenariuszy może być traktowany jako właściwa specyfikacja funkcjonalności systemu.

Przykład UC13 (1)

Identyfikator i nazwa PU:	UC13: Utworzenie grupy wsparcia.		
Źródło pochodzenia PU/Autor:	Stowarzyszenie/Trener		
Odpowiedzialna osoba:	Kowalska		
Aktor główny:	Trener	Aktorzy drugorzędni:	-
Opis:	Trener grupuje podopiecznych do zajęć grupy wsparcia we wskazanym terminie zgodnie z regułami biznesowymi.		
Reguły biznesowe:	Jeżeli wskazany termin jest zajęty przez inne zajęcia podopiecznego, to podopieczny nie może być dodany do grupy. Jeżeli liczba podopiecznych zapisanych do grupy wsparcia przekracza limit miejsc, to wybrany podopieczny nie zostanie dodany do grupy.		
Wyzwalacz:	Trener chce utrwalić w systemie grupę wsparcia we wskazanym terminie.		
Warunki początkowe:	Trener jest zalogowany w systemie.		
Warunki końcowe:	Trener posiada skompletowaną grupę na spotkanie.		

Przykład UC13 (2)

Ciąg podstawowy przepływu:	1. Trener wskazuje termin zajęć zarezerwowany w systemie dla grupy wsparcia. 2. Trener wybiera z grupy bazowej osoby podopiecznych do zajęć we wskazanym terminie. 3. System potwierdza brak zajętości wskazanego terminu u wszystkich wybranych podopiecznych. 4. Trener zatwierdza wybór podopiecznych do spotkania grupy wsparcia. 5. System dodaje wszystkich wybranych podopiecznych do spotkania grupy wsparcia. 6. Trener posiada skompletowaną grupę na spotkanie.
Ciąg alternatywny przepływu:	3a. System wskazuje zajętość wskazanego terminu u części podopiecznych i wskazuje trenerowi możliwość dodania innych osób do wypełnienia limitu. 5a. System wskazuje przekroczenie limitu miejsc w grupie i wskazuje trenerowi konieczność wybrania mniejszej liczby osób.

Przykład UC13 (3)

Wyjątki:	W1. Brak odpowiedzi ze strony systemu po podaniu terminu przez trenera. W2. Brak odpowiedzi ze strony systemu po wskazaniu osób przez trenera. W3. Brak odpowiedzi ze strony systemu po zatwierdzeniu wyboru przez trenera.
Priorytet: Krytyczność:	Wysoki Wysoka
Częstotliwość użycia:	Średnia
Dodatkowe informacje:	Trener musi posiadać konto w systemie. Podopieczny musi być zarejestrowany w grupie bazowej trenera. Logowanie działa prawidłowo.

Scenariusze

- Liczba scenariuszy jest zawsze znacznie większa niż liczba przypadków użycia.
- Średnio skomplikowany system ma ok. kilkudziesięciu przypadków użycia, z których każdy rozwija się nawet do kilkudziesięciu scenariuszy.
- Przykład przypadku użycia: Wybieranie metody wypłaty. Warianty:
  - Wybranie wypłaty w kasie
  - Wybranie wypłaty drogą pocztową
  - Wybranie wypłaty przelewem
  - Pracownik nie znaleziony

Zasady pisania scenariuszy

- Scenariusze pisane językiem naturalnym powinny być w sposób prosty i jednoznaczny. Można stosować gramatykę o składni:
  - podmiot – nazwa aktora lub słowo „System”
  - orzeczenie – nazwa czynności wykonywanej przez podmiot
  - dopełnienie bliższe – nazwa obiektu, który podlega czynności
  - przyimek – „na”, „w”, „z”, itp.
  - dopełnienie dalsze – opcjonalna nazwa innego obiektu

  Przykłady:

  System drukuje raport na podstawie danych ze wskazanego dnia.

  Lekarz wystawia receptę pacjentowi.

Przykład UC33 (1)

Identyfikator i nazwa PU:	UC33: Wypożyczanie książki		
Źródło pochodzenia PU/ Autor:	Regulamin/Bibliotekarz		
Odpowiedzialna osoba:	Nowak		
Aktor główny:	Bibliotekarz	Aktorzy drugorzędni:	Czytelnik
Opis:	Czytelnik wypożycza książkę w bibliotece z klasycznym sposobem komunikowania się z bibliotekarzem za pomocą rewersu.		
Reguły biznesowe:	Jeżeli czytelnik ma przekroczone limity (liczby książek, czasu przetrzymywania), to nie wypożyczy książki.		
Wyzwalacz:	Bibliotekarz otrzymuje od czytelnika wypełniony rewers.		
Warunki początkowe:	Bibliotekarz jest zalogowany do systemu.		
Warunki końcowe:	Książka została wypożyczona.		

Przykład UC33 (2)

Ciąg podstawowy przepływu:	1. Bibliotekarz otrzymuje rewers czytelnika z danymi. 2. Bibliotekarz przekazuje dane czytelnika do systemu. 3. System potwierdza brak przekroczeń limitów. 4. Bibliotekarz wprowadza dane książki dla wypożyczenia. 5. System potwierdza bieżącą możliwość wypożyczenia egzemplarza książki. 6. Bibliotekarz wprowadza fakt wypożyczenia do systemu. 7. System potwierdza rejestrację wypożyczenia.
Ciąg alternatywny przepływu:	3a. System informuje o przekroczeniu limitów. 5a. System informuje o braku możliwości wypożyczenia egzemplarza książki.

Przykład UC33 (3)

Wyjątki:	W1. Brak odpowiedzi ze strony systemu po podaniu danych czytelnika. W2. Brak odpowiedzi ze strony systemu po podaniu danych czytelnika. W3. Brak odpowiedzi ze strony systemu po faktu wypożyczenia.
Priorytet: Krytyczność:	Wysoki Wysoka
Częstotliwość użycia:	Średnia
Dodatkowe informacje:	Czytelnik musi posiadać konto w systemie. Informacje o serwisie i akcjach czytelnika zapisywane są na jego koncie. Logowanie działa prawidłowo. System działa prawidłowo.

Diagram aktywności

- W modelu przypadków użycia diagram aktywności jest stosowany do ilustracji czynności scenariuszowych wykonywanych przez aktorów w celu realizacji zadań zlecanych systemowi.
- Diagram aktywności może być użyty do obrazowania przepływu pracy (workflow), przebiegu sterowania, następstwa zdarzeń, itp..
- Stosowane są diagramy aktywności UML oraz diagramy BPMN.

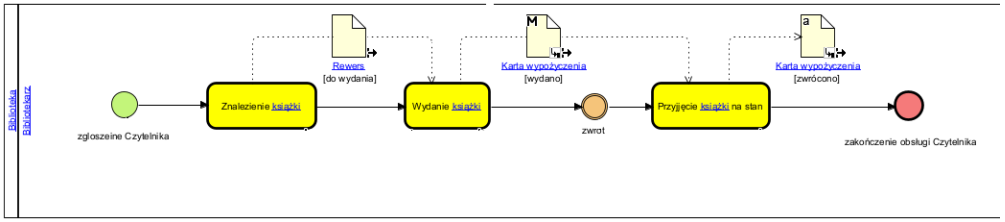


Diagram aktywności UML – przykład (1)

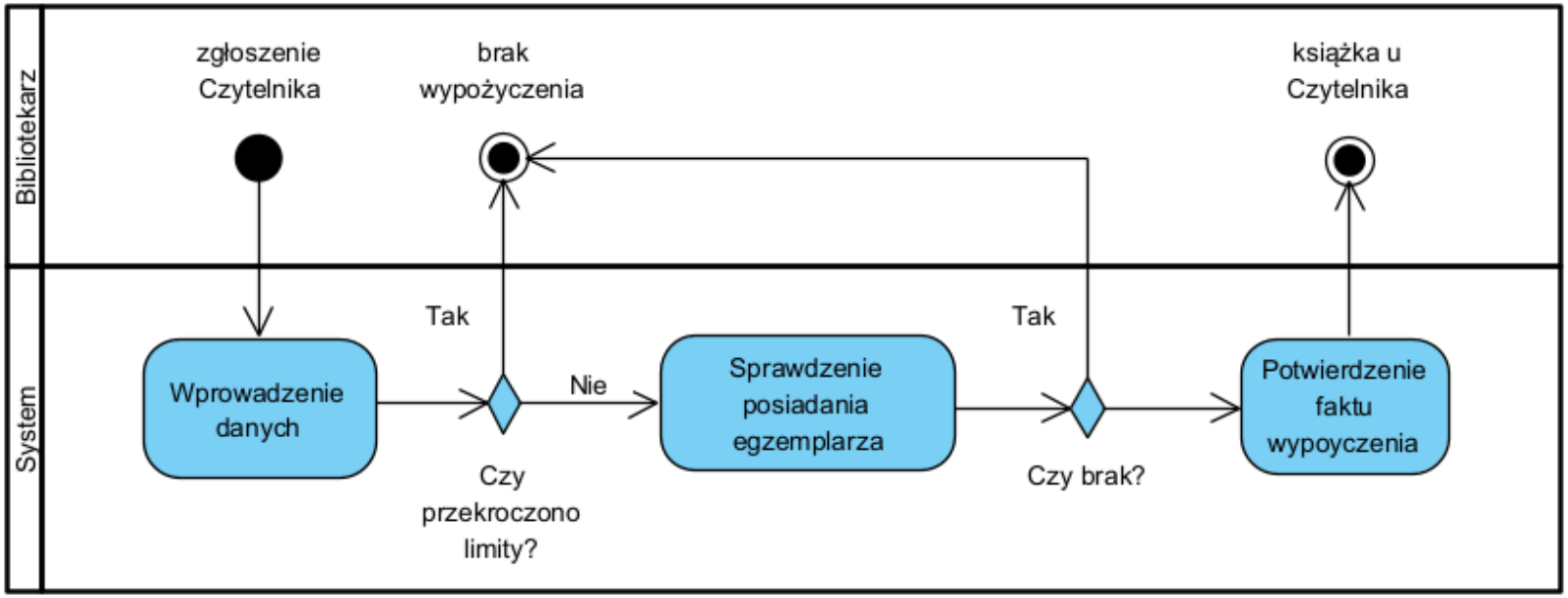
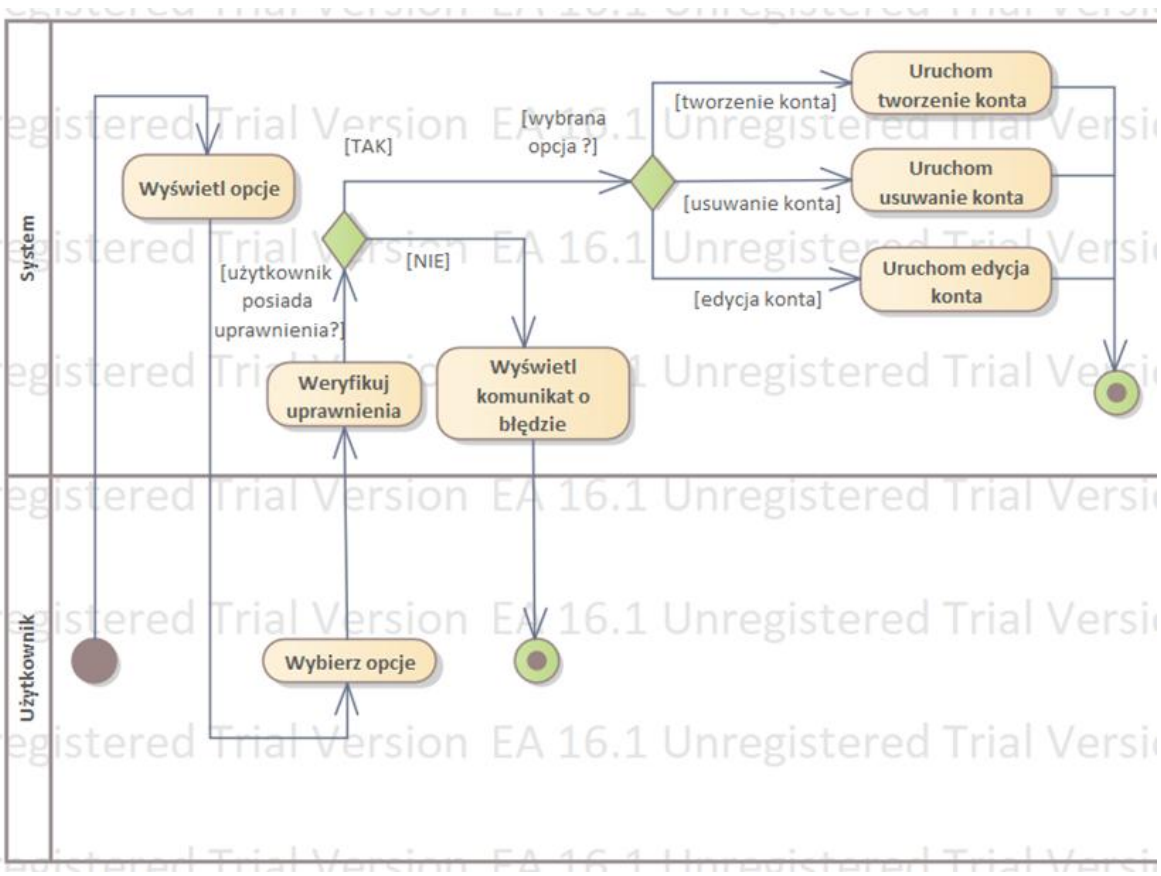
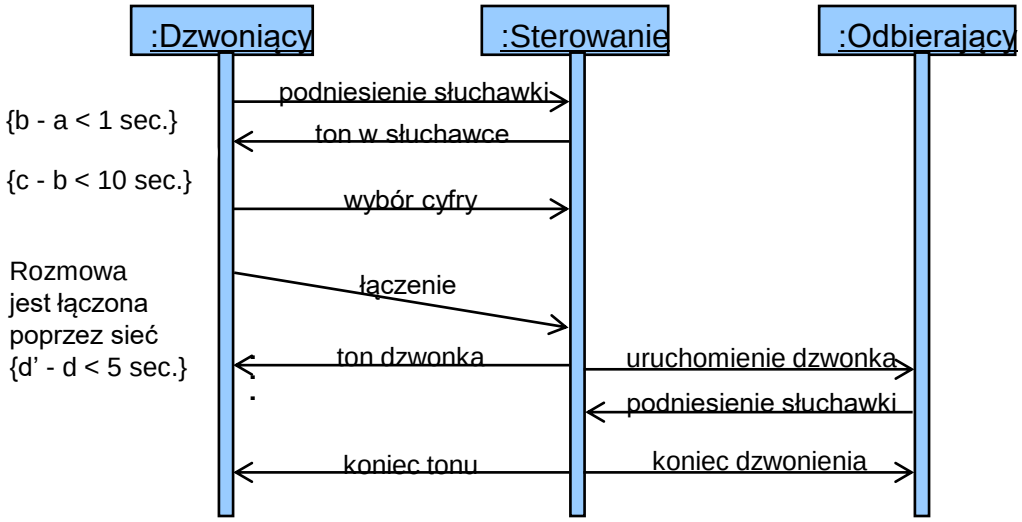


Diagram aktywności UML – przykład (2)



Inne elementy dokumentacji szczegółów przypadków użycia

- Specyfikacja uczestniczących obiektów
- Diagramy interakcji (np.: diagram sekwencji)



Wyodrębnianie obiektów - metoda CRC (Class, Responsibilities, Collaborations)

- Obiekty są instancjami klas mających odpowiedzialność i komunikujących się przez wywoływanie swoich operacji.
- Odpowiedzialność to główna przesłanka istnienia oczekiwanej klasy – idealnie aby była jedna góra dwie odpowiedzialności, a maksymalnie do czterech – to działanie, które obiekt klasy jest zobowiązany wykonać i informacje jakie obiekt posiada, którymi zarządza i które udostępnia.
- Współpracownicy – klasy dopełniające wypełnianie zobowiązań klasy.
- Przykład - na podstawie wymagań i scenariusza pracy biblioteki mamy:
  1. klasa bibliotekarz (odpowiedzialność – A. kontroluje dane wypożyczanych egzemplarzy książek, B. kontroluje stosowane ograniczenia dla wypożyczeń C. kontroluje zwrot egzemplarzy, D. wprowadza nowe zasoby księgozbioru); współpracownicy – egzemplarz, książka
- 2. klasa książka (odpowiedzialność – A. identyfikuje książkę, B. sprawdza zdolność do wypożyczenia egzemplarza); współpracownicy – egzemplarz.

Ocena pracochłonnosci - metoda Use Case Points

- Przypadki użycia jako podstawa szacowania pracochłonności – Gustaw Karner 1993
- Czynniki złożoności środowiska – głównie dotyczą opisu organizacji, tworzącej oprogramowanie
- Czynniki złożoności technicznej – opisują cechy produktu
- Aktorzy
- Przypadki użycia

Use Case Points – 8 czynników środowiskowych (ECF)

Czynniki środowiskowe ECF (*Environmental Complexity Factor*)

- E1 – znajomość projektu (waga 1.5) – jak zespół zna dziedzinę problemu i techniczne aspekty a także jak czuje się w metodyce realizacji projektu
- E2 – doświadczenie zespołu w tworzeniu programowania/aplikacji (waga 0.5)
- E3 - doświadczenie w wykorzystaniu UML i projektowaniu aplikacji typu object-oriented (waga 1.0)
- E4 – umiejętności analityka wiodącego (waga 0.5) – wiedza dziedzinowa i zakres pozyskania wymagań od klienta
- E5 – motywacja (waga 1.0)
- E6 – stabilność wymagań (waga 2.0)
- E7 – pracownicy pracujący w niepełnym wymiarze (waga -1.0)
- E8 – trudności języka programowania (waga -1.0)

Use Case Points - ECF

- $ECF=1.4+(-0.03*\sum wpływ_i*waga_i)$
- $waga_i$ – waga i-tego czynnika E
- $wpływ_i$ – wartość oszacowanego czynnika wpływu w skali 0-5

Use Case Points – 13 czynników technicznych (TCF)

Czynniki techniczne TCF (ang. Technical Complexity Factor)

- T1 – system rozproszony (waga 2.0) – określa czy w systemie wymaga się stosować rozproszone przetwarzanie danych
- T2 – wydajność (waga 1.0) – traktowana w odniesieniu do szybkości przetwarzania i wartości czasu odpowiedzi systemu
- T3 – wydajność dla użytkownika końcowego (waga 1.0) – określa jak ważna dla użytkownika końcowego jest wydajność pracy systemu
- T4 – złożone przetwarzanie wewnętrzne (waga 1.0) – wskazuje czy wymagane jest użycie zaawansowanych algorytmów lub jak skomplikowane są operacje przetwarzania
- T5 – reużywalność (waga 1.0) – czy tworzony kod będzie w przyszłości użyty w podobnych projektach
- T6 – łatwość w instalacji (waga 0.5) – czy jest prosta (np. przez instalator) czy wymaga złożonego środowiska uruchomieniowego i interdyscyplinarnej wiedzy specjalistycznej

Use Case Points – 13 czynników technicznych (TCF)

- T7 – łatwość użycia (waga 0.5) – jak interfejs użytkownika spełnia jego oczekiwania, ergonomia użytkowania i łatwość opanowania
- T8 – przenaszalność (waga 2.0) – czy oprogramowanie ma działać w innych (i jakich) środowiskach
- T9 – łatwość wprowadzania zmian (waga 1.0) – architektura systemu pozwalająca na ewentualną jego rozbudowę, gdy to będzie konieczne
- T10 – współbieżność (waga 1.0) – czy będzie miała zastosowanie
- T11- zabezpieczenia specjalne (waga 1.0) – czy system będzie wymagał specjalnych rozwiązań zabezpieczeń
- T12 – zależność od wewnętrznych bibliotek (waga 1.0) – kompromis między zewnętrznymi bibliotekami (problem utrzymania zgodności oraz rozwoju systemu) a tworzeniem własnych (większa pracochłonność)
- T13 – dodatkowe szkolenia użytkowników (waga 1.0) – czy potrzebne.

Use Case Points - TCF

- $TCF=0.6+(0.01*\sum wpływ_i*waga_i)$
- $waga_i$ – waga i-tego czynnika T
- $wpływ_i$ – wartość oszacowanego czynnika wpływu w skali 0-5

Use Case Points – złożoność aktorów i UAW

Złożoność aktorów wg typów:

- **prosty** (współczynnik **1**) – aktor komunikuje się przez API
- **średni** (współczynnik **2**) – aktor komunikuje się z systemem przez protokół (np. http) lub stanowi źródło danych (baza danych)
- **złożony** (współczynnik **3**) – aktor komunikuje się z systemem przez GUI

Niedostosowana waga aktorów UAW (ang. Unadjusted Actors Weight)

$$UAW = \sum n_i * c_i$$

n_i – liczność i-tego zbioru aktorów

c_i – współczynnik złożoności i-tego zbioru

Use Case Points – wagi przypadku użycia i UUCW

Złożoność przypadku użycia mierzona jest liczbą nierozłącznych akcji (kroków-transakcji) koniecznych do jego realizacji

Zależnie od liczby kroków określamy typ złożoności:

- **prosty** (współczynnik **5**) – 3 lub mniej kroków
- **średni** (współczynnik **10**) – od 4 do 7 kroków
- **złożony** (współczynnik **15**) – 7 lub więcej kroków

Niedostosowana waga przypadków użycia UAW (ang. Unadjusted Use Cases Weight)

$$UUCW = \sum n_i * c_i$$

n_i – liczność i-tego zbioru złożoności przypadków użycia

c_i – współczynnik złożoności i-tego zbioru

Punkty przypadków użycia

Niedostowane punkty przypadków użycia:
UUCP (*Unadjusted Use Case Points*)

$$UUCP = UAW + UUCW$$

Punkty przypadków użycia UCP

$$\mathbf{UCP = UUCP*TCF*ECF}$$

Oszacowanie pracochłonnośći w osobogodzinach

- Pracochłonność = UCP*PF
- Stosuje się przelicznik produktywności PF (*Productivity factor*)
- Średnio liczy się 1PF=20 osobogodzin

Przykład (1)

- Podsystem interfejsu użytkownika – zakładamy hipotetycznie 16 różnych złożonych przypadków użycia w zakresie ergonomii, responsywności, nawigacji itp. w różnych fazach pracy systemu.
- Podsystem obliczeń inżynierskich (2D i 3D) – zakładamy hipotetycznie 14 średnich i 8 prostych przypadków użycia w zakresie obliczeń o przebiegu sekwencyjnym.
- Podsystem obsługi urządzeń peryferyjnych – zakładamy 8 prostych przypadków użycia np. wydruk na ploterze.

UUCW = 16PU*15+14PU*10+16PU*5= 460

Przykład (2)

- Aktorzy komunikujący się przez GUI: administrator, konstruktor, kreślarz, analityk – **4 aktorów**.
- Aktorzy komunikujący się przez protokół sieciowy: **interakcyjne moduły zewnętrzne** i **urządzenia peryferyjne** konieczne do pracy systemu - **12 aktorów**.
- Aktorzy komunikujący się przez zdefiniowane API: moduły wsparcia systemu - **8 aktorów**.

UAW=4*3+12*2+8*1=44

Przykład (3)

- E1=5; E2=5; E3=5; E4=5; E5=5; E6=5
- E7=0; E8=0
- ECF=1.4-0.03*32.5=0.425
- T1=2; T2=5; T3=5; T4=3; T5=3; T6=5; T7=5; T8=0; T9=5; T10=5; T11=2; T12=5; T13=0
- TCF=0.6+0.01*42=1.02
- UCP=(44+460)*0.425*1.02=218.48
- UCP*20=4369,68 osobogodz=~546 ord