Piotr Dobosz

Język UML. Zastosowanie

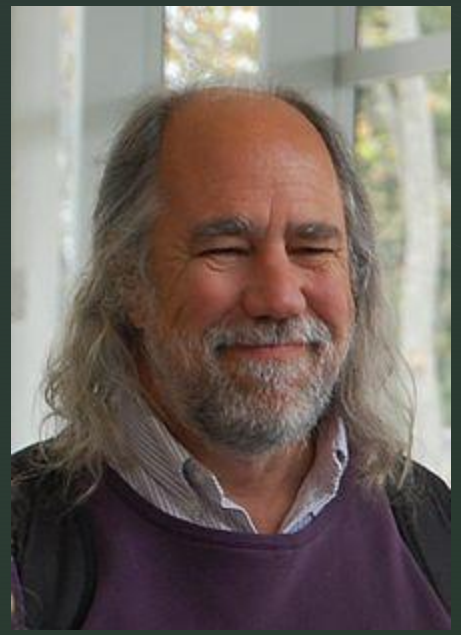
Wstęp

- W jaki sposób programujemy
- Małe projekty i ich charakterystyka
- Duże projekty, duży chaos
- … a dokumentacja do późniejszego rozwoju?

UML

- Unified Modeling Language
- Pozwala na modelowanie systemów (np. informatycznych)
- Język półformalny
- Stworzony przez…

Grady Booch, James E. Rumbaugh oraz Ivar Hjalmar Jacobson



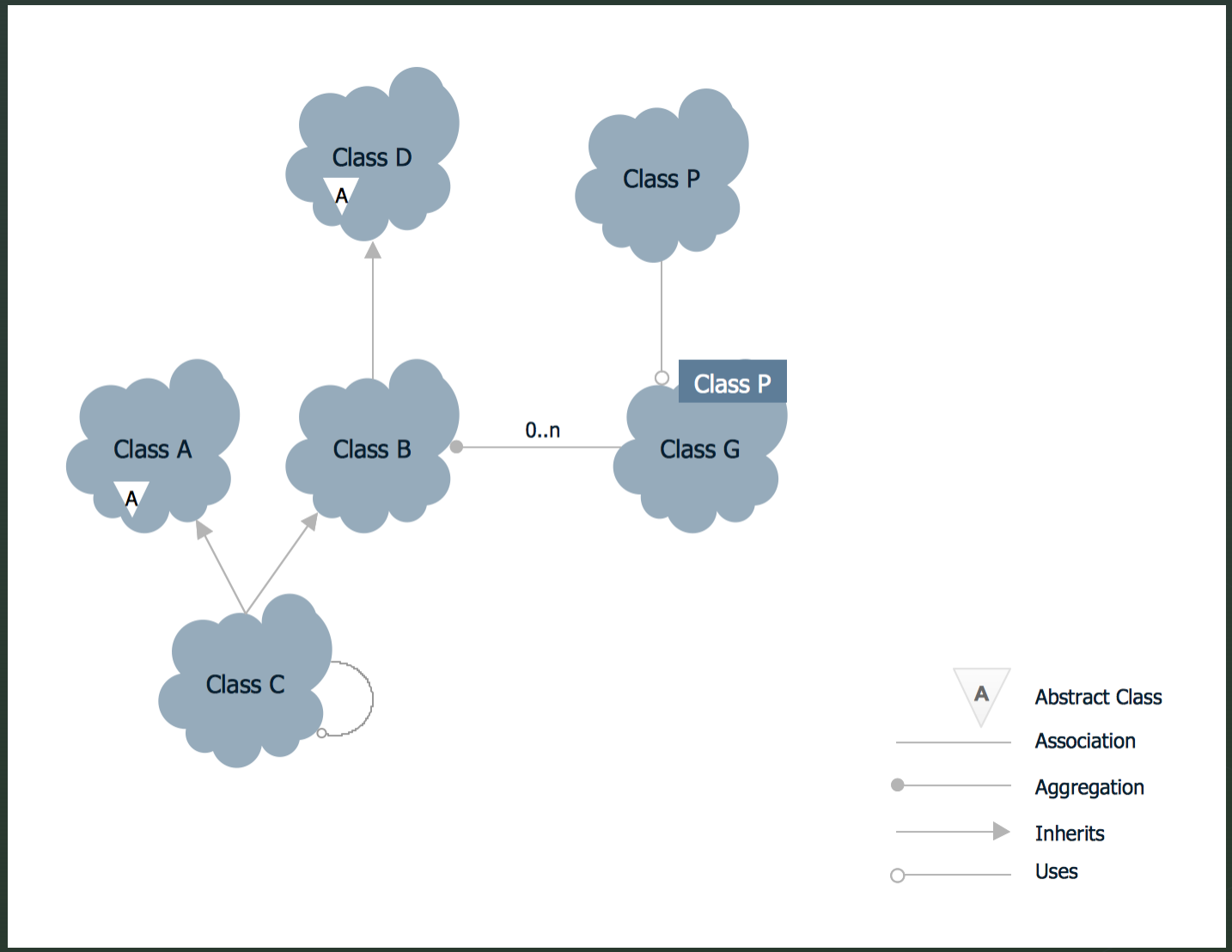
Historia UML

- Modelowanie obiektowe to w zasadzie cel całej inżynierii oprogramowania (od lat 70 XX wieku)
- Powstało wiele odmian modelowania obiektowego (w zasadzie każdy język posiada własne zasady i modele)
- Do lat 90 XX wieku powstało około 50 RÓŻNYCH OD SIEBIE notacji obiektowych

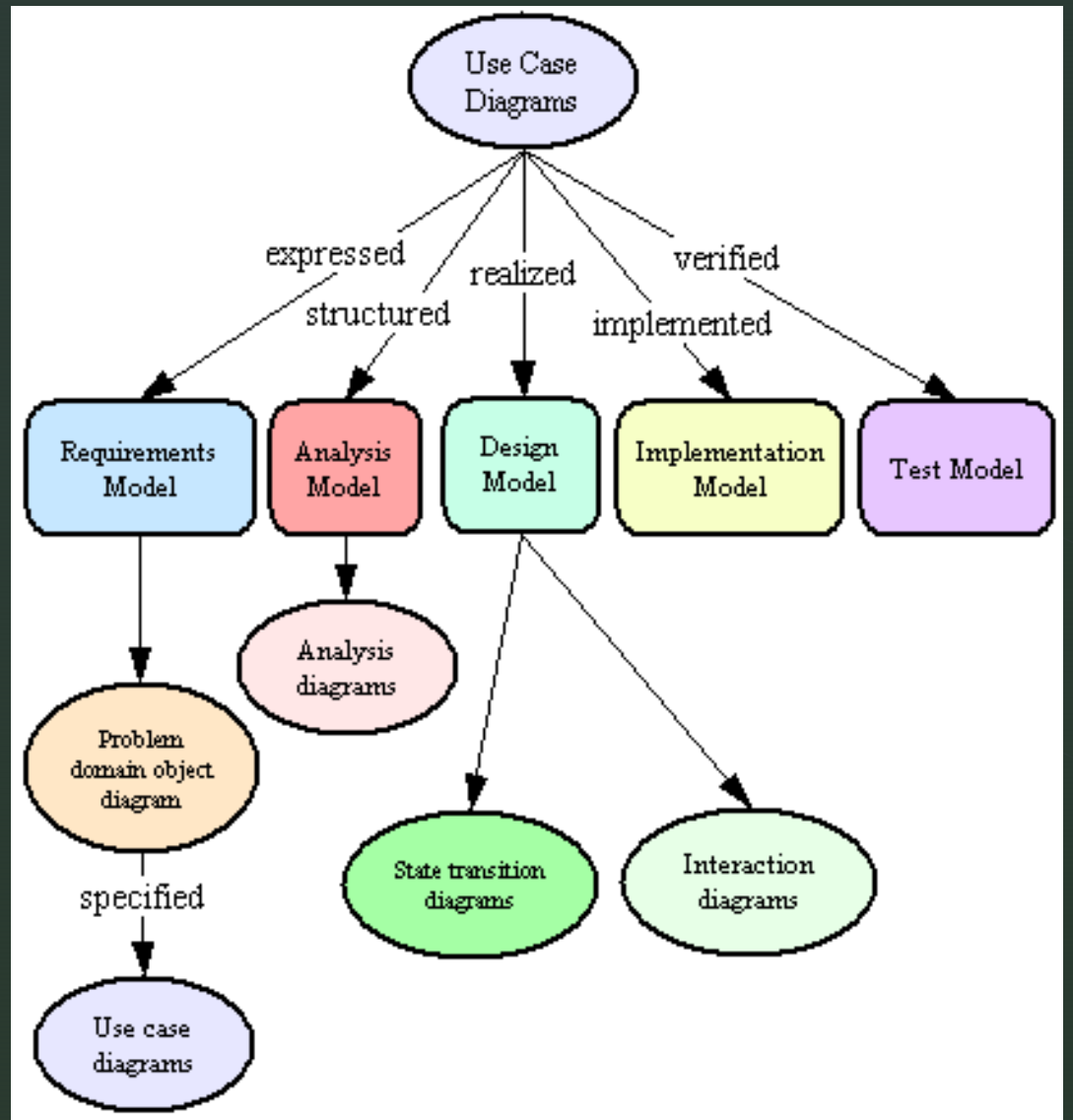
Historia UML

- Tylko kilka metod znalazło szersze zastosowanie
- Najbardziej znane to: Booch method, Object-Oriented Software Engineering, Object Modeling Technique

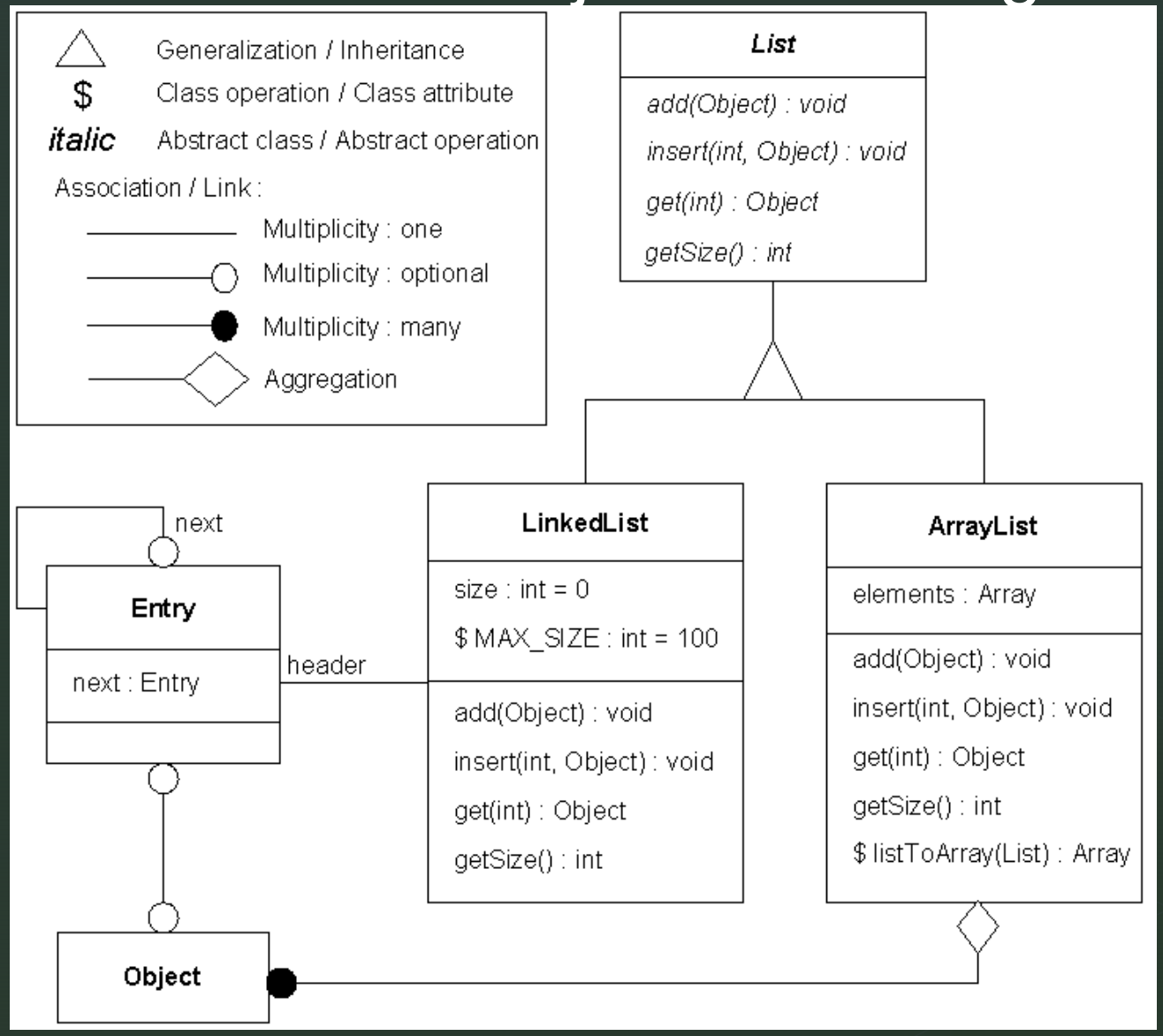
Booch method



Object-Oriented Software Engineering



Object Modeling Technique



Rozwój unifikacji

- Około 1995 roku trzej twórcy wspomnianych rozwiązań nawiązują współpracę
- Rok 1996 – wydana zostaje specyfikacja z numerem 0.9 Unified Method
- Zawiązane konsorcjum UML wydaje w styczniu 1997 roku wersję 1.0 UML

Dalsze losy UML

- Dokumentacje i rozwój przejmuje Object Management Group
- powstają kolejne odsłony UML, w tym wersja 1.42.1 (ISO/IEC 19501)
- Wersja 2.0 - czerwiec 2005
- Wersja 2.4.1 to normalizacja ISO/IEC 19505-1 i 19505-2

Kilka słów o OMG

- Powstało w 1989
- Zrzesza takie firmy i instytycje jak Dell, Microsoft, Fujitsu, Ford MotorCompany, MIT czy NASA (cała lista - https://www.omg.org/cgi-bin/apps/membersearch.pl)
- Nadrzędny cel - utworzyć standard międzyobiektowej komunikacji (teoretycznie... udany – CORBA)
- To oni standaryzują i opiekują się UML

Metody UML

- Sam pół-język nie jest metodą!
- Bo jest podwaliną dla innych metod!
- Przykładowo UML stał się bazą dla Rational Unified Process

Rational Unified Process

- Iteracyjny proces wytwarzania oprogramowania
- To szablon procesu - można go dostosowywać jak chcemy
- Nawiązuje do spiralnego procesu życia oprogramowania (i tworzenia)

RUP – najlepsze praktyki

- iteracyjne wytwarzanie oprogramowania (Iterative Development)
- zarządzanie wymaganiami (Requirement Management)
- używanie architektury bazującej na komponentach (Component-based architecture)
- graficzne projektowanie oprogramowania
- kontrola jakości oprogramowania (Quality Assurance)
- proces kontroli zmian w oprogramowaniu (Change Management)

Iteracyjne wytwarzanie oprogramowania

- Integracja oprogramowania robiona krok po kroku podczas wytwarzania oprogramowania, ograniczając go do mniejszej liczby elementów
- Integracja jest prostsza i mniej kosztowna
- Składowe oprogramowania są projektowane oddzielnie i łatwiej użyć je ponownie
- Łatwiej wykrywać zmiany wymagań i łatwiej nimi zarządzać
- Zagrożenia identyfikowane i atakowane są wcześnie ponieważ każda iteracja pozwala wykryć kolejne zagrożenia
- W iteracjach ulepszana jest architektura oprogramowania

Zarządzanie wymaganiami

- Analiza problemu
- Zrozumienie potrzeb udziałowców (stakeholders)
- Definicja systemu
- Zarządzanie zakresem systemu
- Zawężanie definicji systemu
- Zarządzanie zmianami wymagań

Dyscypliny (tzw. Główne klocki)

- Rola (Roles) – Kto?
- Produkt (Work Products) – Co?
- Zadanie (Tasks)

Dyscypliny inżynierskie

- Modelowanie biznesowe (Business modeling)
- Wymagania (Requirements)
- Analiza i projekt (Analysis and design)
- Implementacja (Implementation)
- Testowanie (Test)
- Wdrożenie (Deployment)

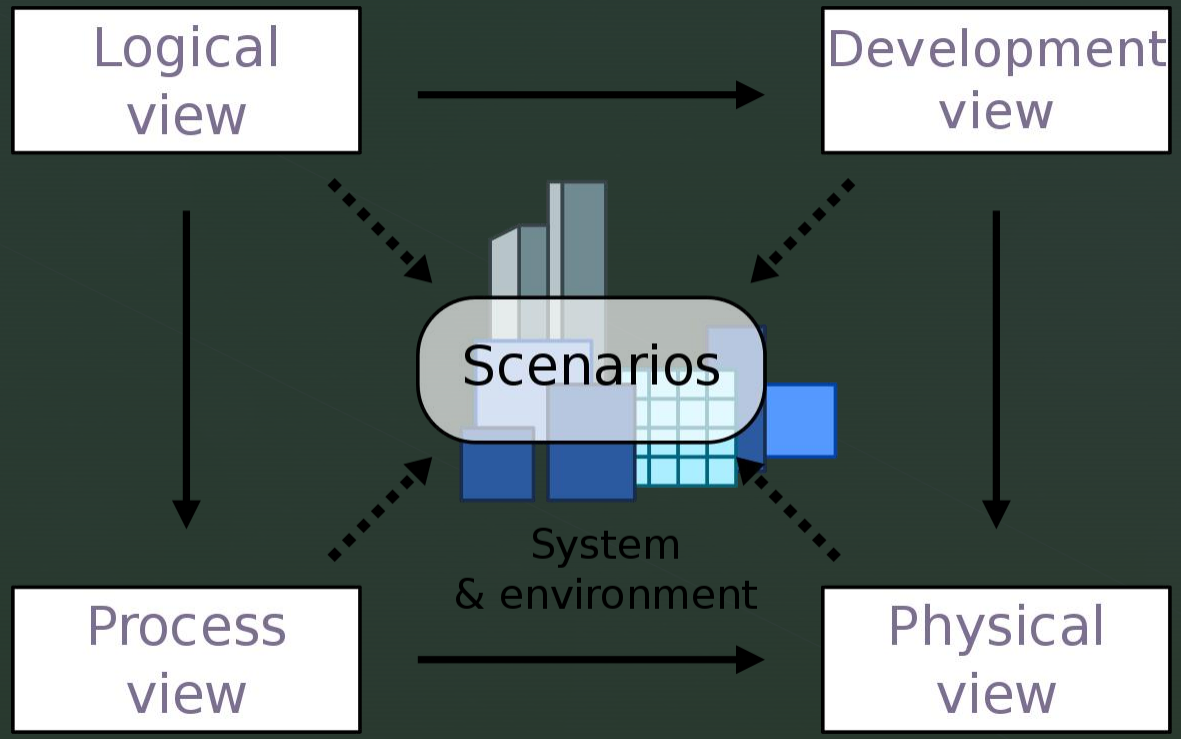
Dyscypliny pomocnicze

- Zarządzanie zmianami oraz konfiguracją (Configuration and change management)
- Zarządzanie projektem (Project management)
- Środowisko (Environment)

Za i przeciw RUP (głównie przeciw?)

- Ciężki i kosztowny (?)
- Nieograniczona możliwość dostosowania to same problemy (?)
- RUP oprata jest na modelu prespektyw architektonicznych 4+1: logiczna, implementacyjna, procesowa, wdrożeniowa i perspektywa przypadków użycia
- perspektywa przypadków użycia - najważniejsza

Model architektury 4+1 (Philippe'a Kruchtena)



Budowa UML

Elementy standardu UML 2.x

- Metamodel – baza superstruktur
- Superstruktury - notacja i semantyki diagramów i ich elementów
- Objektowe ograniczenia językowe (OCL – Object Constraint Language) - definiowanie reguł dla elementów modelu
- Format wymiany plików pomiędzy aplikacjami UML (oraz narzędziami wspomagającymi)

Diagramy

- Wersja UML 2.x wprowadza 14 rodzajów diagramów
- Plus 3 tzw. abstrakcyjne (struktury, zachowania, interakcje)
- Każdy z diagramów posiada odmienne zastosowanie, składające się na całościowe zaprojektowanie w pełni funkcjonalnego systemu (do wdrożenia)

Diagramy struktur

- Klas (najczęściej spotykane, ang. class diagram)
- Obiektów (ang. object diagram)
- Komponentów (ang. component diagram)
- Wdrożenia (ang. deployment diagram)
- Struktur złożonych (ang. composite structure diagram)
- Pakietów (ang. package diagram)
- Profili (ang. profile diagram, nowość wprowadzona w UML 2.2)

Diagramy zachowań

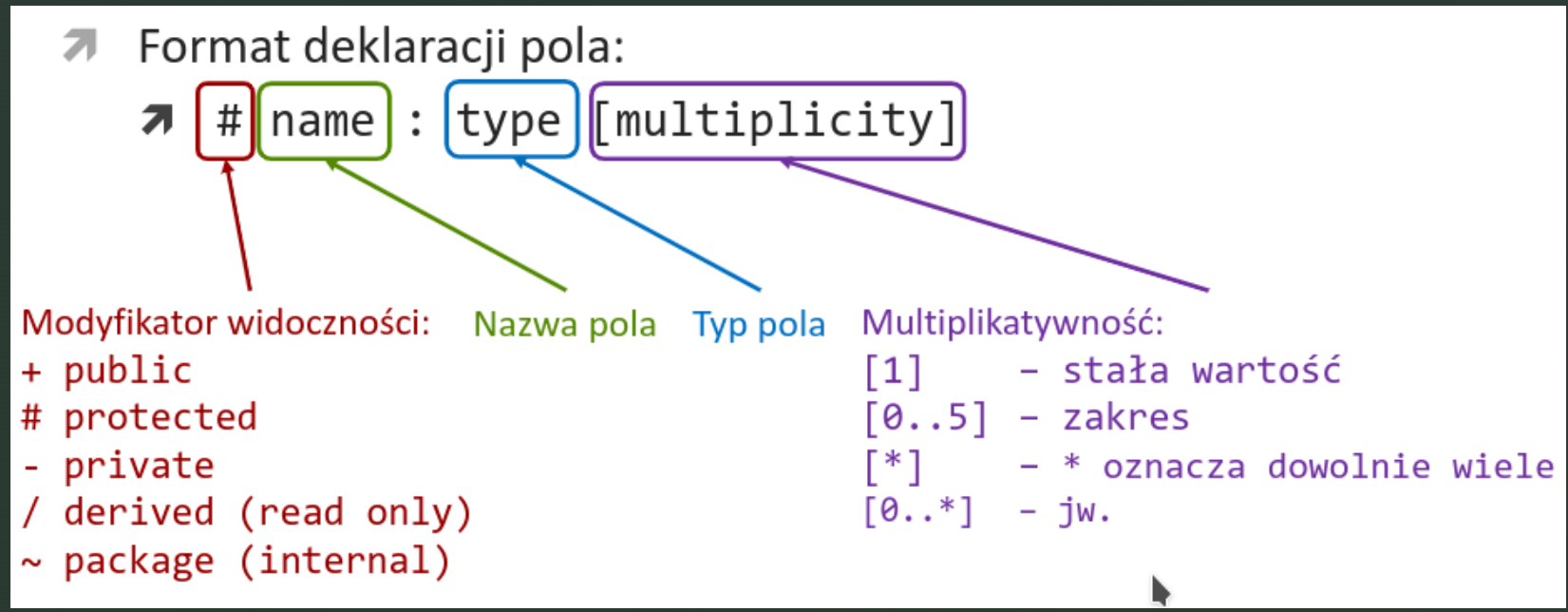
- Czynności (ang. activity diagram)
- Przypadków użycia (ang. use case diagram)
- Maszyny stanów (ang. state machine diagram)
- Interakcji (diagram abstrakcyjny)

Diagam klas

- Przedstawia strukturę systemu w modelach obiektowych
- Przedstawia klasy (typy) obiektów
- używa się do modelowania statycznych aspektów perspektywy projektowej

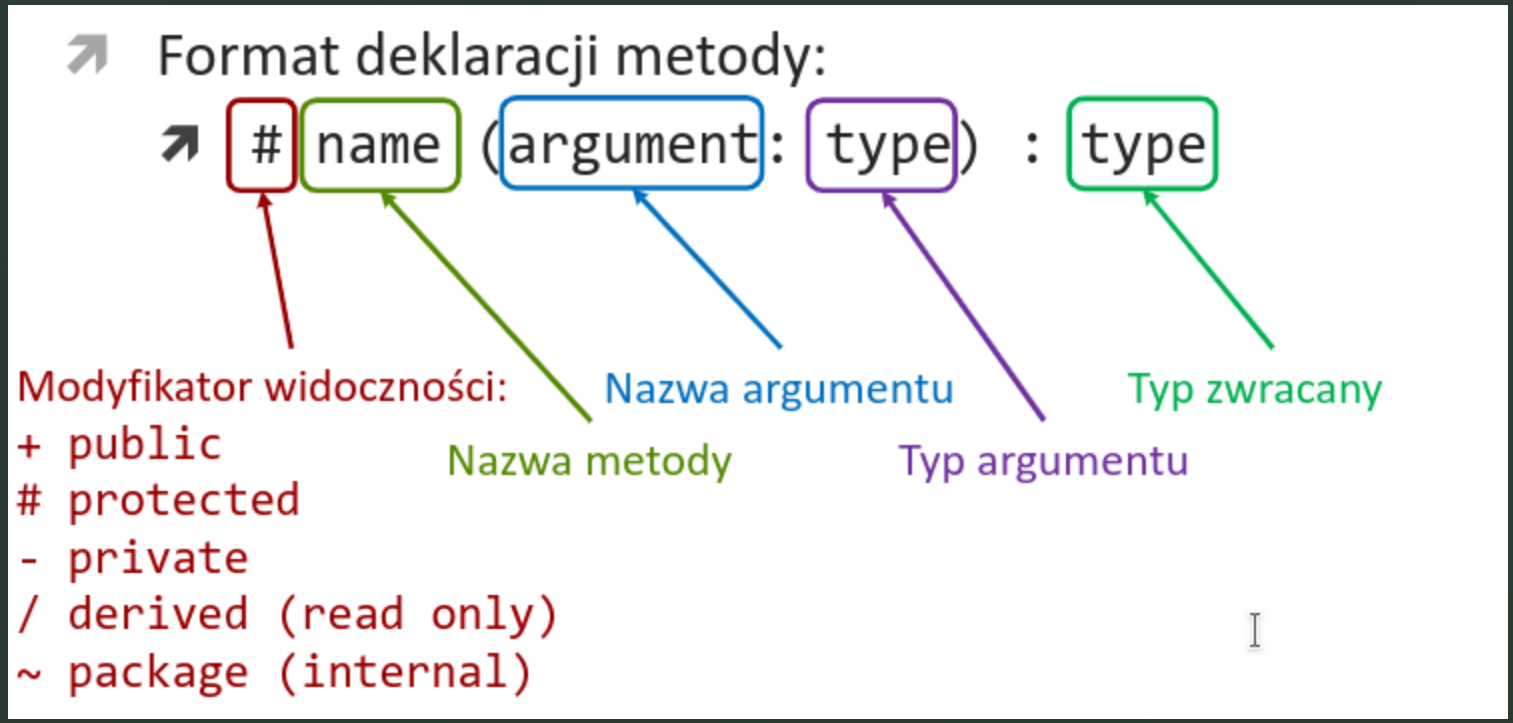
Specyfikacja pól klasy

- Format deklaracji pola:
  - `# name : type [multiplicity]`

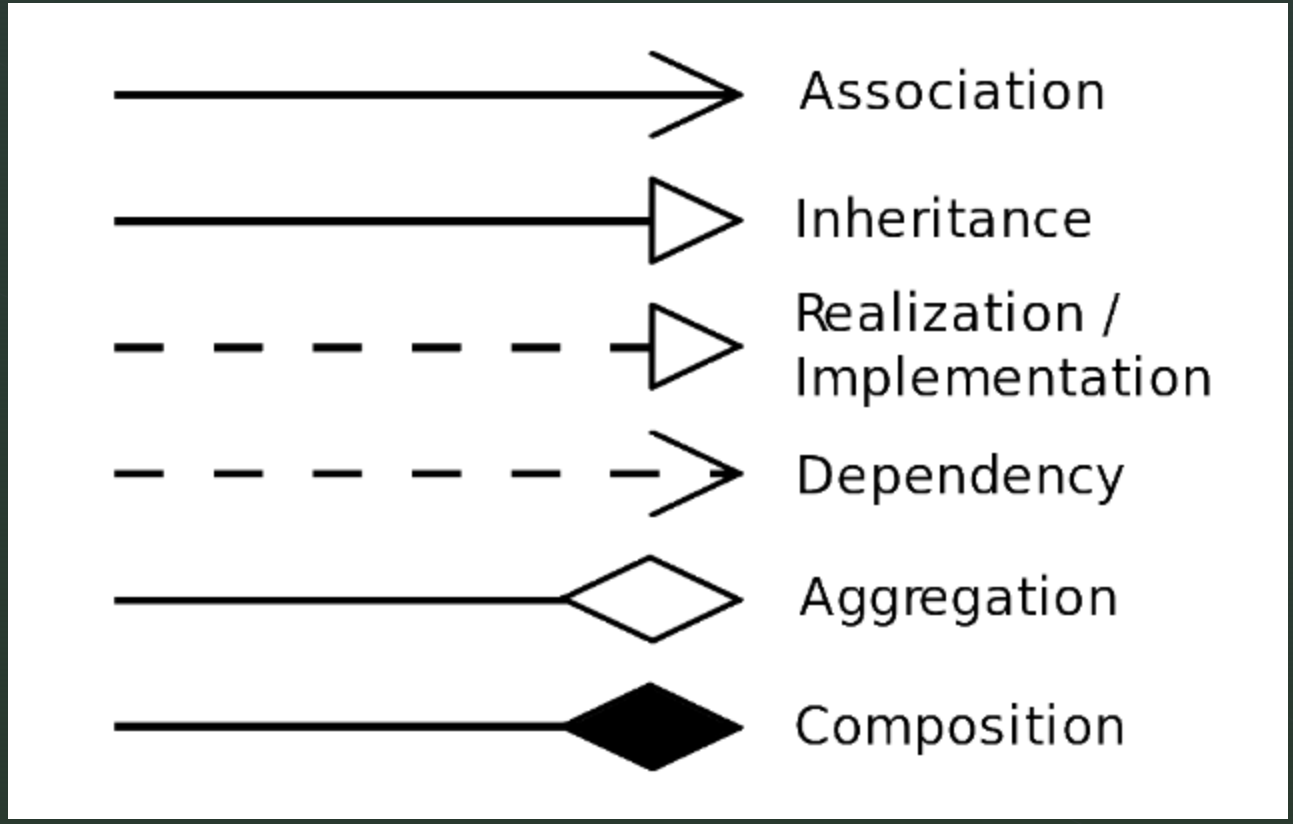


Specyfikacja metod klasy

- Format deklaracji metody:
  - `# name (argument: type) : type`



Związki klas w UML



Opis związków UML

- Zależność
- Asocjacja
- Generalizacja
- Agregacja
- Kompozycja

Przykład diagramu klas

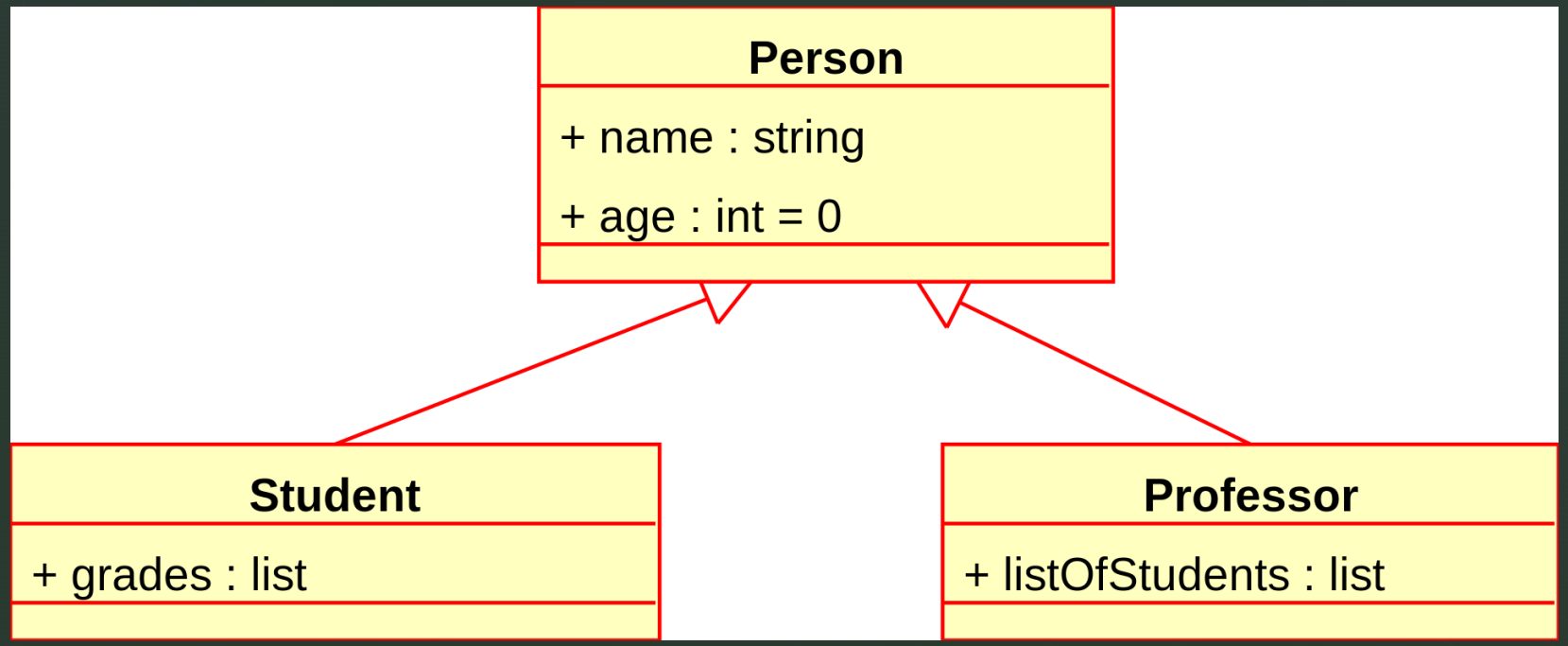


Diagram obiektów

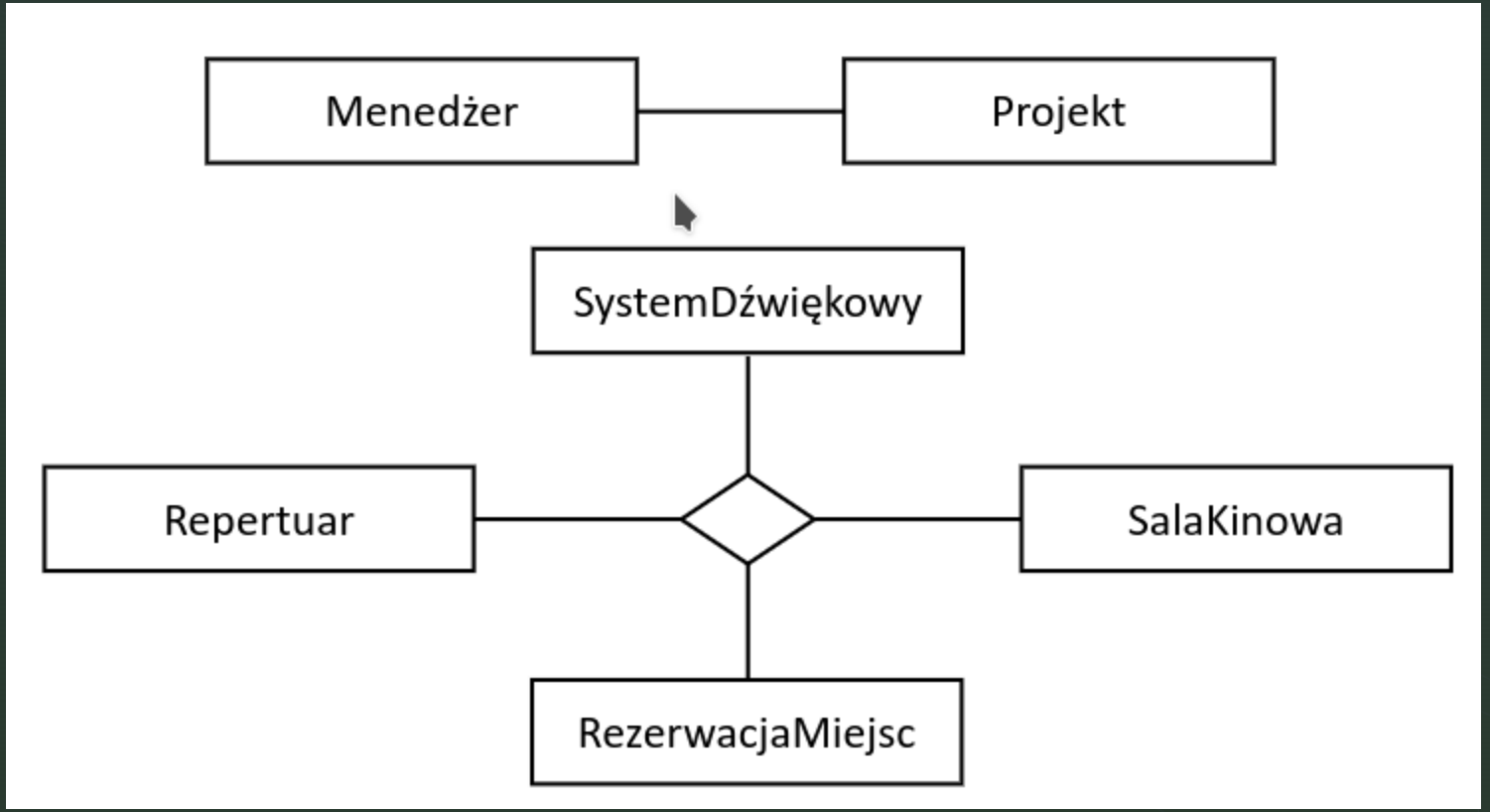


Diagram kompomentów

- Określa jakie komponenty należą do jakich pakietów
- Odpowiada na pytanie dotyczące organizacji komponentów...
- ...oraz powiązań między komponentami

Diagram komponentów

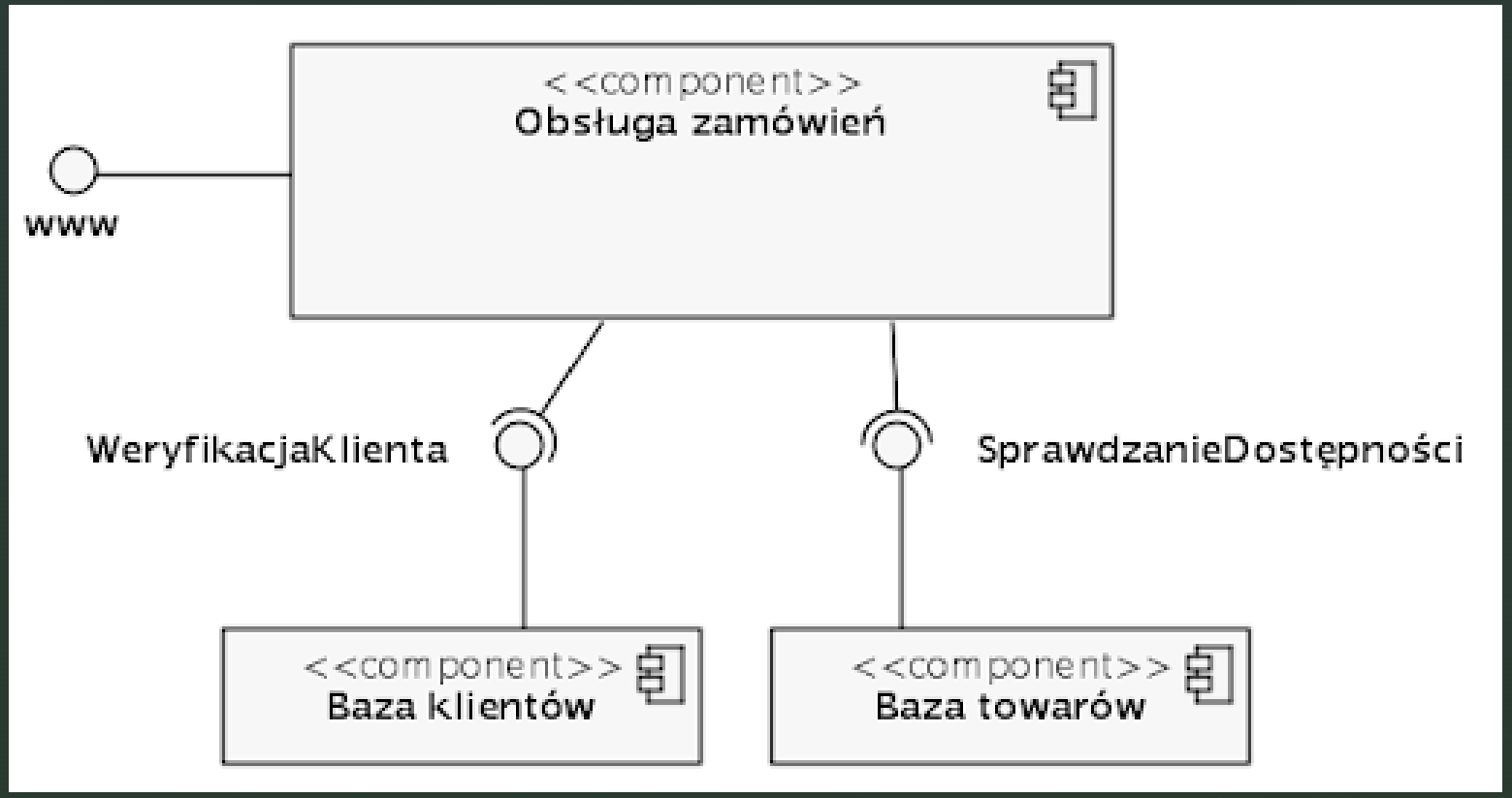


Diagram wdrożenia

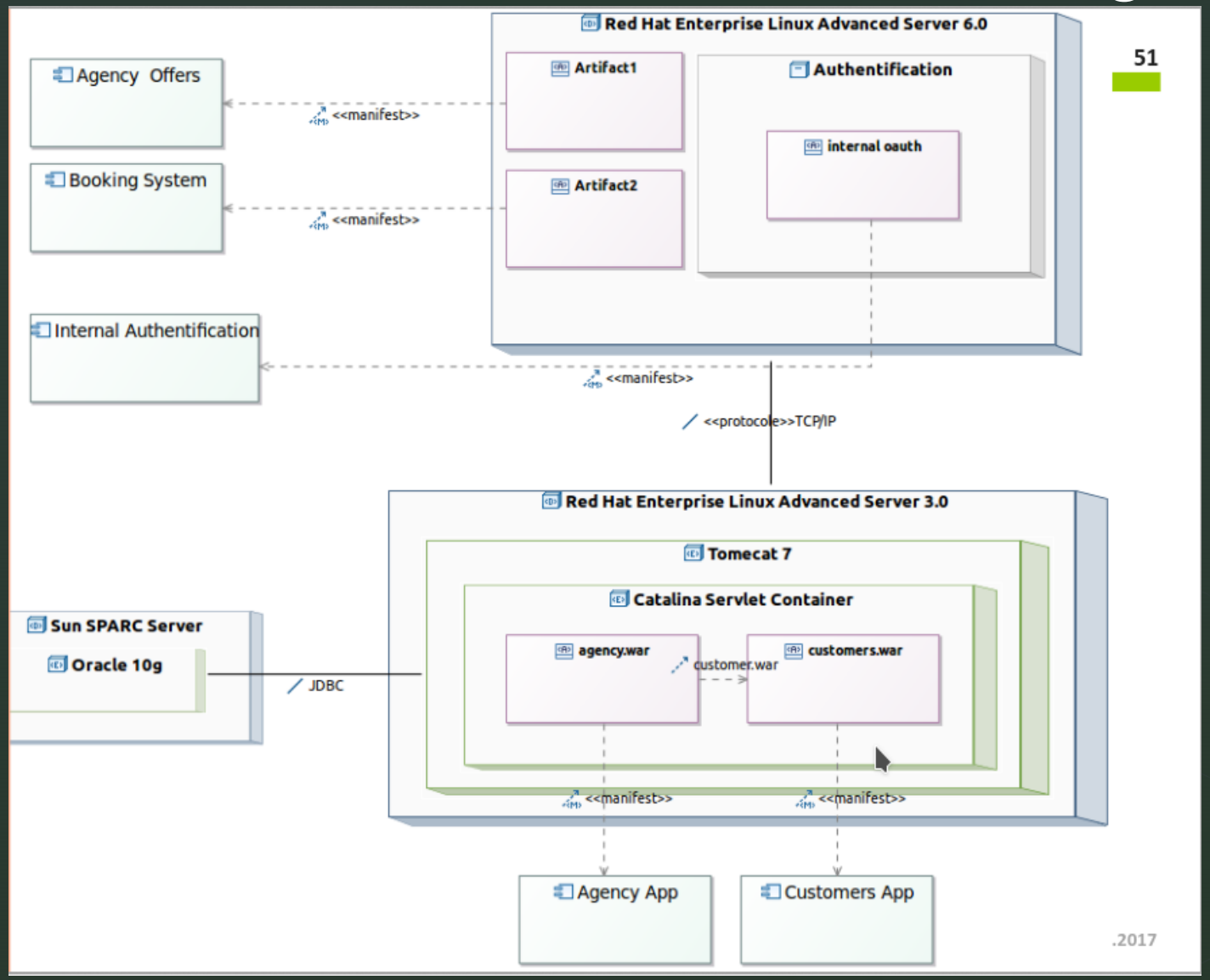


Diagram pakietów

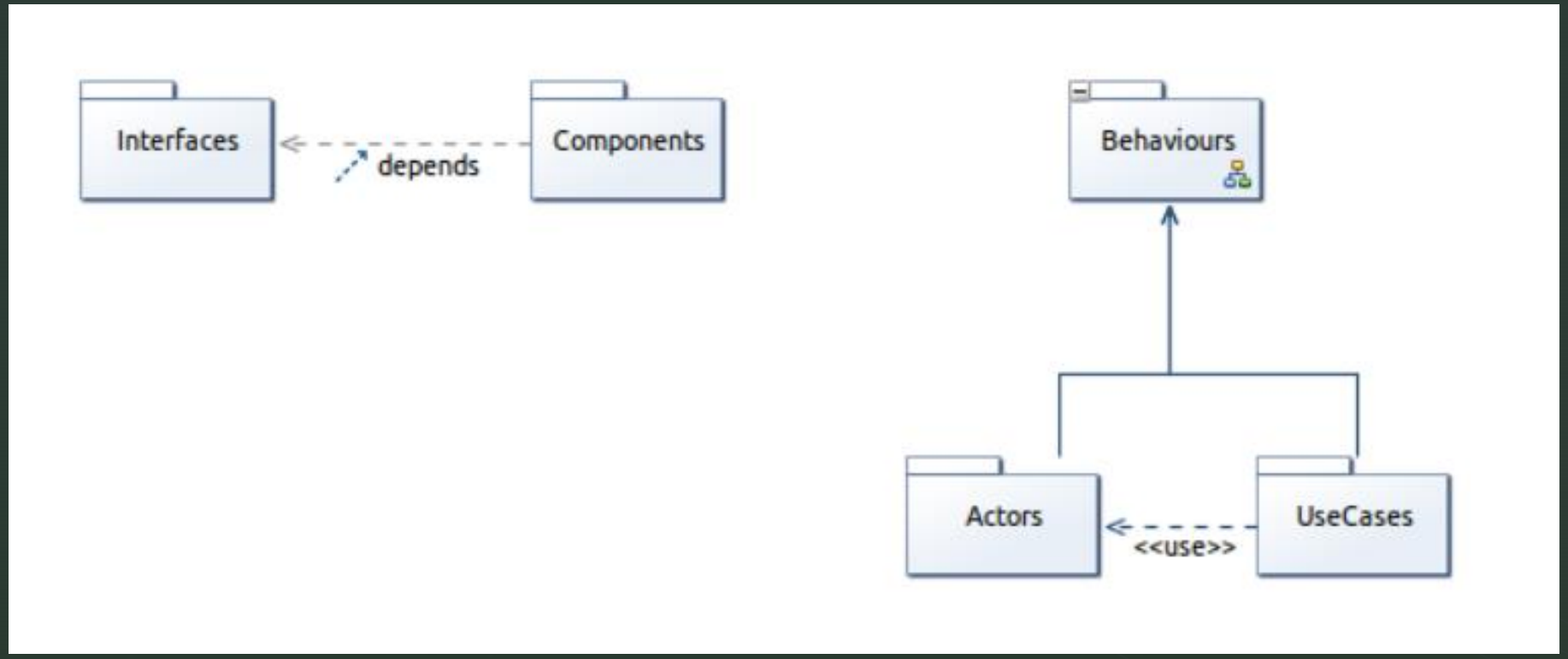


Diagram profili

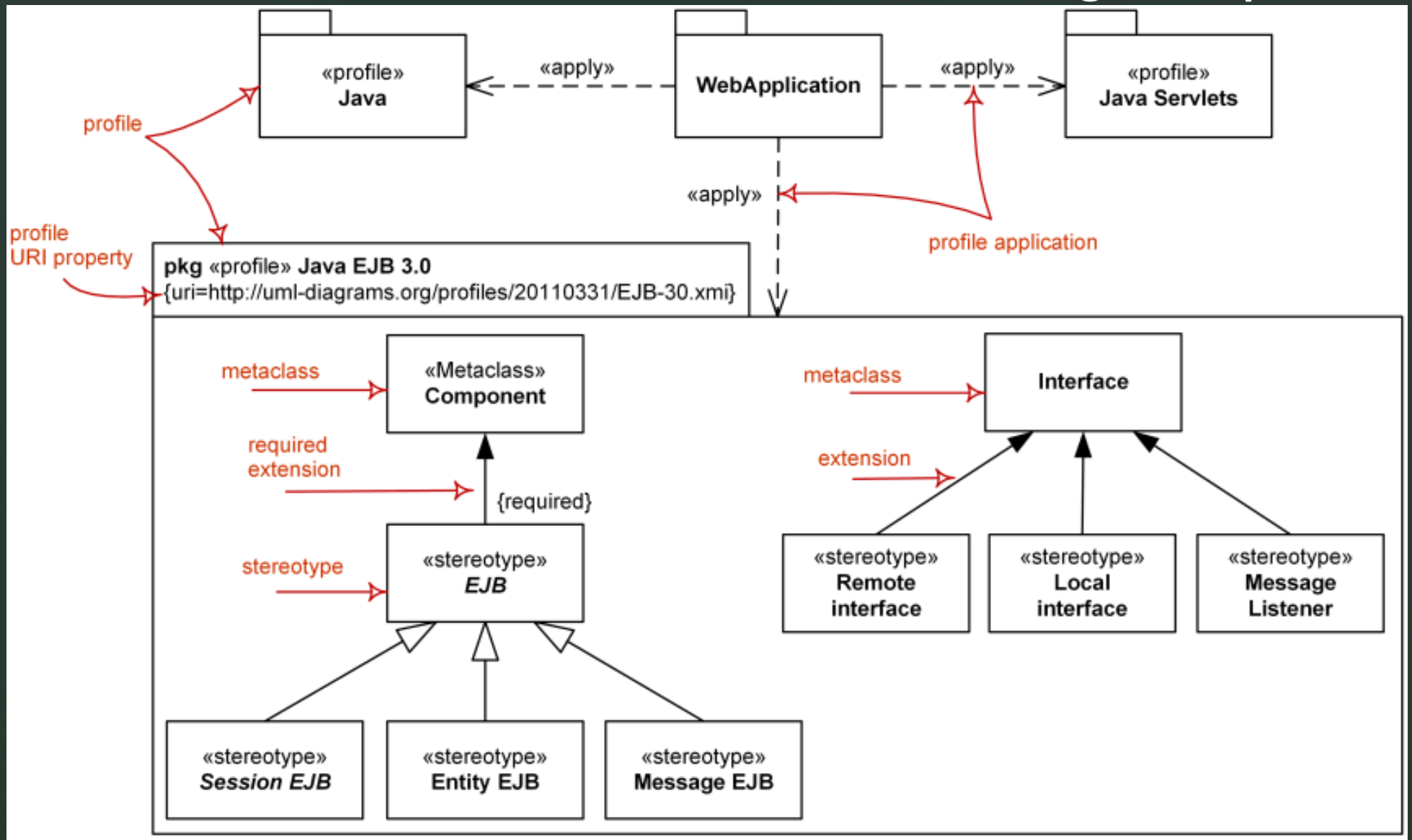


Diagram czynności

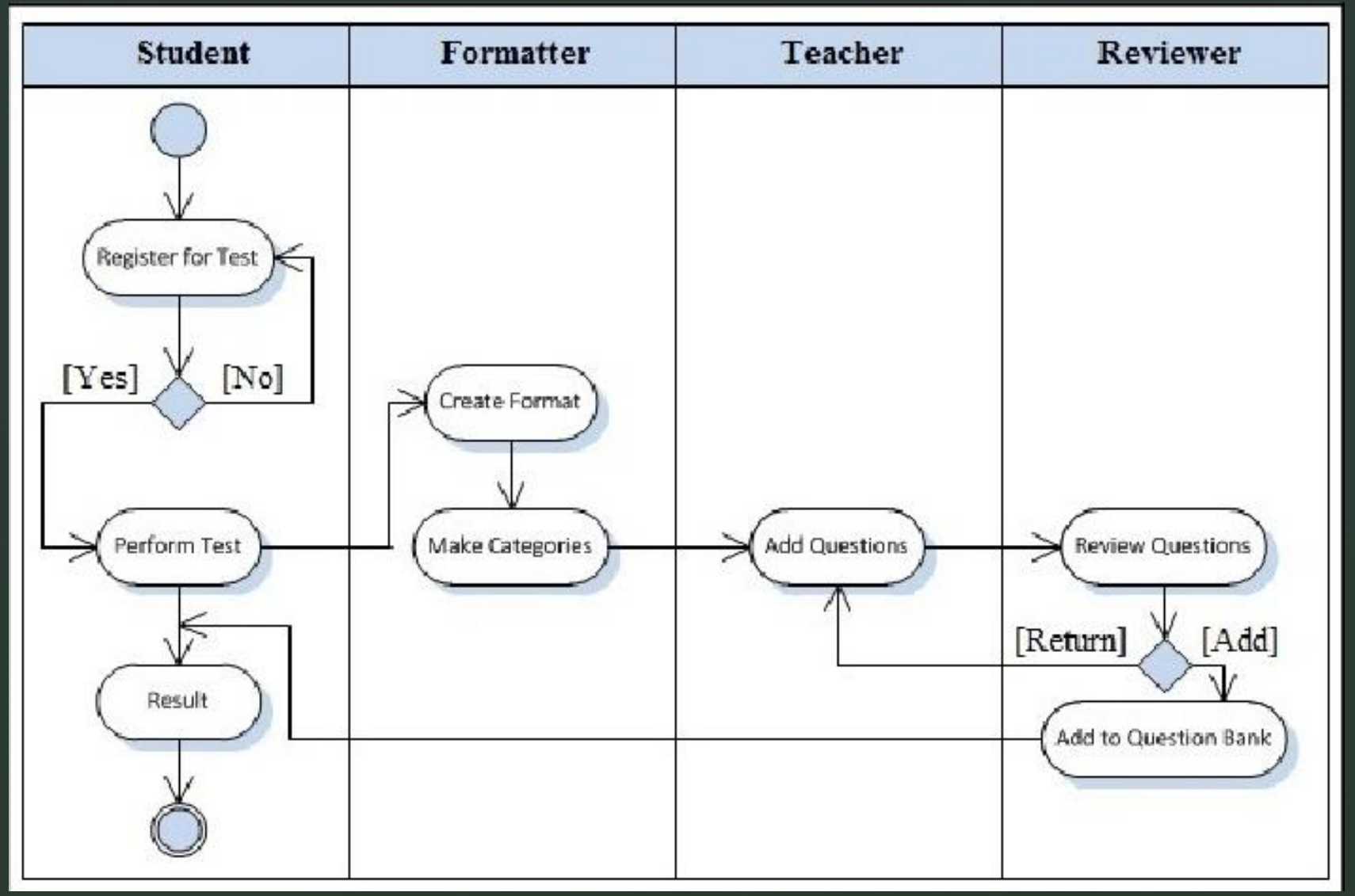


Diagram przypadków użycia

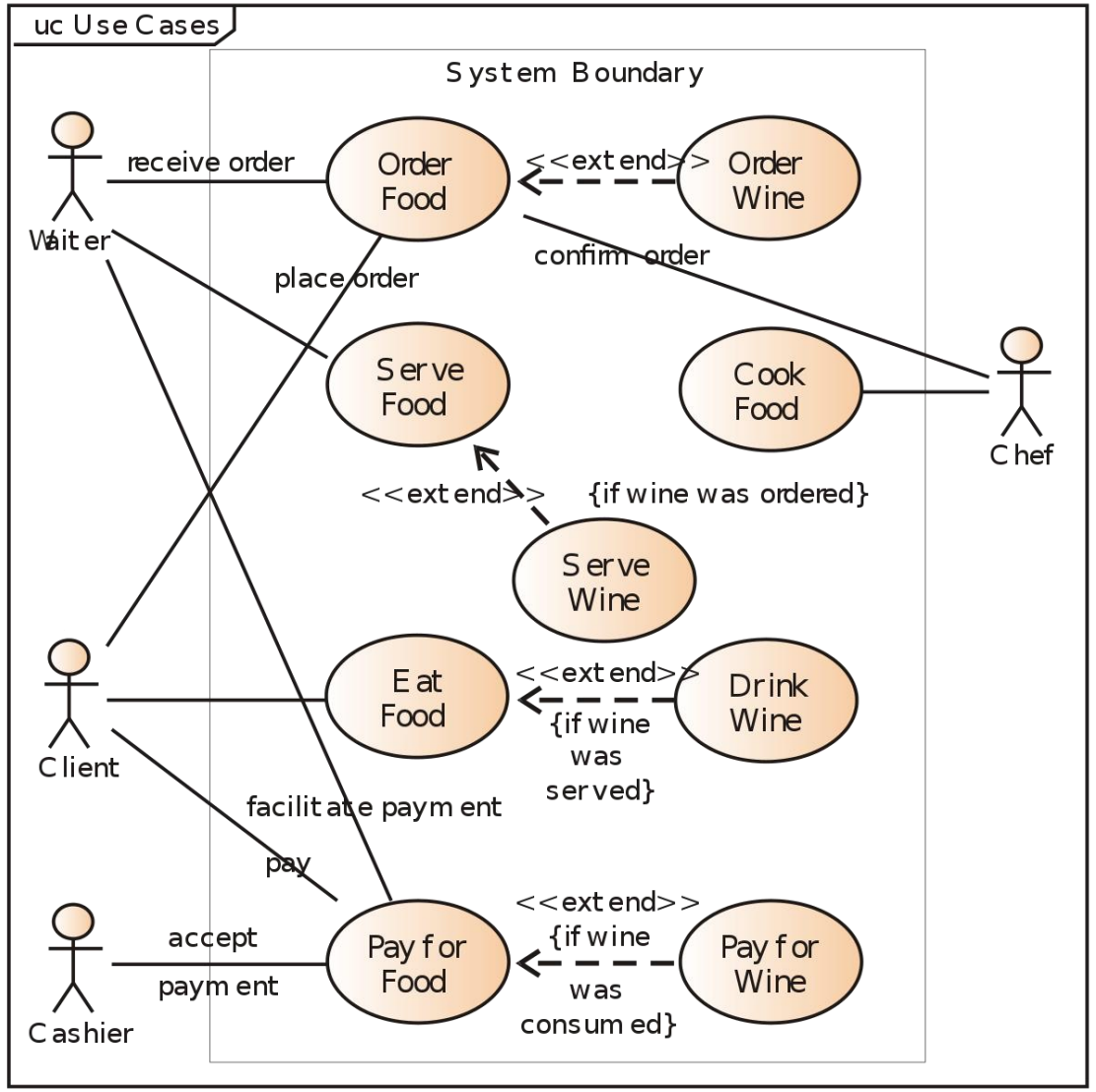


Diagram maszynny stanów

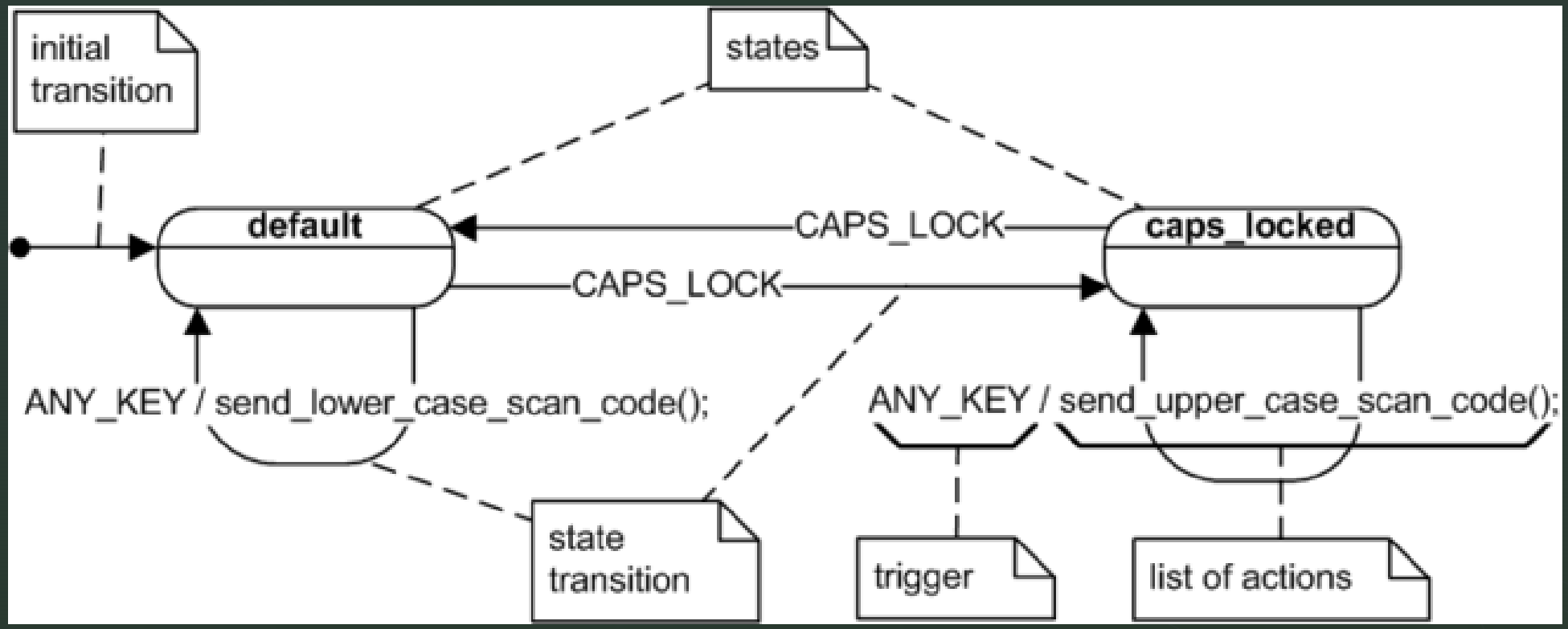


Diagram komunikacji

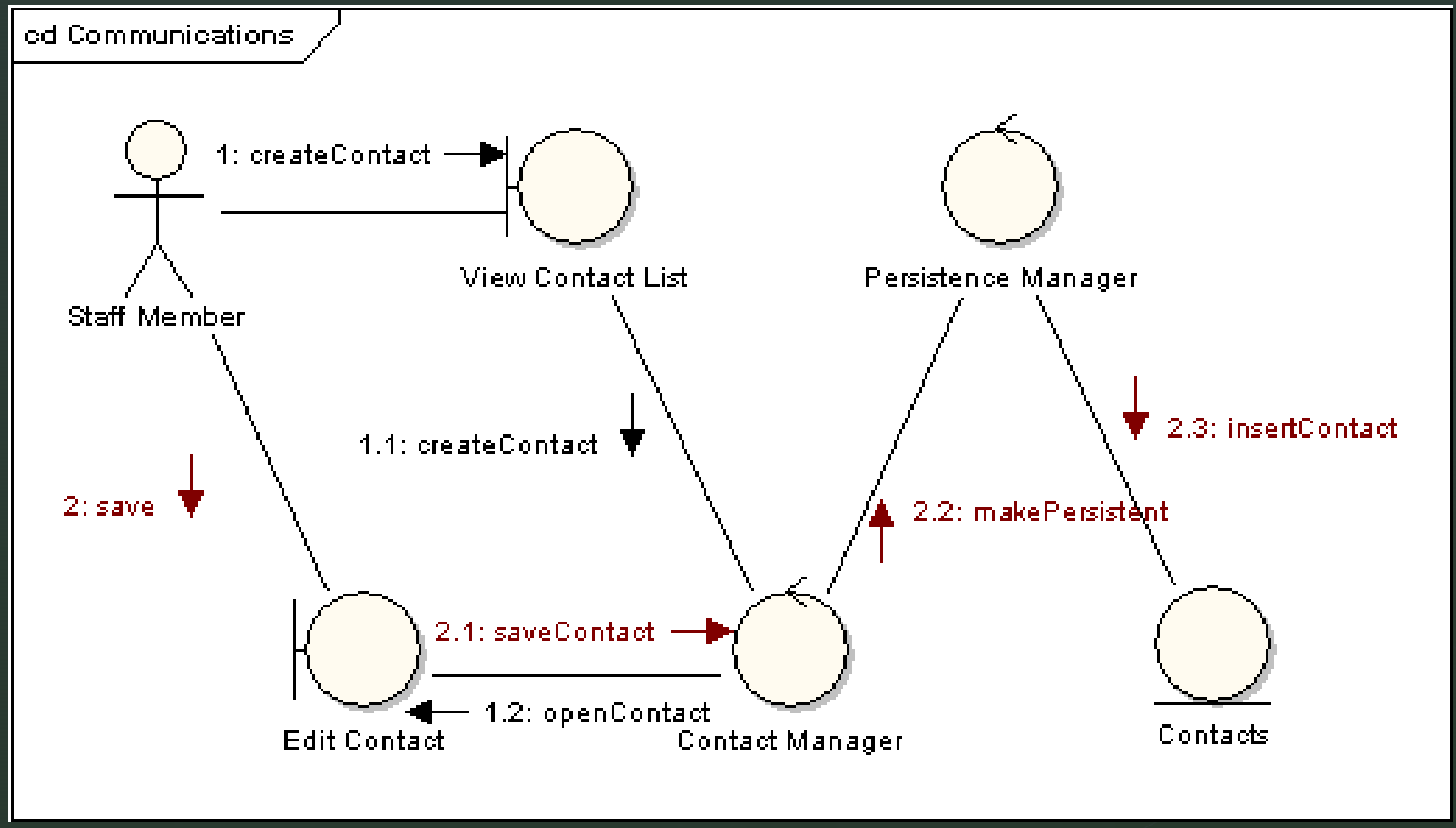


Diagram sekwencji czasowej

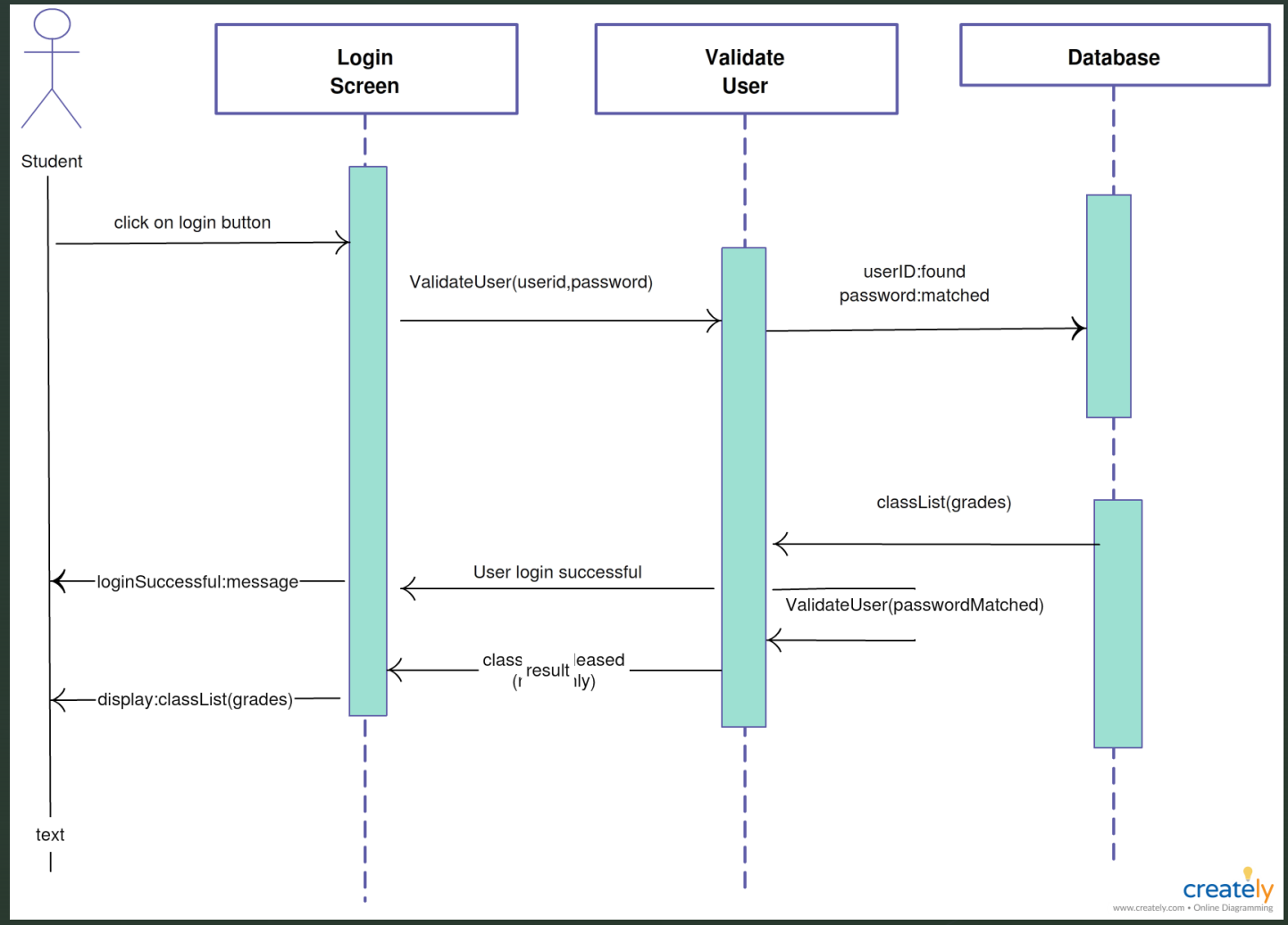


Diagram przebiegów czasowych

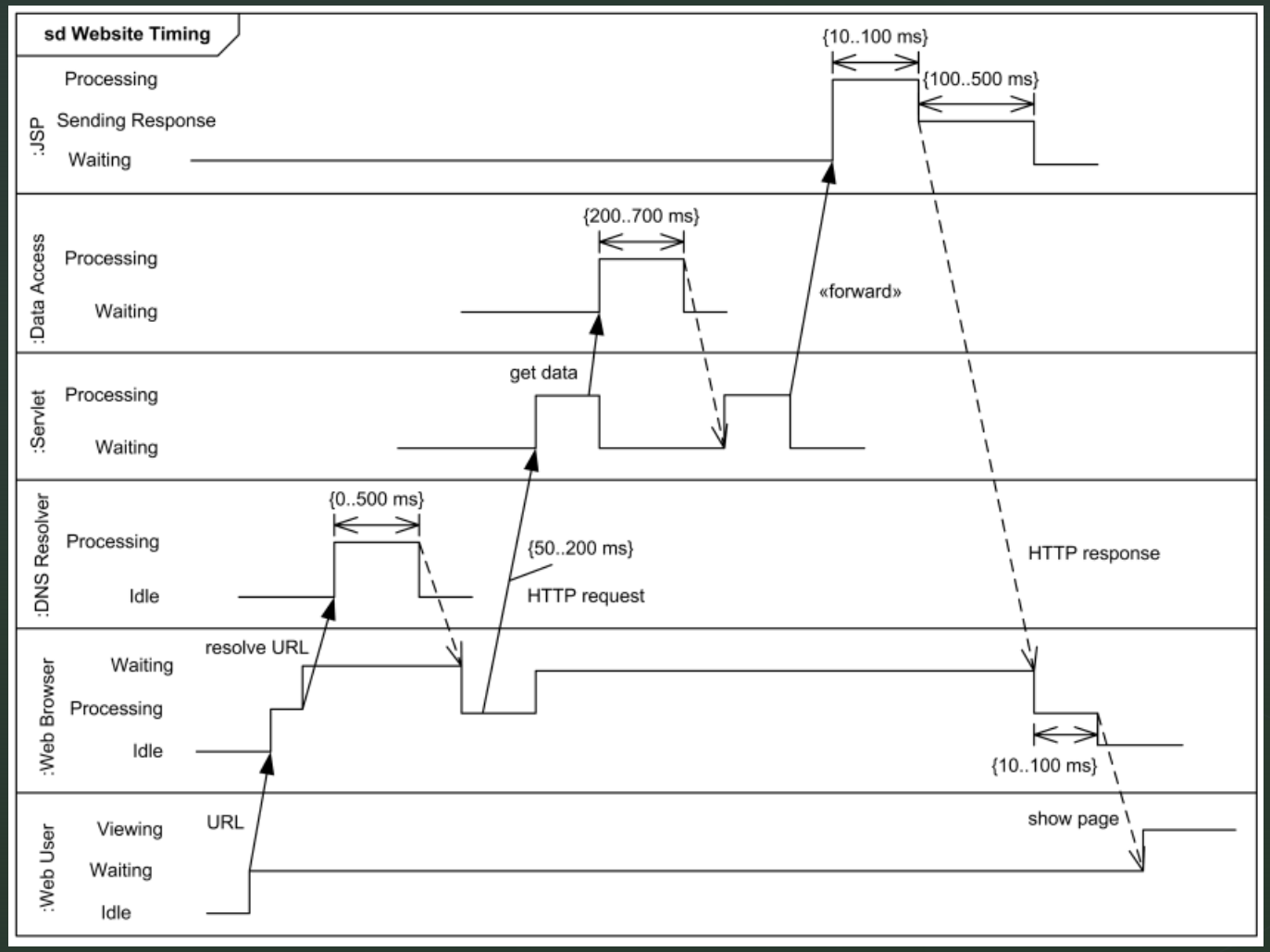
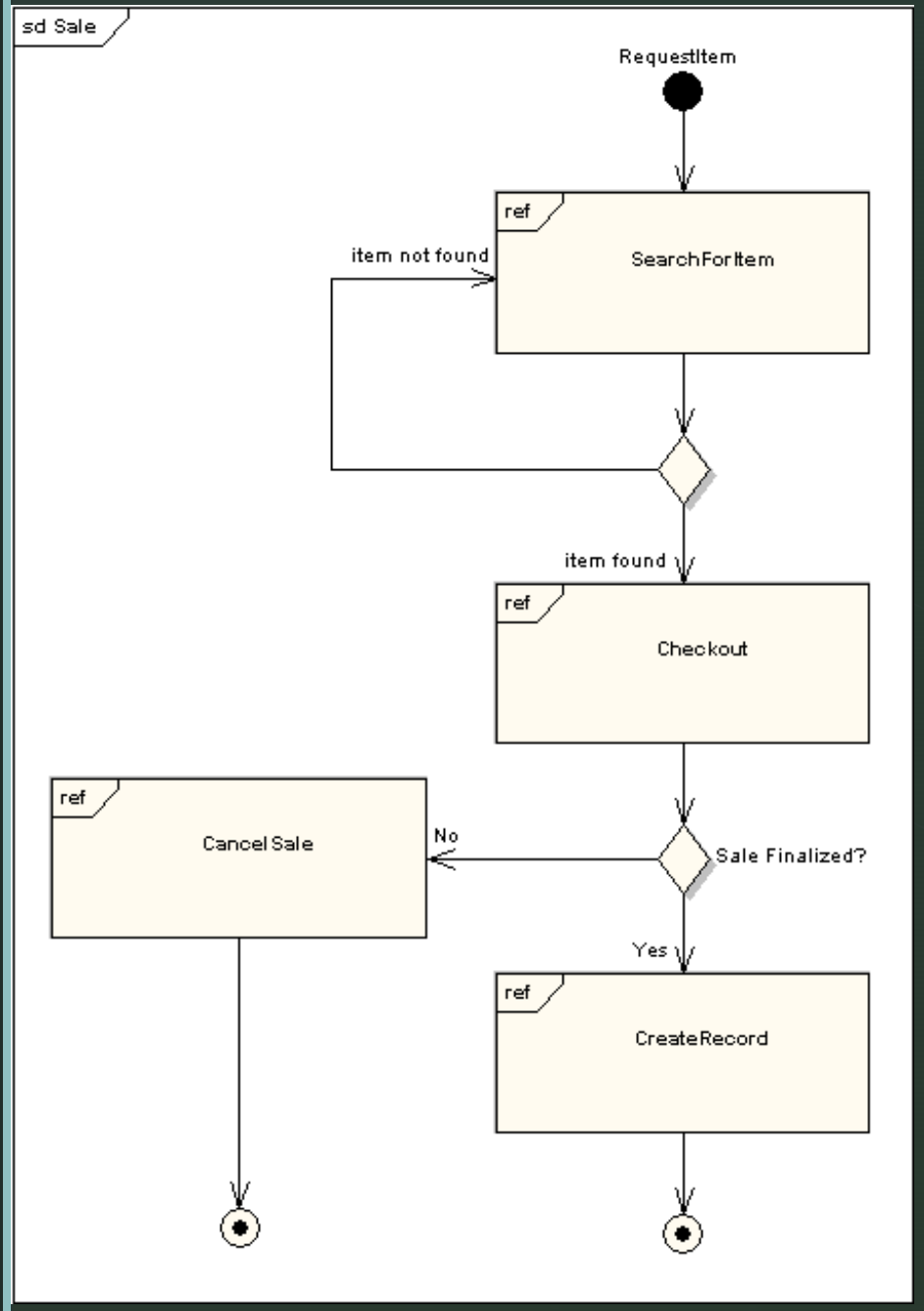


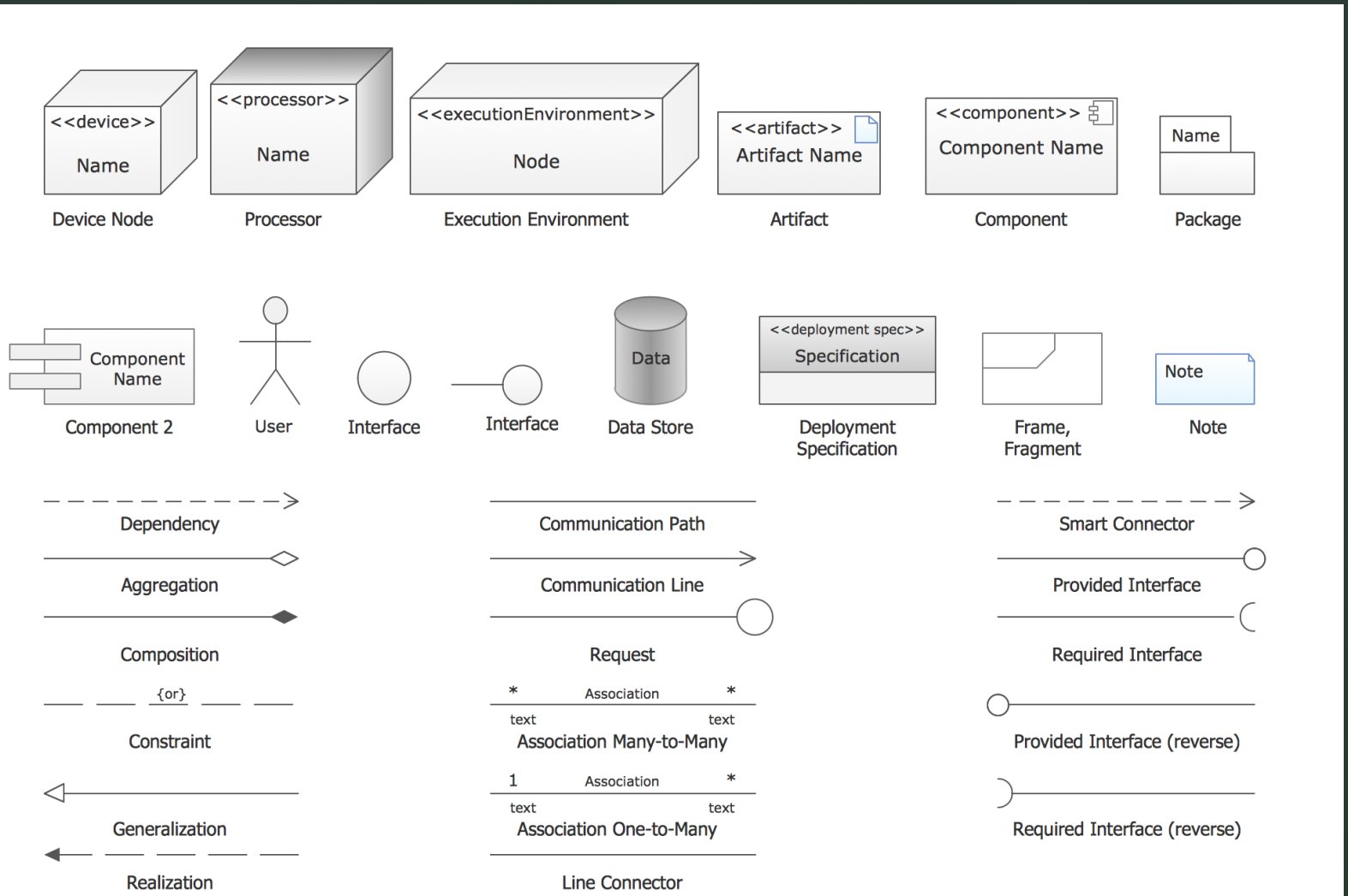
Diagram przeglądu interakcji



Najczęściej stosowane diagramy

- Przypadków użycia
- Sekwencji
- Klas
- Czynności

Elementy UML



Obrazy

- http://tech.sina.com.cn/other/2004-11-11/1024457627.shtml
- https://docplayer.pl/docs-images/73/69090864/images/5-2.jpg
- https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/Grady_Booch%2C_CHM_2011_2_cropped.jpg/220px-Grady_Booch%2C_CHM_2011_2_cropped.jpg
- https://www.conceptdraw.com/How-To-Guide/picture/Booch-OOD-diagram.png
- https://lh3.googleusercontent.com/proxy/sWnAnfcuYoqMaohvHnj50EOvmcj0Z2F-tsu0uvRUcV602YKp45BSaPKGE2l4-1jK97hhpfl7DZw2G3eV9jJnjLnPupE_Vd7MFyJu66fDvPOQqeMFECfPAg
- https://upload.wikimedia.org/wikipedia/commons/9/9d/OMT_object_diagram.png

Obrazy

- https://lh3.googleusercontent.com/proxy/igLcBJlJU5gyuI3aZwg3Ef0NnYNqxst-ttILU36TrjC6LiGgKkEAJKGnKBAH1NxCS5AhDheoY-EwgQAgPlFIbPJ1jwdiR-M2zkGwiUvAPnsrB9GjTn0C2Go
- https://www.conceptdraw.com/How-To-Guide/picture/Design-elements-UML-deployment-diagrams.png
- https://www.uml-diagrams.org/profile-diagrams/profile-diagram-overview.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Use_case_restaurant_model.svg/1200px-Use_case_restaurant_model.svg.png
- https://www.researchgate.net/profile/Muhammad_Rashid_Naeem/publication/262313799/figure/fig3/AS:669279213060099@1536580007115/Activity-Diagram-of-Automatic-Exam-Model.png

Obrazy

- https://upload.wikimedia.org/wikipedia/en/thumb/4/45/UML_state_machine_Fig1.png/660px-UML_state_machine_Fig1.png
- https://sparxsystems.com/images/screenshots/uml2_tutorial/com02.GIF
- https://i.pinimg.com/originals/bd/8c/85/bd8c858b85f9546d9869cec19bea6452.png
- https://sparxsystems.com/images/screenshots/uml2_tutorial/io03.GIF

Materiały i odnośniki

- https://pl.wikipedia.org/wiki/Unified_Modeling_Language
- http://www.temida.si/~bojan/IPIT_2014/literatura/UML_Reference_Manual.pdf
- https://www.omg.org/
- https://pl.wikipedia.org/wiki/Rational_Unified_Process
- https://en.wikipedia.org/wiki/4%2B1_architectural_view_model
- https://docplayer.pl/69090864-Unified-modeling-language.html
- https://docplayer.pl/10737710-Uml-unified-modeling-language.html