Politechnika Wrocławska

Programowanie obiektowe

Wykład 11

Diagramy UML

Prowadzący
dr inż. Paweł Maślak

Plan wykładu

- Czym są diagramy UML
- Rodzaje diagramów UML
- Przykłady diagramów UML
- Narzędzia UML
- Certyfikacja UML

Czym są diagramy UML

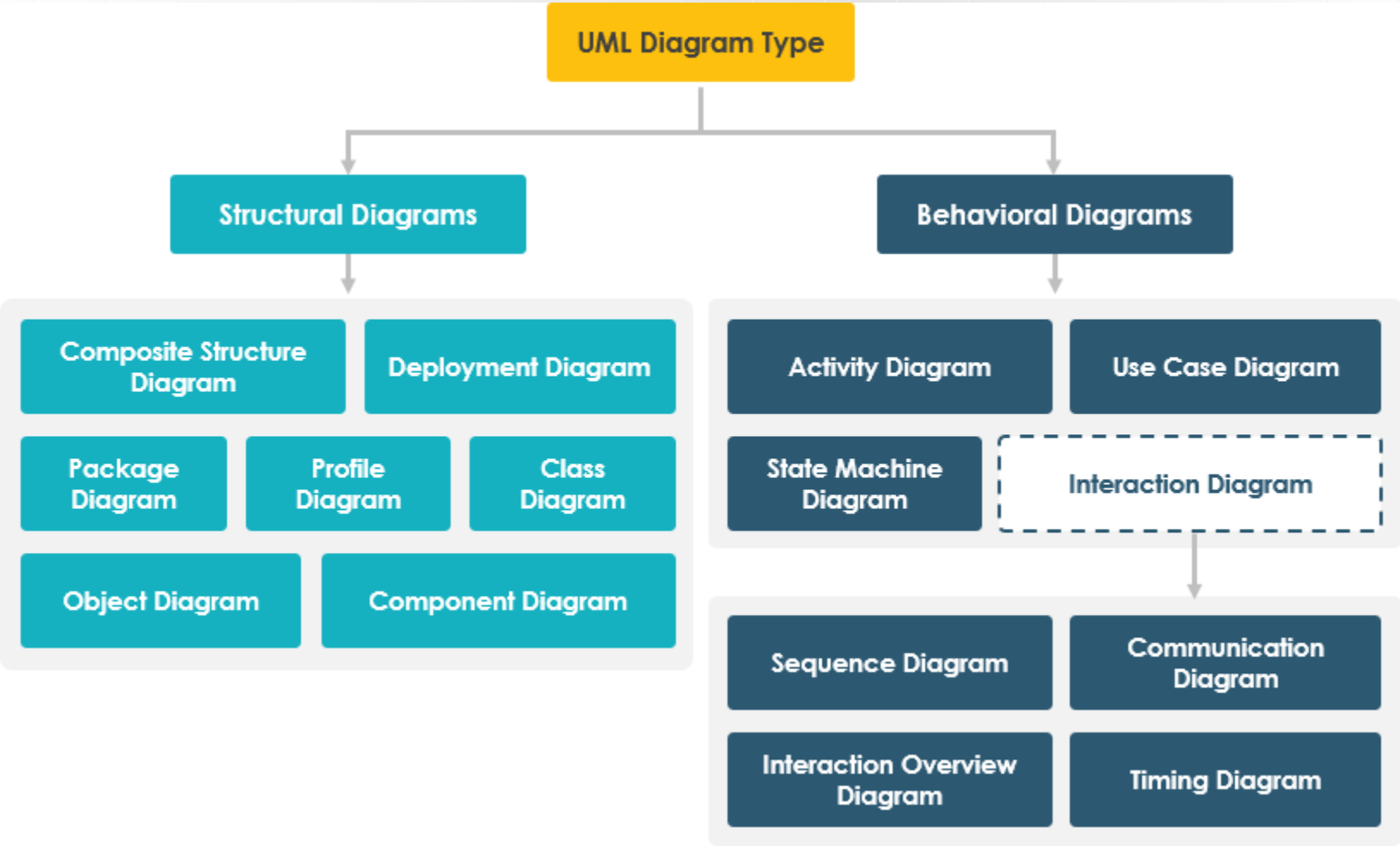
- **UML** to akronim pochodzący od angielskiego określenia ***Unified Modeling Language***. W polskim tłumaczeniu znany jest jako zunifikowany język modelowania.
- **UML** to jasno wyspecyfikowany język składający się z kilkunastu diagramów. Diagramy te pozwalają na formalne opisywanie i modelowanie struktur czy procesów.

Czym są diagramy UML

- Język **UML** przyjmuję w praktyce postać graficznej reprezentacji tworzonego systemu, składającej się z logicznie powiązanych z sobą diagramów. Pozwalają one na opisanie systemu od modeli ogólnych do bardzo szczegółowych. W standardzie UML w wersji 2.0 występuje **czternaście** rodzajów diagramów, które charakteryzują statykę i dynamikę tworzonego systemu.

Rodzaje diagramów UML

- Klasyfikacja diagramów UML



Rodzaje diagramów UML

Diagramy struktur (Structural Diagrams)

1. Klas (class diagram)
2. Obiektów (object diagram)
3. Komponentów (component diagram)
4. Wdrożenia (deployment diagram)
5. Struktur złożonych (composite structure diagram)
6. Pakietów (package diagram)
7. Profili (profile diagram)

Rodzaje diagramów UML

Diagramy zachowań (Behavioral Diagrams)

1. Czynności (activity diagram)
2. Przypadków użycia (use case diagram)
3. Maszyny stanów (state machine diagram)

Diagramy Interakcji (Interaction Diagram)

1. Komunikacji (communication diagram)
2. Sekwencji (sequence diagram)
3. Czasowe (timing diagram)
4. Przeglądu interakcji (interaction overview diagram)

Rodzaje diagramów UML

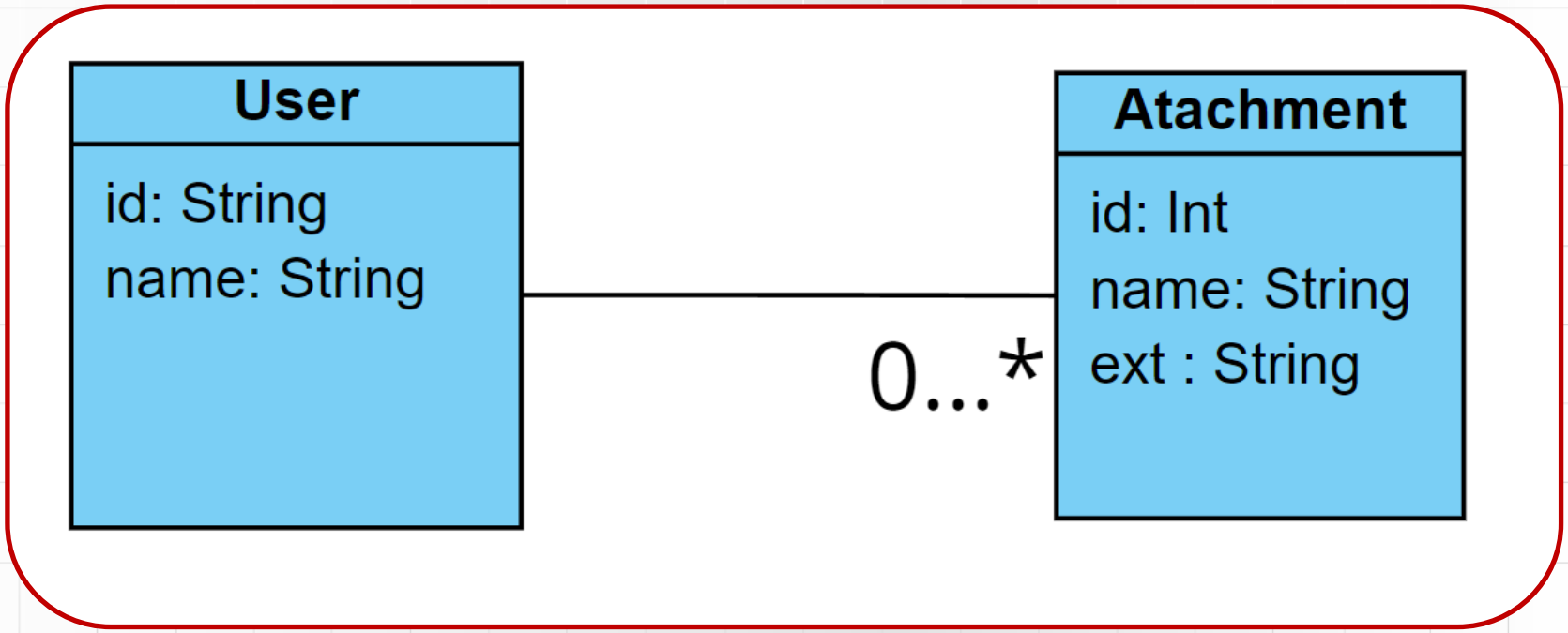
- W praktyce rzadko kiedy trzeba opracowywać wszystkie diagramy i w większości przypadków korzysta się z mniej niż połowy wyżej wymienionych.
- Projektując system informatyczny, rozpoczyna się przeważnie od tworzenia diagramów w następującej kolejności:
  1. Przypadków użycia
  2. Sekwencji
  3. Klas
  4. Czynności
- Są to najczęściej wykorzystywane diagramy. Pozostałe bywają pomijane, zwłaszcza przy budowaniu niedużych systemów informatycznych.

Diagramy Klas (class diagram)

- Diagram klas to graficzne przedstawienie statycznych, deklaratywnych elementów dziedziny przedmiotowej oraz związków między nimi
- Przykłady

Diagramy Klas (class diagram)

- Poniższy przykład diagramu klas reprezentuje dwie klasy — użytkownika i załącznik. Użytkownik może przesłać wiele załączników, aby obie klasy zostały połączone skojarzeniem, przy czym 0..* oznacza wielokrotność po stronie załącznika.

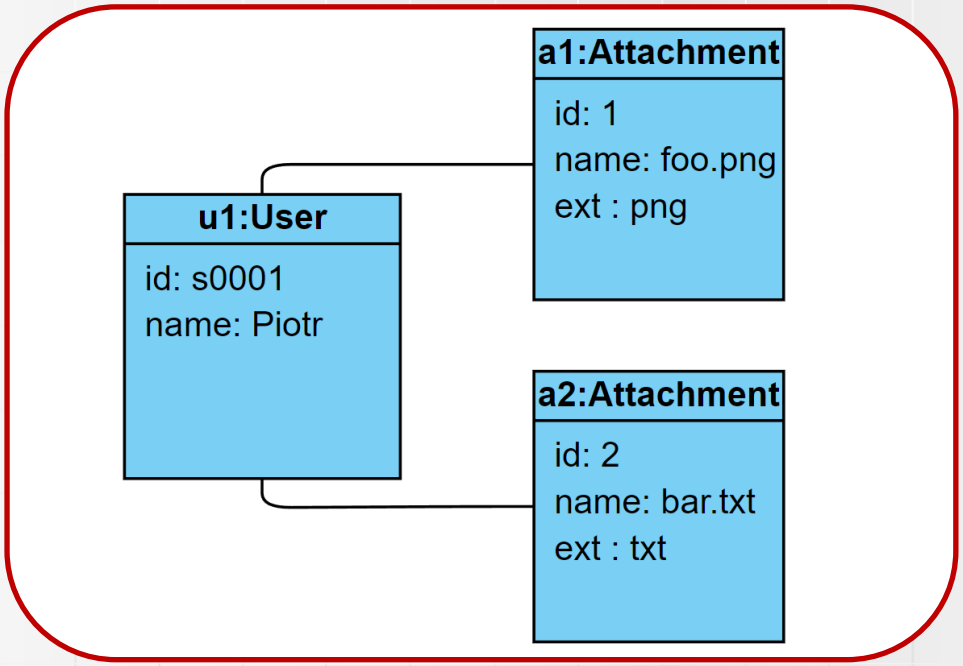


Diagramy Obiektów (object diagram)

- Diagram obiektów to wystapienie diagramu klas, odwzorowujące strukturę systemu w wybranym momencie jego działania

Diagramy Obiektów (object diagram)

- Poniższy przykład Diagramu Obiektu pokazuje, jak „wyglądają” instancje obiektów klasy Użytkownik i Załącznik w momencie, gdy Piotr (tj. użytkownik) próbuje przesłać dwa załączniki. Istnieją zatem dwie specyfikacje instancji dla dwóch obiektów załączników, które mają zostać przesłane.

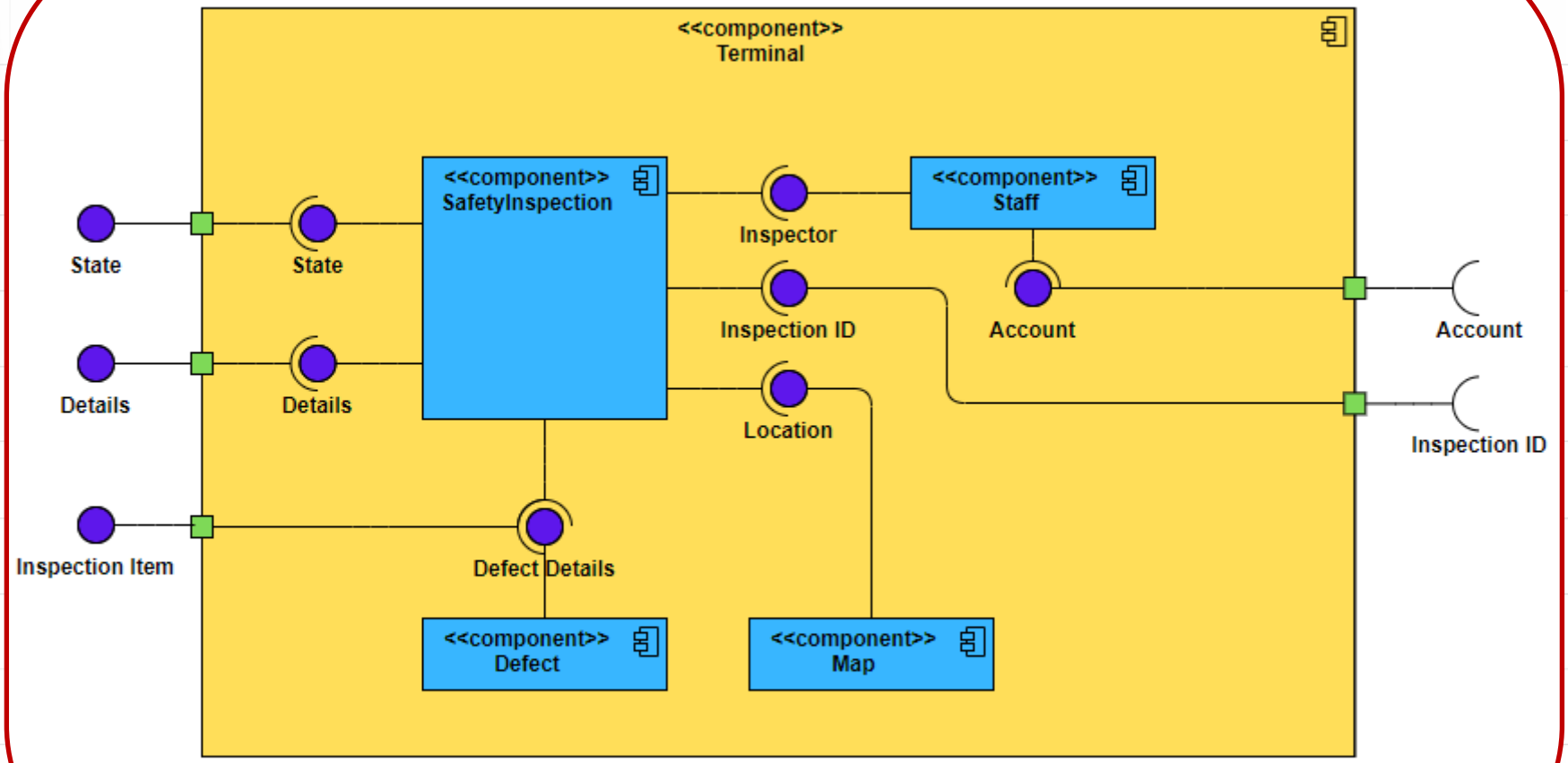


Diagramy Komponentów (component diagram)

- Diagram komponentów to rodzaj diagramu wdrożeniowego, który wskazuje organizację i zależności między komponentami

Diagramy Komponentów (component diagram)

- Przykład

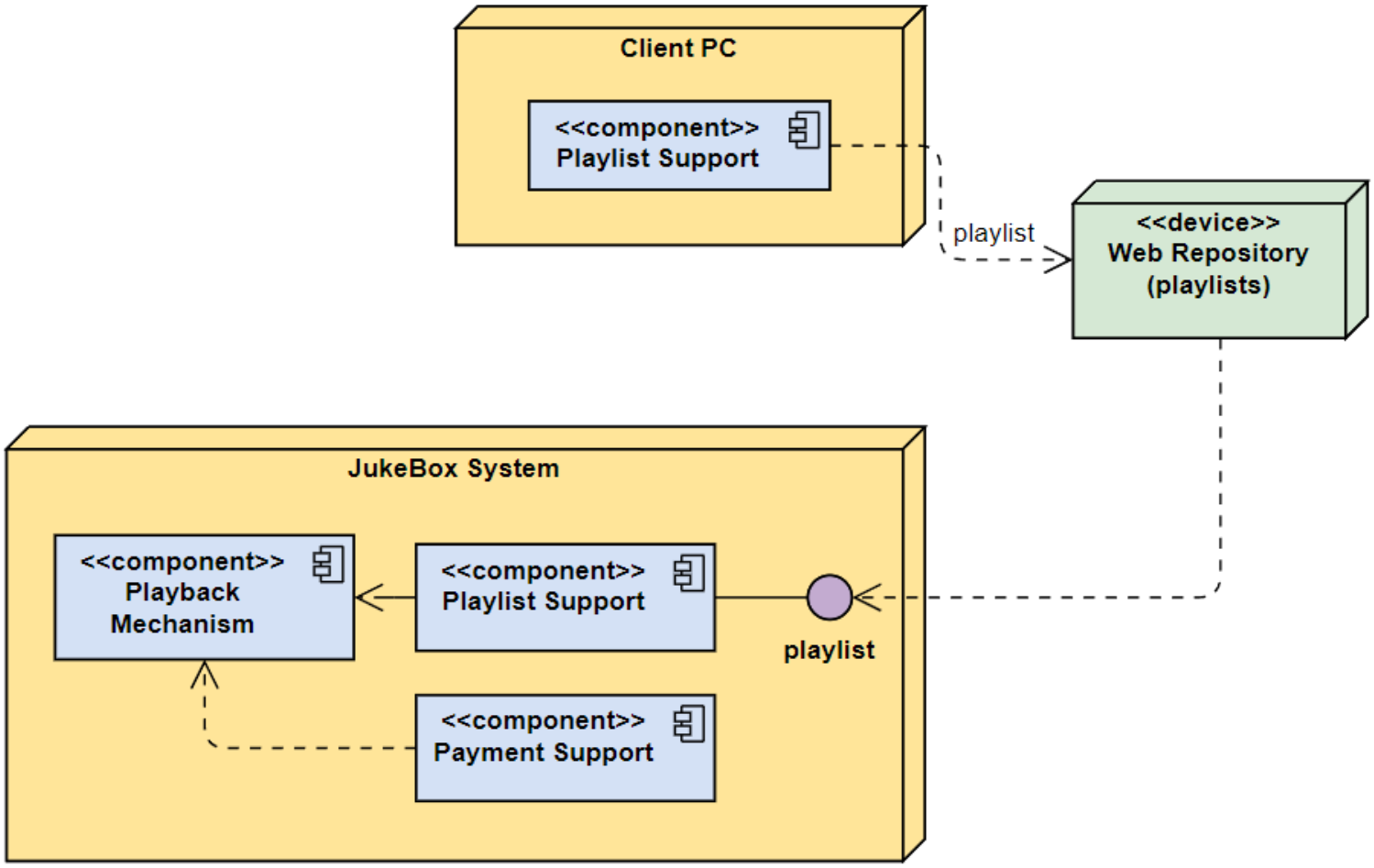


Diagramy Wdrożenia (deployment diagram)

- Diagramy wdrożenia przedstawiają powiązania między oprogramowaniem (artefaktami) i sprzętem (węzłami). Są stosowane przy modelowaniu dużych systemów.

Diagramy Wdrożenia (deployment diagram)

- Przykład



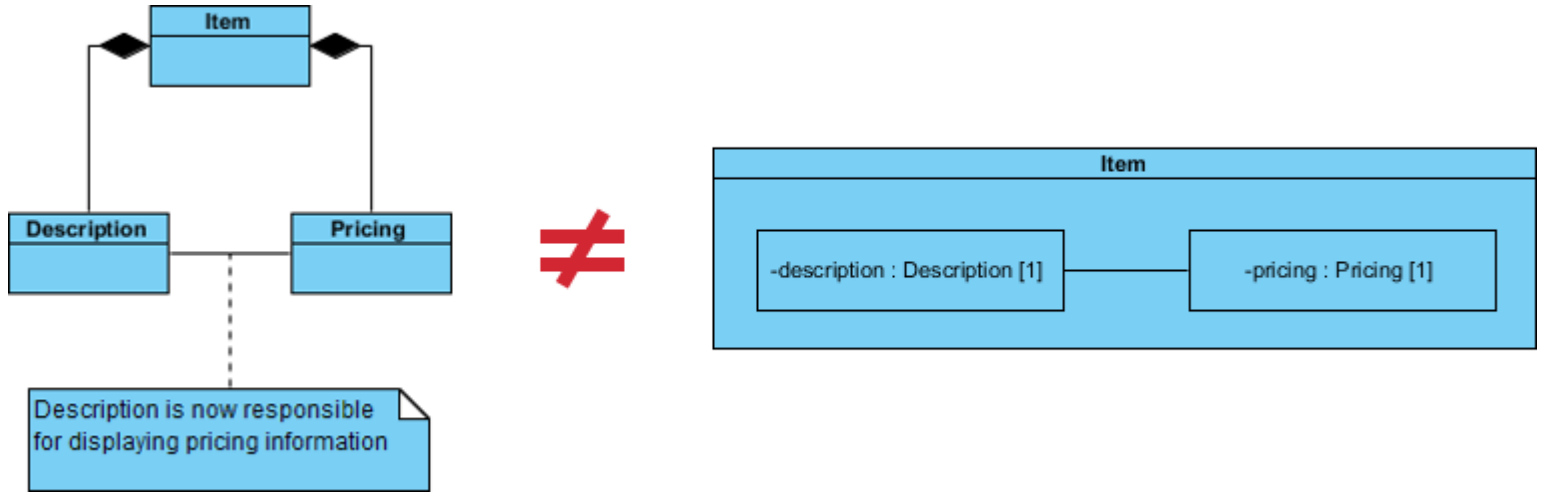
Diagramy Struktur Złożonych (composite structure diagram)

- Diagram struktur złożonych to graficzne przedstawienie wzajemnie współdziałających części dla osiągnięcia pożądanej funkcjonalności współdziałania

Diagramy Struktur Złożonych (composite structure diagram)

- Przykład

- Class Diagram vs. Composite Structure Diagram

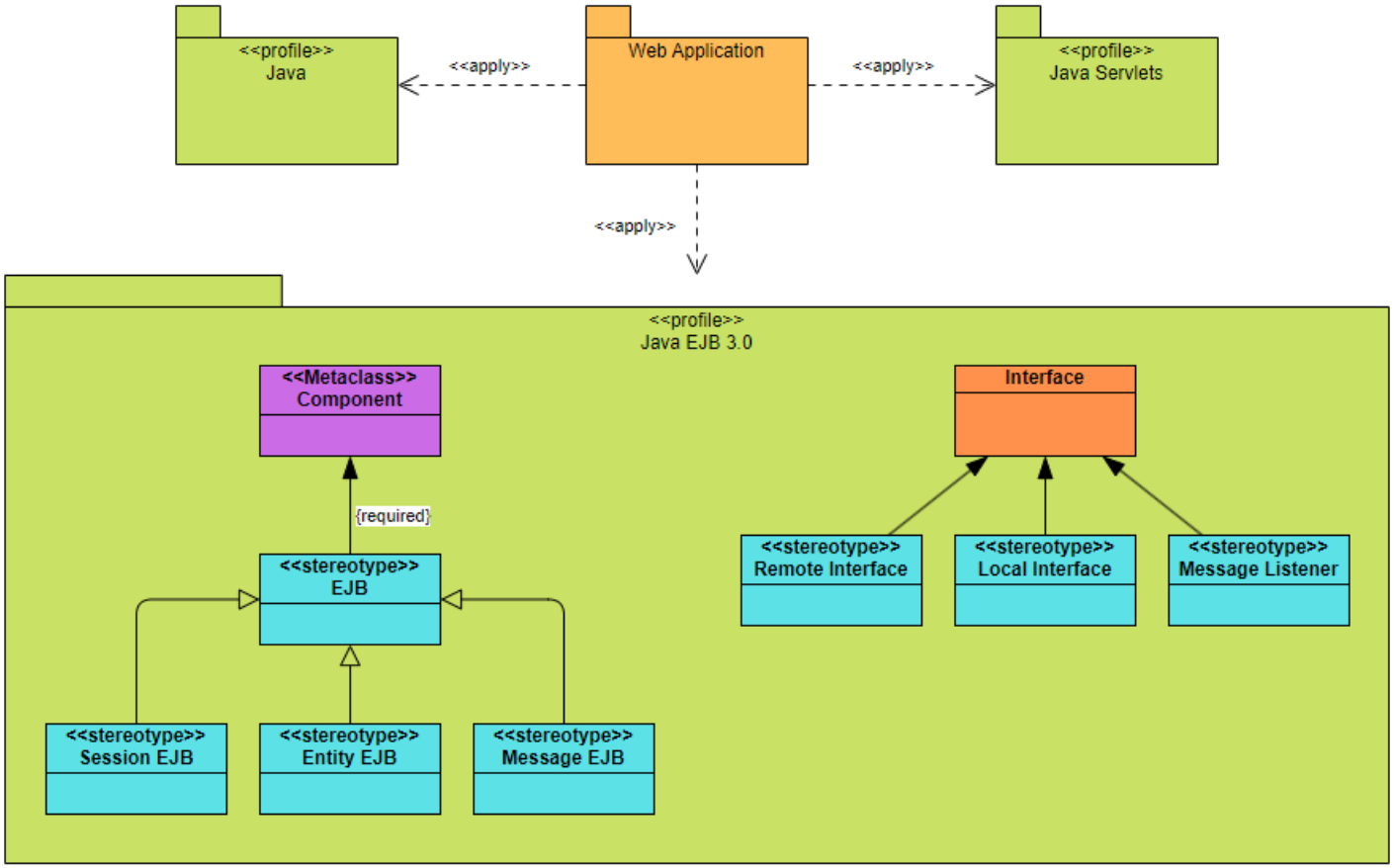


Diagramy Pakietów (package diagram)

- Diagram pakietów to graficzne przedstawienie logicznej struktury systemu w postaci zestawu pakietów połączonych zależnościami i zagnieżdżeniami

Diagramy Pakietów (package diagram)

- Przykład

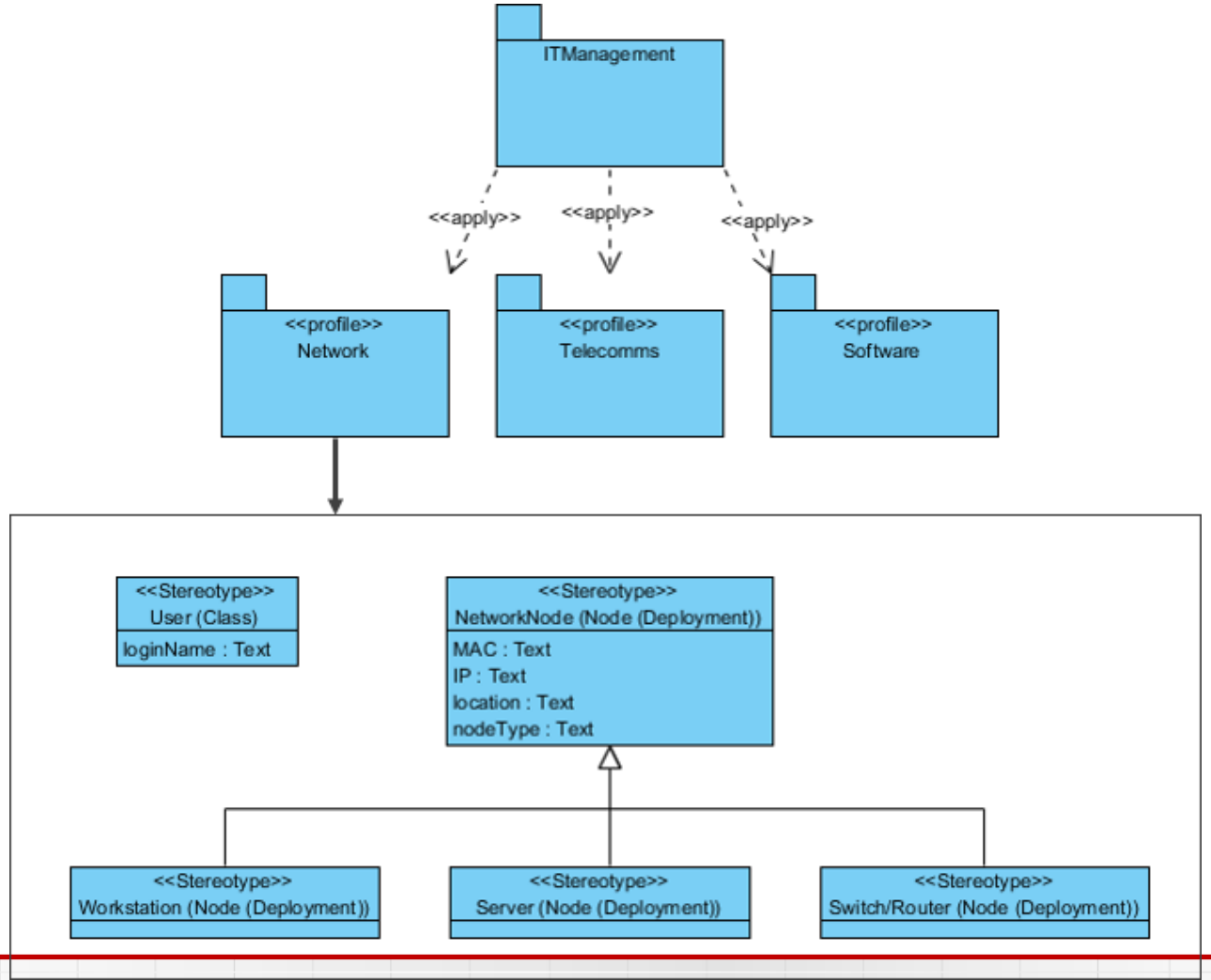


Diagramy Profili (profile diagram)

- Diagram profilu umożliwia utworzenie stereotypów specyficznych dla domeny i platformy oraz zdefiniowanie relacji między nimi. Możesz tworzyć stereotypy, rysując ich kształty i powiązując je z kompozycją lub uogólnieniem za pomocą interfejsu zorientowanego na zasoby. Można także definiować i wizualizować oznaczone wartości stereotypów.

Diagramy Profili (profile diagram)

- Przykład - IT Management

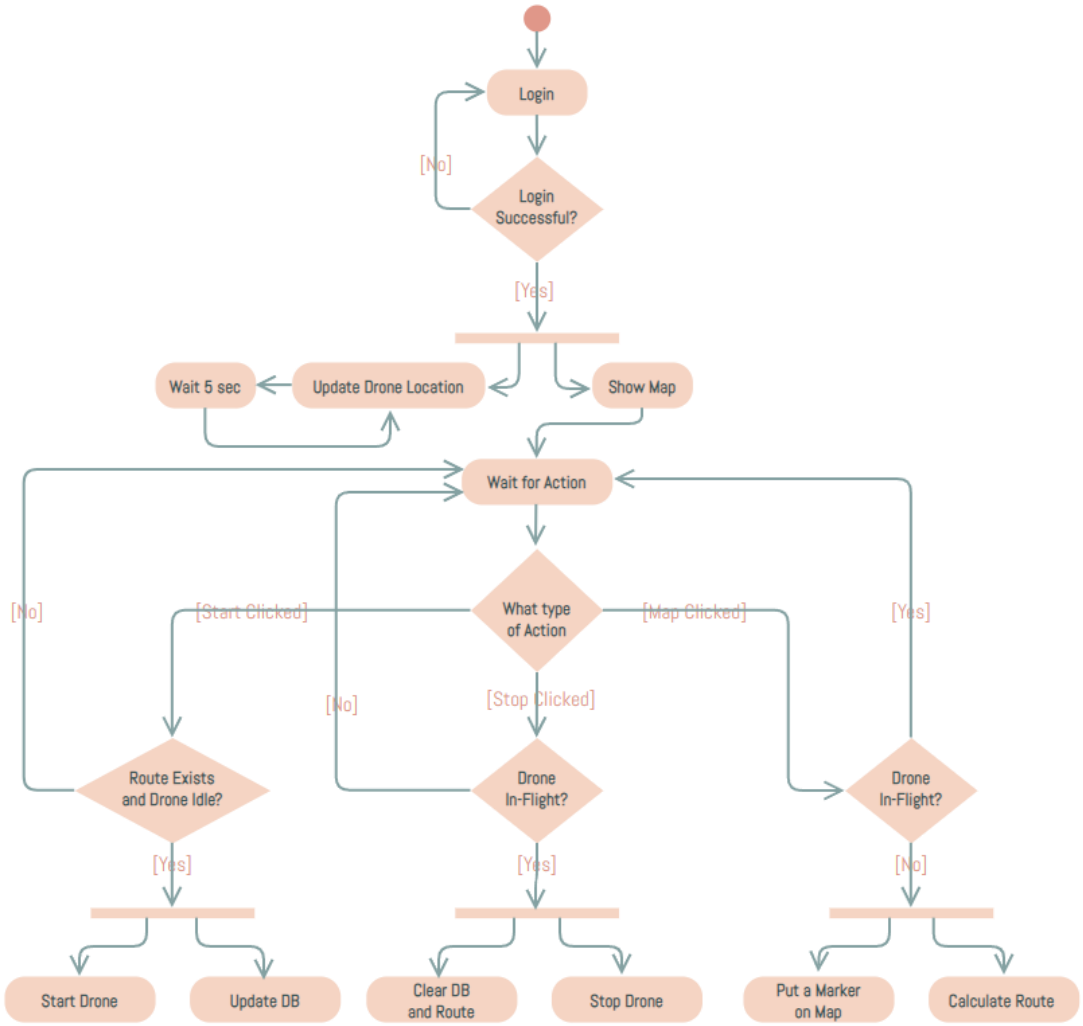


Diagramy Czynności (activity diagram)

- Diagram czynności to graficzne przedstawienie sekwencyjnych i/lub współbieżnych przepływów sterowania oraz danych pomiędzy uporządkowanymi ciągami czynności, akcji i obiektów

Diagramy Czynności (activity diagram)

- Przykład – Android Aplication

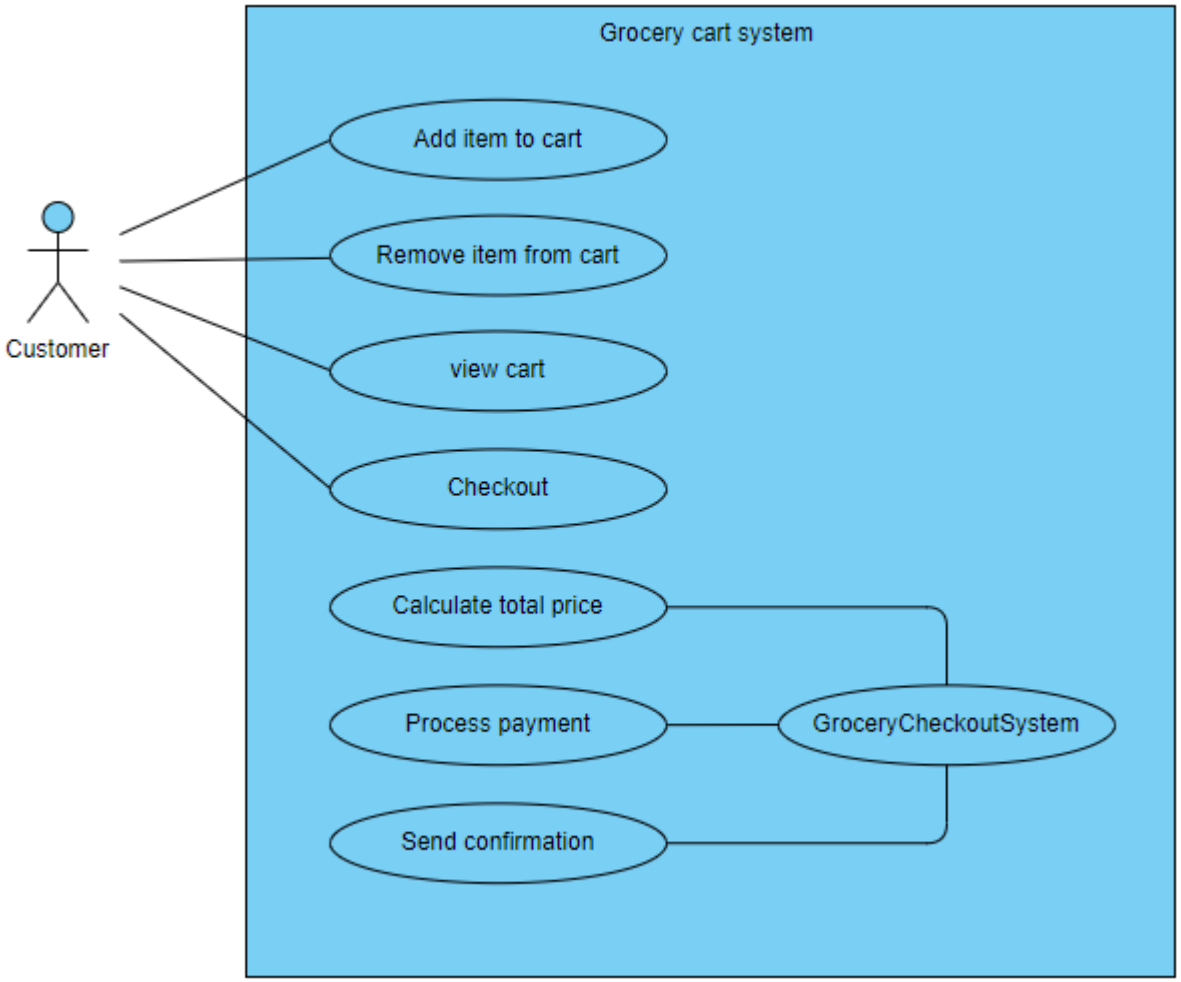


Diagramy Przypadków użycia (use case diagram)

- Diagram przypadków użycia to graficzne przedstawienie przypadków użycia, aktorów oraz związków między nimi, występujących w danej dziedzinie przedmiotowej

Diagramy Przypadków użycia (use case diagram)

- Przykład – sklep spożywczy

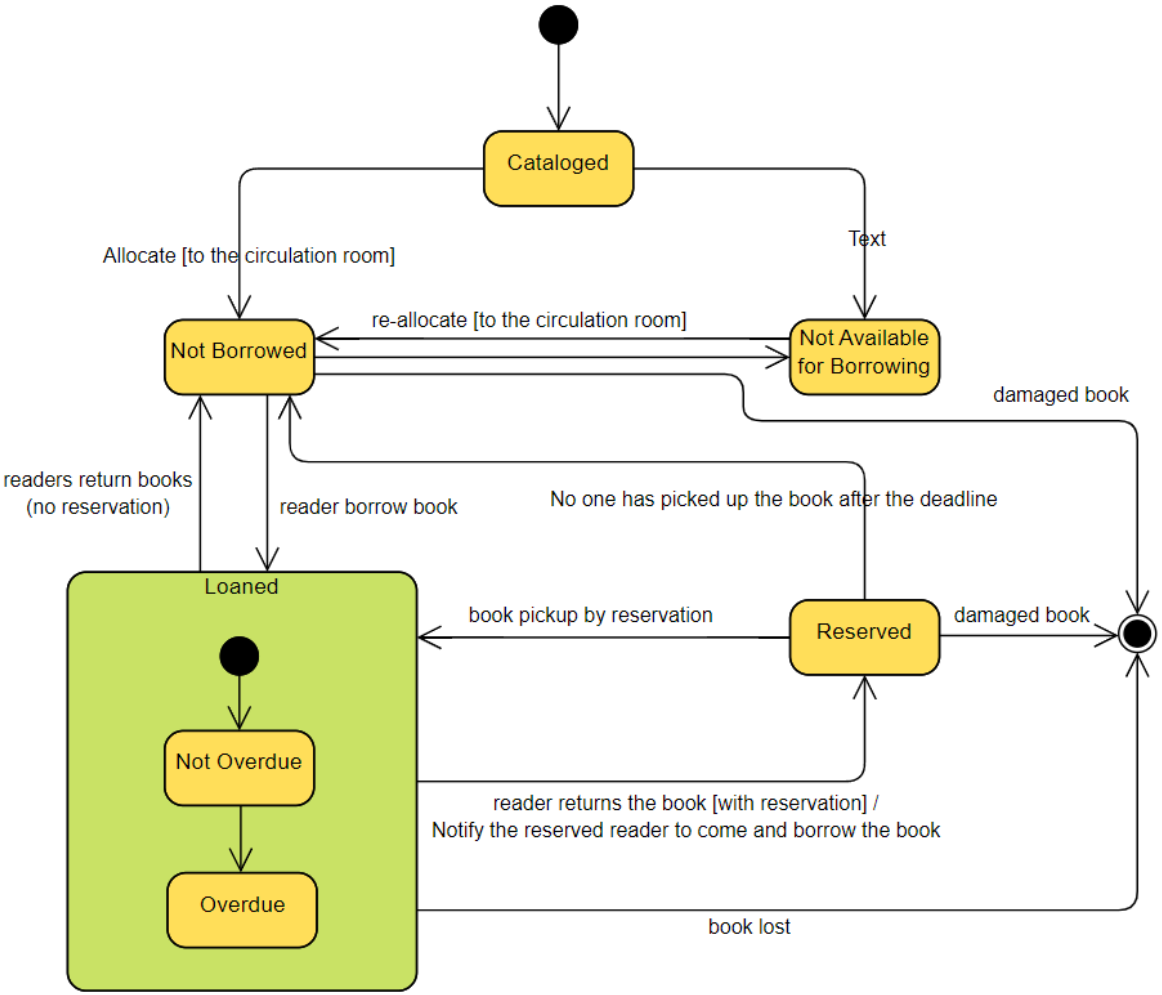


Diagramy Maszyny stanów (state machine diagram)

- Diagram maszyny stanowej to graficzne odzwierciedlenie dyskretnego, skokowego zachowania skończonych systemów stan-przejście

Diagramy Maszyny stanów (state machine diagram)

- Przykład – system biblioteczny

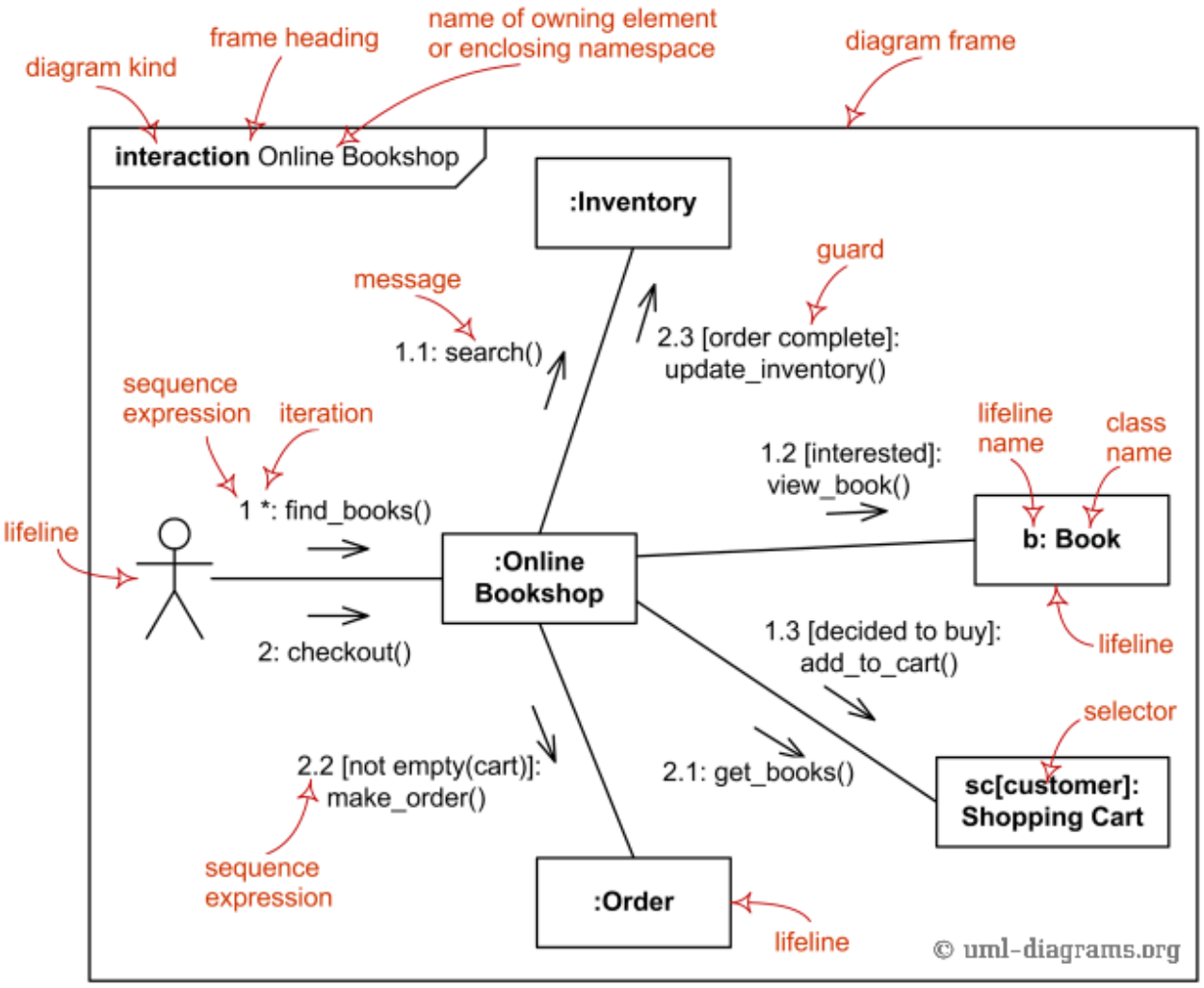


Diagramy Komunikacji (communication diagram)

- Diagram komunikacji jest rodzajem diagramu interakcji, specyfikującym strukturalne związku pomiędzy instancjami klasyfikatorów biorącymi udział w interakcji oraz wymianę komunikatów pomiędzy tymi instancjami

Diagramy Komunikacji (communication diagram)

- Przykład

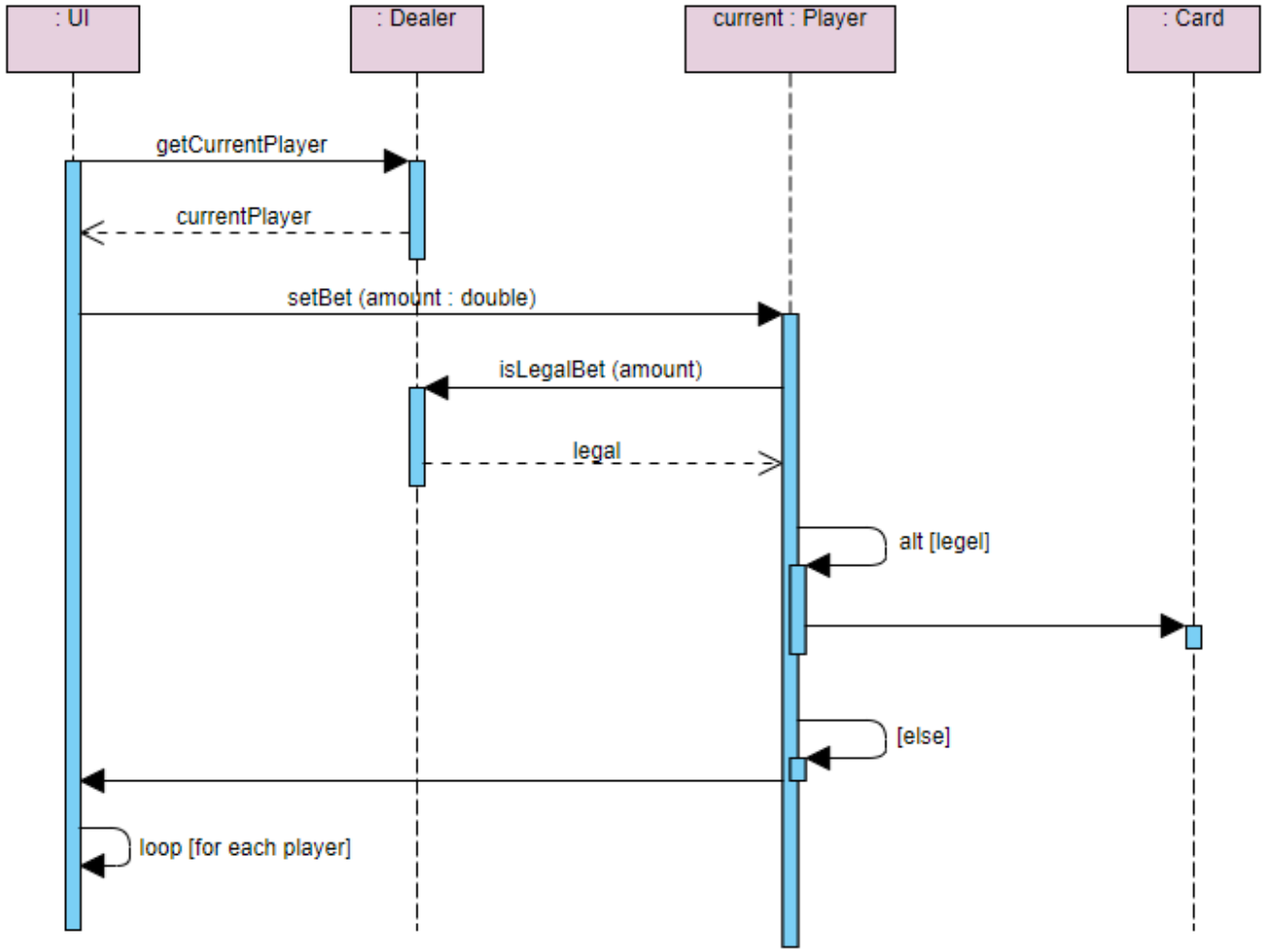


Diagramy Sekwencji (sequence diagram)

- Diagram sekwencji jest rodzajem diagramu interakcji, opisującym interakcje pomiędzy instancjami klasyfikatorów systemu w postaci sekwencji komunikatów wymienianych między nimi

Diagramy Sekwencji (sequence diagram)

- Przykład

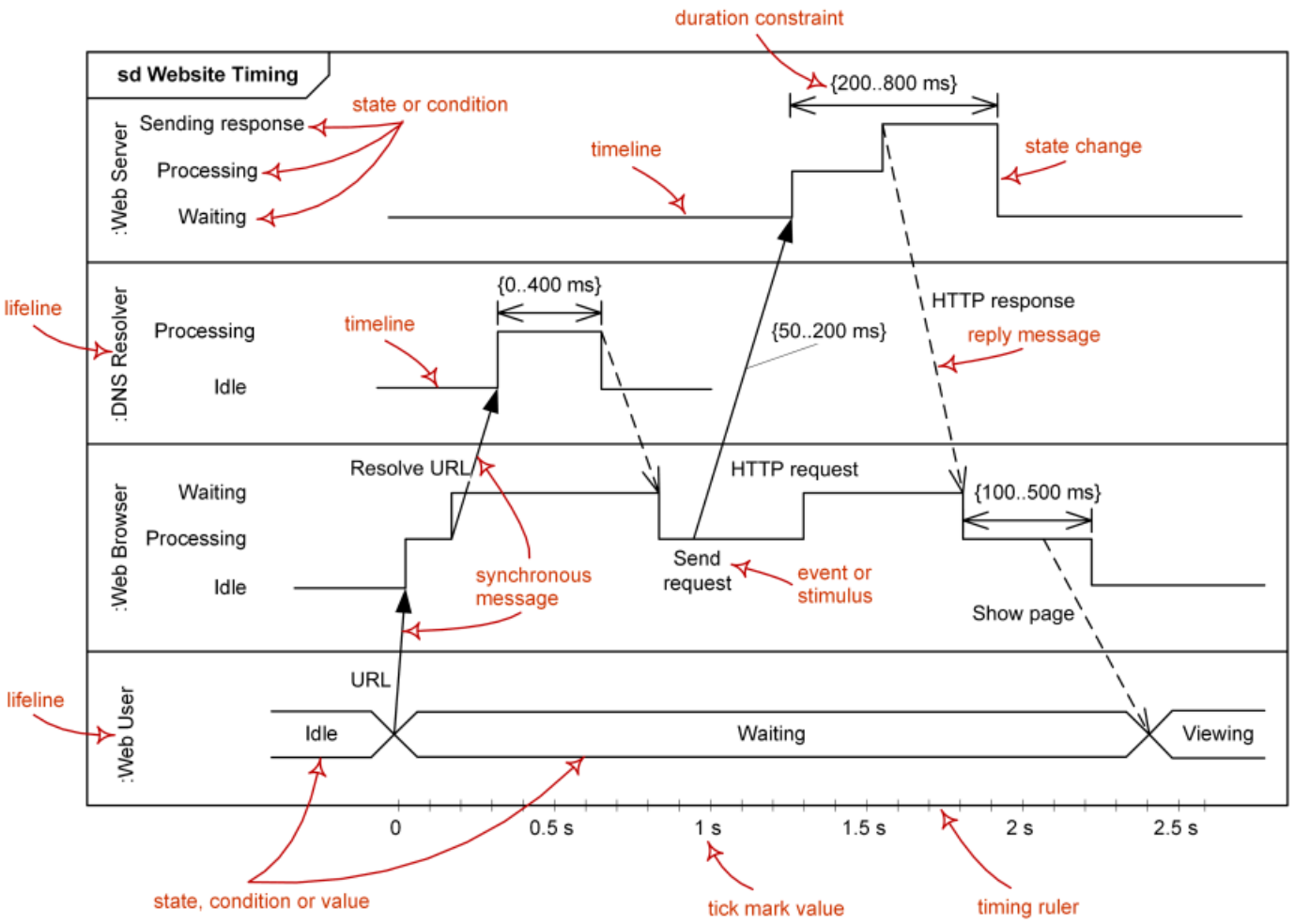


Diagramy Czasowe (timing diagram)

- Diagram czasowy jest rodzajem diagramu interakcji, reprezentującym na osi czasu zmiany dopuszczalnych stanów, jakie może przyjmować instancja klasyfikatora uczestnicząca w interakcji

Diagramy Czasowe (timing diagram)

- Przykład

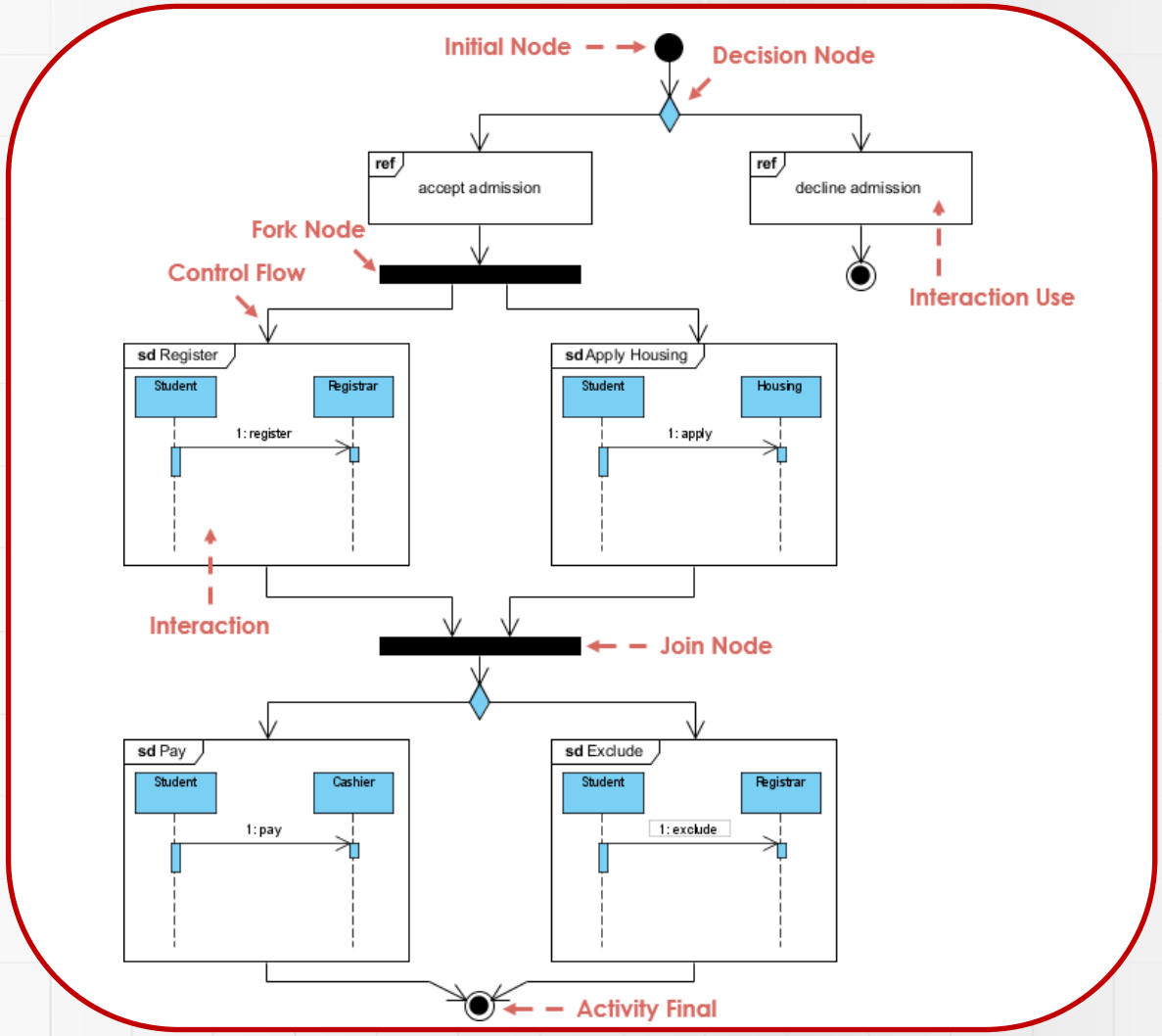


Diagramy Przeglądu interakcji (interaction overview diagram)

- Diagram przeglądu interakcji jest rodzajem diagramu interakcji, dokumentującym przepływ sterowania pomiędzy logicznie powiązanymi diagramami i fragmentami interakcji z wykorzystaniem kategorii modelowania diagramów czynności

Diagramy Przeglądu interakcji (interaction overview diagram)

- Przykład



Narzędzia UML

- Są to narzędzia **CASE** będące de facto implementacją języka **UML**. Niektóre pozwalają także na inżynierię w przód i inżynierię wstecz dla obiektowych języków programowania.

- **CASE** (ang. Computer-Aided Software Engineering) – technika komputerowego wspomagania tworzenia, utrzymywania i rozbudowy oprogramowania[1].
- **Funkcje CASE-a** to analiza, projektowanie i programowanie.
- **Narzędzia CASE** automatyzują metody projektowania, dokumentacji oraz tworzenia struktury kodu programu w wybranym języku programowania, najczęściej w programowaniu obiektowym.

Narzędzia UML - Wolne/otwarte

- Acceleo – system generacji kodu źródłowego z modeli UML oparty na Eclipse i szablonie EMF
- AmaterasUML – plug-in dla Eclipse umożliwiający rysowanie diagramów klas i interakcji; potrafi generować diagramy klas z kodu Javy, natomiast diagramy interakcji z wydruku stosu wywołań
- ArgoUML – napisany w Javie, zaawansowane generowanie kodu i podpowiedzi, ciągle rozszerzany
- ATLAS Transformation Language – narzędzie QVT, pozwalające transformować między innymi modele UML w inne modele UML lub Java itp. ATL jest kompletnym rozwiązaniem open source udostępnionym przez projekt Eclipse GMT (ang. Generative Modeling Tools).
- BoUml – (Obsługuje: C++, Java, IDL, Php, Python) obejmuje UML 2.0, tworzy dokumentację HTML, nie wymaga uprawnień administratora do instalacji, dostępny na licencji GPL do wersji 4.23 ultimate patch 7, wersja 5.0 posiada licencję komercyjną.
- Dia – ogólne narzędzie do rysowania diagramów oparte na GTK+/GNOME, które obsługuje modelowanie UML (licencja GNU GPL)
- ESS-Model – generator diagramów projektów Delphi oraz Java
- Eclipse – z platformą modelowania Eclipse (ang. Eclipse Modeling Framework, EMF) i metamodelem UML 2.0 (bez GUI)
- Fujaba – platforma developerska UML i Java; dostępna też w wersji Eclipse
- Gaphor – środowisko modelowania UML 2.0 oparte na GTK+/GNOME napisane w języku Python
- GenMyModel – środowisko modelowania UML 2.0 oparte na Javascript/HTML5
- MetaUML – Notacja tekstowa dla UML. Renderowanie Diagramów w oparciu o MetaPost, odpowiednie dla systemu składu LaTeX
- MonoUML – bazujące na najnowszym oprogramowaniu Mono, GTK+ i ExpertCoder.
- NetBeans – z „NetBeans IDE 5.5 Enterprise Pack" oraz z NetBeans IDE ≤ 6.7.1 usunięta obsługa UML z projektu
- Software Ideas Modeler – modeler UML napisany w C#
- StarUML – (Obsługuje: C/C++, Java, Visual Basic, Delphi, JScript, VBScript, C#, VB.NET) platforma UML/MDA dla systemu Windows (2000, XP), która umożliwia import projektów z takich komercyjnych aplikacji jak Rational Rose czy Borland Together. Zapewnia *forward* i *reverse engineering* kodu w Javie, C#, C++; dostępna na zmodyfikowanej licencji GNU GPL, napisana głównie w Delphi
- Umbrello – program dla Linuksa, środowisko KDE
- UMLet – łatwe w użyciu narzędzie pozwalające tworzyć diagramy UML, stworzone w Javie (licencja GNU GPL)
- UMLpad – modeler UML napisany w C++/wxWidgets, na licencji GNU GPL
- UML Sculptor – prosty, łatwy w użyciu program do tworzenia diagramów klas

Narzędzia UML - zamknięte

- ARIS Platform – rodzina programów ARIS Platform zapewnia zintegrowane portfolio narzędzi informatycznych, które pozwalają w sposób ciągły doskonalić procesy biznesowe.
- Borland Together – rodzina programów integrujących się z różnymi IDE, istnieje wersja demo
- Enterprise Architect – profesjonalne narzędzie działające na platformach Windows i Linux. Obsługuje UML 2.1.
- IBM Rational Rose (dawniej: Rational Software)
- IBM Rational Software Architect – narzędzie wspierające UML 2.0 oparte na Eclipse
- JUDE – program to tworzenia diagramów UML, diagramów związków encji, diagramów przepływu, map myśli itd.; istnieje darmowa wersja Community, nieco ograniczona (m.in. tylko do diagramów UML oraz importu/eksportu kodu Javy), jednak wciąż o sporych możliwościach i dostępna również do celów komercyjnych.
- Microsoft Visio – program z pakietu MS Office, umożliwiający (między innymi) rysowanie diagramów UML – należy jednak zwrócić uwagę, że nie umożliwia generacji kodu z diagramów, ani sprawdzania ich integralności.
- Rodzina programów iGrafx – narzędzia iGrafx począwszy od iGrafx FlowCharter obsługują tworzenie diagramów UML. Wersja testowa na witrynie iGrafx
- MagicDraw – pakiet przeznaczony również do pracy w sieci, możliwość modelowania w SysML
- Objecteering – do edycji i modelowania diagramów UML, darmowy w wersji podstawowej
- Poseidon for UML – zaawansowane narzędzie bazujące na ArgoUML, darmowa edycja Community, trial 30 dni dla zarejestrowanych użytkowników
- Sybase PowerDesigner – wydajne, rozbudowane i dopracowane narzędzie do tworzenia diagramów UML oraz schematów baz danych, procesów biznesowych i zarządzania wymaganiami. Wersja francuskojęzyczna nosi nazwę PowerAMC.
- Telelogic Tau G2
- Tormigo – polskie narzędzie wspierające zarządzanie wymaganiami w Enterprise Architect (wg informacji ze strony https://modesto.pl co najmniej od 2019 r. nie jest rozwijane przez autora, firmą MODESTO Michał Wolski)
- Visual Paradigm for UML – oprócz wersji płatnych istnieje darmowa wersja Community, która posiada ograniczenie funkcjonalności.
- Visual Paradigm SDE – SDE (środowisko programistyczne). Integruje się ze wszystkimi wiodącymi IDE (Visual Studio, Eclipse/WebSphere, Borland JBuilder, NetBeans/Sun ONE, IntelliJ IDEA, Oracle JDeveloper, BEA WebLogic Workshop). Do użytku niekomercyjnego (do nauki), dostępne w wersji Community, za darmo, z ograniczeniem funkcjonalności.

Narzędzia UML

BPM+
Business Process Management Plus

CORBA®
Common Object Request Broker Architecture™

DDS™
Data-Distribution Service for Real-Time Systems™

FIGI®
Financial Instrument Global Identifier®

IEF™
The Information Exchange Framework™

MOF™
MetaObject Facility Specification™

SYSML®
Systems Modeling Language™

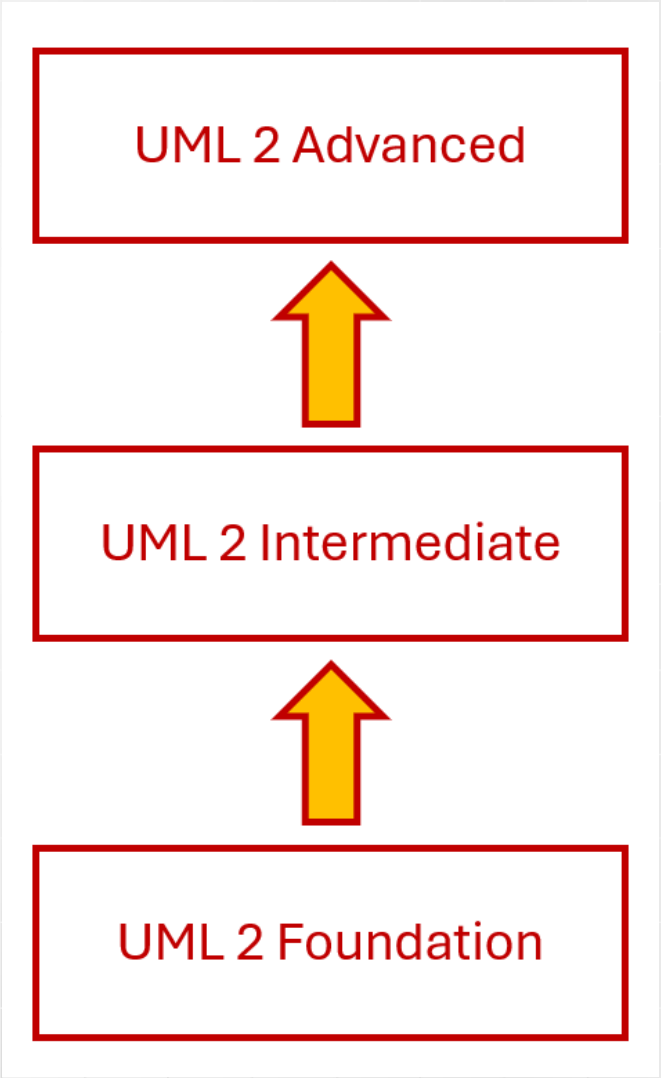
UAF®
Unified Architecture Framework®

UML®
Unified Modeling Language™

https://www.omg.org/about/omg-standards-introduction.htm

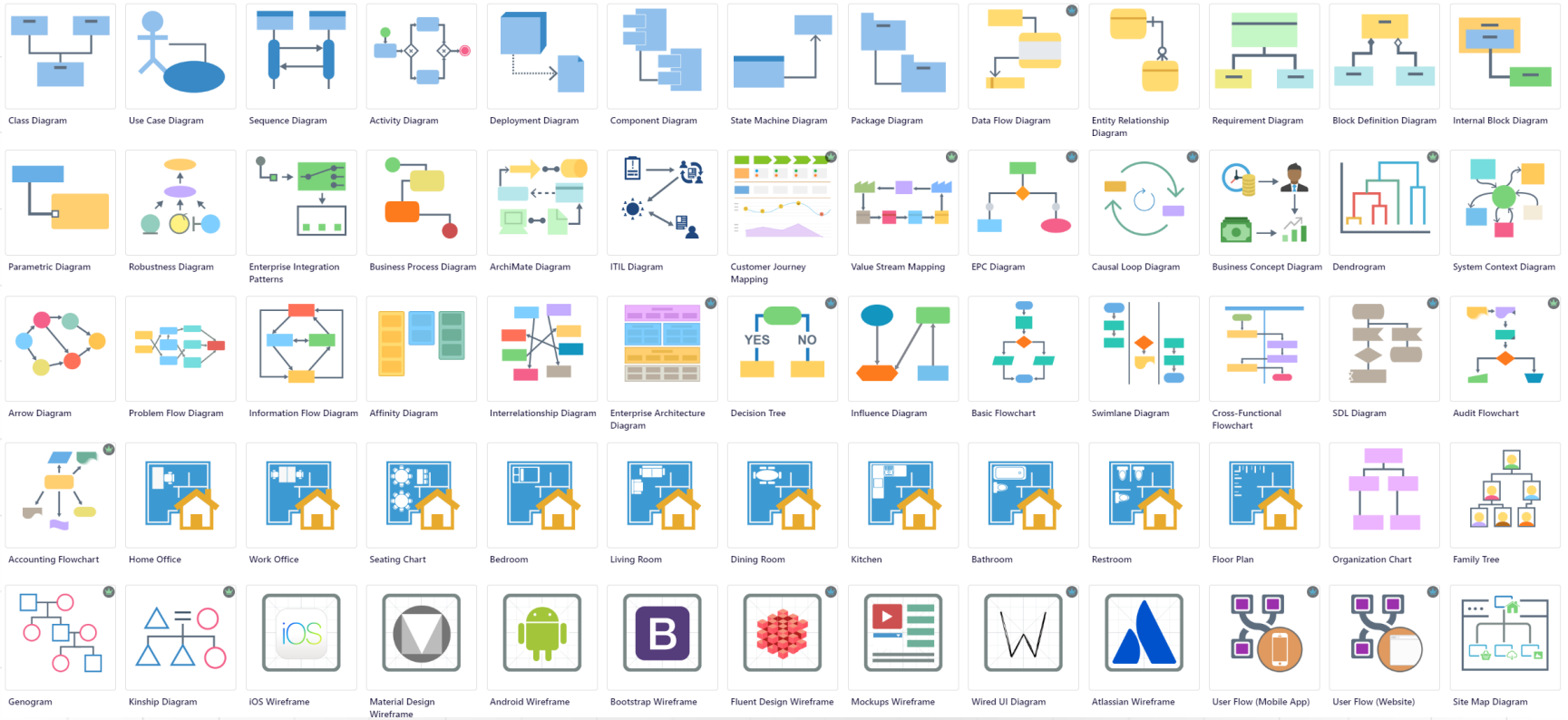
Certyfikacja UML

- https://www.omg.org/certification/uml/



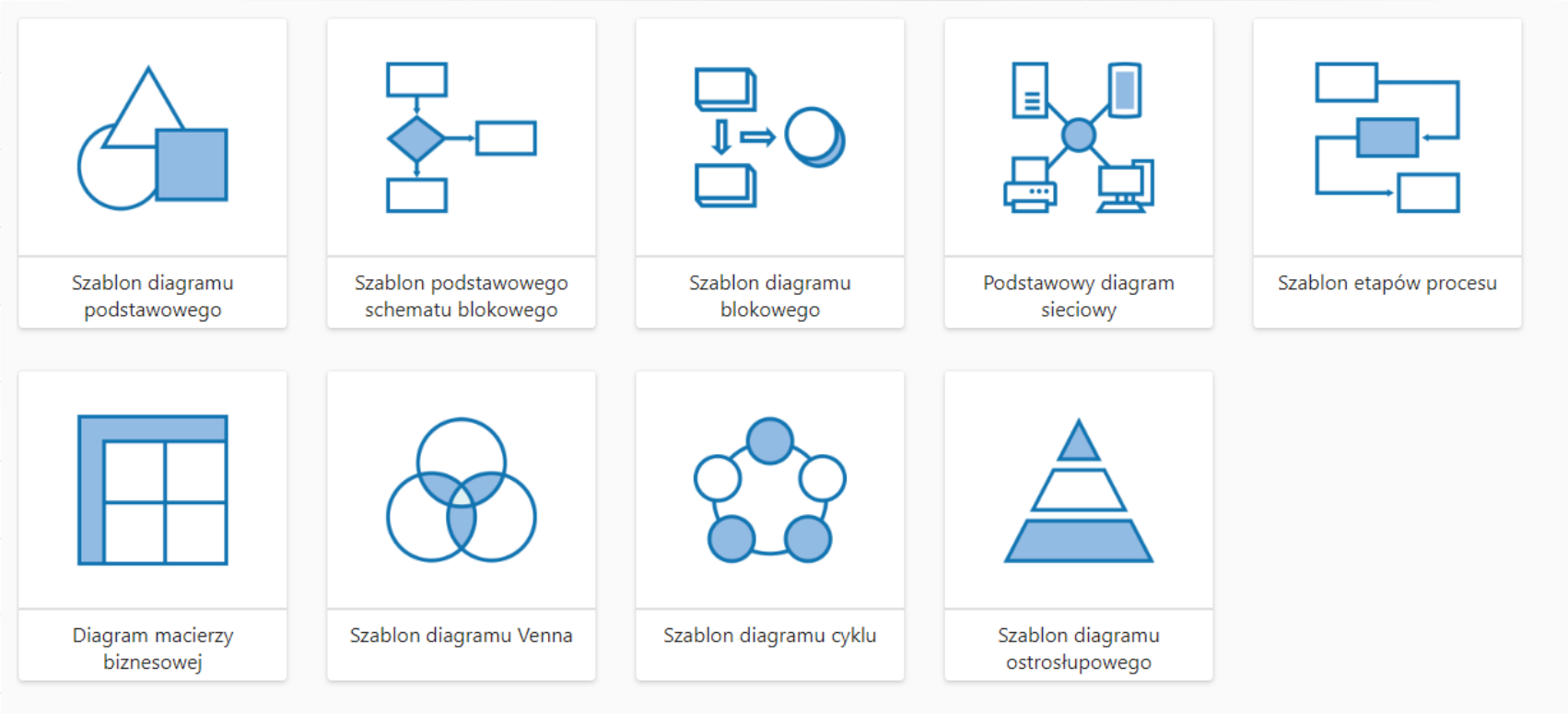
Narzędzia UML - Visual-Paradigm

- https://online.visual-paradigm.com/



Narzędzia UML – Microsoft -Visio

- https://www.microsoft.com/pl-pl/microsoft-365/visio/uml



Politechnika Wrocławska

Dziękuje bardzo za uwagę

Dr inż. Paweł Maślak