

Modelowanie i analiza obiektowa

Wykład 3

Analiza przypadków użycia

Wykład 4

Projektowanie klas

Plan wykładu

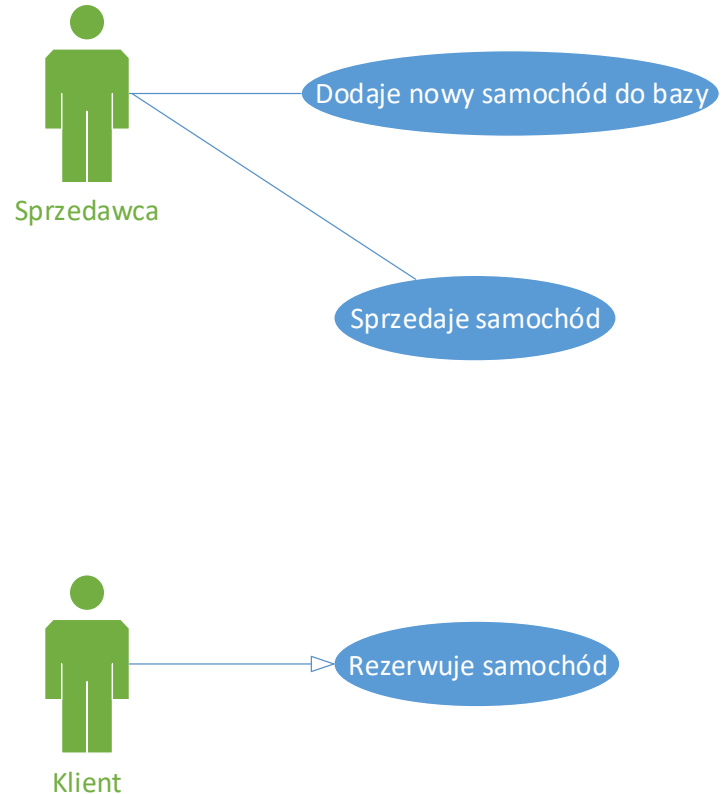
- **Realizacja przypadku użycia**
- **Realizacja przypadku użycia w fazie analizy**
 - **Klasy analizy**
 - **Analiza wymiany informacji na diagramach interakcji**
 - **Faza analizy a wymagania нефunkcjonalne**

Składniki P.U.



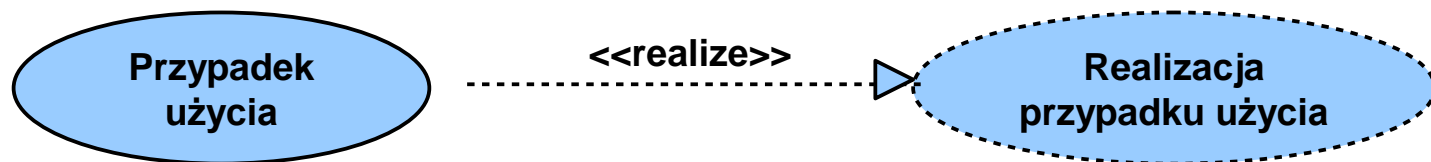
Aktor 1

- nie jest częścią systemu;
- reprezentuje rolę, w którą może wcielić się użytkownik;
- może reprezentować człowieka, urządzenie bądź inny system;
- może aktywnie wymieniać informacje z systemem;
- może dostarczać informacje;



Realizacja przypadku użycia

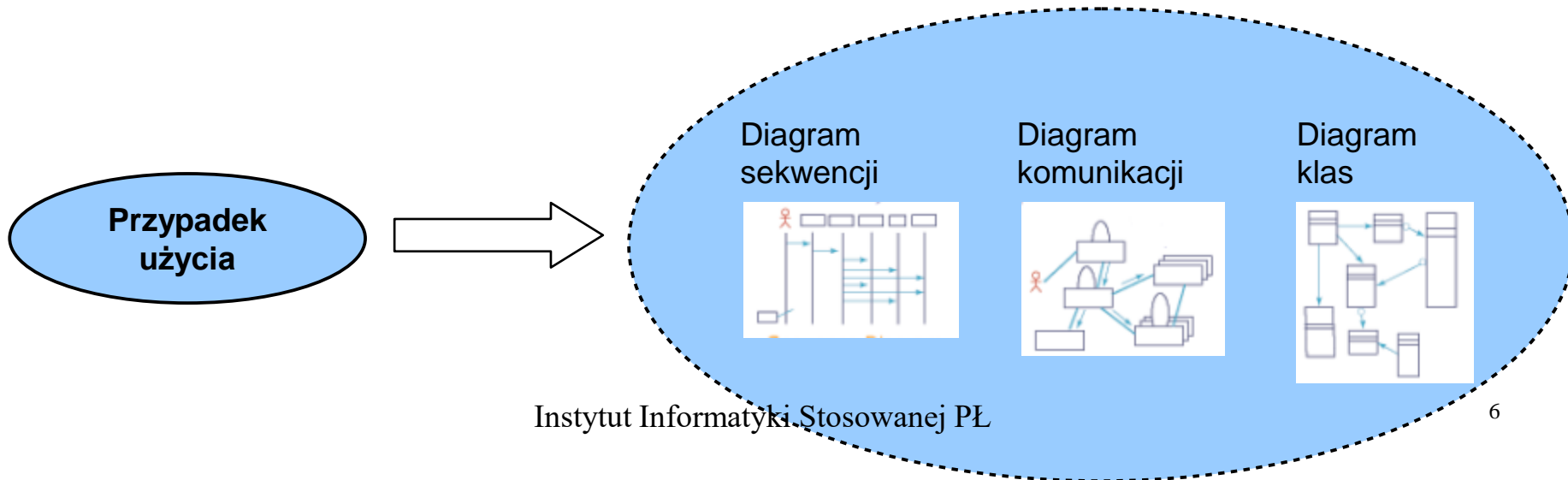
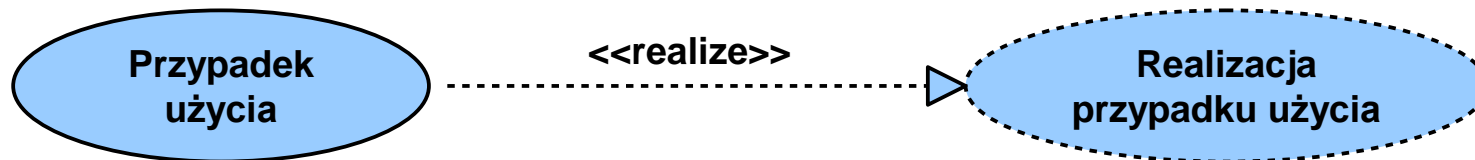
- Opis sposobu, w jaki konkretny przypadek użycia jest realizowany w modelu projektowym.
- Określa powiązanie pomiędzy modelem przypadków użycia a modelem projektowym.
- Określa, jakie klasy muszą zostać zbudowane, aby zaimplementować dany przypadek użycia.



Elementy realizacji przypadku użycia

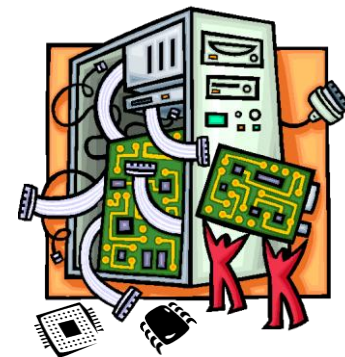
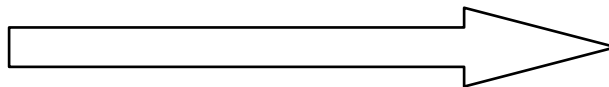
Model przypadków użycia

Model projektowy



Analiza przypadku użycia

- Jest to pierwszy etap realizacji przypadku użycia
- Określa, jak zewnętrzne zachowanie systemu przekłada się na współdziałanie jego elementów logicznych.
- Elementami logicznymi są klasy analizy



Kroki analizy przypadków użycia

1: Dla każdej realizacji Przypadku Użycia

1.1: Znalezienie klas na podstawie analizy zachowania

1.2: Przydzielenie klasom odpowiedzialności

2: Określanie własności klas analizy

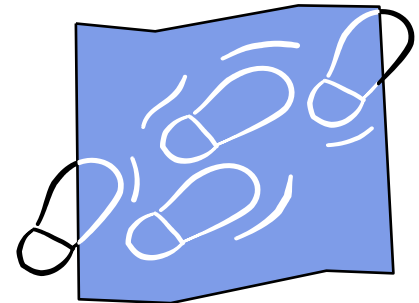
2.1: Opisanie odpowiedzialności

2.2: Opisanie atrybutów i asocjacji

2.3: Przypisanie mechanizmu analizy

Krok 3: Unifikacja klas

Krok 4: Kontrola



Krok 1.1: Wyszukiwanie klas analizy dla każdej realizacji przypadku użycia

Gdzie szukać kandydatów na klasy?

Specyfikacja przypadku użycia

Słownik pojęć

Dokument wymagań klientów



Twórczy proces, który w pewnych sytuacjach powinien odbywać się w obecności ekspertów z dziedziny problemowej

Klasy powinny odzwierciedlać modelowaną dziedzinę biznesową oraz słownik systemu używany do projektowania

Nazwa klasy

- Powinna pochodzić z dziedziny problemowej i odzwierciedlać to co reprezentuje (Student, Wykładowca, Kurs)
- Nazwa powinna być jednoznaczna i unikatowa (rzeczownik – osoba, miejsce, rzecz)
- Potencjalne nazwy klas znajdują się w dokumentach wymagań

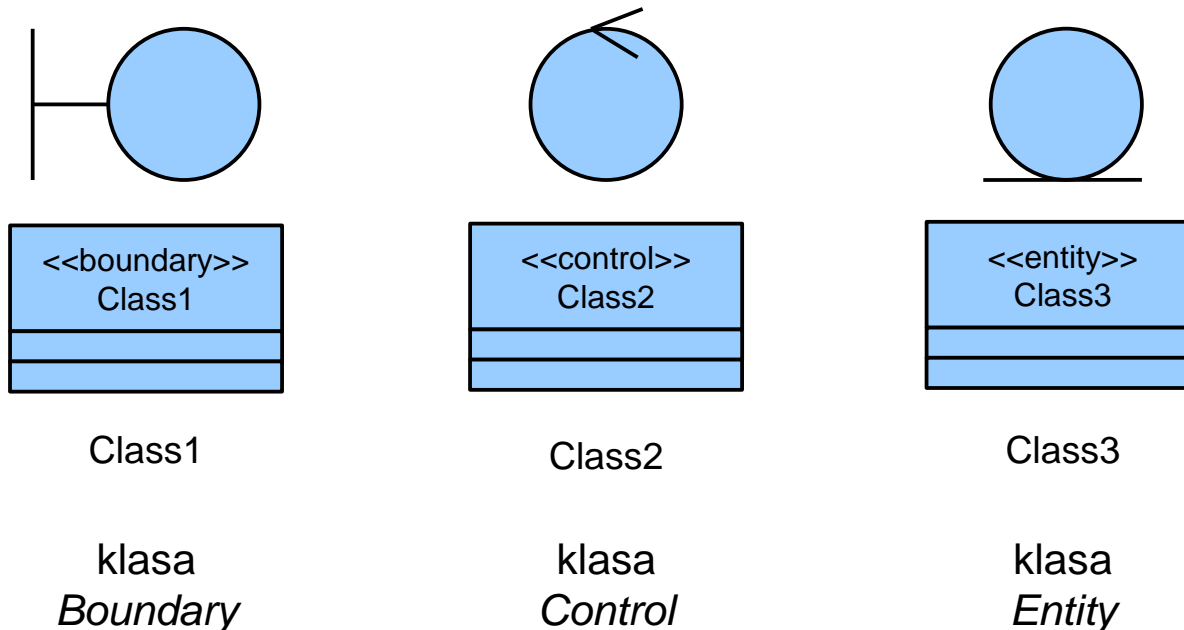


Czym są klasy analizy

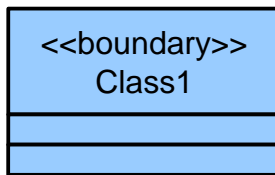
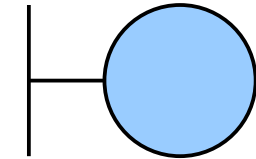
- Klasy analizy reprezentują koncepcyjny model elementów systemu, które posiadają pewne zachowanie i zadania. Jest to pierwszy szkic modelu systemu.
- Klasy analizy reprezentują trzy aspekty systemu, które często podlegają zmianom:
 - Granice pomiędzy systemem i aktorami,
 - Dane, których system używa.
 - Sposób sprawowania kontroli na systemem (przetwarzaniem)
- Klasy analizy są swego rodzaju prototypami klas, część z nich stanie się podsystemami, komponentami, klasami, inne zostaną odrzucone w dalszym procesie projektowym.

Klasy analizy

Modelowanie zachowania przypadku użycia w fazie analizy podzielone jest pomiędzy trzy typy klas:

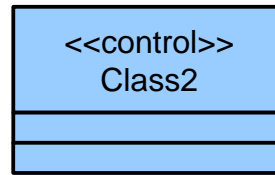
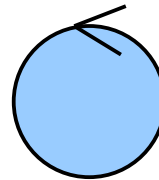
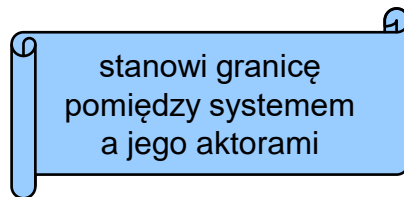


Klasy analizy



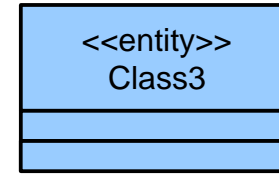
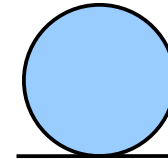
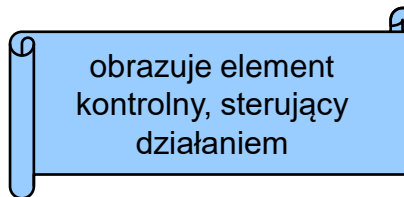
Class1

klasa
Boundary



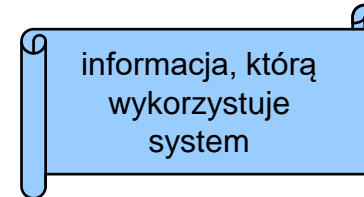
Class2

klasa
Control



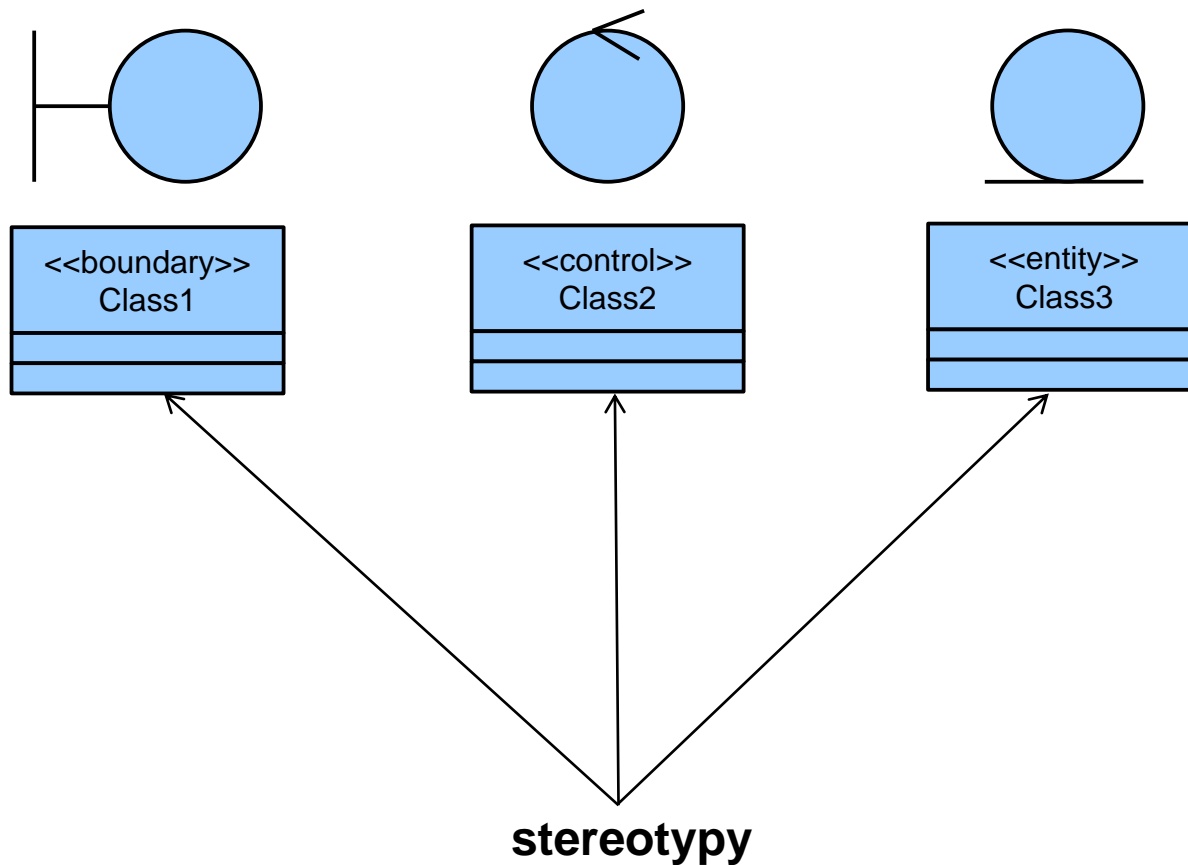
Class3

klasa
Entity



Klasy analizy

Reprezentacje graficzne



Przypadek użycia a klasy analizy

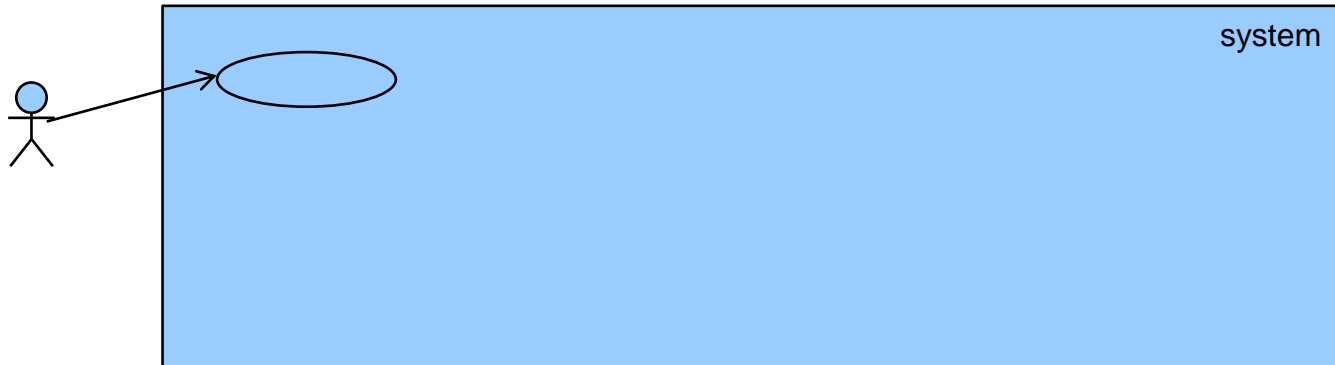


Diagram przypadków użycia

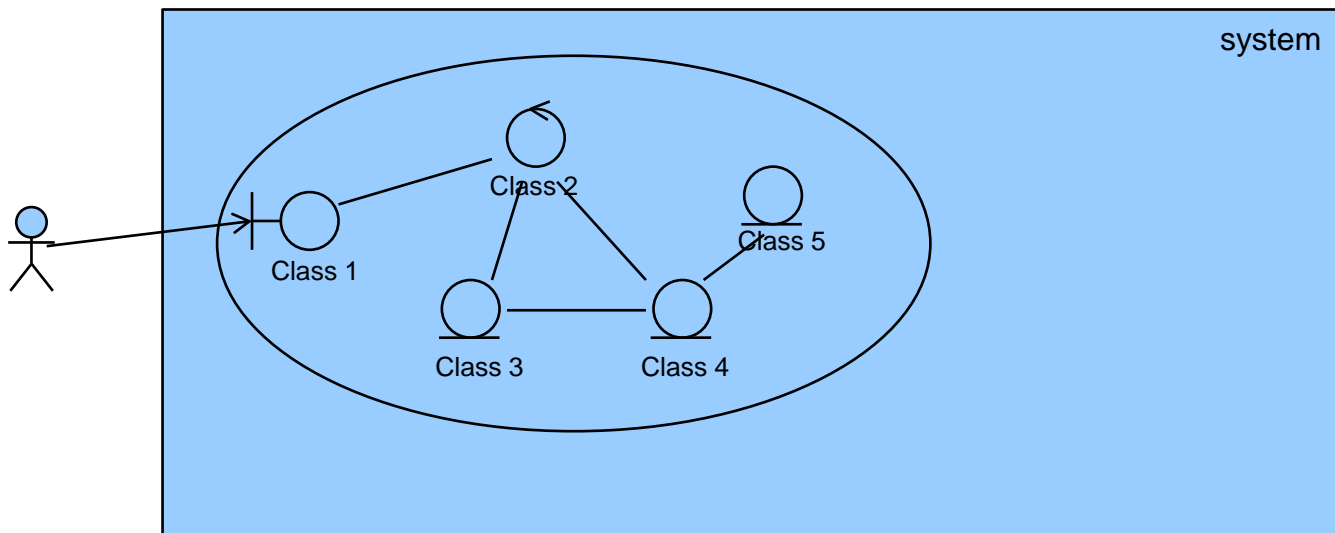
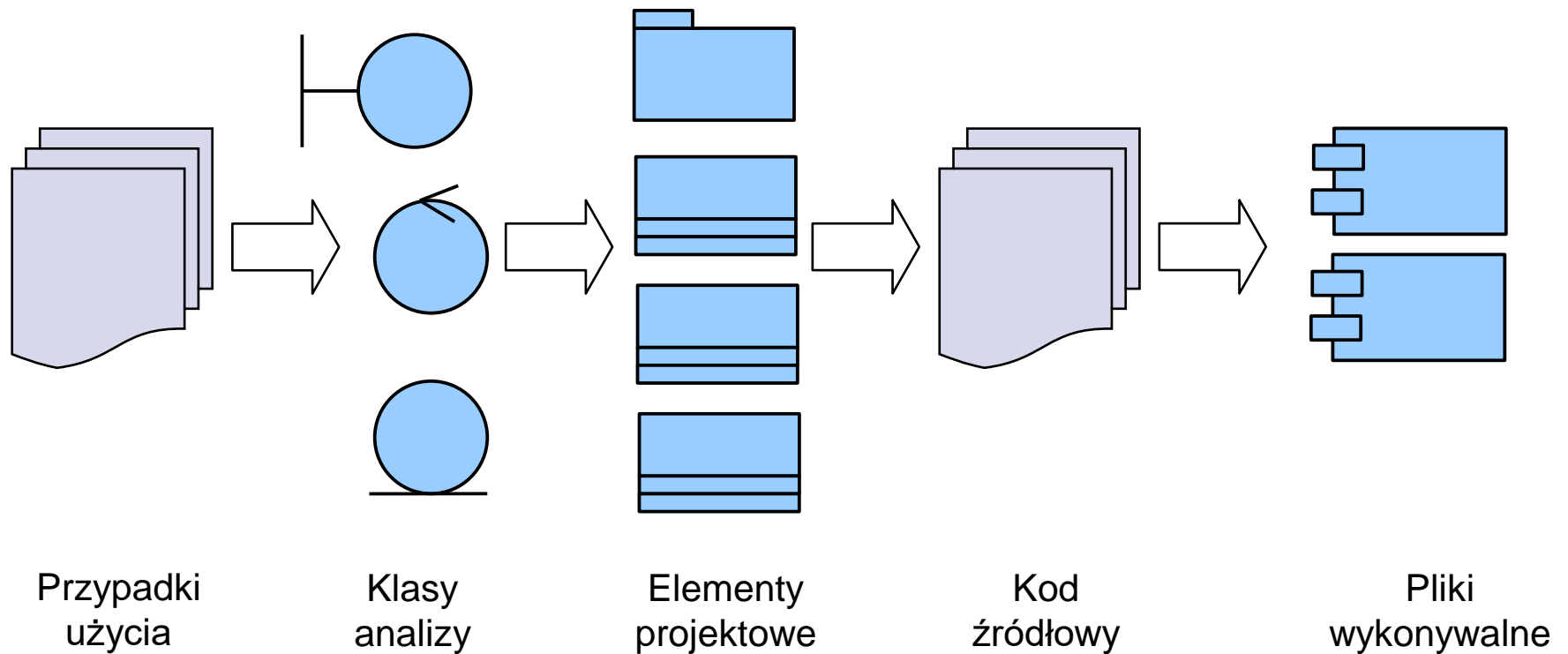
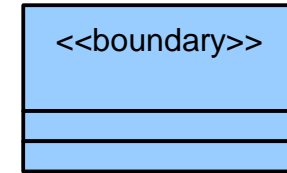
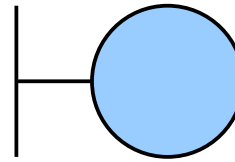


Diagram realizacji przypadku użycia przy pomocy klas analizy

Czym są klasy analizy?

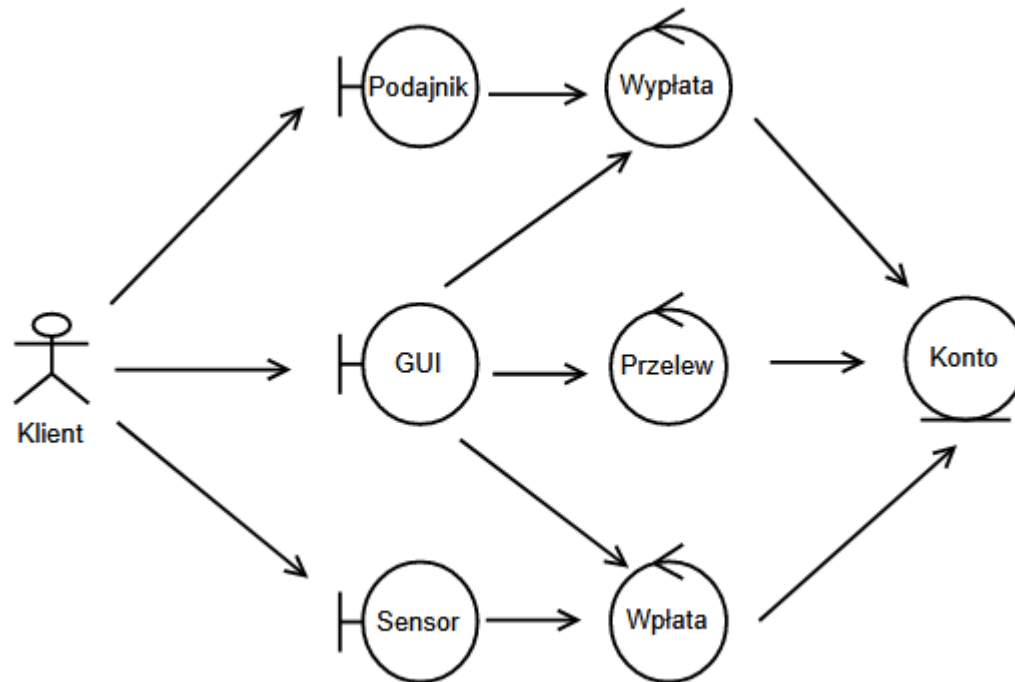


Klasa **Boundary**



- Różne typy (Zależne od środowiska)
 - Klasy interfejsu użytkownika
 - Klasy interfejsu z innym systemem
 - Klasy interfejsu sprzętowego
- Założenie
 - Jedna klasa **Boundary** dla pary:
Aktor – Przypadek użycia.

Przykład BCE dla bankomatu



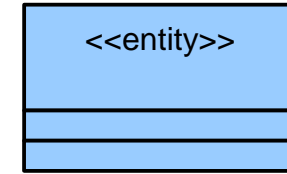
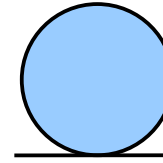
Klasa **Boundary** – cechy

- Modeluje interakcje pomiędzy otoczeniem a systemem.
- Modeluje te części systemu, które są zależne od otoczenia
- Izolują wnętrze systemu od wpływu środowiska (i vice versa)

Odkrywanie klas Boundary - wskazówki

- Klasy interfejsu użytkownika
- Skup się na tym jak informacja jest prezentowana użytkownikowi (nie zwracaj uwagi na szczegóły)
- Klasy interfejsu do innego systemu lub interfejsu urządzenia
- Jaki protokół komunikacji musi być zdefiniowany (nie koncentruj się na tym w jaki sposób go zaimplementować)

Klasa Entity



- Logiczna struktura danych.
- Reprezentuje i zarządza informacjami przechowywanymi przez system.
- Zazwyczaj trwała
- Niezależna od środowiska zewnętrznego systemu (aktora)
- Zazwyczaj nie odpowiada tylko jednej realizacji przypadku użycia.



Klasa Entity

Źródła, w których należy wyszukiwać klasy Entity:

- Słownik
- Model dziedziny biznesowej
- Przepływ zdarzeń przypadku użycia
- Podstawowe abstrakcje



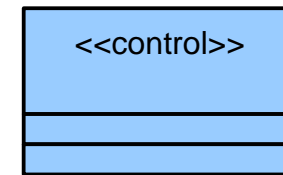
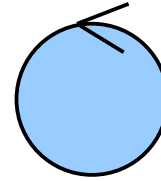
Klasa Entity

Wyszukiwanie klasy Entity:

- Wyszukanie rzeczowników (fraz rzeczownikowych) w przepływie zdarzeń (specyfikacja przypadku użycia).
- Filtrowanie listy (eliminacja rzeczowników oznaczających to samo, niejednoznacznych, zbyt ogólnych, usunięcie aktorów, atrybutów, operacji)



Klasa Control

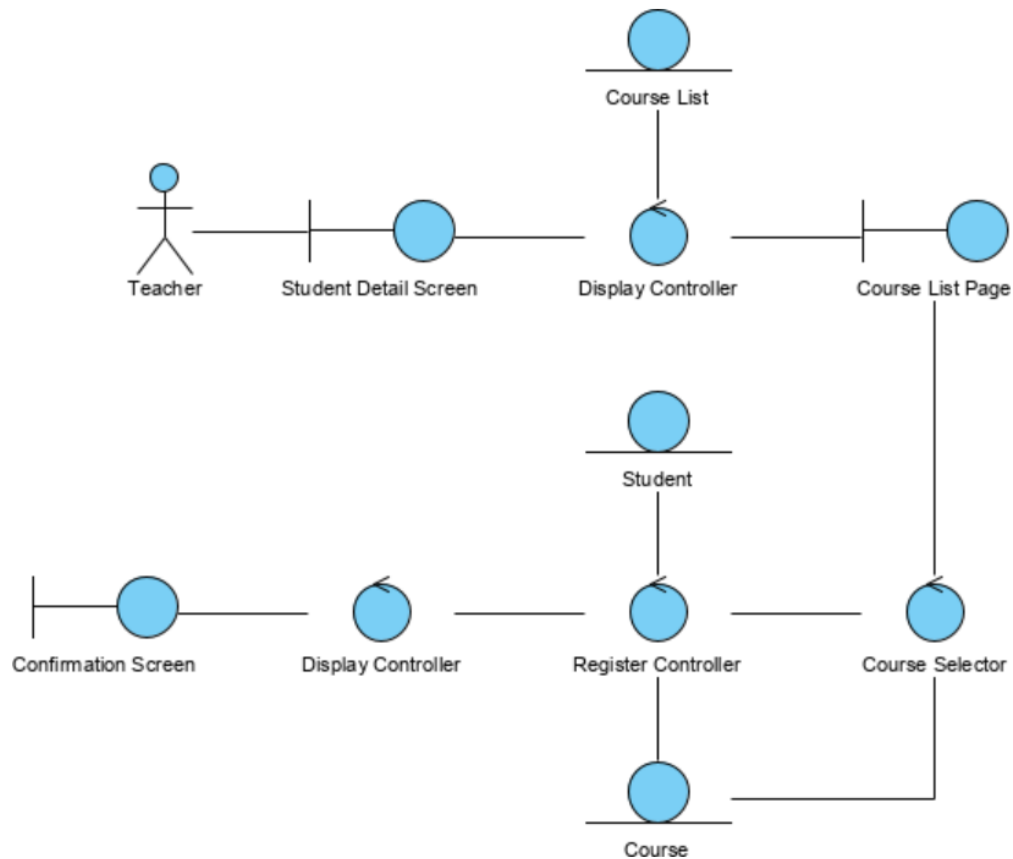


- Koordynator przypadku użycia
- Definiuje logikę działania (kolejność przesyłania komunikatów) mechanizm obsługi błędów czy transakcji dla przypadku użycia
- Rozdziela klasy Boundary i Entity, dzięki czemu łatwiej będzie w przyszłości wprowadzić zmiany
- Może być jedna klasa Control na Przypadek Użycia
- Zależna od Przypadku Użycia, niezależna od środowiska zewnętrznego
- Zazwyczaj ma kilka stanów zależnych od miejsca w przepływie zdarzeń.



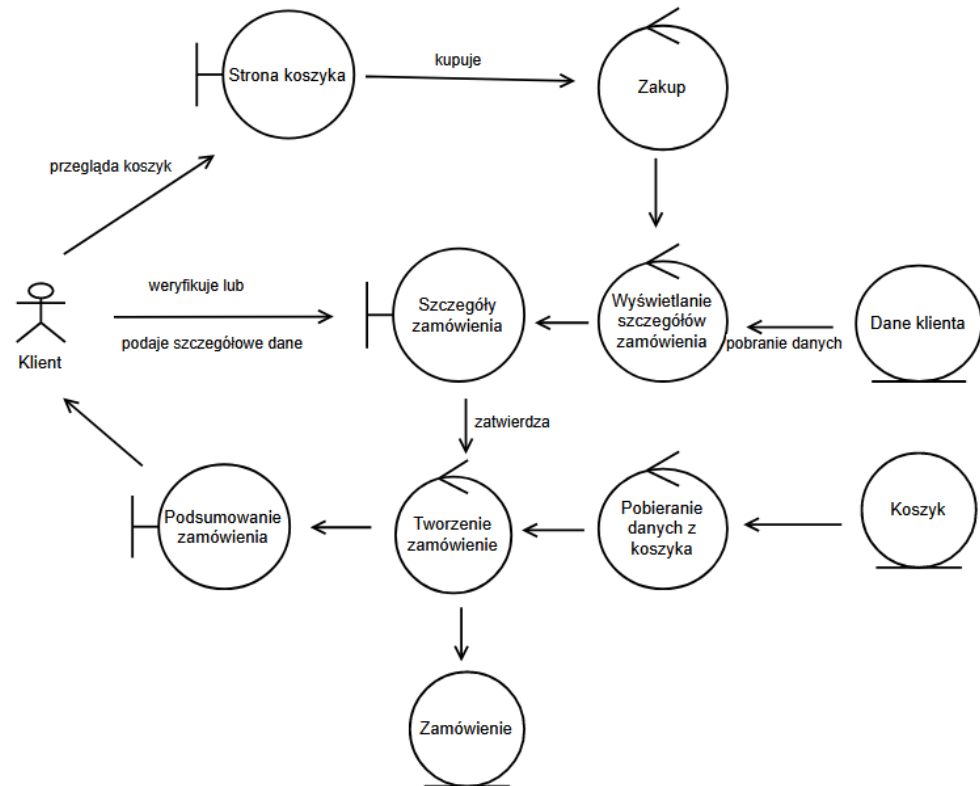
Przykład

Źródło: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/robustness-analysis-tutorial/>



Przykład diagramu Robustness

1. Klient przegląda zawartość koszyka.
2. Klient kupuje i decyduje się realizować zakup.
3. System wyświetla formularz Szczegóły zamówienia, w którym należy podać dane wymagane do realizacji zamówienia (imię i nazwisko, numer telefonu, adres e-mail, adres do wysyłki i adres rozliczeniowy, metoda płatności).
- 3.a System pobierze informacje o kliencie w celu wypełnienia listy wartości domyślnych.
4. Klient wypełnia dane i przesyła formularz szczegółów zamówienia, po zatwierdzeniu.
5. System rejestruje zamówienie.
- 5.a System pobiera istniejące elementy koszyka, aby dodać je do zamówienia.
6. System wyświetla stronę „Utworzone zamówienie”.



Kroki analizy przypadków użycia

1: Dla każdej realizacji Przypadku Użycia

1.1: Znalezienie klas na podstawie analizy działania systemu

1.2: Przydzielenie klasom odpowiedzialności

2: Określanie własności klas analizy

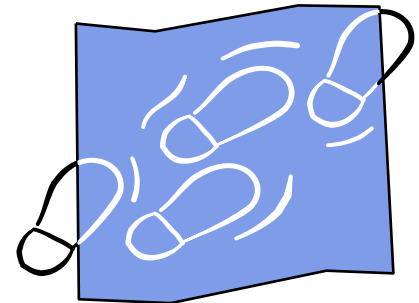
2.1: Opisanie odpowiedzialności

2.2: Opisanie atrybutów i asocjacji

2.3: Przypisanie mechanizmu analizy

Krok 3: Unifikacja klas

Krok 4: Kontrola



Przydzielenie klasom odpowiedzialności - wskazówki

- Klasy Boundary
 - zachowanie, z którym związana jest komunikacja z aktorem
- Klasy Entity
 - Wymiana danych, przechowywanych w systemie
- Klasy Control
 - Zachowanie specyficzne dla Przypadku Użycia

Nie należy modelować interakcji pomiędzy aktorami, są oni poza granicami systemu.

Diagramy interakcji

Diagramy interakcji, stanowiące jeden z rodzajów diagramów dynamicznych, pozwalają na utworzenie opisu interakcji obiektów systemu podczas realizacji danego zadania: Przypadku Użycia czy też jednego konkretnego ciągu zdarzeń dla danego Przypadku Użycia

UML posiada dwa rodzaje diagramów interakcji

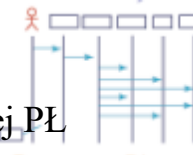
diagramy komunikacji (*communication diagrams*, w UML 1.x *collaboration diagrams*)

Diagram komunikacji



diagramy sekwencji (ang. *sequence diagrams*)

Diagram sekwencji



Diagramy interakcji

- Konstruowanie diagramów interakcji nie musi być stosowane dla wszystkich Przypadków Użycia. Mogą się one okazać szczególnie użyteczne do:
 - komunikacji wewnątrz zespołu projektowego (jak zresztą wszystkie rodzaje diagramów),
 - rozważenia opcjonalnych realizacji w “trudnych przypadkach”.
- Ponadto, niektóre narzędzia CASE potrafią wykorzystać te diagramy do generacji kodu, co może stanowić ważny powód dla ich konstruowania.

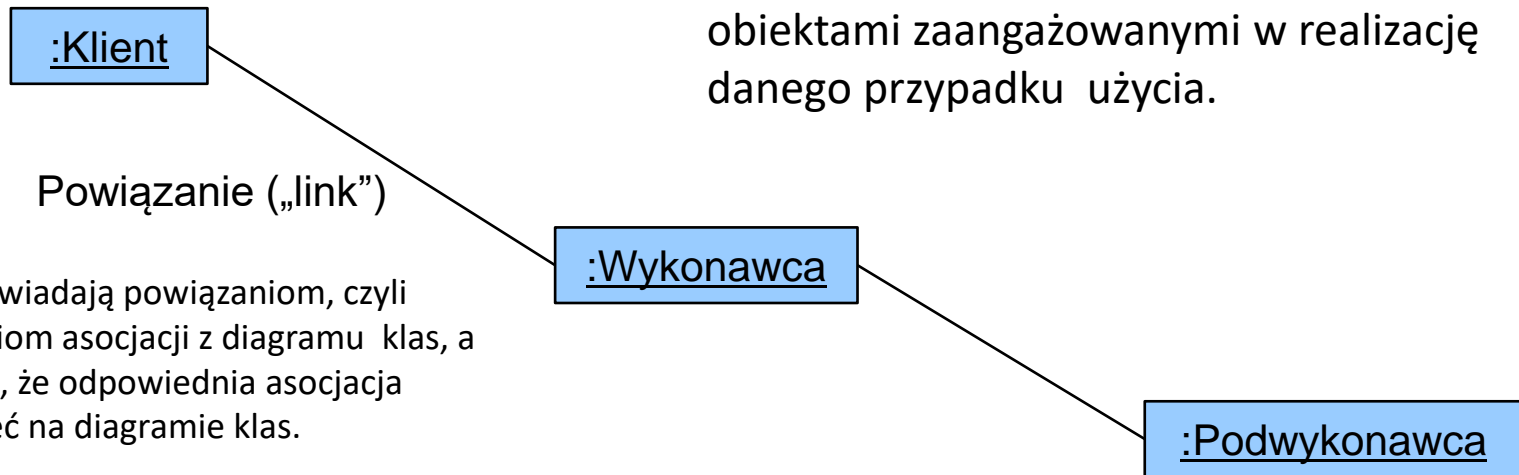
Diagramy interakcji

- Oba rodzaje diagramów, bazując na danym diagramie klas, pokazują prawie tą samą informację, ale w nieco inny sposób.
- Decyzja, który rodzaj diagramów konstruować, zależy od pożądanego aspektu interakcji.

Diagram komunikacji (współpracy)*

Diagramy komunikacji pokazują w jaki sposób system realizuje dany przypadek użycia i jakie elementy (**obiekty**) są zaangażowane w tę realizację.

Diagramy komunikacji mogą dodatkowo pokazywać interakcje zachodzące między obiektami zaangażowanymi w realizację danego przypadku użycia.



Linki odpowiadają powiązaniom, czyli wystąpieniom asocjacji z diagramu klas, a to oznacza, że odpowiednia asocjacja musi istnieć na diagramie klas.

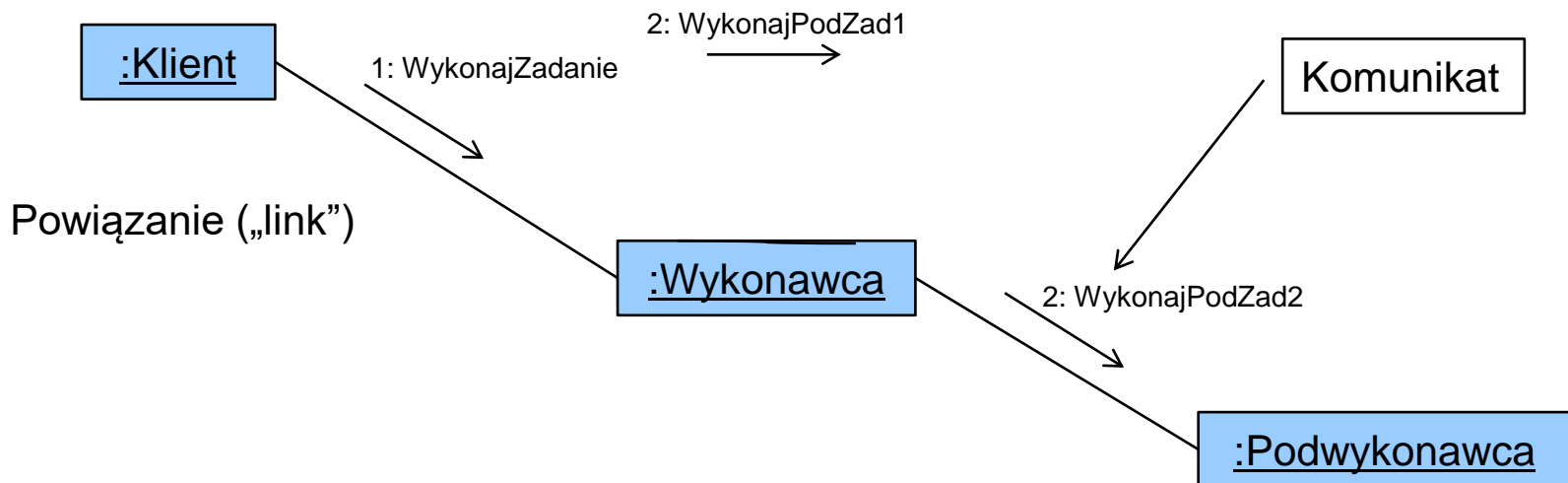
*) w UML 1.x znany jako diagram współpracy (kolaboracji).

Diagram komunikacji

- Prosty diagram komunikacji, bez uwidaczniania interakcji między obiektami, stanowi coś w rodzaju “wystąpienia fragmentu diagramu klas” – pokazuje aktora, znaczące obiekty i powiązania między nimi.
- Możliwe jest pokazanie więcej niż jednego obiektu danej klasy.

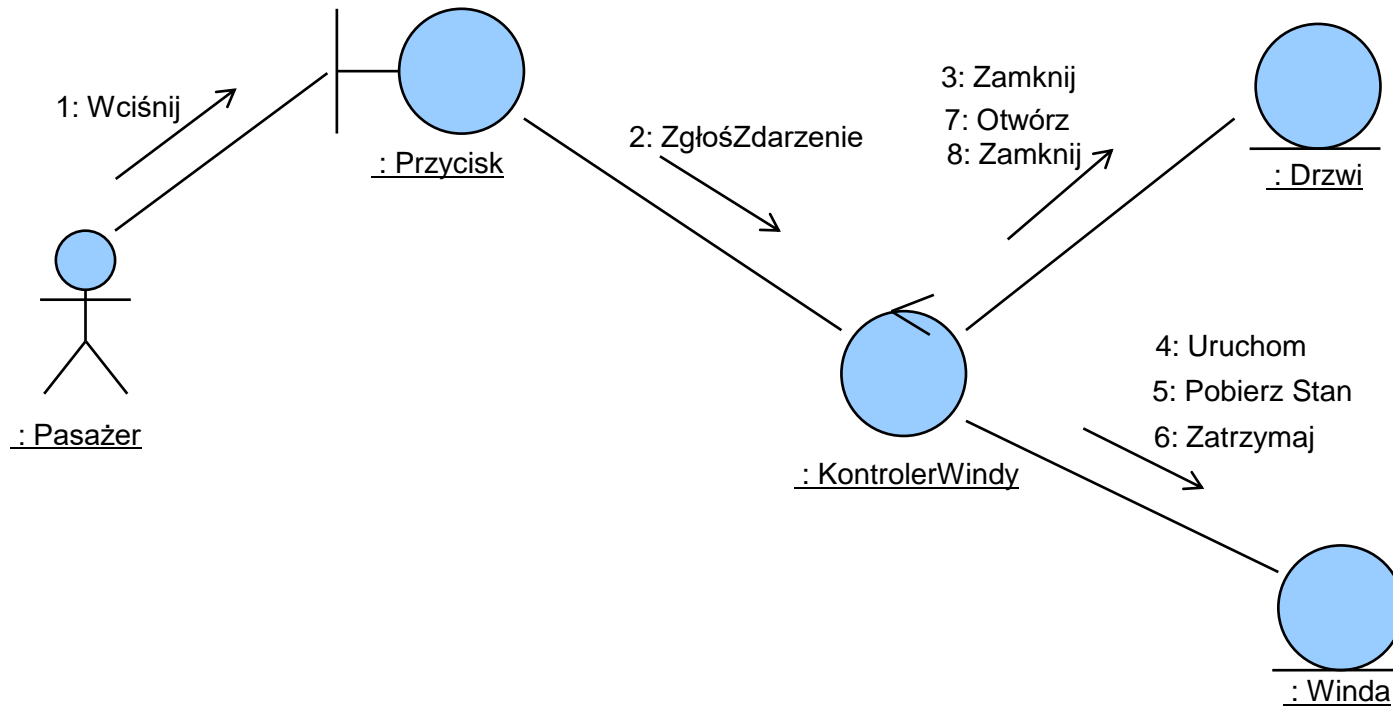
Interakcja na diagramach komunikacji

Diagramy komunikacji mogą dodatkowo pokazywać interakcje zachodzące między obiektami zaangażowanymi w realizację danego **przypadku użycia**. Sekwencja interakcji oznacza tu sekwencję komunikatów przesyłanych między współpracującymi obiektami.



Komunikaty przedstawiane są tu w postaci etykiet strzałek rysowanych wzdłuż linków między współpracującymi obiektami.

Diagram komunikacji - winda



Rodzaje interakcji

Sekwencyjna:

- Tylko jeden aktor może zainicjować sekwencję komunikatów i w danym momencie tylko jeden obiekt może “działać”.
- Obiekt rozpoczyna tzw. “aktywne życie” (*live activation*) w momencie otrzymania komunikatu.
- Może wysłać odpowiedź do nadawcy komunikatu.
- W międzyczasie może prowadzić obliczenia czy też wysyłać komunikaty do innych obiektów.
- Wysyłając komunikat do innego obiektu nadal pozostaje w aktywnym stanie, ale jego własna działalność zostaje zawieszona do czasu otrzymania odpowiedzi na wysłany komunikat – wysyłanie komunikatu związane jest tu z przekazywaniem sterowania do odbiorcy komunikatu.
- W każdym momencie istnieje w systemie stos aktywnych obiektów:
 - na szczycie stosu znajduje się ten obiekt, który aktualnie “działa”,
 - wysłanie odpowiedzi na komunikat powoduje zdjęcie obiektu ze stosu.

Rodzaje interakcji

Współbieżna:

- Wolna od ograniczeń interakcji sekwencyjnej.
- Pozwala na aktywne działanie wielu obiektów, które mogą wysyłać wiele komunikatów jednocześnie.
- Takie asynchroniczne działanie jest wykorzystywane w systemach wielowątkowych (np. rozproszonych).

Diagramy sekwencji

- Diagramy sekwencji **nie pokazują** powiązań między współpracującymi obiektami, ale można to wydedukować w oparciu o zaznaczone komunikaty.
- Diagramy sekwencji, wyraźniej niż diagramy komunikacji, pokazują przekazywanie sterowania.
- Istotnym elementem opisu jest czas, który chronologicznie segreguje komunikaty.

Diagramy sekwencji

Oznaczenia (notacja) interakcji:

Synchroniczna (*synchronous*),



Nadawca zawiesza działanie oczekując na odpowiedź odbiorcy

Asynchroniczna (*asynchronous*),



Nadawca nie oczekuje na odpowiedź, pracuje dalej (w tym wysyła inne komunikaty)

Jednostronna (*flat*),



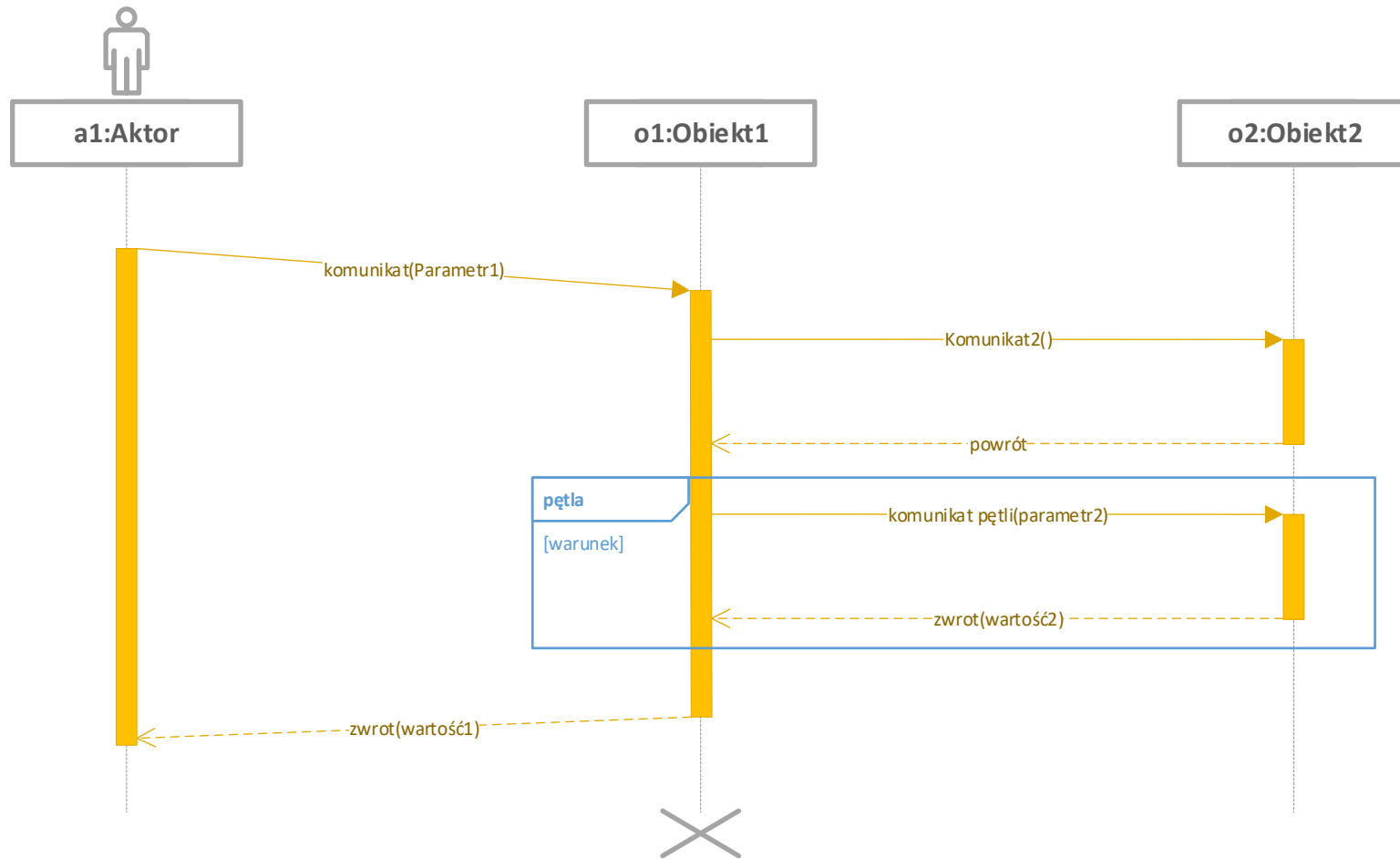
Nadawca wysyła komunikat do odbiorcy i kończy działanie nie oczekując na odpowiedź

Powrót (*return*).

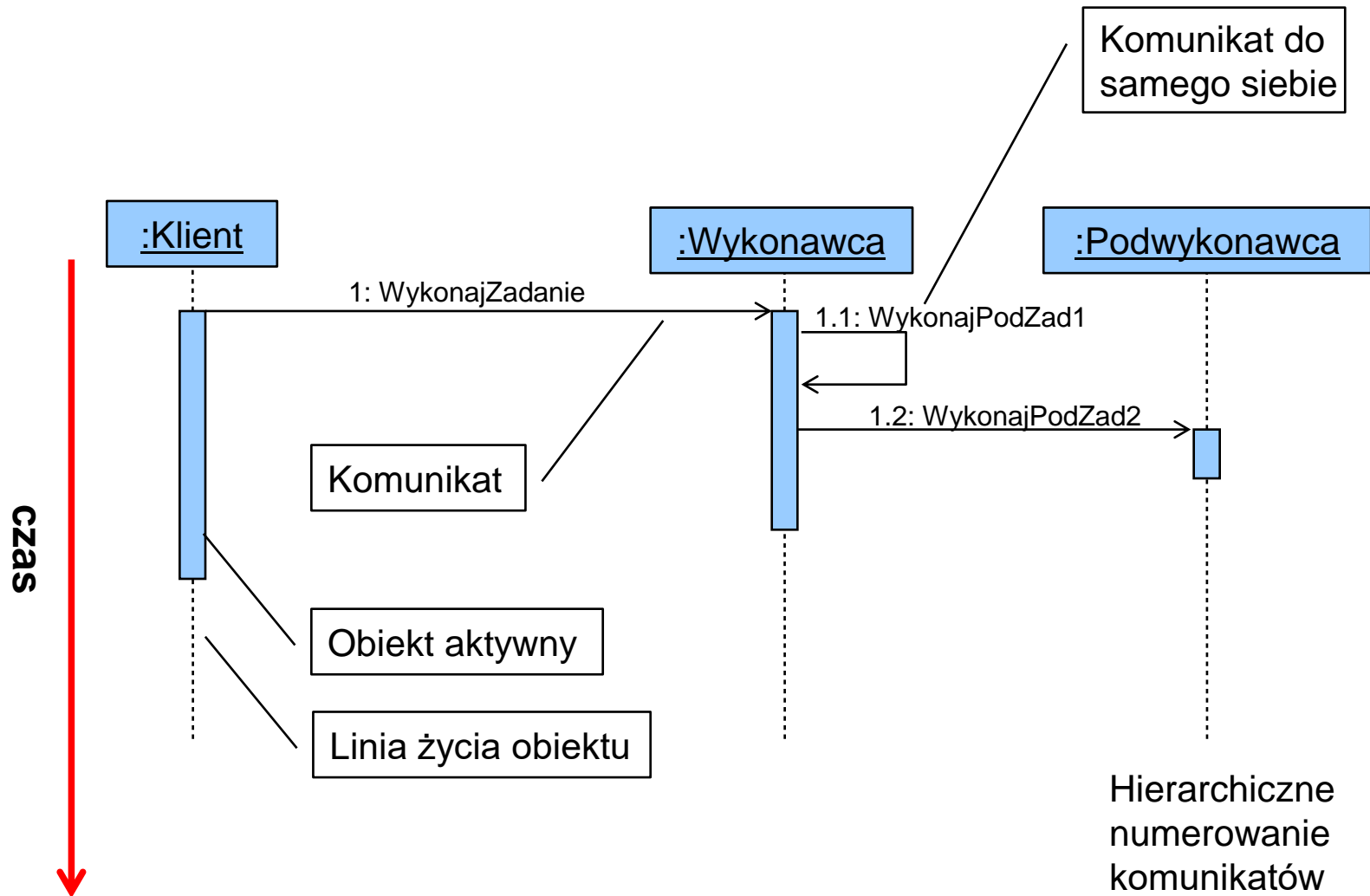


Zakończenie komunikatu i przekazanie sterowania do odbiorcy.
Powrót nie jest komunikatem.

Diagramy sekwencji



Diagramy sekwencji



Nakładanie ograniczeń na przepływ czasu

Główna przewaga diagramów sekwencji nad diagramami współpracy przejawia się w ich zdolności do graficznego prezentowania przepływu czasu, a nawet do podawania ograniczeń czasowych. Taka możliwość może mieć duże znaczenie dla opisu systemów czasu rzeczywistego.

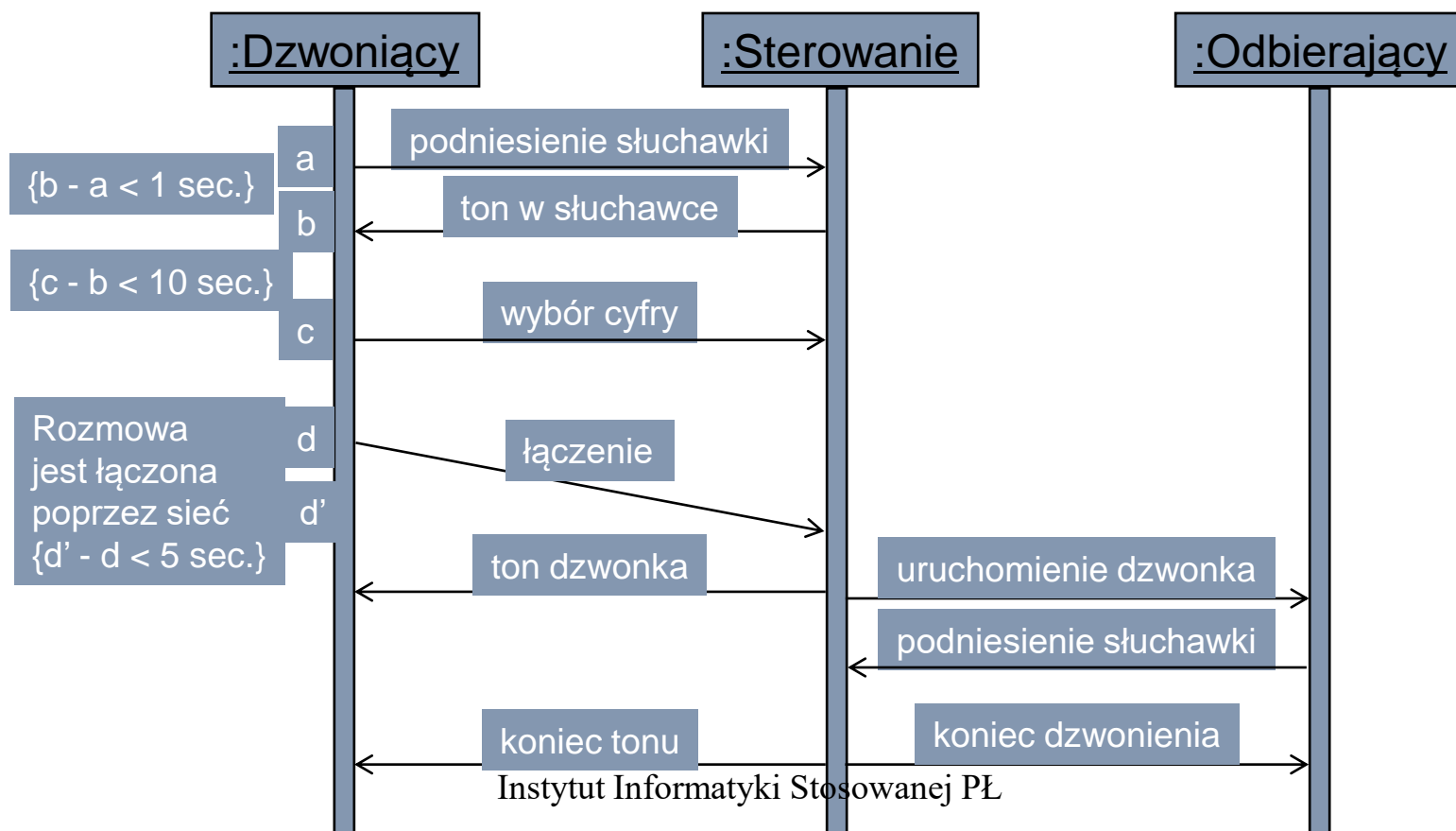
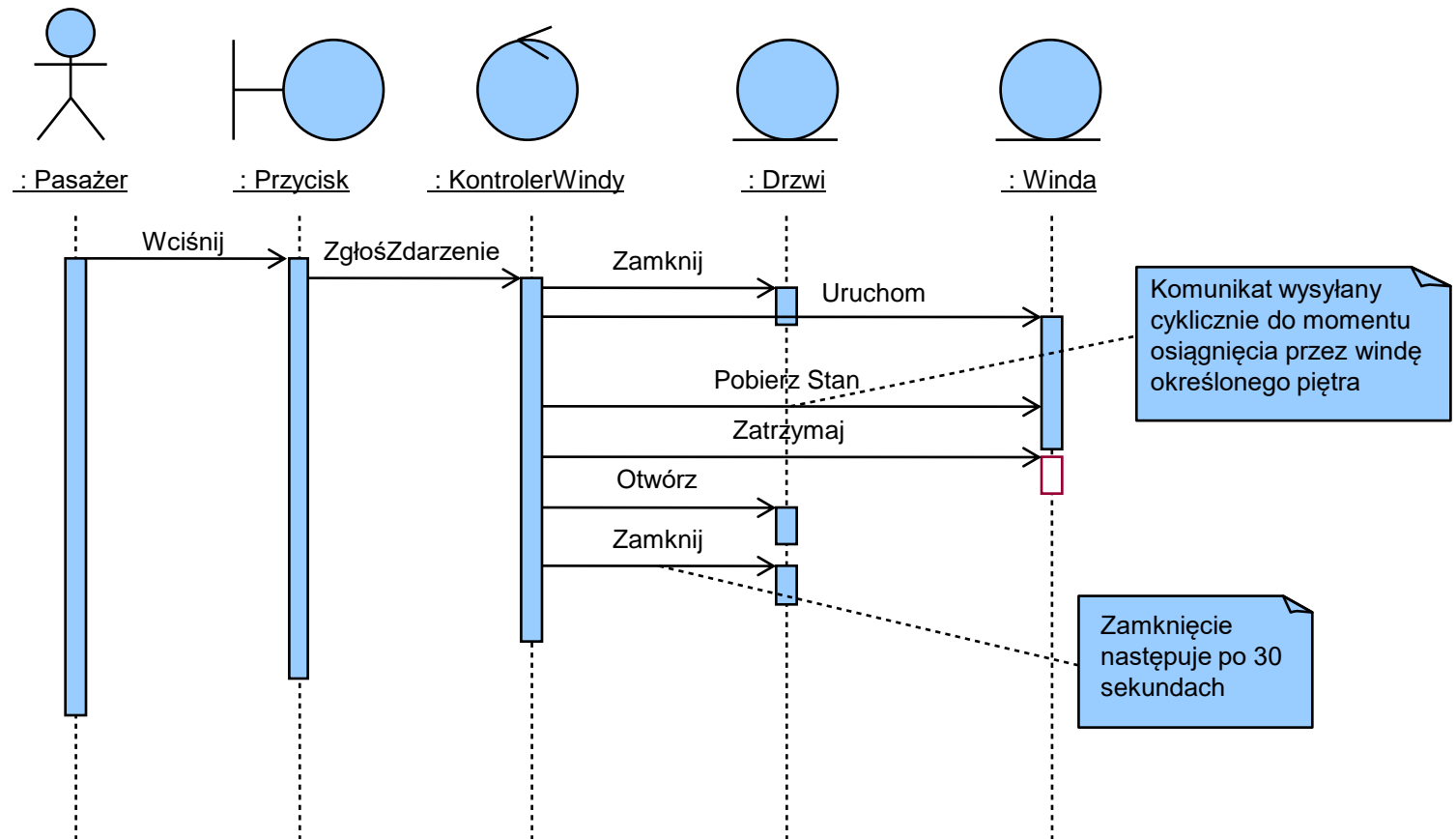


Diagram sekwencji - przykład



Diagramy sekwencji a diagramy komunikacji

Diagramy są semantycznie równoważne, jednak pokazują informacje w różny sposób.

Diagramy sekwencji:

- Bezpośrednio pokazują kolejność komunikatów
- Uwypuklają przepływ sterowania
- Lepiej modelują złożone scenariusze oraz specyfikację dla systemów czasu rzeczywistego.

Diagramy komunikacji:

- Kładą nacisk na powiązania
- Uwypuklają wzorzec współpracy (kto z kim)
- Uwypuklają efekty działania danego obiektu
- Łatwiejsze do narysowania w trakcie burzy mózgów (np. z wykorzystaniem kart CRC), której wynikiem jest opis działania systemu podczas realizacji przypadku użycia

Krok 2: Określanie klas analizy

Kroki analizy przypadków użycia

1: Dla każdej realizacji Przypadku Użycia
1.1: Znalezienie klas na podstawie analizy działania systemu

1.2: **Przydzielenie klasom odpowiedzialności**

2: Określanie własności klas analizy

2.1: Opisanie odpowiedzialności

2.2: Opisanie atrybutów i asocjacji

2.3: Przypisanie mechanizmu analizy

Krok 3: Unifikacja klas

Krok 4: Kontrola



Dokumentowanie co klasa posiada i co robi.

Opisanie odpowiedzialności klas

Odpowiedzialność klasy

- działanie, które obiekt klasy jest zobowiązany wykonać.
- Informacje jakie obiekt posiada, którymi zarządza i które udostępnia

Źródła

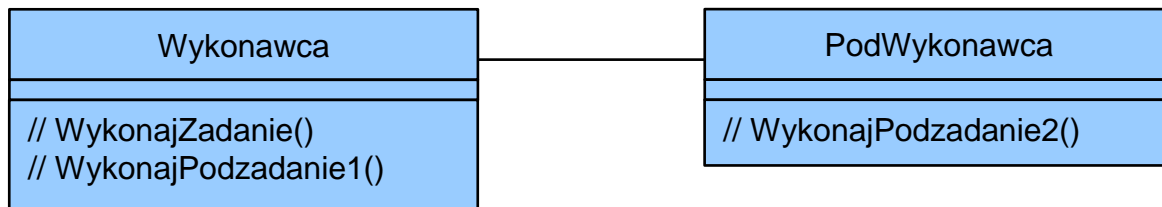
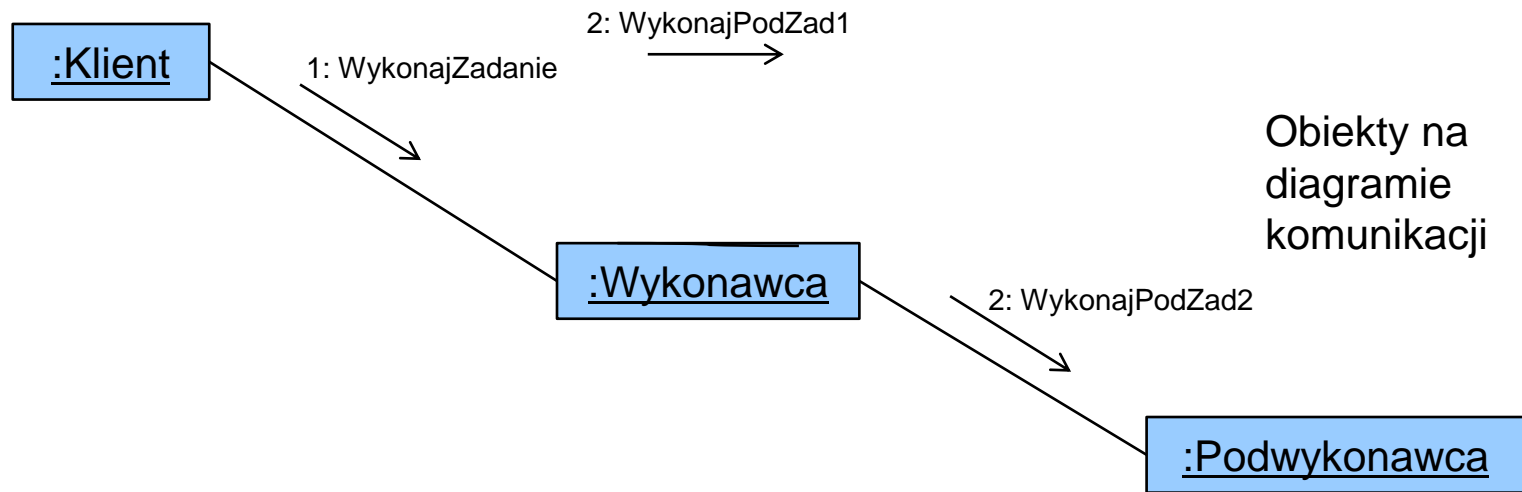
- Diagramy interakcji
- Wymagania нефunkcjonalne

Należy pamiętać o tym, że w procesie projektowania wyniki analizy (klasy, ich odpowiedzialności) w wielu przypadkach ulegną zmianie.

Odpowiedzialności klas a interakcja na diagramach interakcji

- Obiekt, adresat komunikatu musi go „rozumieć”, co oznacza, że klasa której jest wystąpieniem musi dostarczyć (definiować) tę operację.
- Konstruowanie diagramów interakcji może pomóc w identyfikowaniu zarówno asocjacji między klasami, jak i operacji w klasach, a przez to może prowadzić do korekty diagramu klas.

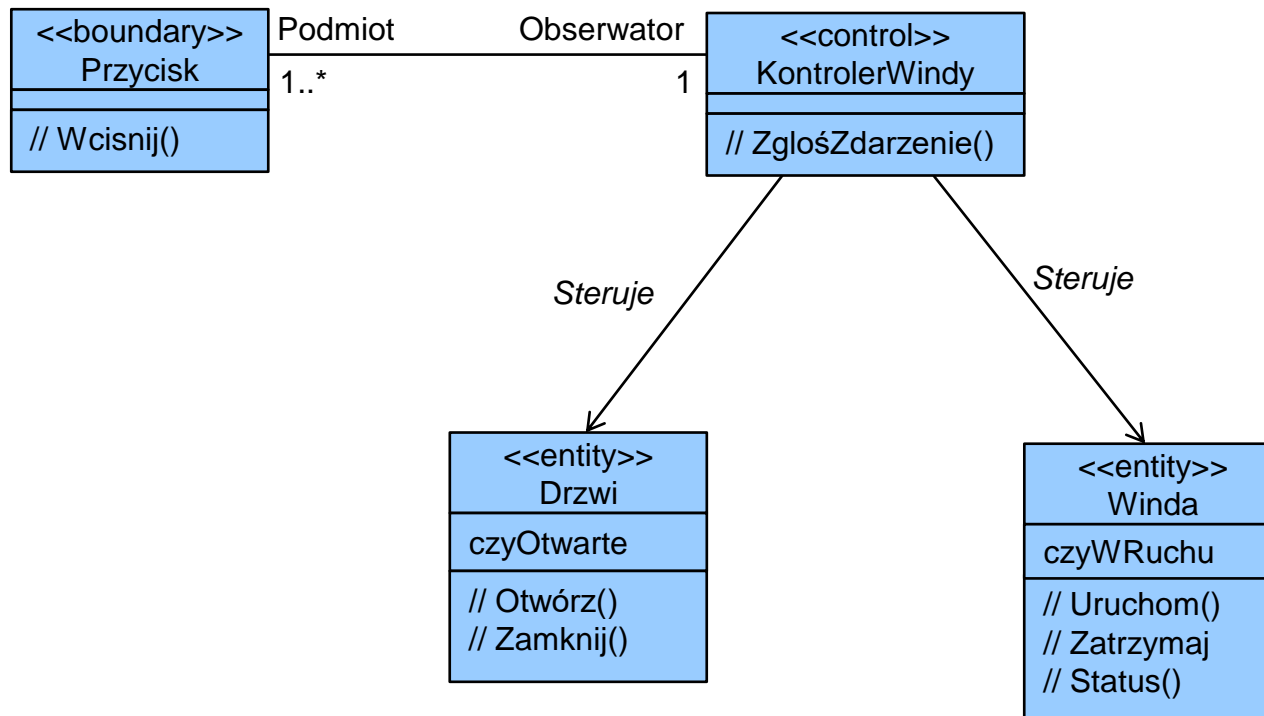
Diagram komunikacji a diagram klas



Krok 2.2: Opisanie atrybutów i asocjacji

- Atrybuty
 - przechowują informacje
 - Są atomowe, nie posiadają odpowiedzialności
- Relacje pomiędzy dwoma (lub większą liczbą klas)
 - Asocjacja
 - Agregacja
 - Generalizacja

Przykład diagramu klas z powiązaniem i atrybutami



Krok 2.3: Przypisanie mechanizmów analizy

Mechanizm analizy (konceptyjny):

- Opisuje podstawowe aspekty rozwiązania zagadnienia w sposób niezależny od konkretnej implementacji.
- Używany jest podczas analizy w celu zmniejszenia złożoności problemu na tym etapie.
- Pewne złożone zachowanie opisujemy określonym stereotypem nie przejmując się szczegółami implementacyjnymi.

Własności mechanizmu analizy opisują нефunkcjonalne wymagania związane z systemem

Mechanizm analizy – przykłady

- Trwałość (ang. *persistence*)
- Typ komunikacji (IPC, RPC)
- Rozproszenie (ang. *distribution*)
- Zarządzanie transakcjami (ang. *transaction management*)
- Synchronizacja (ang. *process synchronization*)
- Bezpieczeństwo (ang. *security*)
- Obsługa błędów (ang. *error detection, handling, reporting*)
- Nadmiarowość (ang. *redundancy*)
- Interfejsy spadkowe (ang. *legacy interface*)

Właściwości mechanizmu analizy

- przykłady

- Trwałość
- Ziarnistość
- Natężenie
- Czas trwania
- Mechanizm dostępu
- Częstotliwość dostępu
- Niezawodność
- Komunikacja międzyprocesowa
- Opóźnienie
- Synchronizacja
- Rozmiar komunikatu
- Protokół



Właściwości mechanizmu analizy – przykłady cd.

Interfejs spadkowy

- Opóźnienie
- Czas
- Mechanizm dostępu
- Częstotliwość dostępu

Bezpieczeństwo

- Ziarnistość danych
- Ziarnistość użytkowników
- Reguły bezpieczeństwa
- Typy uprawnień

Przykład:

Klasa: **Terminarz Studenta**

Mechanizm analizy:

- Trwałość
- Ziarnistość (rozmiar danych): 1 – 10 kb
- Objętość (liczba rekordów): do 2000
- Częstotliwość dostępu:
- Tworzenie: 500 dziennie
- Odczyt: 2000 na godzinę
- Aktualizacje: 1000 dziennie
- Usuwanie: 50 dziennie

Proces opisu mechanizmu analizy

Zbierz wszystkie mechanizmy analizy w postaci listy

Takie same mechanizmy analizy mogą występować w różnych sytuacjach pod różnymi nazwami (trwałość, baza danych, repozytorium, składowanie) (komunikacja międzyprocesowa, zdalne wywołanie procedury, przekazywanie komunikatów)

Proces opisu mechanizmu analizy

Realizacja mapy powiązań pomiędzy klasami i mechanizmami analizy

- klasy,
- nazwy mechanizmów,
- graficzna prezentacja klasy z odpowiednimi mechanizmami

Proces opisu mechanizmu analizy

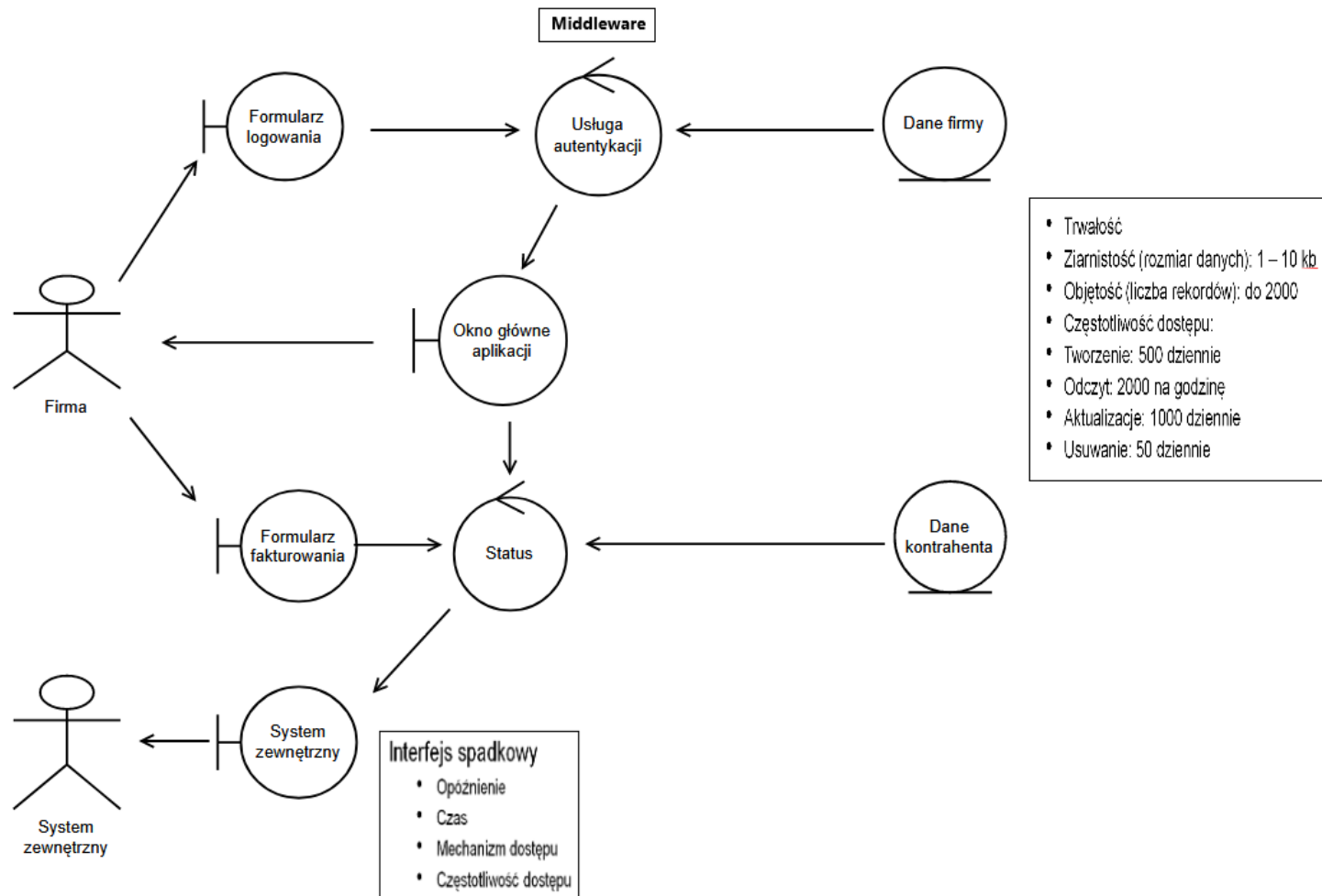
Określ właściwości mechanizmów analizy

Aby ułatwić w procesie projektowym wybór odpowiedniego rozwiązania projektowego należy opisać podstawowe właściwości mechanizmów.

Proces opisu mechanizmu analizy

Kiedy mechanizmy zostały już zidentyfikowane i nazwane, należy je zamodelować w postaci współpracujących ze sobą klas. Niektóre z tych klas nie udostępniają żadnej funkcjonalności specyficznej dla tworzonej aplikacji, ale jedynie służą wsparciu działania aplikacji (często umieszcza się je potem w specjalnej warstwie oprogramowania zwanej – middleware).

Przykład opisu mechanizmów analizy



Krok 3: Unifikacja klas

- Dotychczas tworzyliśmy osobne modele dla każdego **przypadku użycia**.
- Teraz trzeba je zunifikować, tak aby wyodrębnić te, które mają unikatowa odpowiedzialność i połączyć te, których zadania są wspólne.
- Nazwa klasy analizy powinna odzwierciedlać istotę roli, jaką klasa gra w systemie. Nazwa musi być unikatowa.
- Należy pamiętać, że po uaktualnieniu klas należy uaktualnić opis realizacji przypadku użycia. Czasem uzupełnienie może dotyczyć oryginalnych wymagań (przypadków użycia) ale **należy pamiętać o tym, że wymagania są częścią umowy z klientem i każda zmiana powinna być potwierdzona i kontrolowana.**

Krok 4: Kontrola poprawności modelu

- Czy klasy analizy mają sens?
- Czy nazwa klasy odzwierciedla rolę tej klasy w systemie?
- Czy klasa reprezentuje pojedynczą, dobrze zdefiniowaną abstrakcję?
- Czy atrybuty klasy oraz jej odpowiedzialności są ze sobą funkcjonalnie powiązane?
- Czy klasa udostępnia wystarczającą funkcjonalność?
- Czy klasa spełnia wszystkie wymagania?



Krok 5: Kontrola poprawności modelu

Realizacja przypadków użycia:

- Czy wszystkie przepływy zostały zrealizowane (włącznie z alternatywnymi i wyjątkowymi)?
- Czy wszystkie obiekty zostały znalezione?
- Czy działanie wykonywanie w przepływach zostało jednoznacznie rozdzielone pomiędzy obiekty.
- Czy działanie zostało przypisane odpowiednim obiektom?
- Jeżeli dla przepływu zostało zdefiniowanych kilka diagramów interakcji to czy związki pomiędzy nimi są czytelne?

