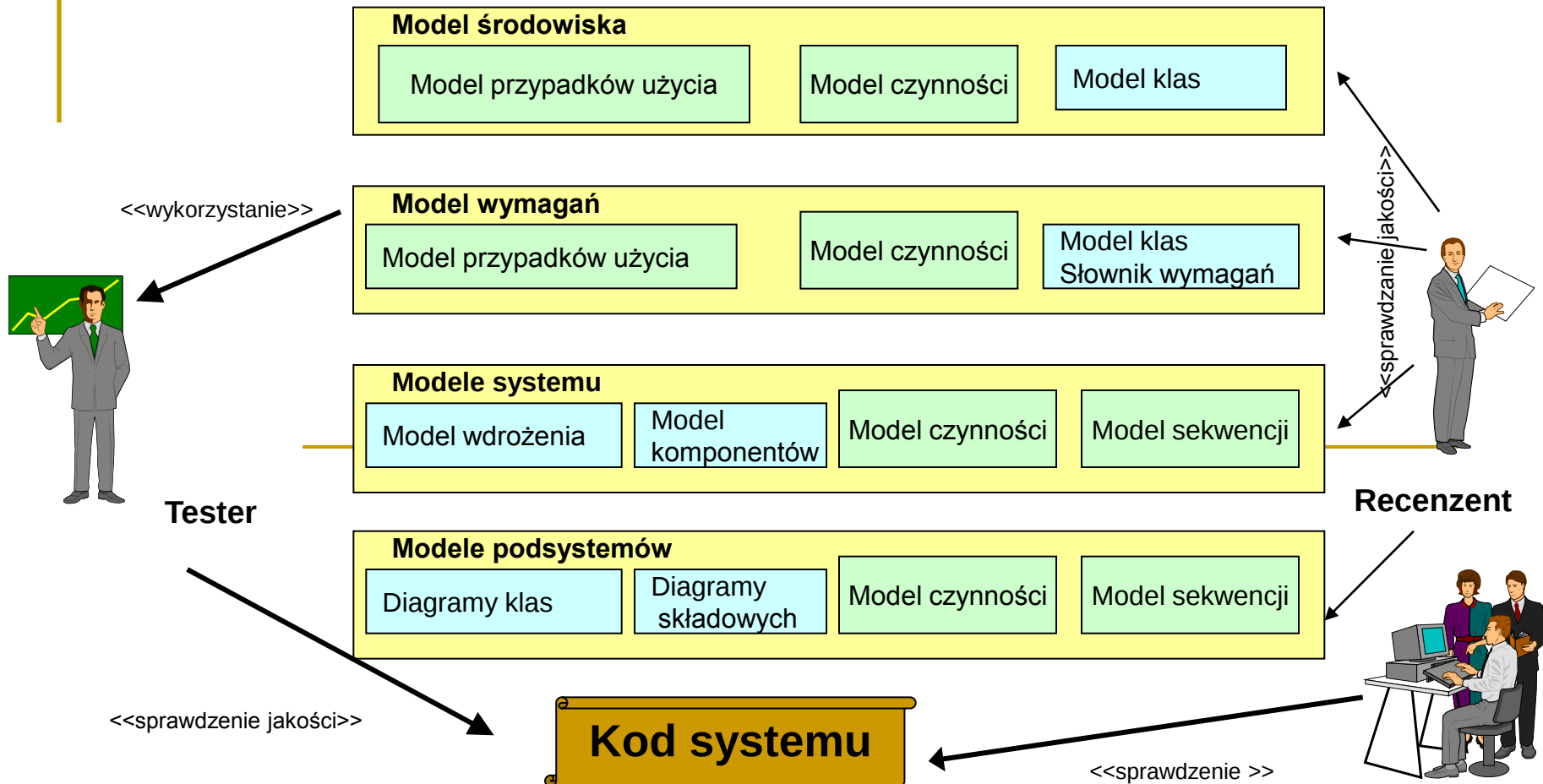


KONTROLA JAKOŚCI

Zapewnienie jakości w projekcie tworzenia oprogramowania



- **Wpływ ponownego wykorzystania na jakość projektu**
- **Tworzenie modeli projektowych (podsystemów) wzorce projektowe**
- **Szkielety architektoniczne – wzorce, modele wymagań - wzorce.**

PRZEGLĄDY

W trakcie opracowywania przypadków użycia
dokonujemy przeglądów – recenzja dokumentów

(Identyfikacja problemów – wcześniejsza to tańsza)

ZAINTERESOWANI:

klienci, doradcy

użytkownicy – zwłaszcza tam gdzie są aktorami!!! Np. interfejs użytkownika

przegląd architektury systemu, spisu zagrożeń

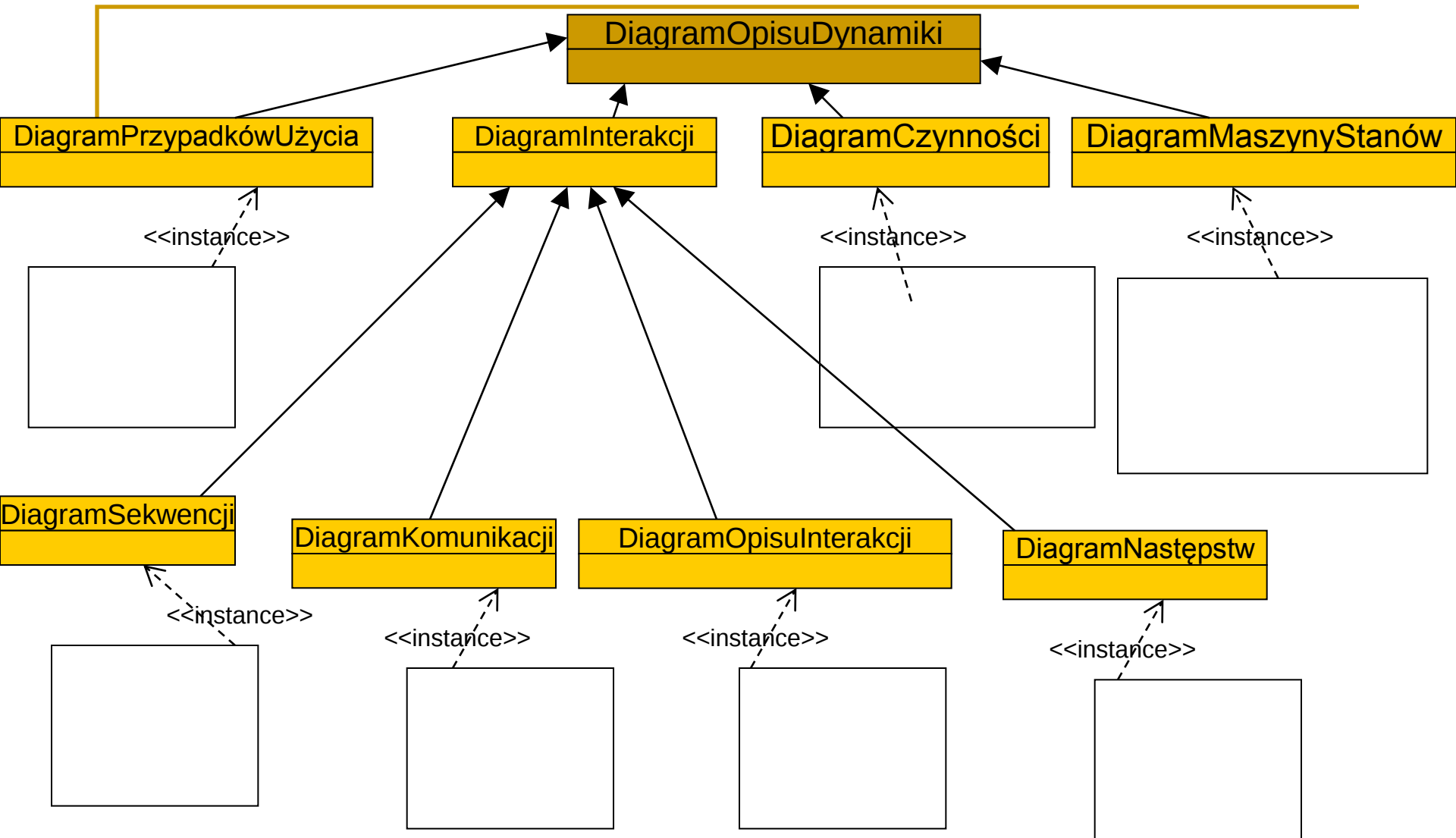
sprawdzenie pełności, przejrzystość, wykonalność przyjętych założeń

RODZAJE PRZEGLĄDÓW:

- **Kontrola pełności – przegląd dokumentacji przez architekta systemu i analityka dla sprawdzenia ,
jej pełności i spójności na danym etapie prac.**
- **Kontrola w celu znalezienia potencjalnych problemów**
- **Przeglądy z użytkownikami docelowymi**
- **Przeglądy z klientami**
- **Przeglądy z programistami**
- **Przeglądy z recenzentami**

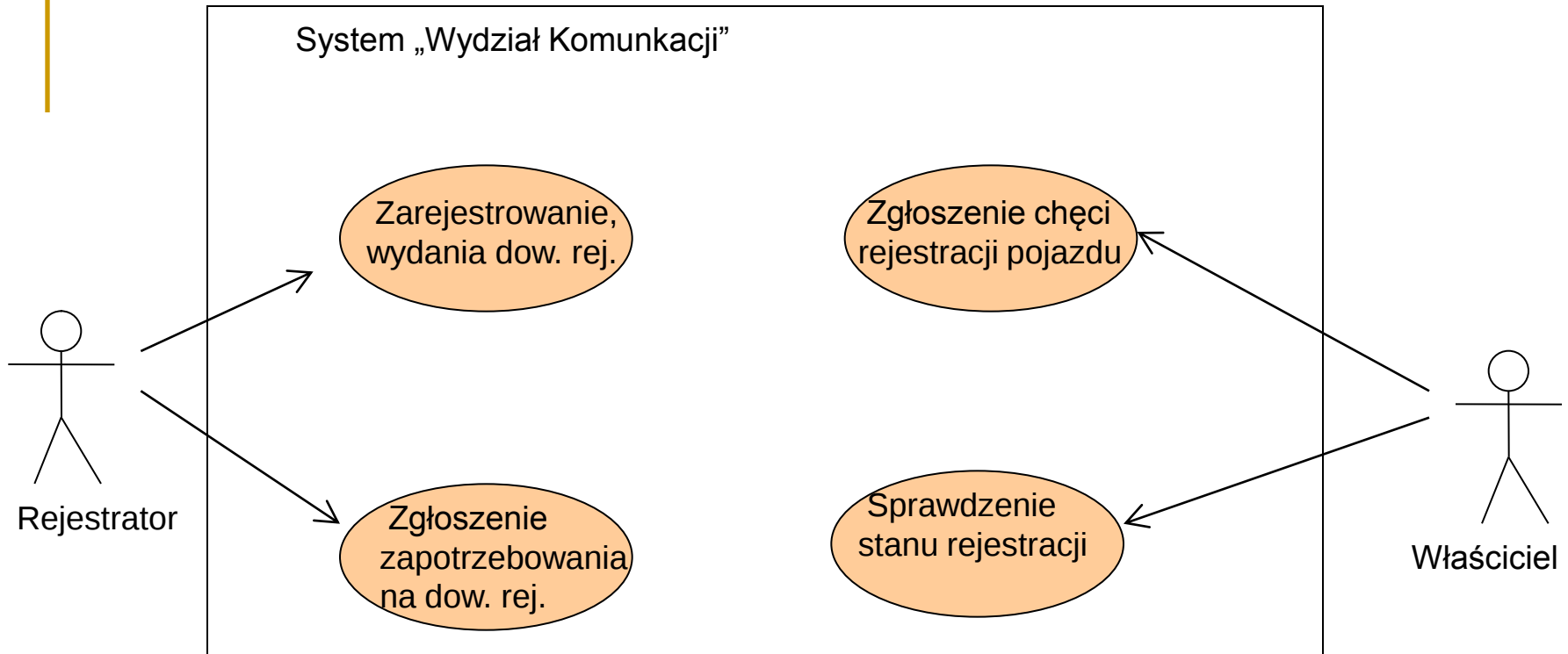
CZĘSTO SPOTYKANE BŁĘDY?

PRZEGLĄDY DYNAMIKI



PRZEGLĄDY

diagram przypadków użycia



- **Przypadki użycia** – jednostki opisu dynamiki systemu – **zestawy scenariuszy**
- **Scenariusze** – sekwencja interakcji podlegających określonym warunkom i prowadzącym do określonego celu (w opisie prezentowana jako tekst lub **diagramy czynności**)
- **Actor** – obiekt (klasa obiektów) na zewnątrz systemu, często użytkowników systemu

PRZEGLĄDY

Kontrola pełności *(pełność i spójność dokumentacji)*

→ Opis przedsięwzięcia

- *Czy opis przedsięwzięcia w dalszym ciągu jest zrozumiały?*
- *Odkryte nowe czynniki wpływające na przedsięwzięcie?*
- *Czy zagrożenia są takie same, czy są nowe?*

→ Opis przypadków użycia i diagramów

- *Czy odkryłeś nowych aktorów, lub niektórzy są niepotrzebni?*
- *Czy jacyś aktorzy zostali przeniesieni do wnętrza systemu?*
- *Czy coś z wnętrza zostało wyciągnięte na zewnątrz*
- *Czy nazwy i opisy aktorów odpowiadają ich rolom?*
- *Czy powstały nowe przypadki użycia, a jakieś zostały usunięte?*
- *Czy każdy przypadek użycia zawiera przynajmniej opis głównego ciągu zdarzeń?*
- *Czy diagramy ilustrujące interfejs użytkownika pasują do przypadków użycia?*
-

PRZEGLĄDY

Kontrola w celu znalezienia potencjalnych problemów

- **Architekt oraz analityk systemu przeglądają zagrożenia, czynniki rynkowe i warianty rozwiązania**
- *np.. W systemie obsługi zamówień wiele scenariuszy dotyczy przypadków niedostępności systemu magazynowego Lub księgowego. Potrzebny jest więc mechanizm przechowywania zleceń do czasu ich dostępności – stworzenie mechanizmów właściwych dla całego systemu*
- *tutaj, uwaga unikamy szczegółów implementacyjnych np.. W miejsce stwierdzenia, że klienci potrzebują uzyskać dostęp do systemu przez przeglądarkę www zanotuj potrzebna jest metoda dostępu klienta do systemu i opisz cechy tego interfejsu. O tym jak go zrealizować zdecydujesz później.*

PRZEGLĄDY

Przeglądy z użytkownikami (ludźmi korzystającymi z systemu)

- *Czy system robi to, czego się od niego oczekuje?*
- *Czy czegoś brakuje?*
- *Czy jakieś części systemu są niepotrzebne?*
- *Czy wszystko co system robi jest zrozumiałe?*
- *Czy sądząc na podstawie przebiegów zdarzeń i ilustracji interfejsu użytkownika, system będzie wygodny w użyciu*
- *Czy system zachowuje się tak, jak powinien?*

PRZEGLĄDY Z KLIENTAMI

- *Czy zgadzają się na poczynione założenia?*
- *Co należy zmienić?*
- *Czy dobrze zrozumieliśmy ich potrzeby?*
- *Czy system jest tym, czego oczekiwali?*
- *Czy rozumieją co system ma robić?*

PRZEGLĄDY

Przeglądy z programistami

- *Czy wszyscy znają cel przedsięwzięcia?*
- *Czy to wszystko ma sens?*
- *Czy mając przygotowaną dokumentację, programiści są w stanie stworzyć system?*
- *Co jeszcze muszą wiedzieć, zanim rozpoczną pracę.*
- *Czy są entuzjastycznie nastawieni do przedsięwzięcia?*

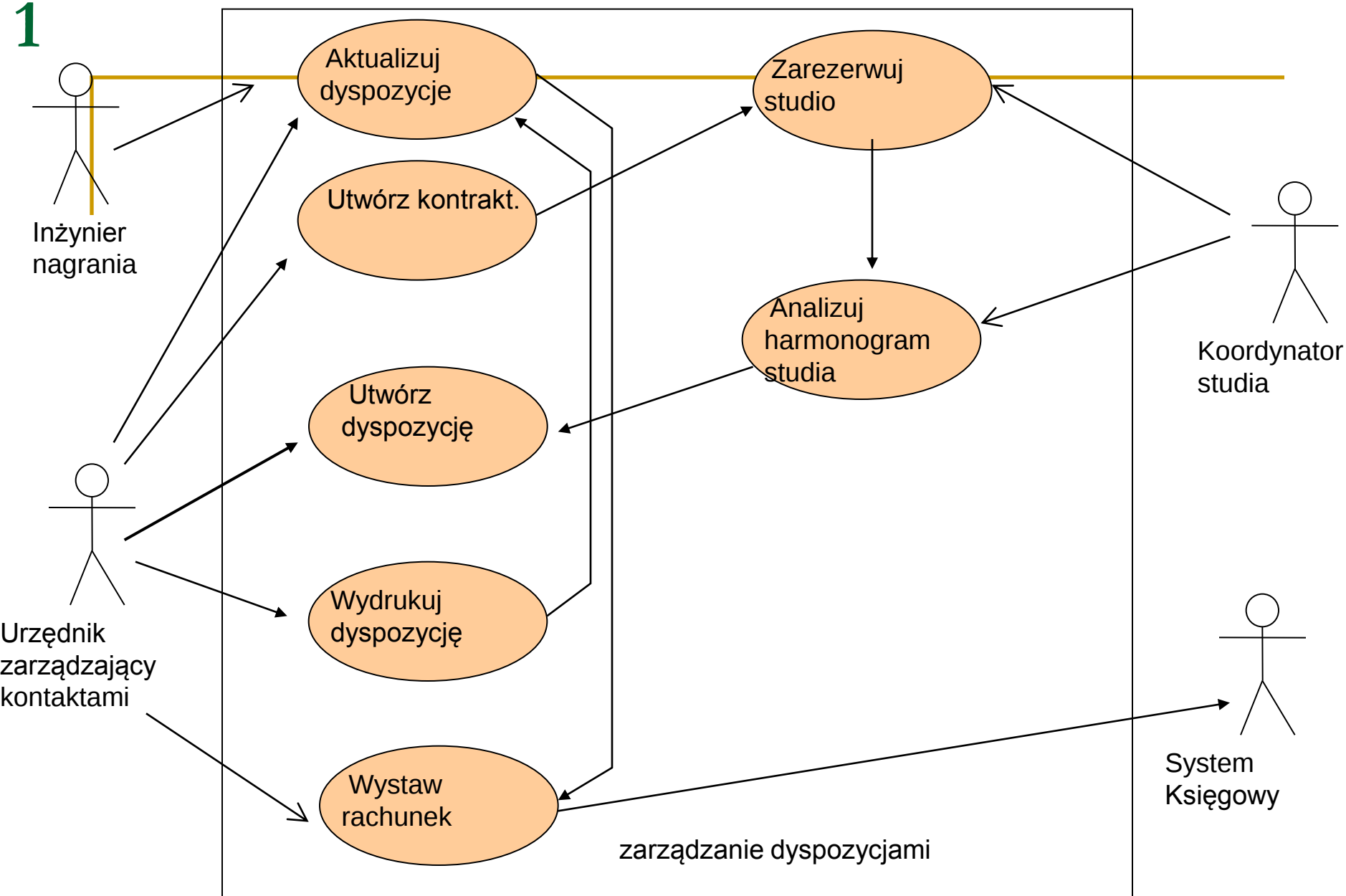
Recenzenci

- *Jedni odpowiadają na powyższe pytania sami – w zespole?*
- *Inne zespoły przez kilka tygodni wykonują formalne prezentacje dla klientów, niezależnych zespołów weryfikujących, zarządów.*
- *szczerzy recenzenci – bezinteresowni i niezbyt mili!! Chodzi o znalezienie błędów*
- *Najlepsi obcy recenzenci – nie mają uprzedzeń i szybko odnajdą miejsca w których coś nie jest jasne, lub jest sprzeczne.*

PRZEGLĄDY

System – studio nagrań, sprzedaje klientom czas pracy w studio, taśmy które powstają oraz usługi, jak np. montaż taśm PRZEPŁYW CZYNNOŚCI NA DIAGRAMIE?

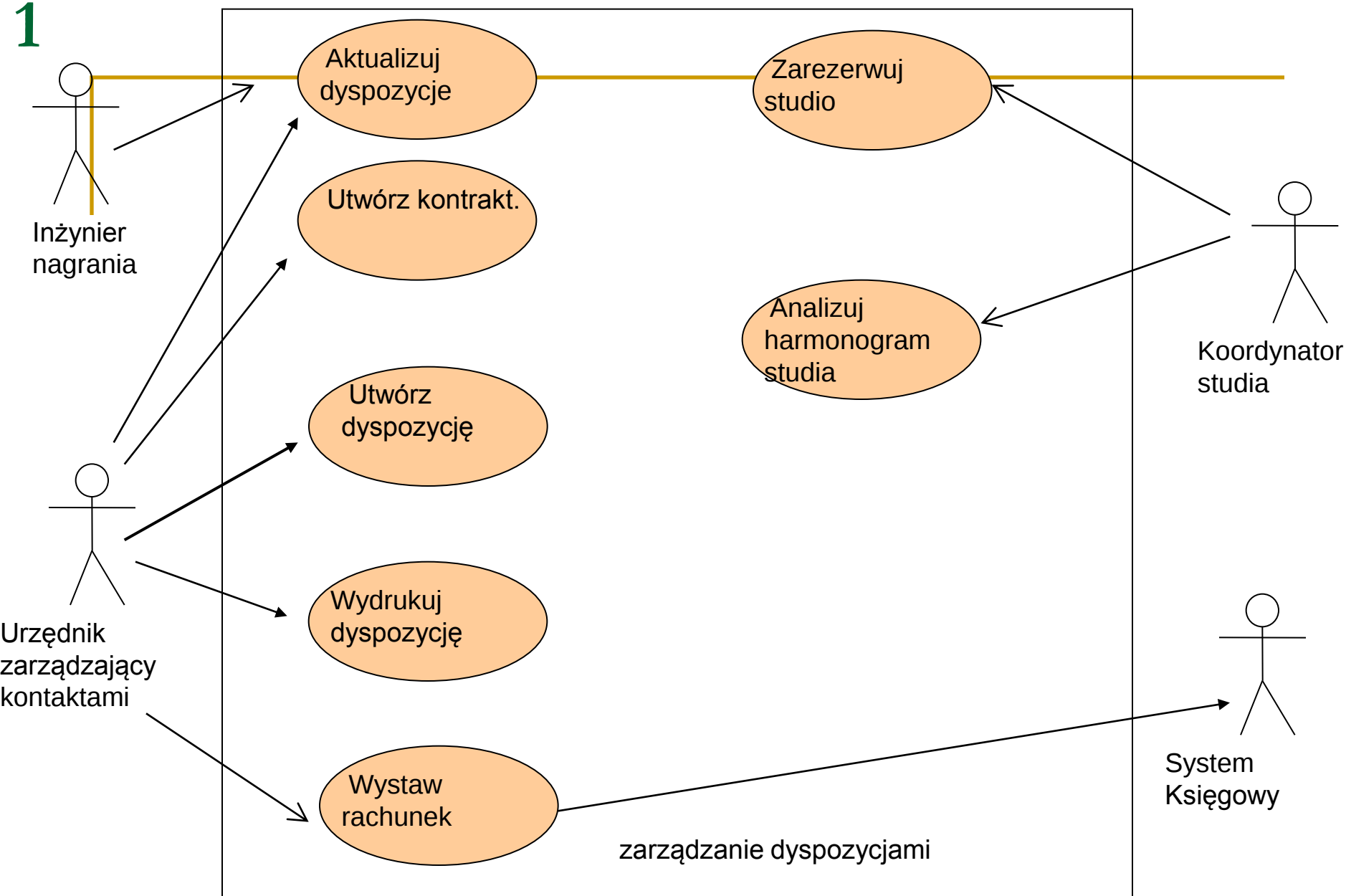
1



PRZEGLĄDY

System – studio nagrań, sprzedaje klientom czas pracy w studio, taśmy które powstają oraz usługi, jak np. montaż taśm

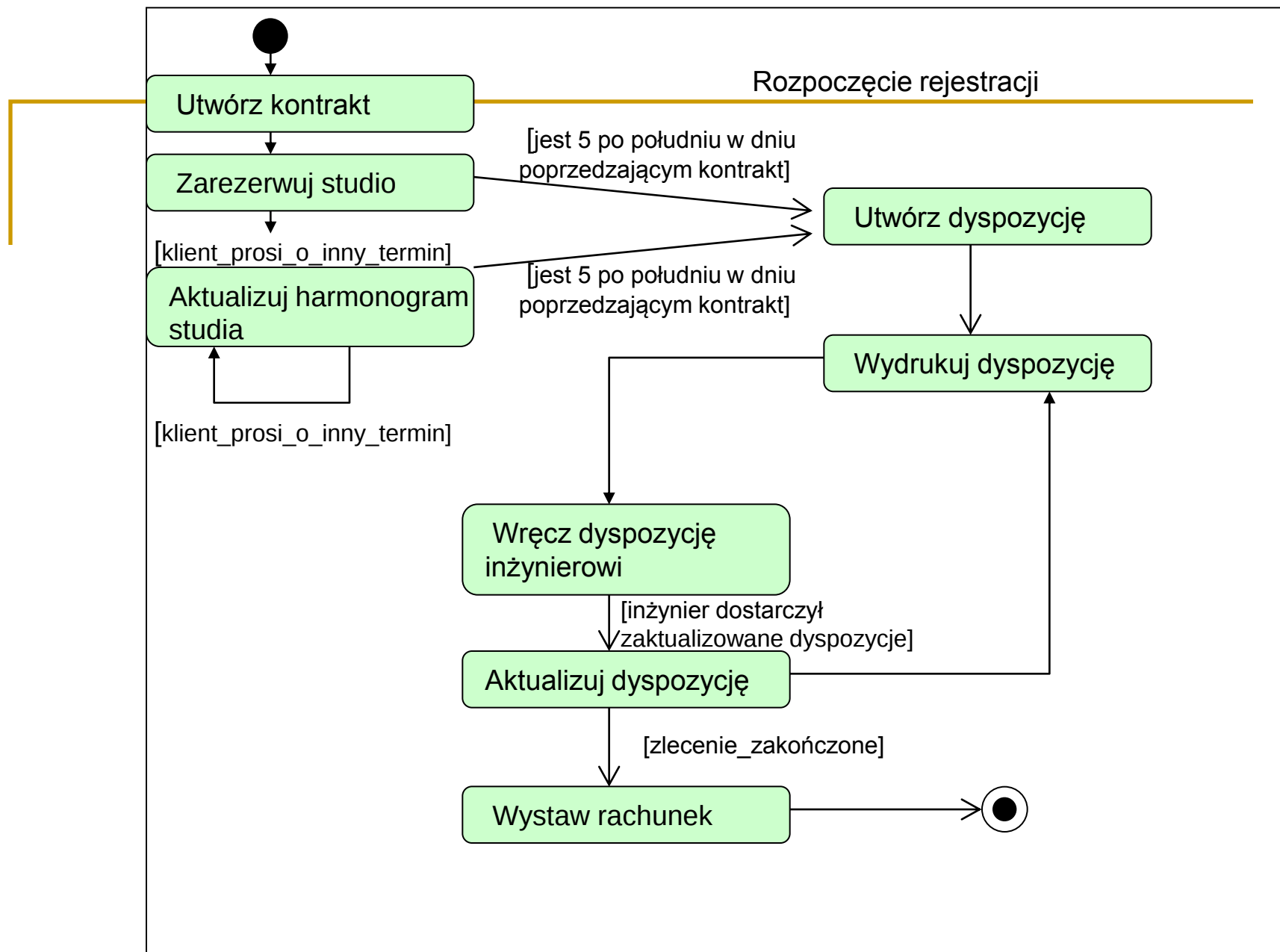
1



PRZEGLĄDY

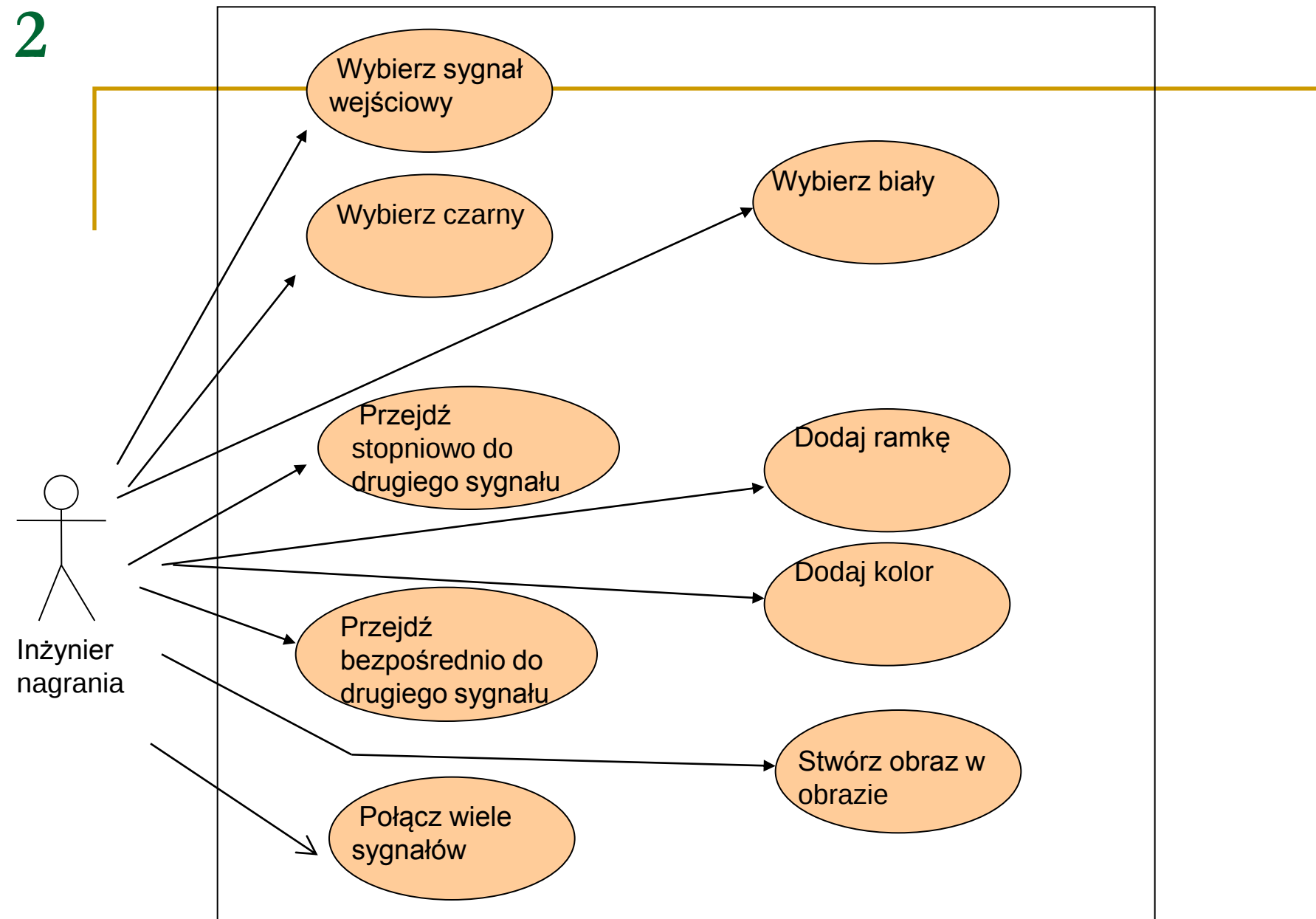
System – studio nagrań, diagram czynności dla
studia nagrań – na poziomie procesów biznesowych

1



PRZEGLĄDY Firma produkuje mikser – na wejściu przyjmuje kilka sygnałów wizyjnych a na wyjściu umieszcza tylko jeden. Umożliwia przejścia pomiędzy sygnałami, przenikanie obrazu, dodanie tła lub zmianę kolorów. ZBYT SZCZEGÓŁOWE

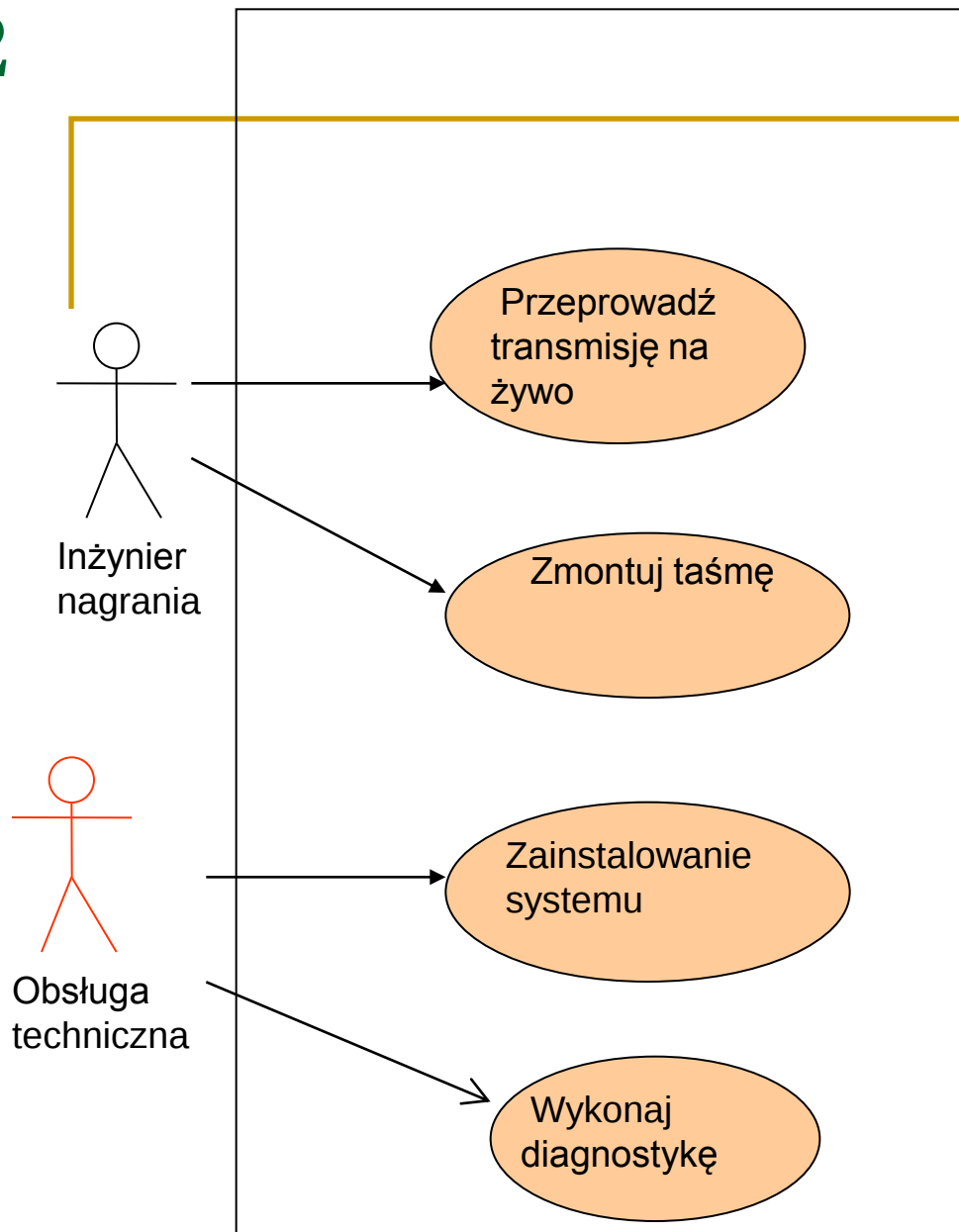
2



PRZEGLĄDY

Firma produkuje mikser – dwa przypadki użycia

2



- Inżynier może wybrać Koniec w dowolnej chwili po rozpoczęciu nadawania i przypadek użycia się kończy.

-Przypadek użycia jest rozpatrywany na wysokim poziomie abstrakcji – system jest opisywany jednym przypadkiem

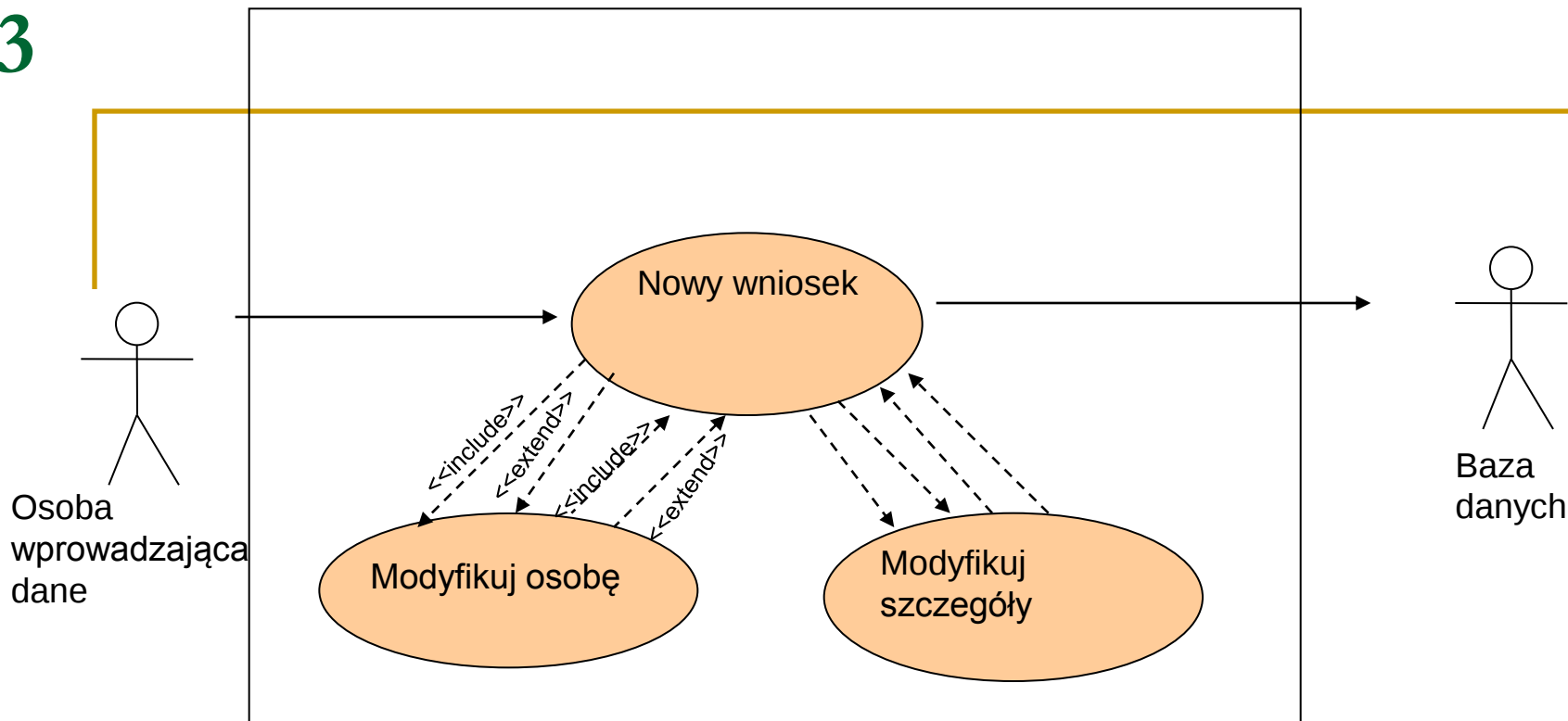
-- niewielkie systemy na poziomie procesów biznesowych (z zewnątrz) dają się tak opisać.

WNIOSKI:

-Osoba wykonuje kilka czynności w krótkim czasie, może są one krokami większego przypadku użycia

-Przypadki użycia na poziomie procesów biznesowych często opisują cały system

-Nie wszystkie przypadki użycia zostały właściwie odczytane.

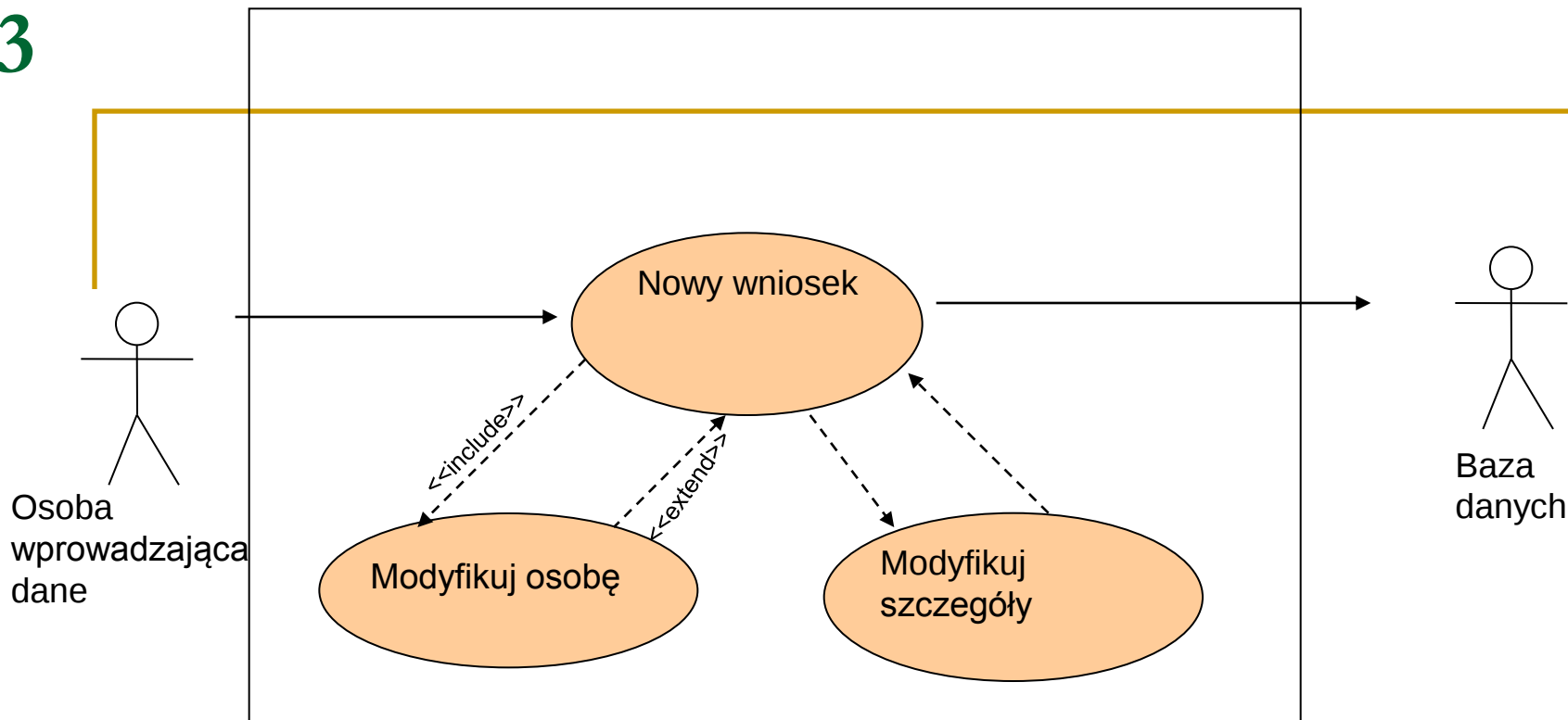


Związek <<include>> zawieranie się – odrębny przypadek użycia wykorzystywany wielokrotnie

Związek <<extend>> rozszerzenia – funkcjonalności opcjonalne

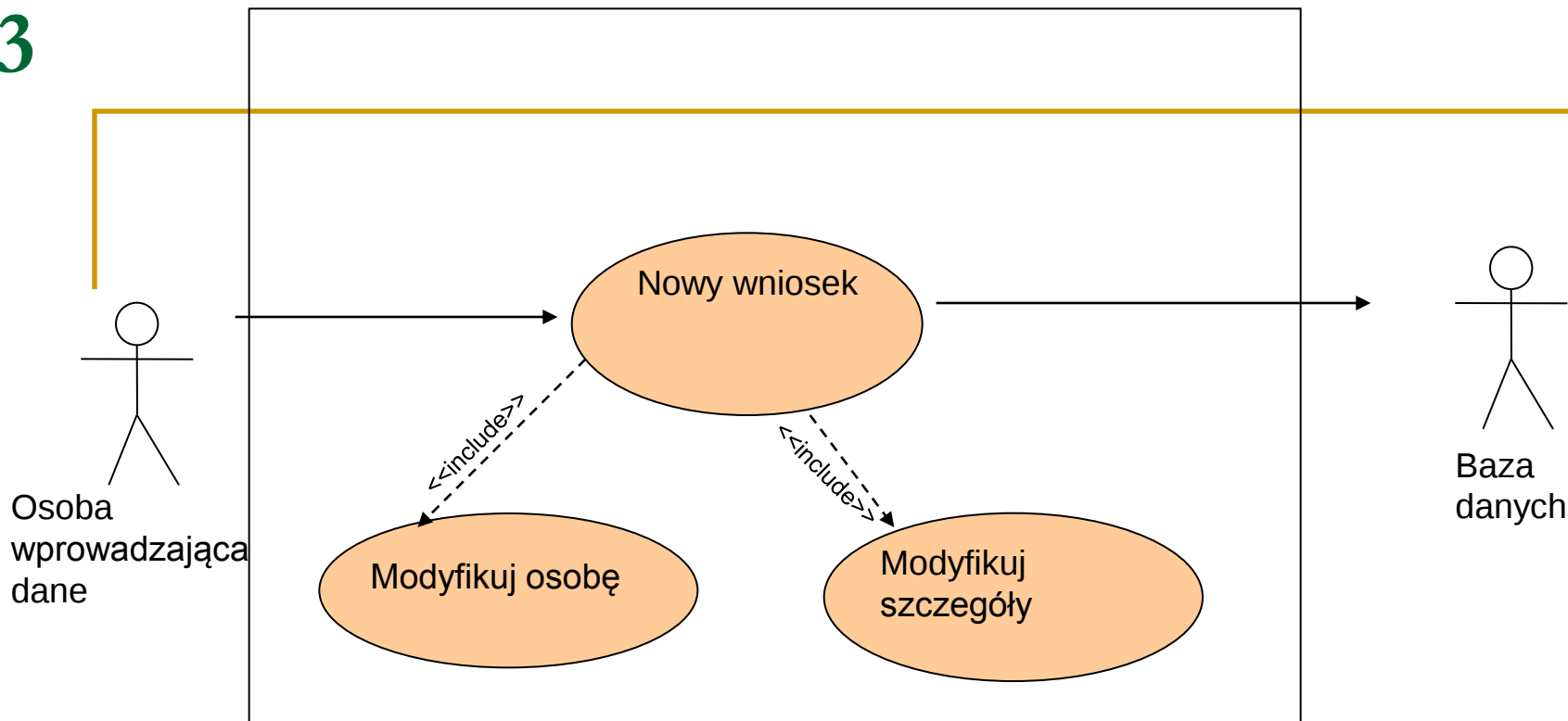
Tutaj przypadek oznacza nieskończona pętlę

Jest możliwe, że dwa przypadki się rozszerzają



Jednoczesne zawieranie i rozszerzanie, choć możliwe oznacza jedynie ten sam związek

Nowy wniosek musi włączać Modyfikuj osobę jak i modyfikuj dane (nie dotyczy modyfikacji danych)



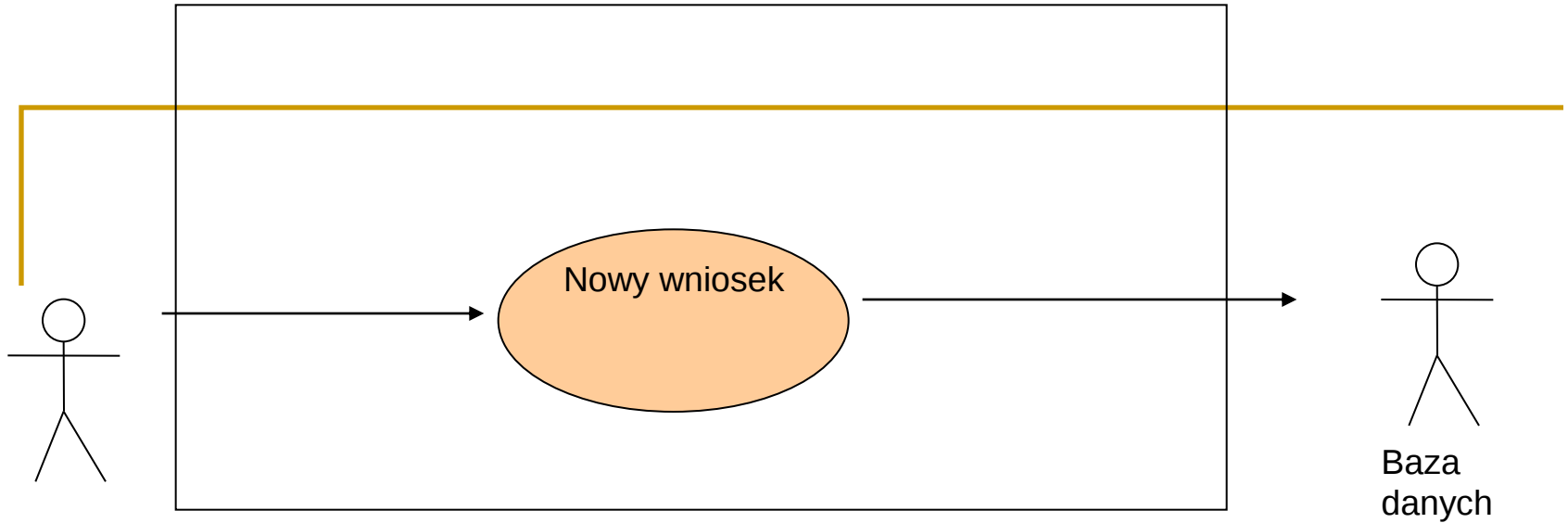
Powrót z przypadku użycia Modyfikuj osobę i Modyfikuj szczegóły jest automatyczny!!!

Przypadki <<include>> modyfikuj osobę modyfikuj szczegóły mogą być przypadkami użycia bez włączenia

PRZEGLĄDY

Formularze jako przypadki użycia – program do wypełniania na ekranie wniosków ubezpieczeniowych

3



Osoba
wprowadzająca
dane

Baza
danych

Ostateczny przypadek użycia dla programu ubezpieczeniowego.

PODZIAŁ WIELKICH SYSTEMÓW

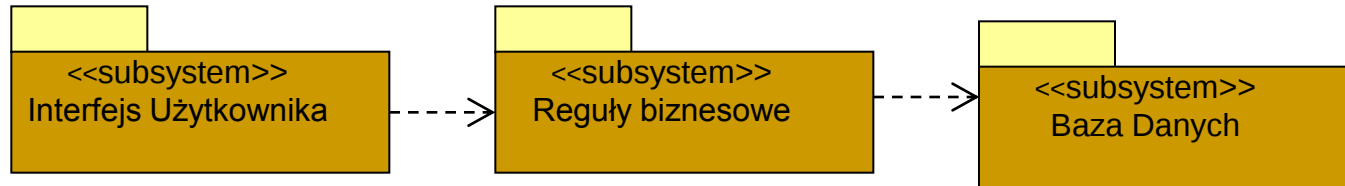
Wzorce architektoniczne

- Wzorce architektoniczne – uogólnione spojrzenie na strukturę systemu
- Są wykorzystywane przy tworzeniu głównych części systemu – widocznych na pierwszy rzut oka
- Pozwalają na generacje podsystemów
- Związki pomiędzy podsystemami pozwalają dopasować opracowywany system do któregoś z wzorców architektonicznych

- Architektura trójwarstwowa
- Wzorzec potoków i filtrów
- Wzorzec architektury obiektowej

PODZIAŁ WIELKICH SYSTEMÓW

Architektura trójwarstwowa (system obsługi zamówień)



Trzy warstwy - aby reguły biznesowe działały – muszą korzystać z bazy danych

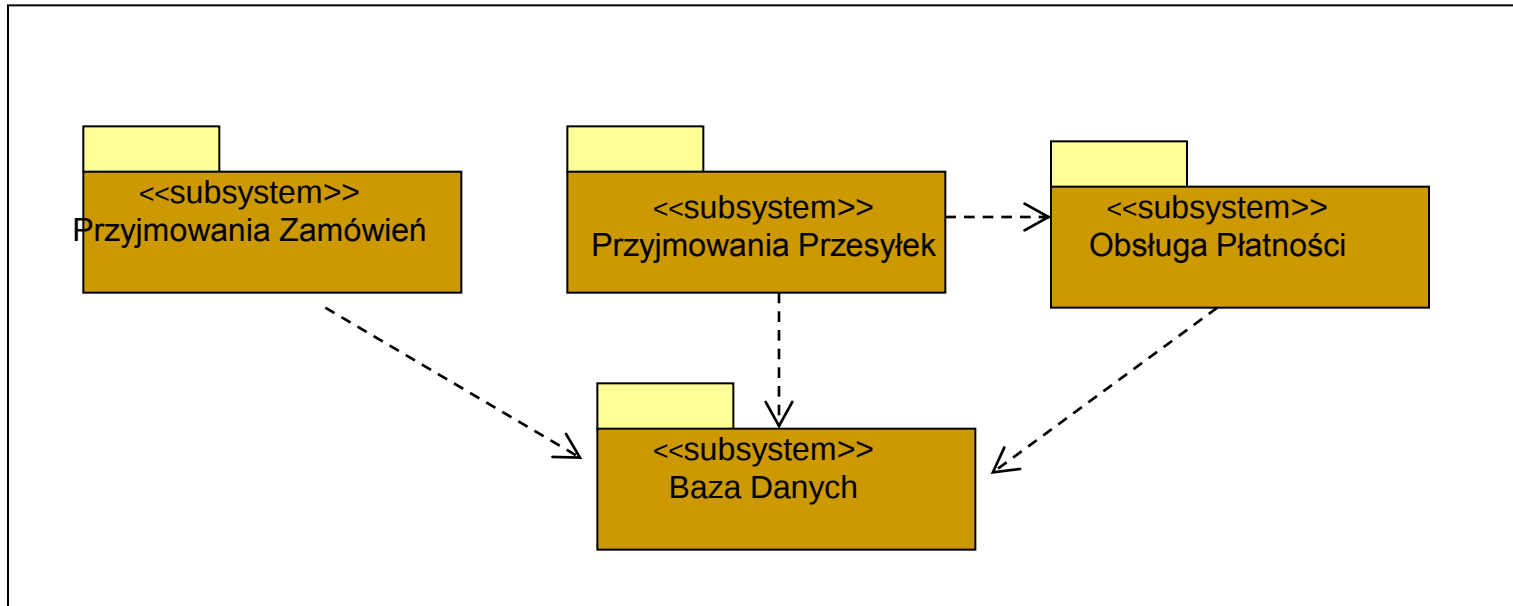
Przypisanie poszczególnych części systemu do kolejnych warstw: Formularz – część Interfejsu Użytkownika; Dane zebrane w związku z zamówieniem – warstwa Bazy Danych; Proces pobierania zamówienia – warstwa Reguł Biznesowych

Korzyści: Jedna baza danych; zestaw procesów biznesowych - spójny

PODZIAŁ WIELKICH SYSTEMÓW

Wzorzec potoków i filtrów

(system obsługi zamówień)



Pewien system przyjmuje dane, w pewien sposób je przetwarza, a następnie je zwraca, potem kolejny element pobiera zwrócone informacje, przetwarza itd.

Elementy są niezależne i tak naprawdę nic nie wiedza o sobie.

Przyjmowanie zamówień wykonuje prace i umieszcza ją na stosie zamówień

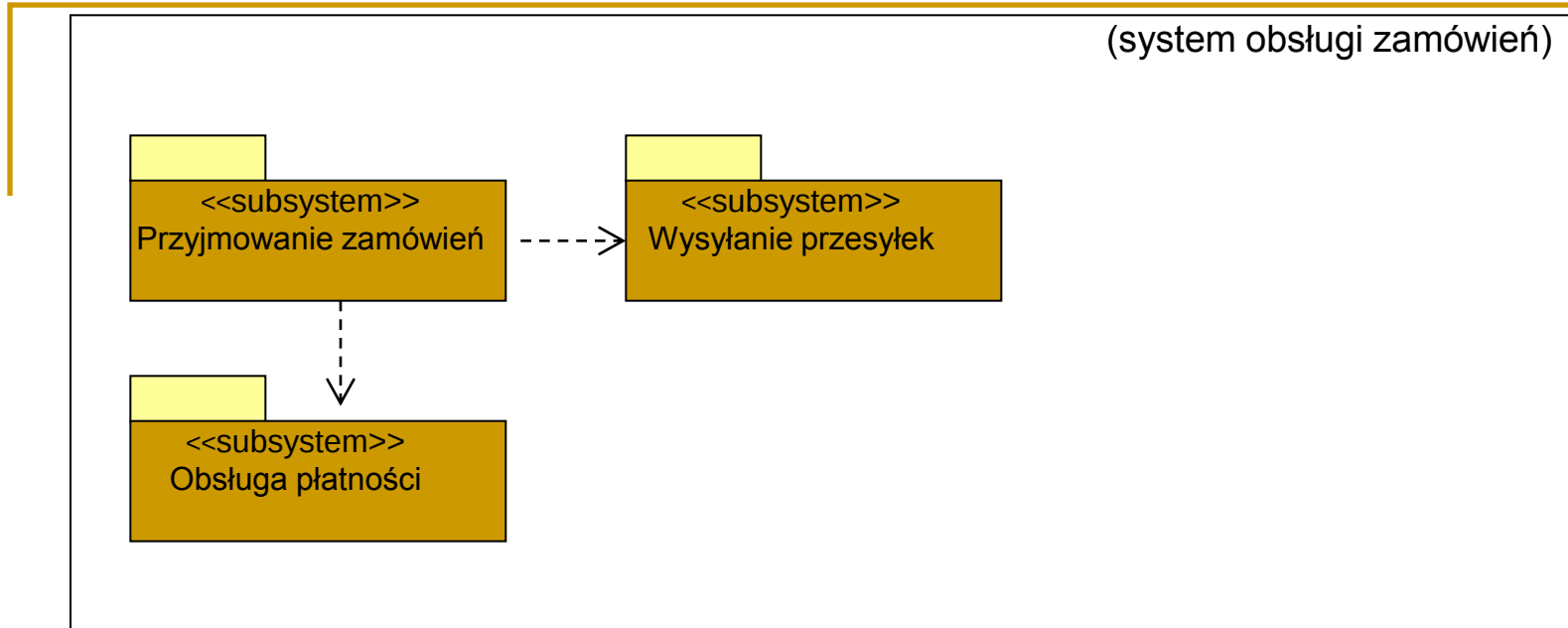
Wysyłanie przesyłek pobiera stos zamówień, przygotowuje je i wysyła tworząc stos płatności

Te są pobierane przez Obsługę płatności i są realizowane;

**DUŻA NIEZALEŻNOŚĆ PODSYSTEMÓW – ZALEŻNOŚĆ OD STRUMIENIA DANYCH ZAPISYWANYCH
SIOS**

PODZIAŁ WIELKICH SYSTEMÓW

Wzorzec architektury obiektowej



Podsystemy zdefiniowane wokół danych i związanych z nimi funkcjami

Pomiędzy podsystemami istnieją zależności, ale podsystemy nie wiedzą nic o sobie:

Przyjmowanie zamówień zawiera zamówienia i funkcje które je przetwarzają

Obsługa płatności zajmuje się opłatami, kredytami i kontami

Wysyłanie przesylek zajmuje się przesyłkami i informacjami adresowymi.

Tutaj każda funkcja jest niezależna i lokalizowana w jednej części – w trójwarstwowej każda funkcja istnieje w trzech warstwach – w potokach i filtrach funkcje komunikują się za pomocą danych.