

Michał Adamczyk

# Język UML

# UML

## I. Czym jest UML

- Po co UML

## II. Narzędzia obsługujące UML, edytory UML

## III. Rodzaje diagramów UML wraz z przykładami

- Zastosowanie diagramu
- Podstawowe elementy diagramu
- Przykładowy diagram

# Omawiane diagramy:

- Diagram przypadków użycia
- Diagram czynności
- Diagram klas
- Diagram maszyny stanowej
- Diagram sekwencji
- Diagram komunikacji
- Diagram harmonogramowania
- Diagram sterowania interakcją
- Diagram wdrożeniowy

# Czym jest UML?

UML=*Unified Modelling Language*

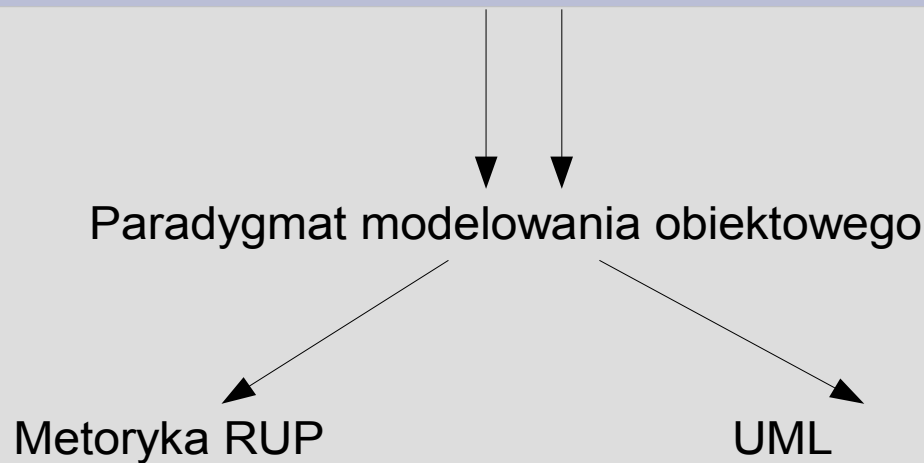
Język UML jest graficznym językiem wizualizacji, specyfikowania, tworzenia i dokumentowania składników i systemów informatycznych

# Rys historyczny

- 1995 – Pierwsza wersja UML:
  - Połączenie metodyk G. Boocha i J. Jacobsona i J. Rumbaugh
  - Rozwój obiektowej analizy i projektowania
- 2003 – wersja 2.0
- Maj 2010 - aktualna wersja: 2.3, rozwijana przez *Object Management Group*

# Po co UML?

Modelowanie nowoczesnych systemów informatycznych



- UML to praktyczne wdrożenie paradygmatu modelowania obiektowego, przejrzysta i wygodna forma projektowania systemów informatycznych

# Narzędzia/edytory UML

## NetBeans IDE

- Wtyczka do obsługi UML dostępna od wersji 6.7.1
- Posiada możliwość generowania kodu JAVA na podstawie niektórych diagramów.

## ArgoUML

- Obsługuje standardowe diagramy UML 1.4
- darmowy

# narzędzia/edytory UML cd.

## Visual Paradigm for UML

- Dostępny w darmowej wersji COMMUNITY z ograniczoną funkcjonalnością
- Obsługuje diagramy UML 2.0

## Enterprise Architect

- Obsługuje UML 2.1
- Profesjonalne narzędzie modelowania



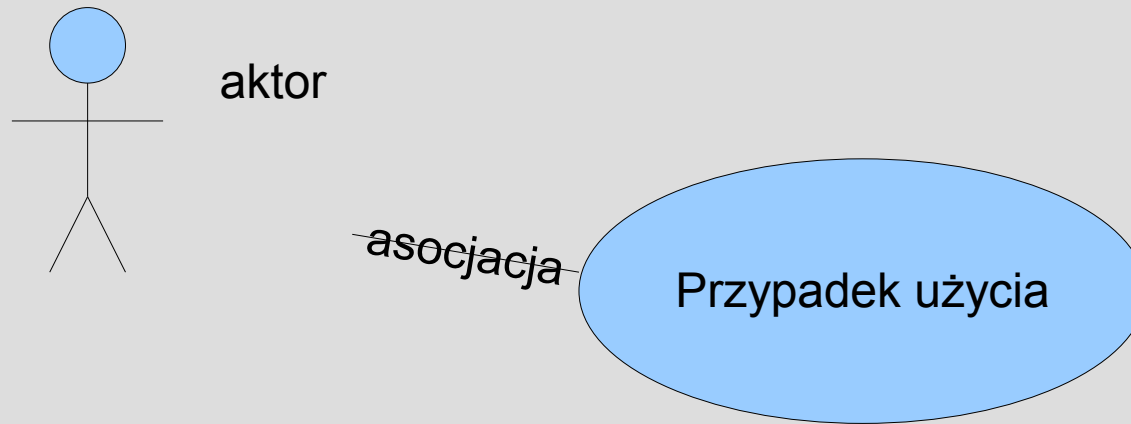
# Diagram przypadków użycia (use case diagram)

- DPU to służy do graficznego przedstawiania przypadków użycia, aktorów (użytkowników, administratorów...) oraz związków zachodzących między nimi w danej dziedzinie przedmiotowej. Określa funkcjonalność analizowanego systemu i sposoby komunikowania się użytkowników z systemem.
- Przypadki użycia określają usługi, świadczone przez system wobec użytkowników

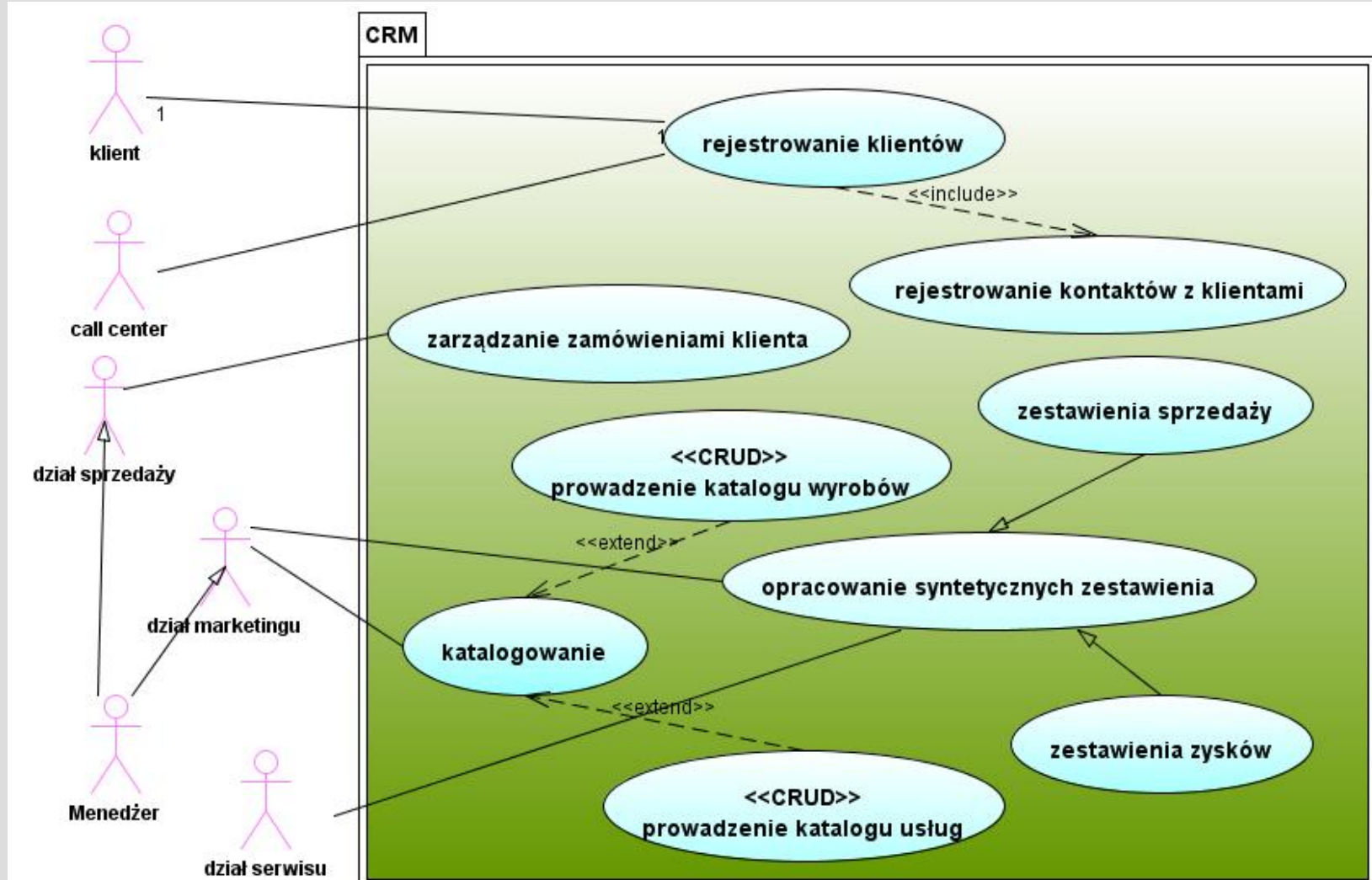
# DPU -podstawowe elementy:

- **Przypadek użycia** = ciąg akcji i ich wariantów wykonywanych przez system w wyniku interakcji z użytkownikami
- **Aktor** = spójny zbiór ról odgrywanych przez użytkowników w czasie interakcji z systemem, klasa użytkowników (ze względu na wykonywane z systemem interakcje)
- **Związki** między aktorami i przypadkami użycia (asocjacja, uogólnienie, zależność, realizacja)


# DPU – elementy graficzne



# DPU - przykład



# Diagram czynności (activity diagram)

-  Jest graficzną reprezentacją sekwencyjnych lub współbieżnych przepływów sterowania oraz danych pomiędzy ciągami czynności, akcji i obiektów

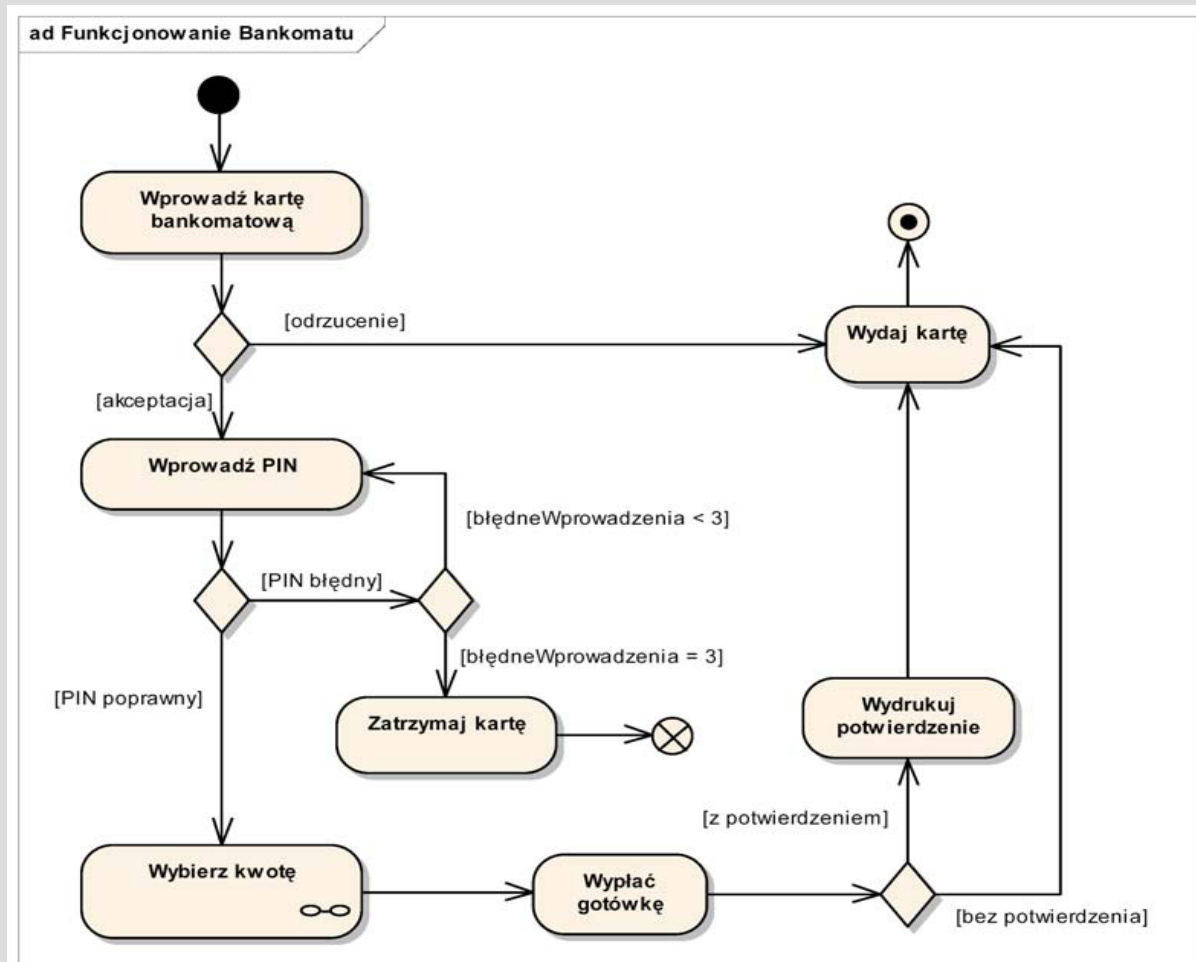
# DC – zastosowania w modelowaniu:

- Wysokopoziomowych procesów biznesowych
- Systemów oraz podsystemów
- Scenariuszy przypadków użycia
- Procesów systemowych charakteryzujących się dużą liczbą równoległych czynności i sytuacji precyzyjnych
- Operacji
- Algorytmów

# DC – podstawowe elementy:

- Czynności i akcje = procesy biznesowe i ich podprocesy;
- Przepływ sterowania – relacja między dwoma czynnościami, reprezentowana strzałką wskazującą na kierunek przekazania sterowania;
- Początek – punkt rozpoczęcia przepływu sterowania
- Koniec – punkt zatrzymania wszystkich przepływów sterowania
- Zakończenie przepływu sterowania

# DC - przykład

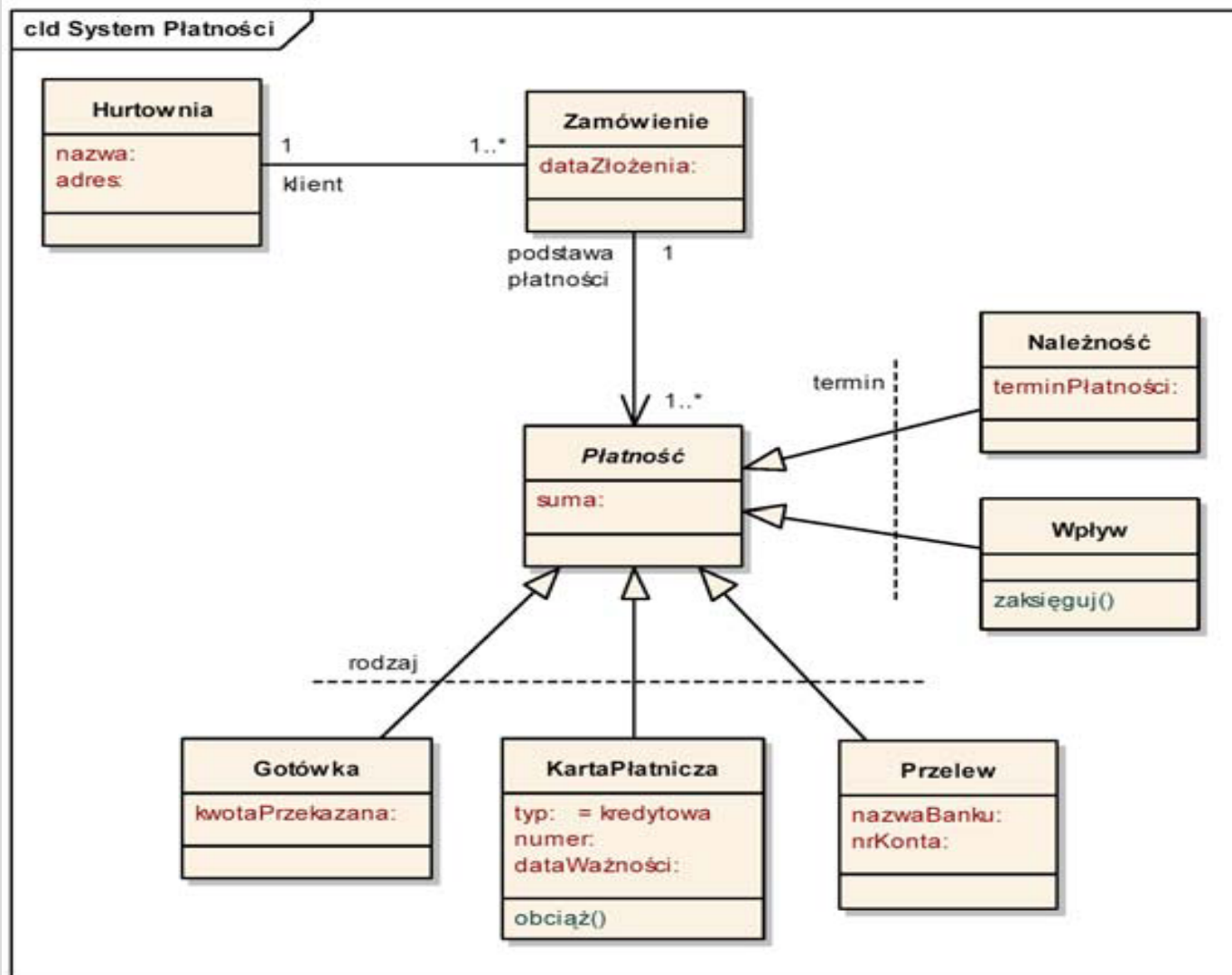




# Diagram klas (class diagram)

- Przedstawiają statyczne, deklaratywne elementy systemu
- Stanowią podstawę obiektowej bazy danych
- Podstawowe elementy: klasy i ich związki:
  - Asocjacja
  - Uogólnienie
  - Zależność
  - realizacja

# DK - przykład



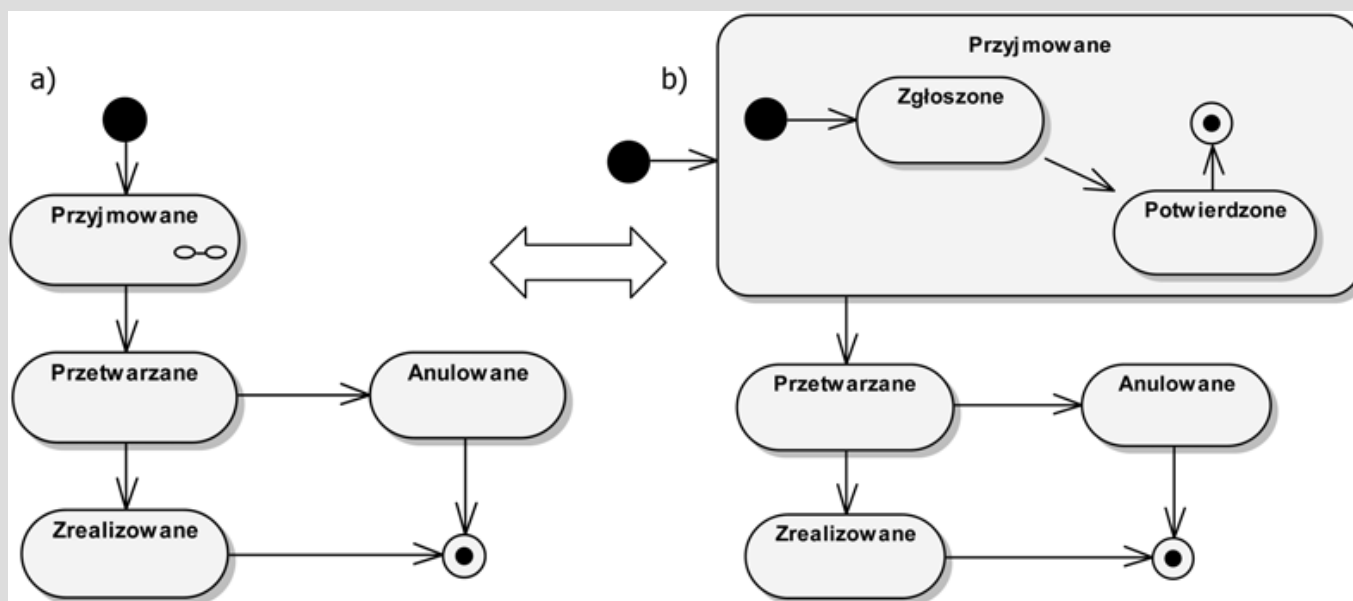
# Diagram maszyny stanowej (state machine diagram)

- DMS to graficzne odzwierciedlenie dyskretnego zachowania systemów stan-przejście
- Wykorzystuje 2 notacje – klasyczną i zakładkową
- Podstawowe pojęcie: stan, przejście, stan początkowy, stan końcowy

# Podstawowe elementy DMS

- Stan = okoliczność lub sytuacja w jakiej znajduje się obiekt w cyklu swojego rzycia spełniając określony warunek, wykonując czynność lub czekając na zdarzenie
- Przejście = relacja między dwoma stanami wskazująca, że obiekt znajdujący się w pierwszym z nich wykona pewne akcje i przejdzie do drugiego stanu, ilekroć będą spełnione określone warunki
- Stan początkowy – zainicjowanie MS
- Stan końcowy – zakończenie MS

# DMS- przykład



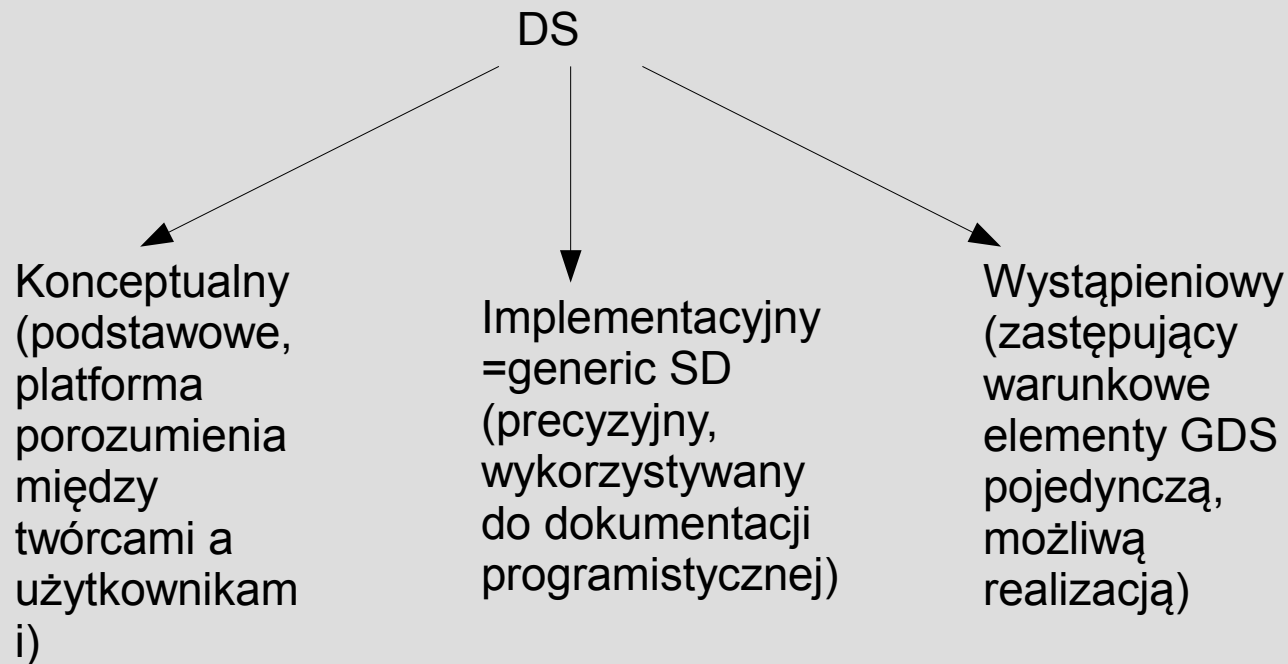
# Diagram Sekwencji (sequence diagram)

- Jest rodzajem diagramu interakcji opisujących interakcje między elementami (instancjami el.) systemu w postaci sekwencji wymienianych między nimi komunikatów
- Właściwości DS odpowiadają koncepcji modelowania obiektowego systemów informatycznych i pozwalają na bezpośrednie generowanie kodu źródłowego w językach obiektowych

# DS

- Interakcja w DS jest opisywana na dwóch poziomach:
  - Poziomym – jako oś na której umieszczono instancje obiektów biorących udział w interakcji (cahrakter statyczny)
  - Pionnowym – jako oś czasu reprezentująca ułożone chronologicznie komunikaty (charakter dynamiczny)

# Rodzaje DS

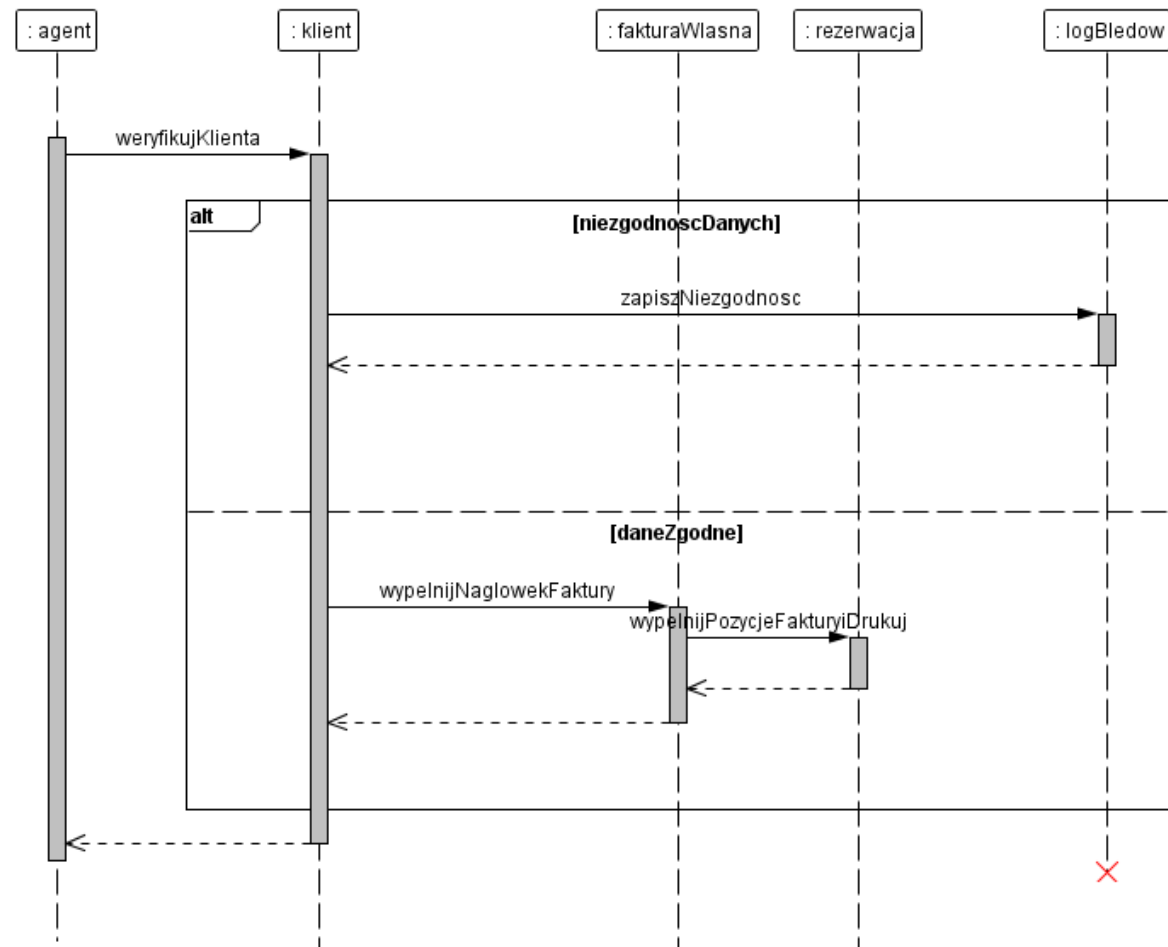




# Podstawowe elementy DS

- Klasyfikator = obiekt (uogólnienie kolekcji elementów o tych samych cechach)
- Komunkat- specyfikacja wyiany informacji między obiektami zawierająca zlecenia wykonania określonej operacji
- Linia życia- powiązana z konkretnym obiektem linia wskazująca okres jego istnienia
- Ośrodek sterowania – specyfikacja wykonywania czynności, operacji lub innego typu zachowania w trakcie interakcji

# DS - przykład



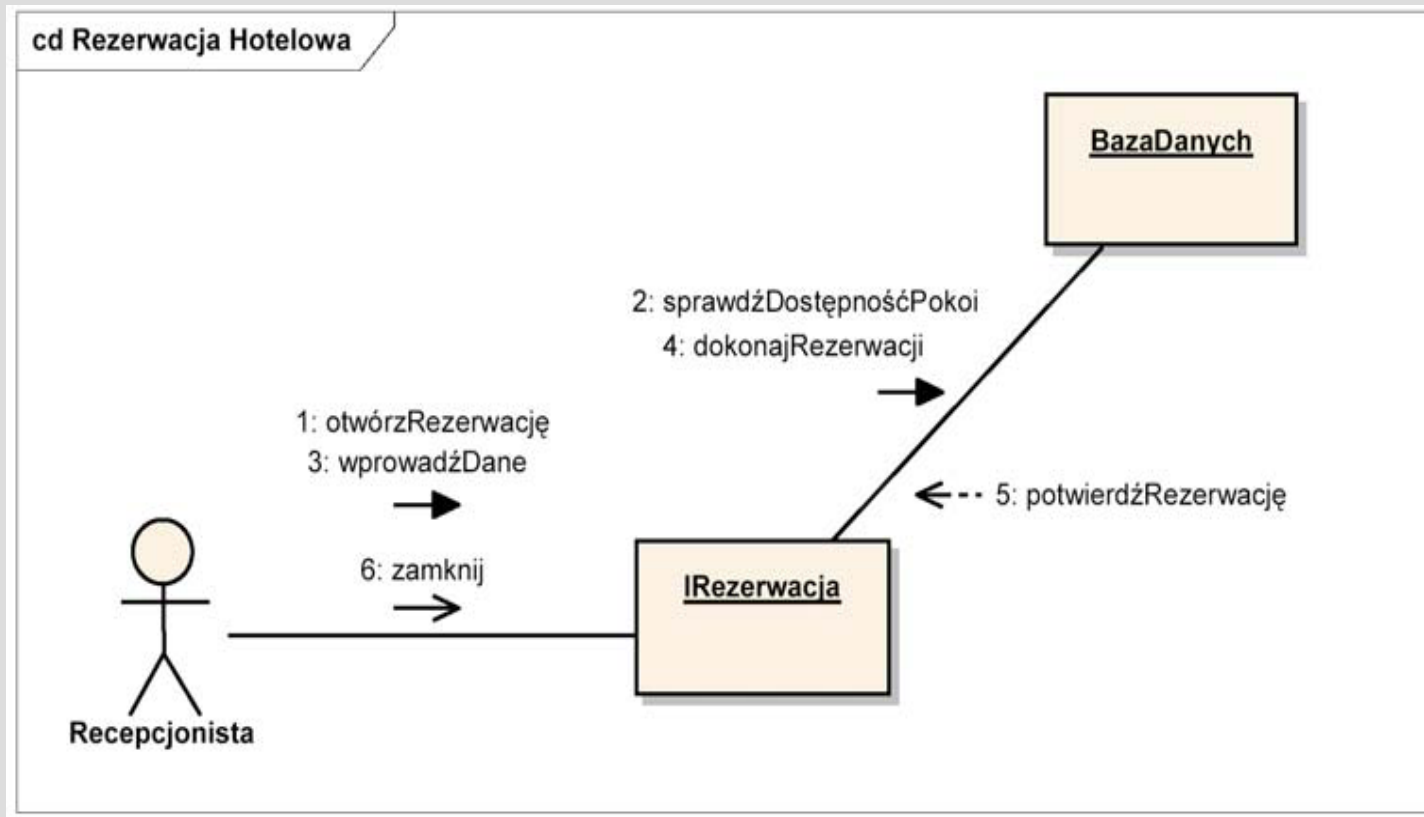
# Diagram komunikacji (communication diagram)

- Rodzaj diagramu interakcji specyfikujący strukturalne związki między obiektami biorącymi udział w interakcji oraz wymianę komunikatów między tymi obiektami
- Posiada 2 składniki:
  - Strukturalną organizację obiektów wyrażoną asocjacjami
  - Interakcje między obiektami realizowane przez komunikaty przyporządkowane do asocjacji

# Dkom – podstawowe elementy

- Klasyfikator (obiekt)
- Asocjacja
- Komunikat
  - Niezbędna numeracja oraz podanie kierunku i typu komunikatu

# DKom - przykład



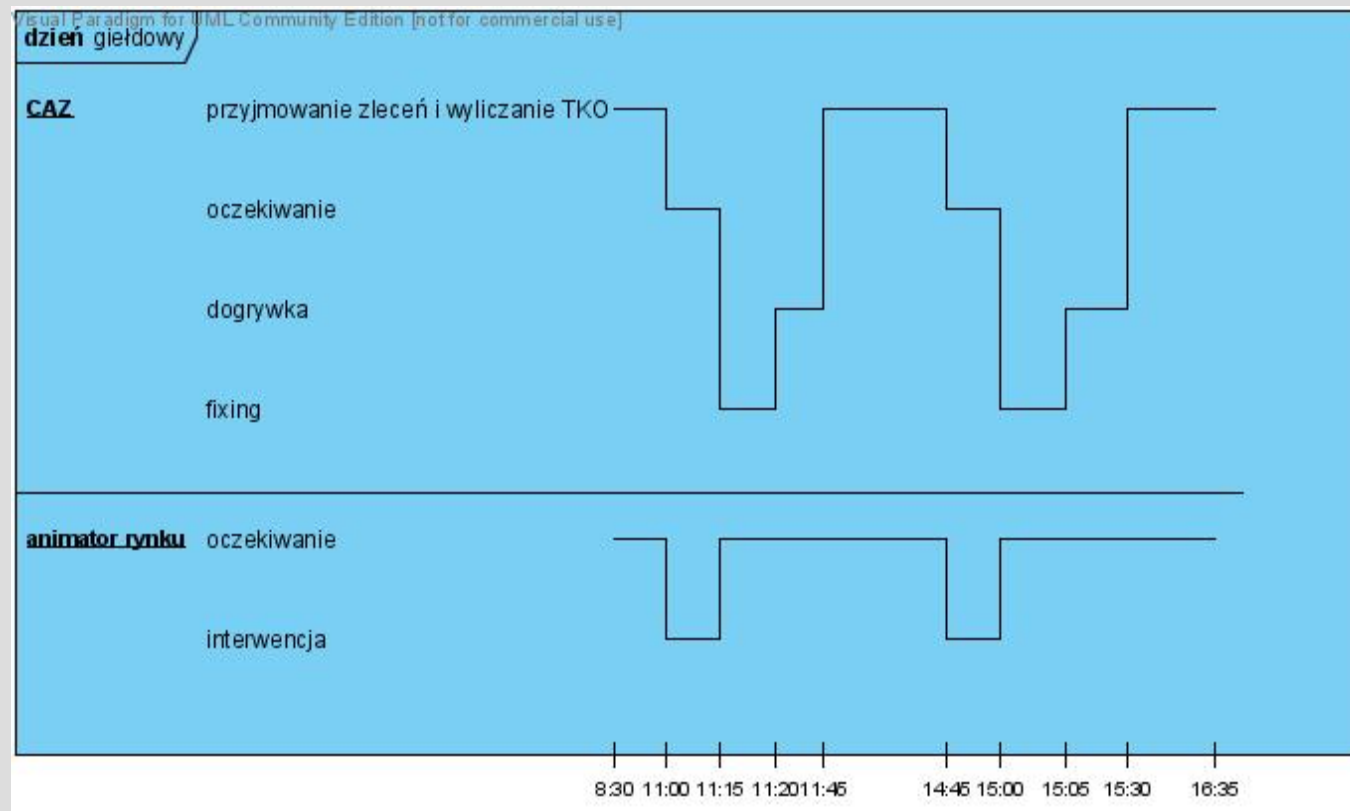
# Diagram harmonogramowania (timing diagram)

- Rodzaj diagramu interakcji reprezentującym na osi czasu zmiany dopuszczalnych stanów, jakie może przyjmować obiekt uczestniczący w interakcji
- Służy do sporządzenia harmonogramów interakcji

# DH – podstawowe elementy

- Klasyfikator (obiekt)
- Nazwa stanu (bezczynność, czuwanie, oczekiwanie, wykonywanie, obliczanie ...)
- Linia zmiany stanów obiektu

# DH - przykład





# Diagram sterowania interakcją (interaction overview diagram)

- Rodzaj diagramu interakcji dokumentujący przepływ sterowania między logicznie powiązanymi diagramami i fragmentami interakcji z wykorzystaniem kategorii modelowania diagramów czynności.

# DSI - elementy

- Podstawowe i wybrane zawansowane elementy diagramów czynności
- Wybrane kategorie pojęciowe poszczególnych diagramów interakcji (sekwencji, komunikacji, harmonogramowania) - np poprzez odwołanie do gotowych poddiagramów interakcji

[illegible]

# Diagramy wdrożeniowe

- Opisują aspekty fizyczne systemu informatycznego. Rozróżniamy 2 rodzaje takich diagramów:
  - Komponentów – pozwalają na modelowanie elementów oprogramowania i związków między nimi
  - Rozlokowania – pozwalają na modelowanie rozmieszczenia infrastruktury sprzętowej oraz platform użytkowania systemu
- Jest to abstrakcyjna kategoria diagramów

# Diagram komponentów

- Komponenty = moduły oprogramowania realizujące jego usługi za pomocą interfejsów
- Podstawowe elementy:
  - Komponent
  - Interfejs udostępniający
  - Interfejs pozyskujący
  - port/port złożony
  - Zależność, realizacja,
  - Konektor delegowany, składany

# Diagram rozlokowania

- Dokumentuje fizyczne rozlokowanie elementów systemu
- Rozlokowanie = alokacja artefaktów (sztucznie wytworzonych produktów) lub ich instancji w określonym węźle
- DR przedstawia sieć połączonych ścieżkami komunikowania węzłów z umieszczonymi w nich artefaktami

# Literatura/źródła

- S. Wrycza, B. Marcinkowski, K. Wyrzykowski, „Język UML 2.0 w modelowaniu systemów informatycznych”, Helion, 2005
- P. Graessle, H. Baumann, P. Baumann, „UML 2.0 w akcji. Przewodnik oparty na projektach”, Helion, 2006
- <http://www.uml.org/>
- <http://www.omg.org/spec/UML/2.2/> - dokumenty związane z UML 2.2, specyfikacje