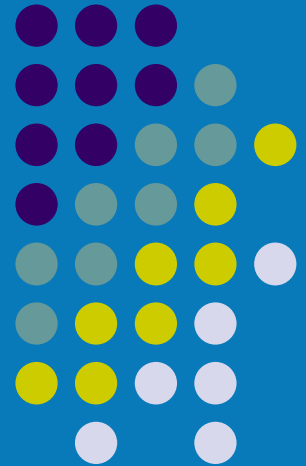


Podstawy Inżynierii Oprogramowania



Modele systemu

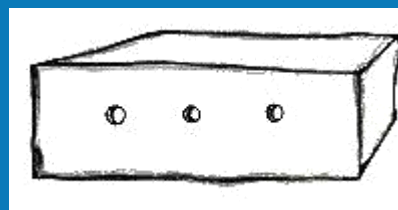


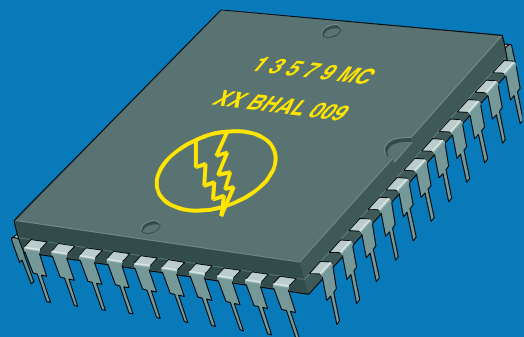




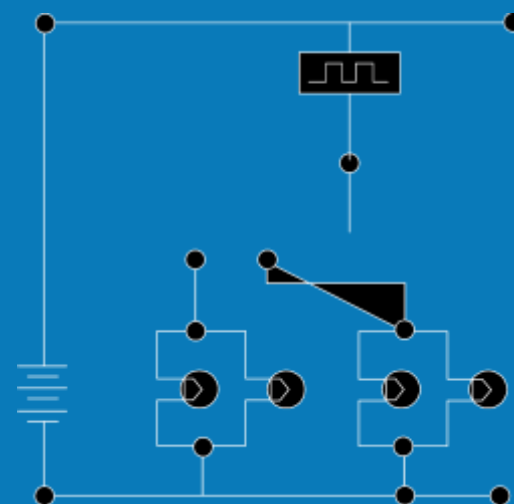


$$E = \frac{(n_1^2 + n_2^2 + n_3^2)h^2}{8mL^2}$$

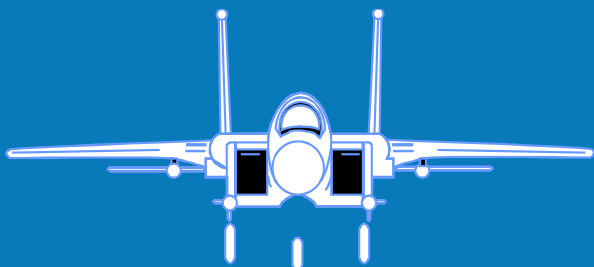
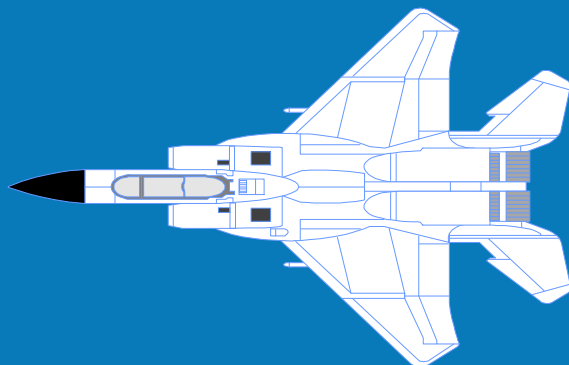
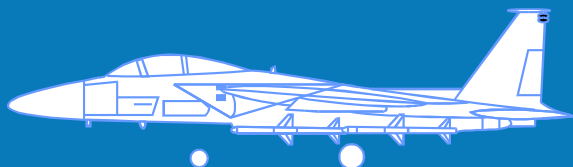




Dla klienta



Dla projektanta



Agenda

- Modelowanie pojęciowe
- Modele kontekstowe
- Modele zachowania, danych, obiektowe.
- Język UML (*ang. Unified Modeling Language*)
- Narzędzia CASE wspierające modelowanie systemu.



Modelowanie pojęciowe



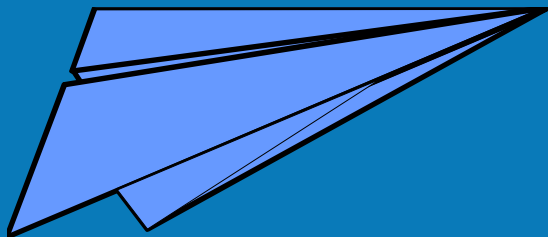
- Projektant i programista muszą dokładnie wyobrazić sobie problem oraz metodę jego rozwiązania. Zasadnicze procesy tworzenia oprogramowania zachodzą w ludzkim umyśle i nie są związane z jakimkolwiek językiem programowania.
- Pojęcia **modelowania pojęciowego** (*ang. conceptual modeling*) oraz **modelu pojęciowego** (*ang. conceptual model*) odnoszą się do procesów myślowych i wyobrażeń towarzyszących pracy nad oprogramowaniem.
- Modelowanie pojęciowe jest wspomagane przez środki wzmacniające ludzką pamięć i wyobraźnię. Służą one do przedstawienia rzeczywistości opisywanej przez dane, procesów zachodzących w rzeczywistości, struktur danych oraz programów składających się na konstrukcję systemu.

Czy zawsze musimy modelować?



Mniej istotne

Ważniejsze



Papierowy samolot



Myśliwiec

Dlaczego zespoły programistów nie modelują?



- Wiele zespołów podchodzi do wytwarzania złożonego oprogramowania jak do budowy papierowego samolociku.
 - Zaczynają pisać kod na podstawie wymagań.
 - Pracują długie godziny i piszą dłuższy kod programu.
 - Architektura nie jest planowana
 - ... Mówi się, że nad projektami informatycznymi wisi fatum porażki (*ang. doom of failure*).
 - Często jest to związane z traktowaniem myśliciela w kategoriach papierowego samolociku.

Cztery zasady modelowania



- Wybór modelu ma wpływ na to jak będziemy postrzegać rzeczywistość.
- Każdy model może być wyrażony na dowolnym poziomie szczegółowości
- Najlepsze model są bezpośrednio odnoszą się do rzeczywistości
- Pojedynczy model jest niewystarczający.

Perspektywy w modelowaniu pojęciowym

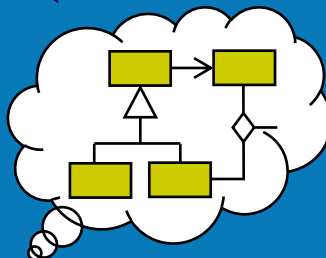


odwzorowanie

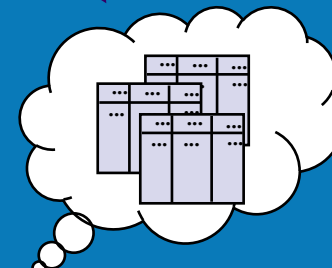
odwzorowanie



Percepcja
rzeczywistego
świata



Analityczny
model
rzeczywistości



Model
struktur danych
i procesów SI

Trwałą tendencją w rozwoju metod i narzędzi projektowania oraz konstrukcji SI jest dążenie do minimalizacji luki pomiędzy myśleniem o rzeczywistym problemie a myśleniem o danych i procesach zachodzących na danych.

Modelowanie analityczne systemu



- Modelowanie systemu pomaga analitykowi w zrozumieniu funkcjonalności systemu oraz w komunikacji z klientem.
- Modele mogą prezentować system z różnych punktów widzenia
 - **Zewnętrznego** – pokazuje kontekst lub środowisko systemu;
 - **Zachowania** – modeluje zachowanie systemu;
 - **Strukturalnego** – modeluje architekturę systemu lub strukturę przetwarzania danych.

Typy modeli



- **Model przetwarzania danych** – jak dane są przetwarzane na różnych etapach pracy systemu.
- **Model kompozycji** – jak elementy systemu komponowane są z mniejszych fragmentów.
- **Model architektoniczny** – z jakich zasadniczych podsystemów składa się system.
- **Model klasyfikacji** – wspólne cechy poszczególnych elementów systemu.
- **Model bodziec-reakcja** – w jaki sposób system reaguje na zdarzenia zachodzące tak poza nim jak i w jego wnętrzu.

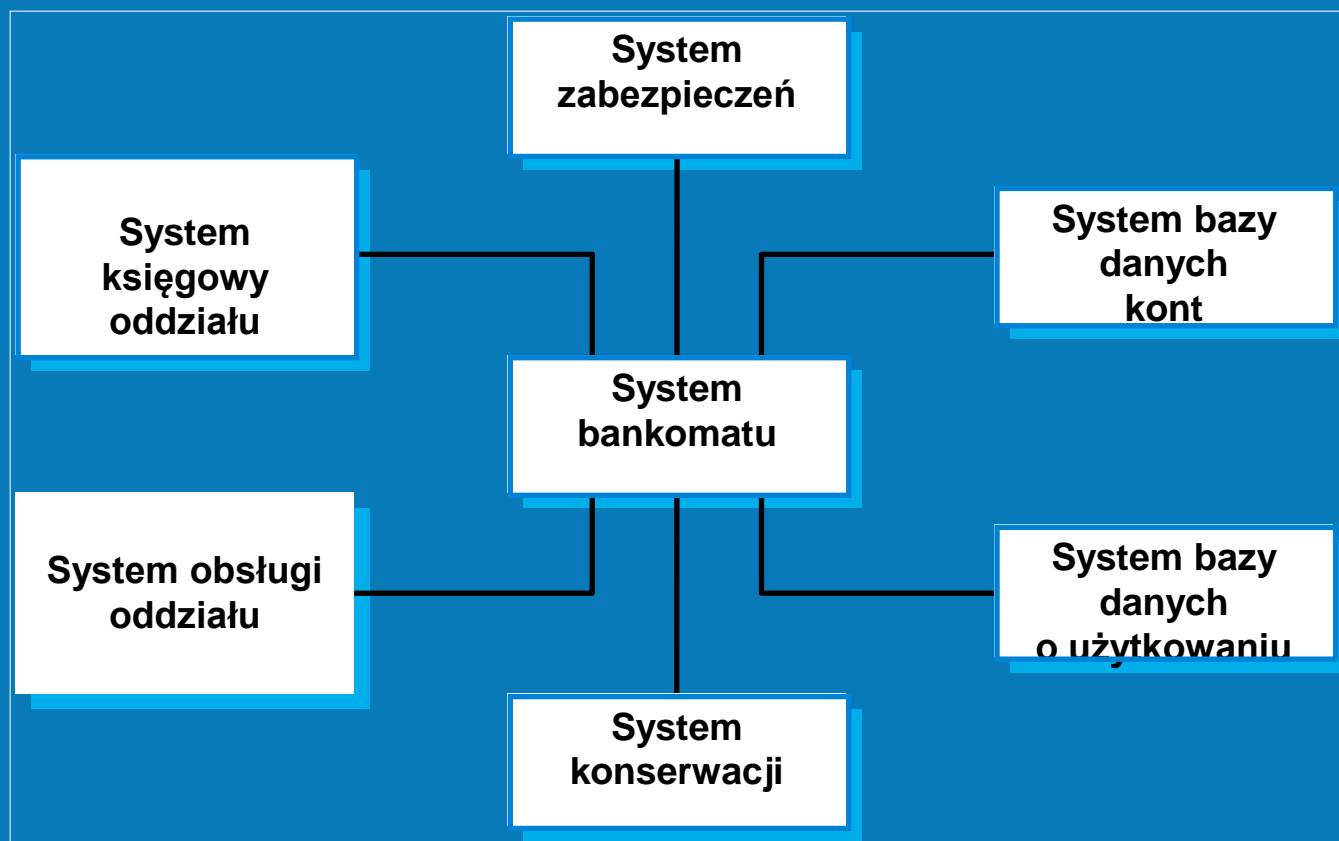
Modele kontekstowe

Context models



- Ilustrują operacyjny kontekst systemu-otoczenie.
- Kwestie społeczne i organizacyjne mogą mieć wpływ na ustalenia co do systemu należy a co jest poza jego granicami.
- Modele architektoniczne pokazują system i jego powiązania z innymi systemami.

Kontekst systemu bankomatu



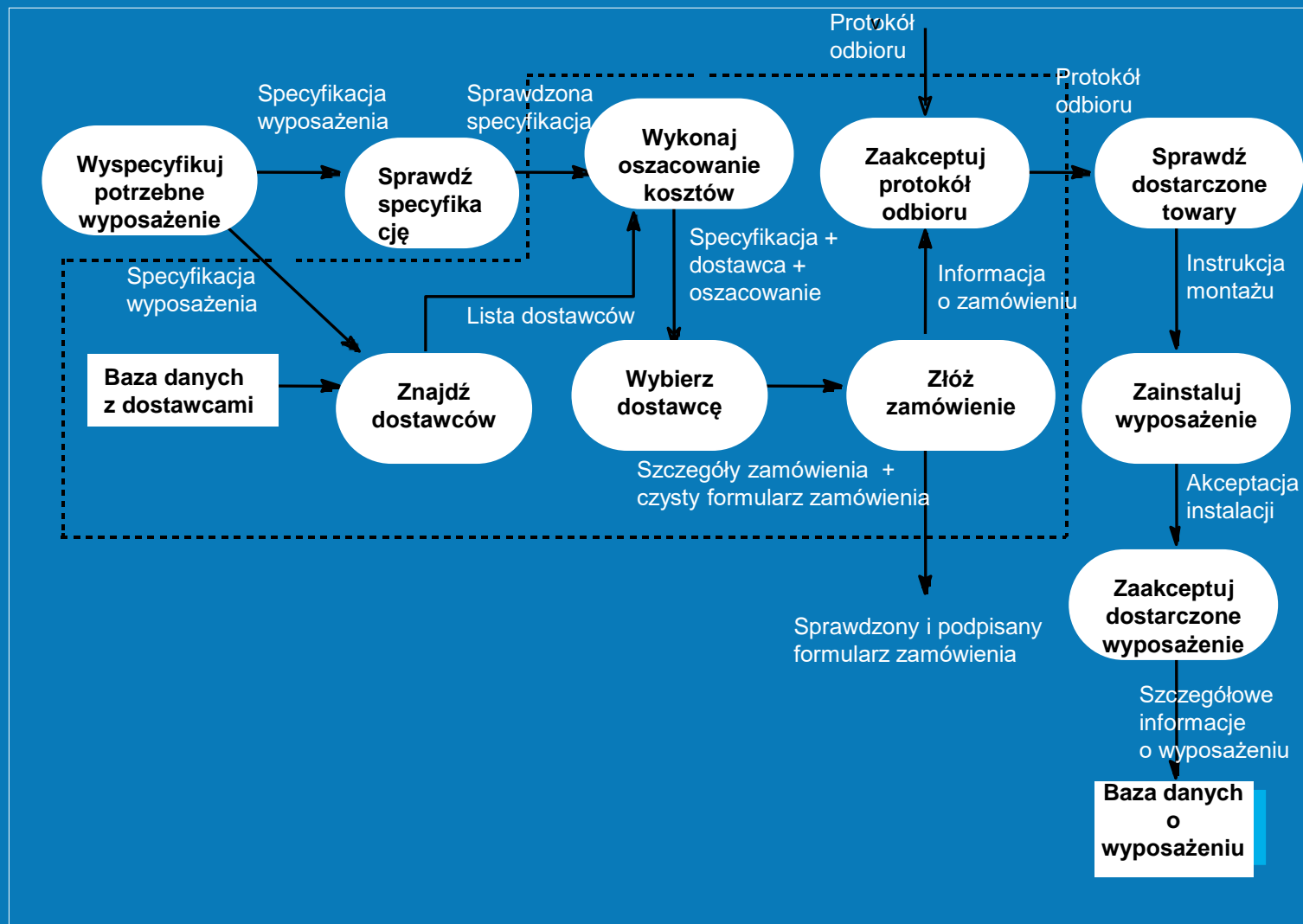
Modele procesu

Process models



- Pokazują czynności wspierane przez system.
- Uzupełniają model kontekstowy i pomagają w podjęciu decyzji, które czynności będą wykonywane automatycznie.

Model procesu zakupu wyposażenia dla firmy



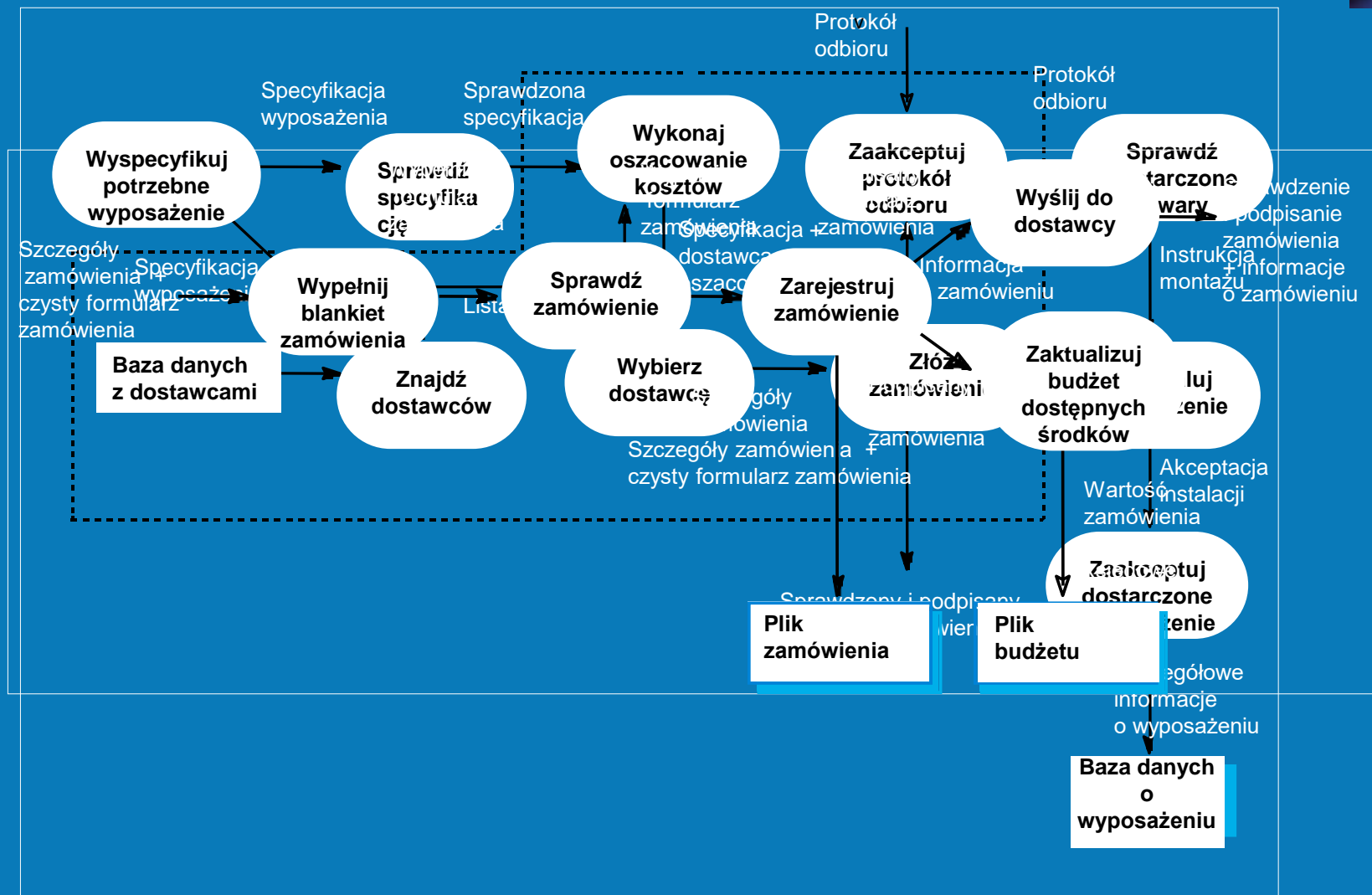
Modele zachowania

Behavioural models

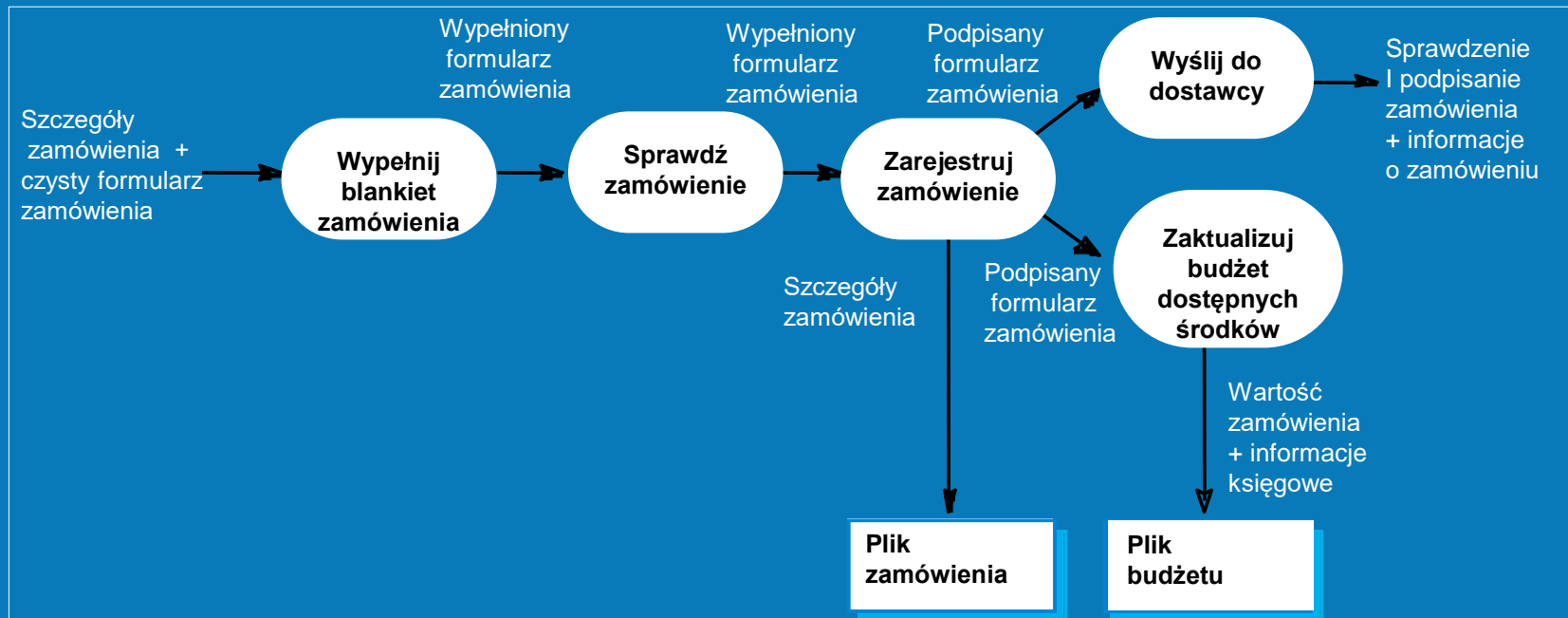


- Opisują ogólne zachowanie systemu.
- Typy:
 - **Modele przetwarzania (przepływu) danych** – pokazują sposób w jaki dane są przetwarzane i jak przepływają przez system;
 - **Modele stanów** (maszyna stanów) pokazujące reakcje systemu na zdarzenia.
- Pozwalają spojrzeć na zachowanie systemu z różnych punktów widzenia.

Model procesu zakupu wyposażenia dla firmy



Obsługa zamówienia – diagram przepływu danych



Maszyny stanów

State machine models



- Opisują zachowanie systemu w reakcji na wewnętrzne lub zewnętrzne zdarzenia.
- Pokazują odpowiedzi systemu na określone stymulacje, dlatego często wykorzystywane są do modelowania systemów czasu rzeczywistego.
- Maszyna stanu pokazuje system w postaci zbioru stanów oraz możliwych przejść pomiędzy nimi wraz ze zdarzeniami, które to przejście powodują.
- Do graficznego przedstawienia maszyny stanów wykorzystuje się diagramy stanów.

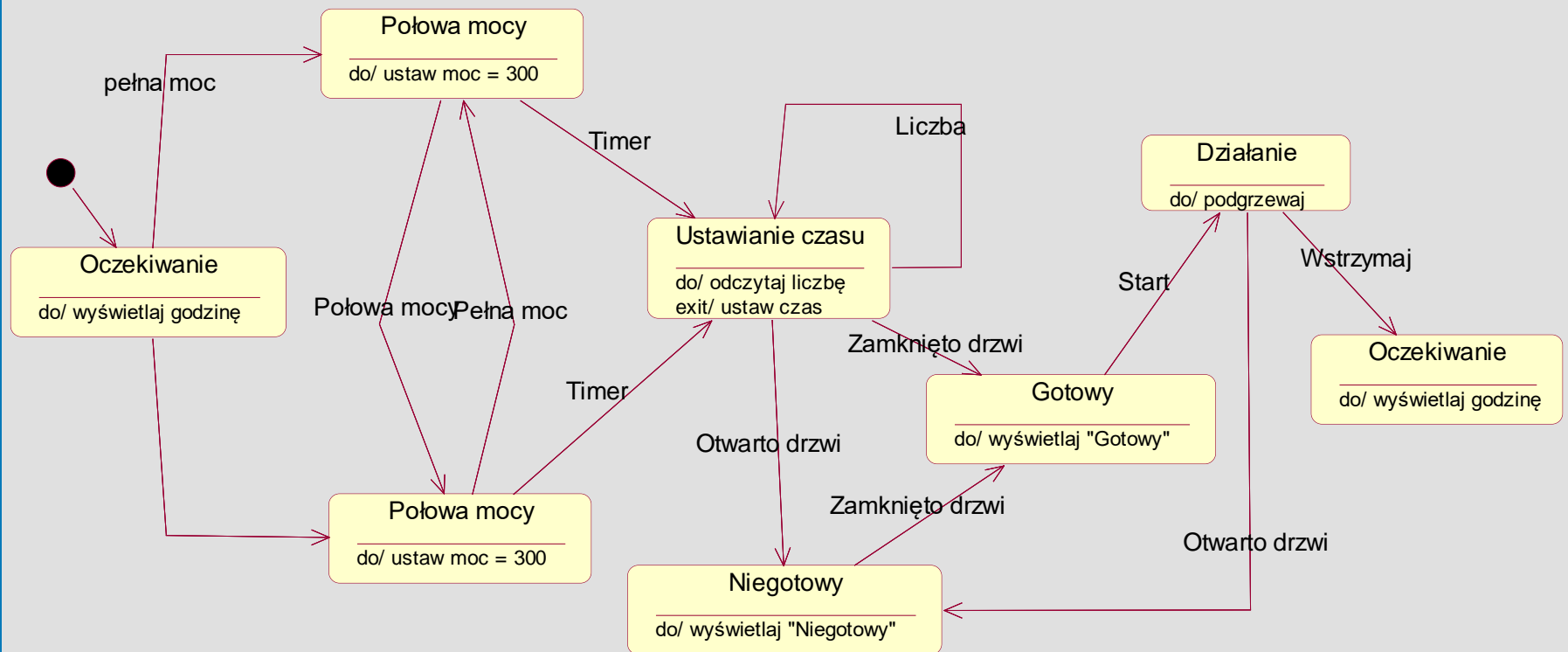
Diagramy stanów

Statecharts

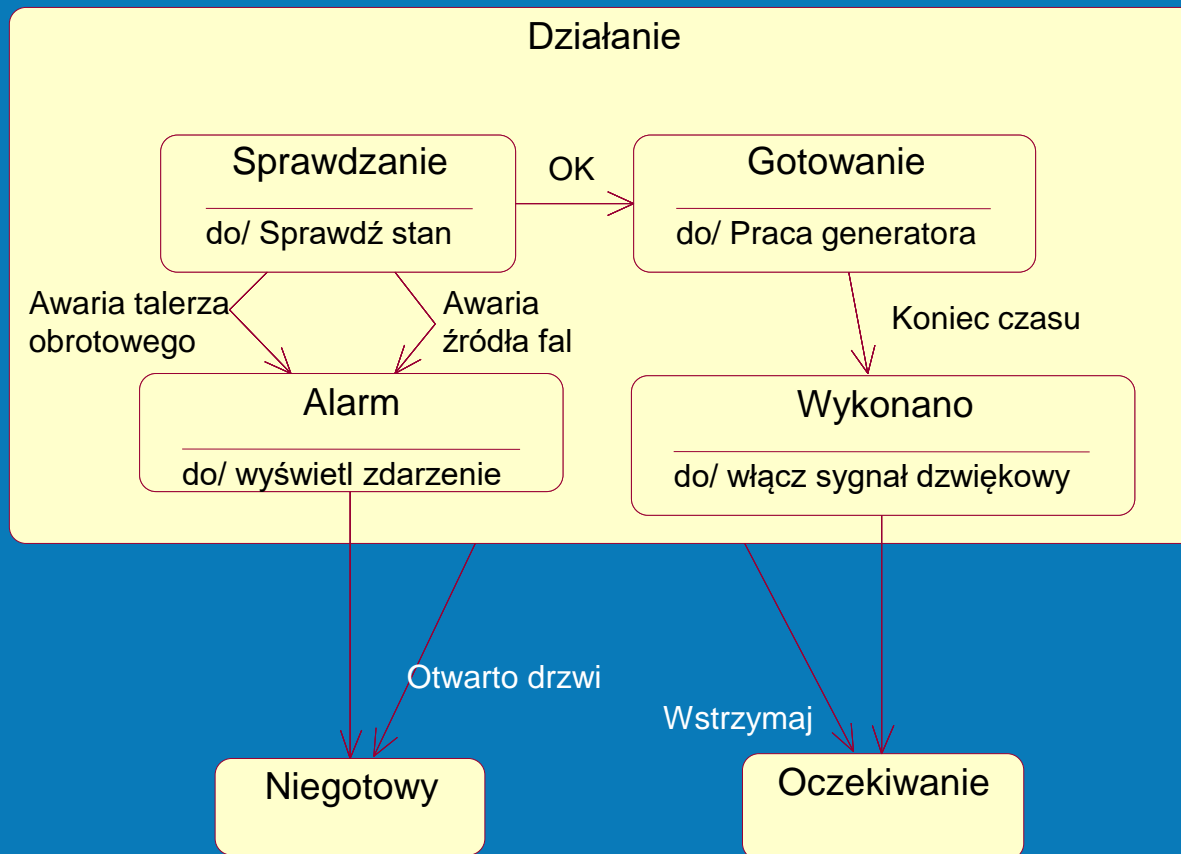


- Umożliwiają poziomowanie modelu (dekompozycję na pod-modele).
- W każdym stanie można opisać akcję która jest wykonywana.
- Mogą być wspomagane tabelami szczegółowo opisującymi stany i pobudzenia.

Diagram stanów dla mikrofalówki



Poziomowanie - superstan



Modele danych

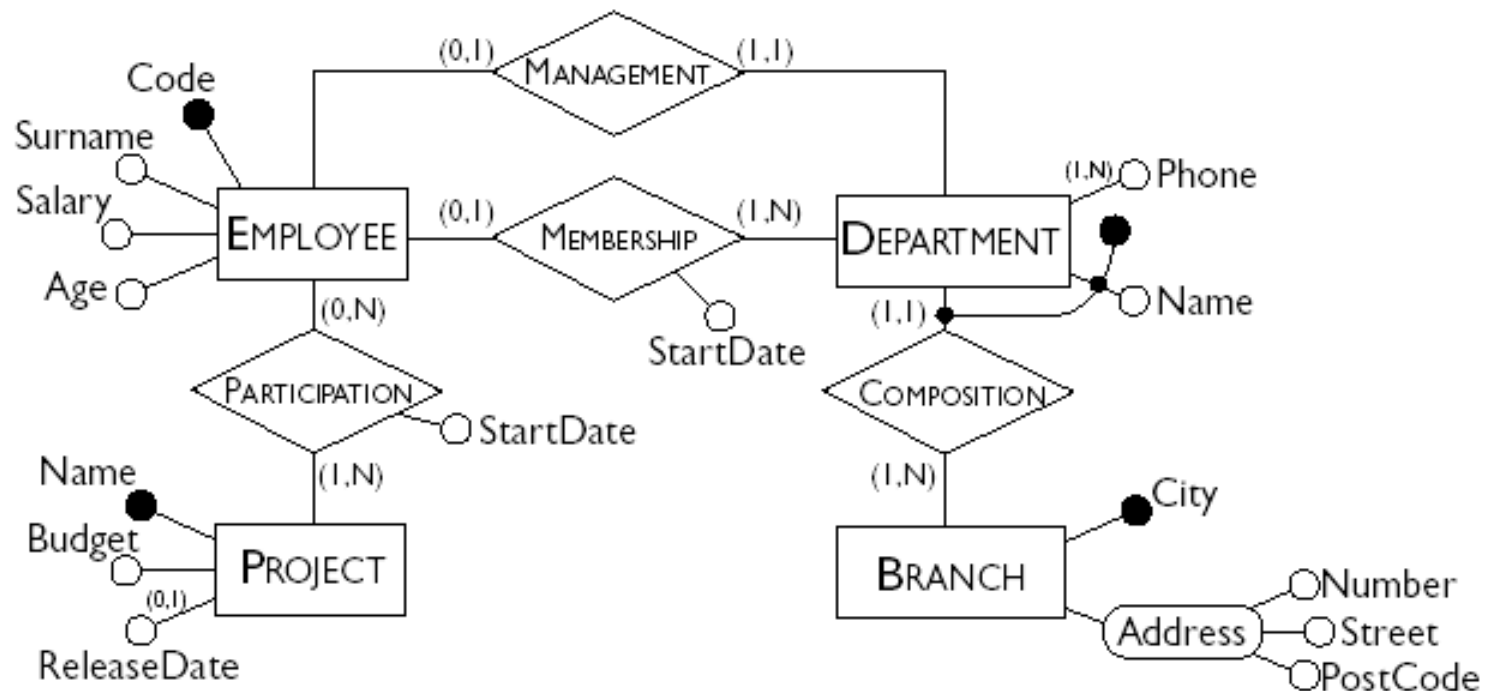
Data models



- Opisują logiczną strukturę danych, które przetwarza system (np. model Encja-Związek ang. Entity–Relationship i jego odmiany).
 - **Encja** – koncepcyjnie niezależna jednostka (obiekt)
 - **Związek/relacja** – koncepcyjny związek pomiędzy encjami.
 - **Atrybut** – własność encji
- Szeroko stosowane w projektowaniu baz danych (mogą być łatwo przekształcone w model relacyjny).

Diagram Encja-Związek

Entity-Relationship diagram



Słowniki danych



- Alfabetyczna lista nazw używanych w różnych modelach systemu, również elementów modelu danych (encji, związków, atrybutów)
- Zalety
 - Wspiera proces zarządzania nazwami i umożliwia unikanie duplikatów;
 - Przechowuje wiedzę nabywaną podczas projektu; wiąże ze sobą różne poziomy modelowania oraz implementację;
- Proces tworzenia słownika jest często wspierany przez narzędzia CASE.

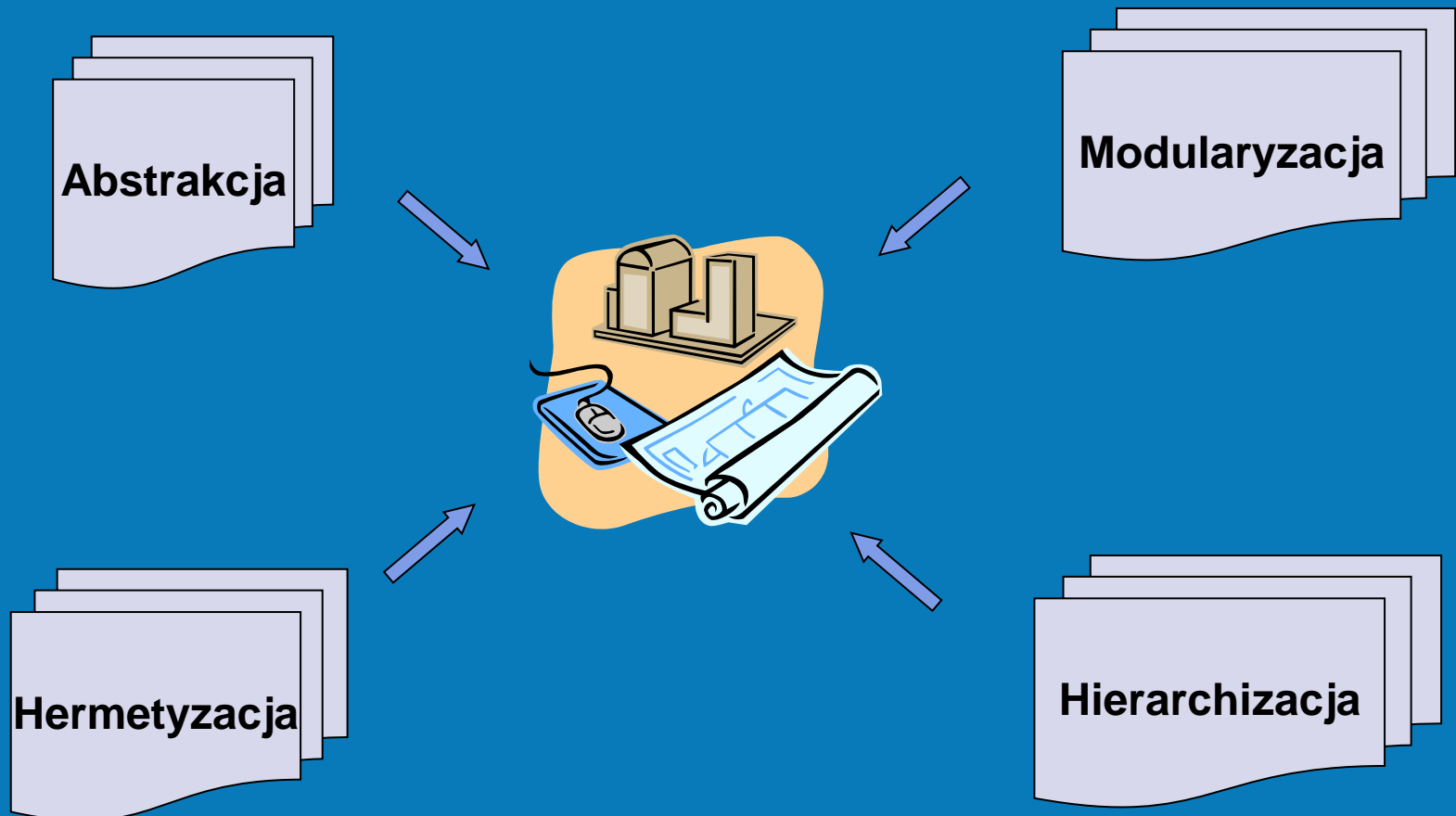
Metodyki modelowania obiektowego



- Obecnie najbardziej popularne.
- Oparte o podejście obiektowe.
- Rozwijane od lat 80-tych, oparte na wyróżnianiu obiektów łącznie z operacjami.
- Na terenie analizy i projektowania obiektowego istnieje wiele metodyk obiektowych oraz notacji graficznych służących modelowaniu (UML, OODA, OMT, OOSE, Objectory, OOSA, OOA/OOD, Fusion, OSA, OORAM, BON, MOSES/OPEN).

Obiektość

- Obiektość zmniejsza lukę pomiędzy myśleniem o rzeczywistości (dziedzinie problemowej) a myśleniem o danych i procesach, które zachodzą na danych.



Zasada abstrakcji



- Eliminacja lub ukrycie mniej istotnych szczegółów rozważanego przedmiotu lub mniej istotnej informacji.
- Wyodrębnianie cech wspólnych i niezmiennych dla pewnego zbioru bytów i wprowadzanie pojęć lub symboli oznaczających takie cechy.
- Abstrakcja definiuje granice zależne od perspektywy obserwatora.

Przykłady abstrakcji

- Student
- Profesor
- Kurs
- Ruchomy pojazd silnikowy, transportujący ludzi z miejsca na miejsce.
- Urządzenie do bezprzewodowego odbioru sygnałów



Zasada hermetyzacji – ukrywanie informacji



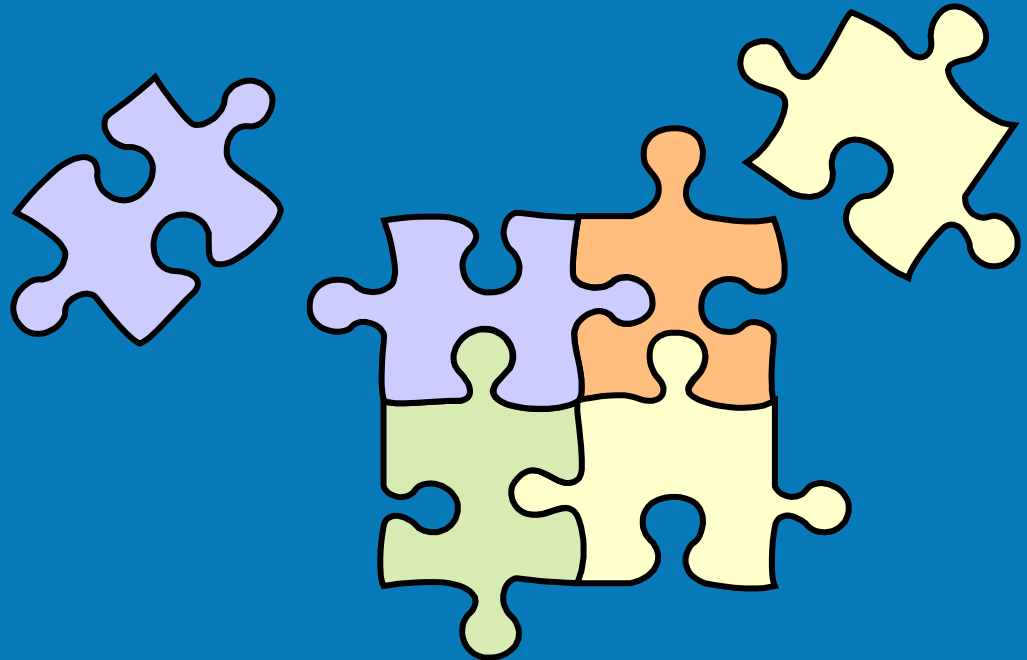
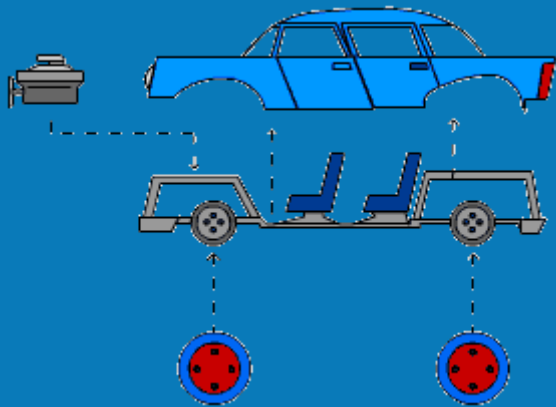
- Zasada inżynierii oprogramowania (Parnas, 1972): programista ma tyle wiedzieć o obiekcie programistycznym, ile potrzeba, aby go efektywnie użyć. Wszystko, co może być przed nim ukryte, powinno być ukryte.
- Klient zależy od interfejsu.
- Hermetyzacja i ukrywanie informacji jest podstawą pojęć modułu, klasy i ADT.



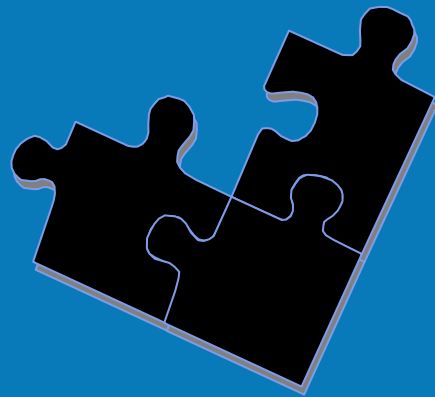
Zasada modularyzacji - dekompozycja



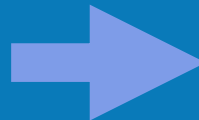
- Rozdzielenie czegoś złożonego na małe łatwiejsze do zarządzania fragmenty.
- Pomaga w zrozumieniu złożonych systemów.



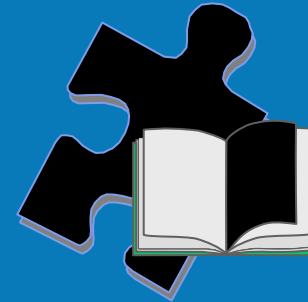
Przykład



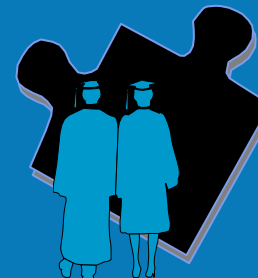
**Course Registration
System**



**System
Płacowy**



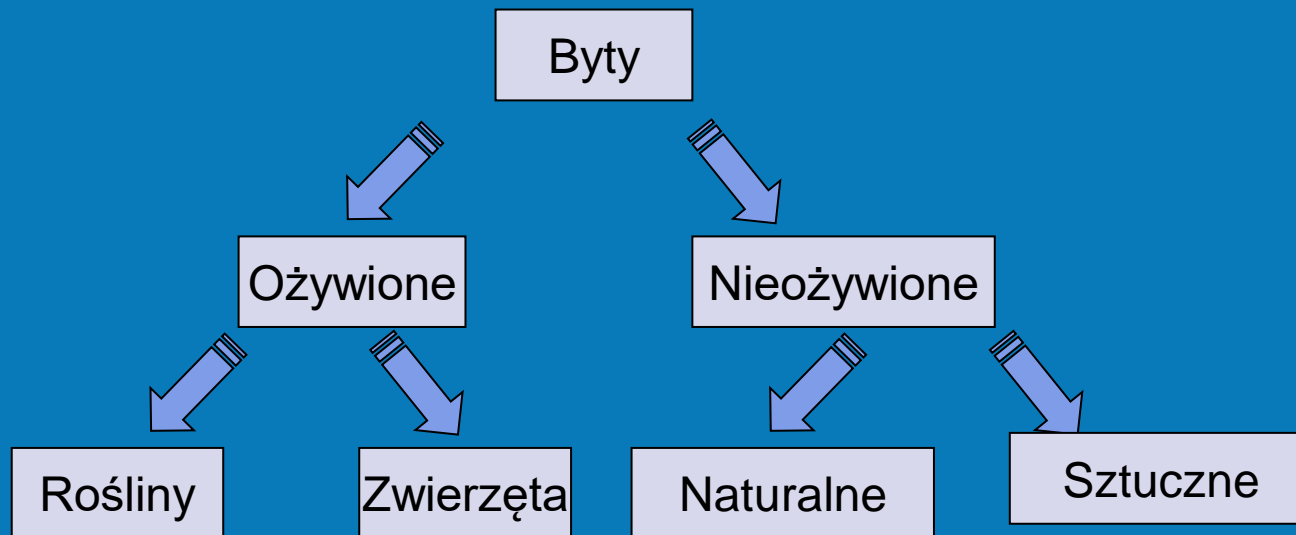
**Katalog
Kursów**



**Zarządzanie
Studentami**

Zasada hierarchizacji

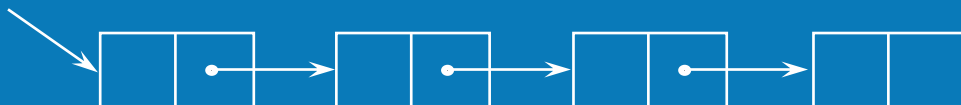
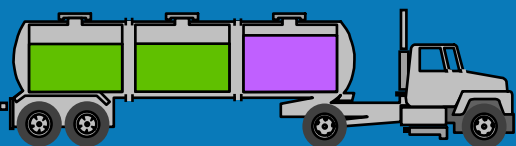
- Porządkowanie (szeregowanie) abstrakcji w strukturę drzewiastą. Rodzaje: hierarchia agregacji, klas, dziedziczenia, typów (Słownik terminów obiektowości, Friesmith, 1995)



Obiekt



- **Nieformalnie**, obiekt jest to byt obserwowalny w rzeczywistości (jej wycinku), koncepcyjny obraz tej rzeczywistości lub jednostka oprogramowania.
- **Formalnie**, obiekt jest jednostką z dobrze zdefiniowanymi granicami, który hermetyzuje stan (*ang. state*) oraz zachowanie (*ang. behavior*).



Podstawowe własności obiektu



Obiekt jest charakteryzowany poprzez:

1

Tożsamość, która odróżnia go od innych obiektów. Tożsamość obiektu jest niezależna zarówno od wartości atrybutów obiektu, jak i od lokacji bytu odwzorowywanego przez obiekt w świecie rzeczywistym czy też lokacji samego obiektu w przestrzeni adresowej komputera.

(W praktyce: tożsamość = trwały **wewnętrzny identyfikator** obiektu)

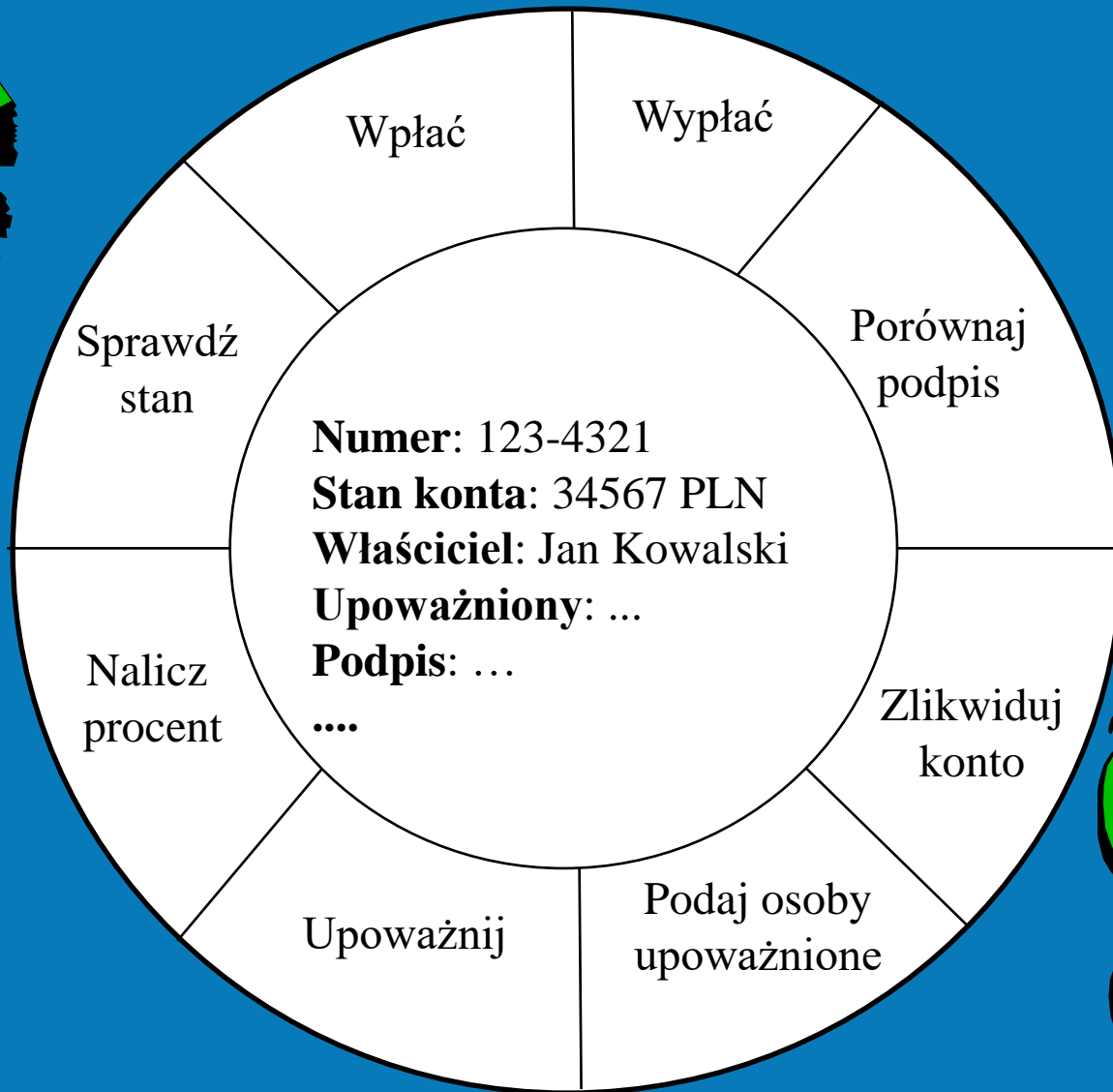
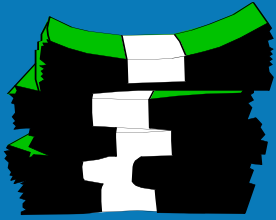
2

Stan, który może zmieniać się w czasie (bez zmiany tożsamości obiektu). Stan obiektu w danym momencie jest określony przez aktualne wartości jego atrybutów i powiązań z innymi obiektami.

3

Obiekt ma przypisane **zachowanie**, tj. zestaw operacji, które wolno stosować do danego obiektu.

Przykład obiektu



Obiekt
KONTO



Modele obiektowe

Object models



- Naturalnie odzwierciedlają elementy rzeczywistości, którymi manipuluje system.
- Opisują system w terminach klas obiektów i relacji pomiędzy nimi.
- Obiekty mogą być rzeczywiste lub abstrakcyjne.
- Identyfikacja klas obiektów jest uważana za trudny proces (wymaga głębokie znajomości dziedziny aplikacji).
- Klasy opisujące obiekty z dziedziny problemowej mają duży potencjał ponownego użycia.

Unified Modeling Language



- UML 0.8-0.9, styczeń 1995 - wrzesień 1996
- UML 1.0, styczeń 1997, przesłany do OMG
- UML 1.1, koniec 1997, zatwierdzony jako składnik standardu OMG
- UML 1.3, 1999 - 2000
- UML 1.5, Marzec 2003
- UML 2.0 2004
- UML 2.1.2 2007

Połączone siły trzech znanych metodologów oprogramowania:



Grady Booch



Ivar Jacobson



James Rumbaugh

UML: Krótka charakterystyka (1)



- UML cieszy się aktualnie bardzo dużą popularnością. Prawdopodobnie przez wiele najbliższych lat będzie dominował w obszarze analizy i projektowania.
- UML nie jest metodyką projektowania.
- Notacja UML, która opiera się o podstawowe pojęcia obiektowości może być wykorzystana w dowolnej metodyce.
- Pojęcia UML, wynikające z doświadczenia jej twórców, mają w założeniu przykrywać większość istotnych aspektów modelowanych systemów.

UML: Krótka charakterystyka (2)



- Wady i zalety metodyk, których autorami są twórcy UML:
 - OMT (Rumbaugh): dobry do modelowania dziedziny przedmiotowej. Nie przykrywa dostatecznie dokładnie zarówno aspektu użytkowników systemu, jak i aspektu implementacji (konstrukcji).
 - OOSE (Jacobson): dobrze podchodzi do kwestii modelowania użytkowników i cyklu życiowego systemu. Nie przykrywa dokładnie modelowania dziedziny przedmiotowej jak i aspektu implementacji (konstrukcji).
 - OOAD (Booch): dobrze podchodzi do kwestii projektowania, konstrukcji i związków ze środowiskiem implementacji. Nie przykrywa dostatecznie dobrze fazy rozpoznania i analizy wymagań użytkowników.
- Celem UML jest przykrycie również tych aspektów.

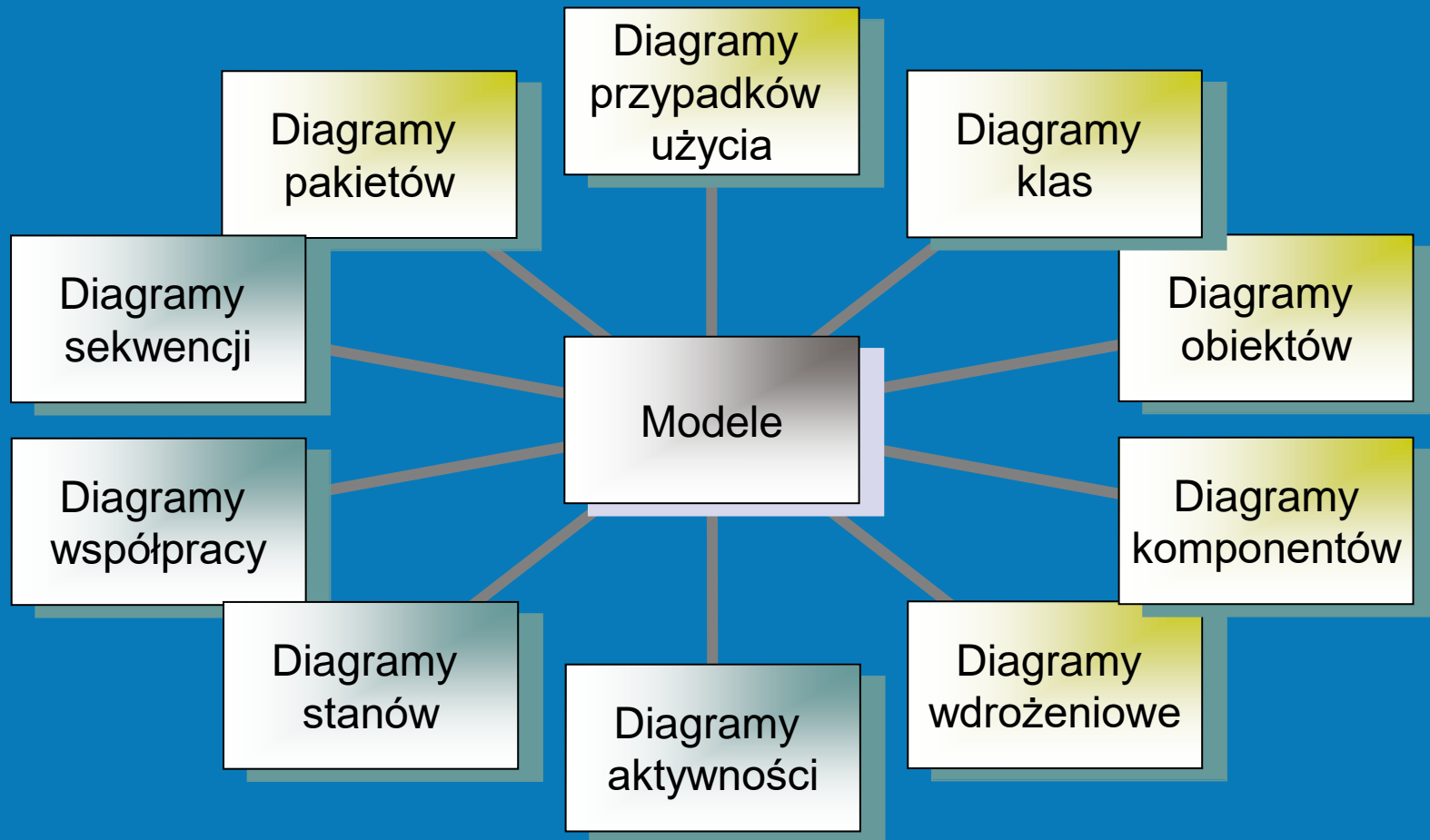
Perspektywy modelowania w UML



UML jest określany jako język modelowania z 4+1 perspektywą :

- **Perspektywa przypadków użycia** – opisuje funkcjonalność systemu widzianą przez użytkowników
- **Perspektywa logiczna** – sposób realizacji funkcjonalności, struktura systemu widziana przez projektanta
- **Perspektywa implementacyjna** – zawiera moduły i interfejsy, przeznaczona dla programisty
- **Perspektywa procesowa** – podział systemu na czynności i jednostki wykonawcze (wątki, procesy, współbieżność) – służy głównie programistom i instalatorom
- **Perspektywa wdrożeniowa** – fizyczny podział elementów systemu i ich rozmieszczenie w infrastrukturze, ważna dla instalatorów

Diagramy definiowane w UML



Model a diagram. Modele w UML



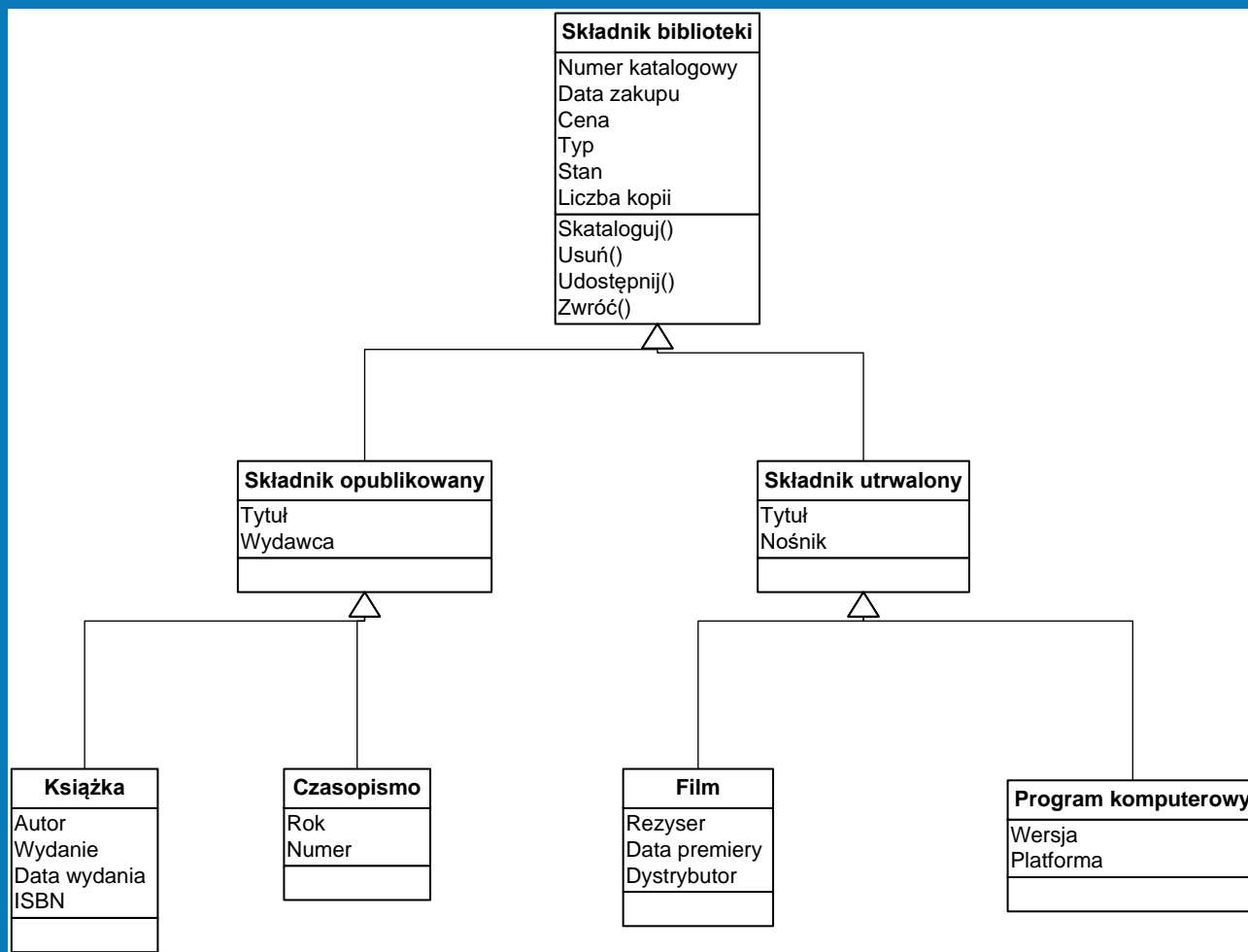
- **Model** - pewna abstrakcja projektowanego systemu, widziana z określonej perspektywy, na określonym poziomie szczegółowości.
- **Diagram** - środek służący do opisu modelu. Dany model może być opisany przy pomocy wielu diagramów. Dany element modelu może pojawiać się na wielu diagramach jednego modelu.
- Dwa najważniejsze modele w UML, wykorzystywane w fazie analizy (modelowania), to:
 - **model przypadków użycia** opisujący system widziany z perspektywy jego przyszłego użytkownika (za pomocą diagramów przypadków użycia),
 - **model obiektowy** przedstawiający statyczną budowę, czyli strukturę systemu (za pomocą diagramów klas i diagramów obiektów). Diagram klas może zawierać obiekty. Diagram obiektów nie zawiera klas, ale wyłącznie obiekty.

Modele obiektowe a UML

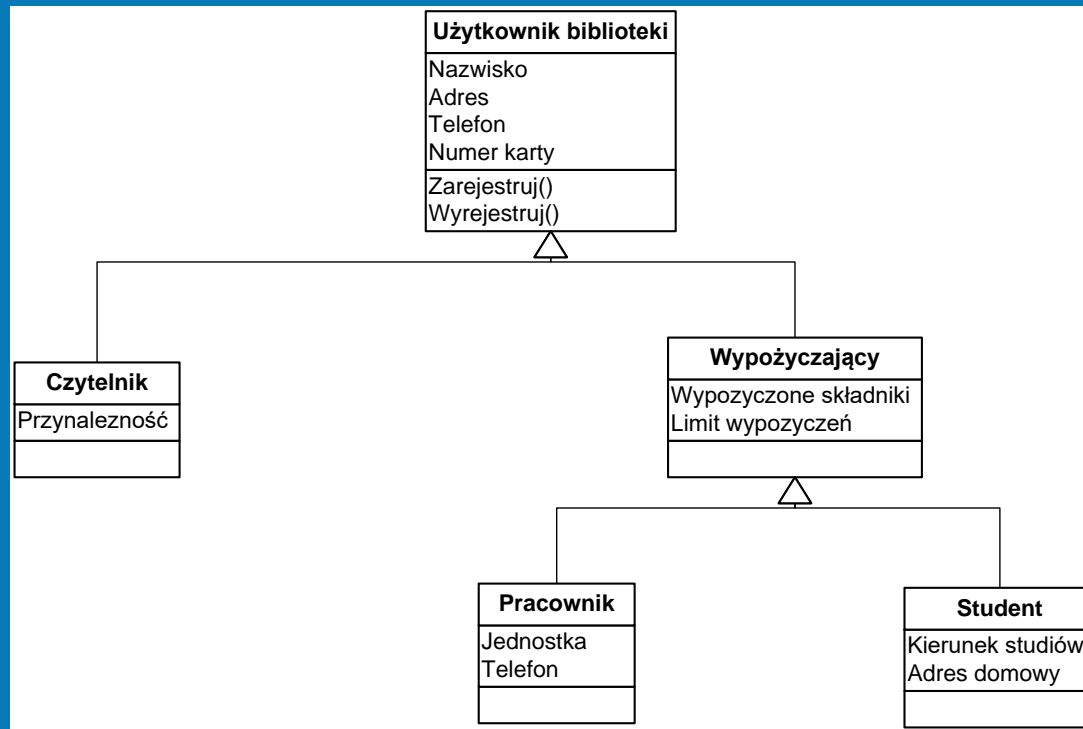


- Notacja:
 - Klasy obiektów reprezentowane są jako prostokąty
Opis podzielony jest na trzy części
 - Nazwa
 - Atrybuty
 - Operacje
 - Powiązania pomiędzy klasami (zwane asocjacjami) reprezentowane są przez linie łączące klasy;
 - Dziedziczenie, określane mianem związku generalizacji modeluje związek „jest rodzaju”
 - Agregacja modelująca zależności typu całość - część

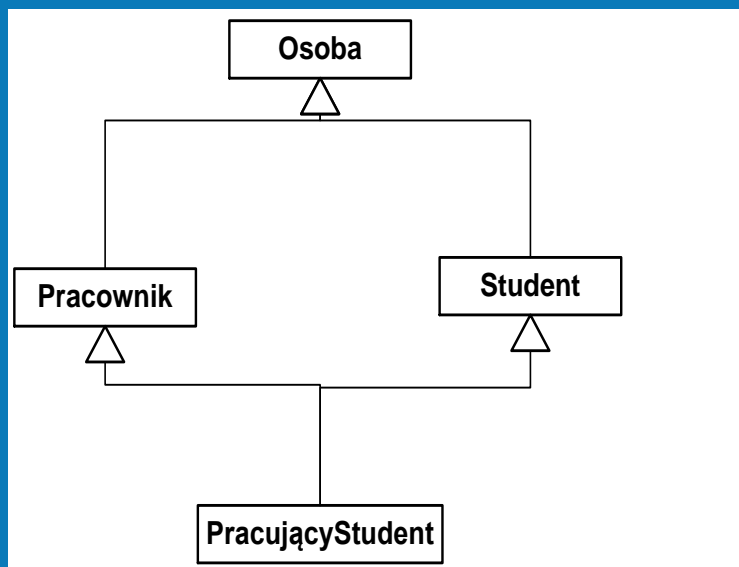
Hierarchia klas w systemie biblioteki



Hierarchia klas reprezentujących użytkowników



Dziedziczenie wielokrotne



- W systemie wspierającym wielokrotne dziedziczenie klasa może dziedziczyć z więcej niż jednej nad-klasy.
- Może to prowadzić do konfliktów znaczeniowych gdy w nad-klasach znajdują się takie same nazwy atrybutów/usług o różnej semantyce.
- Dziedziczenie wielokrotne utrudnia ewentualną reorganizację hierarchii.

Agregacja obiektów



- Model agregacji pokazuje jak klasy będące kolekcjami są komponowane z innych klas.

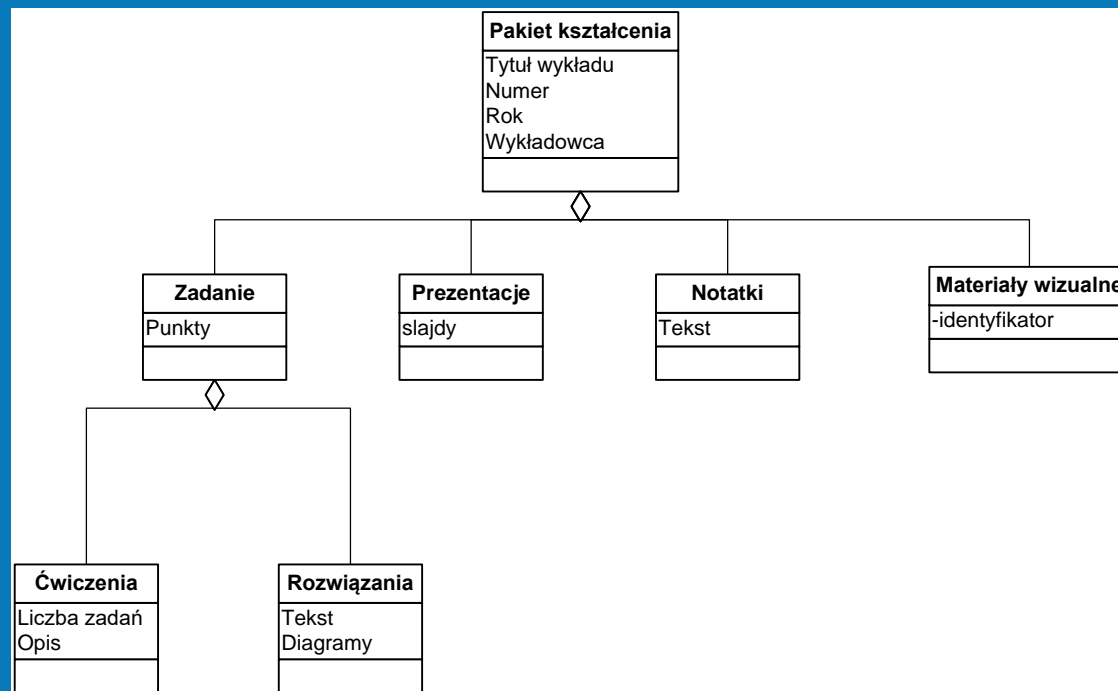
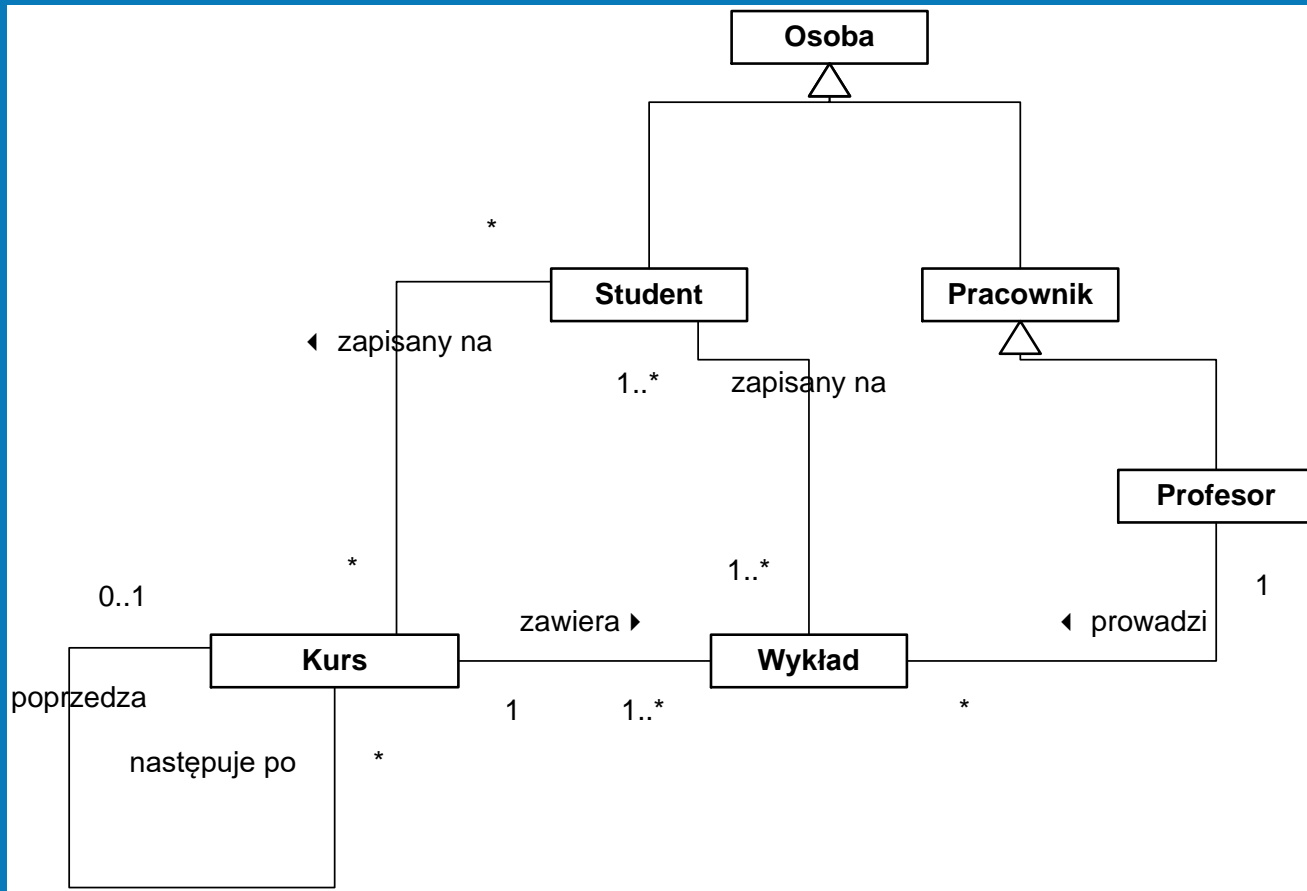


Diagram klas z dziedziczeniem i asocjacjami

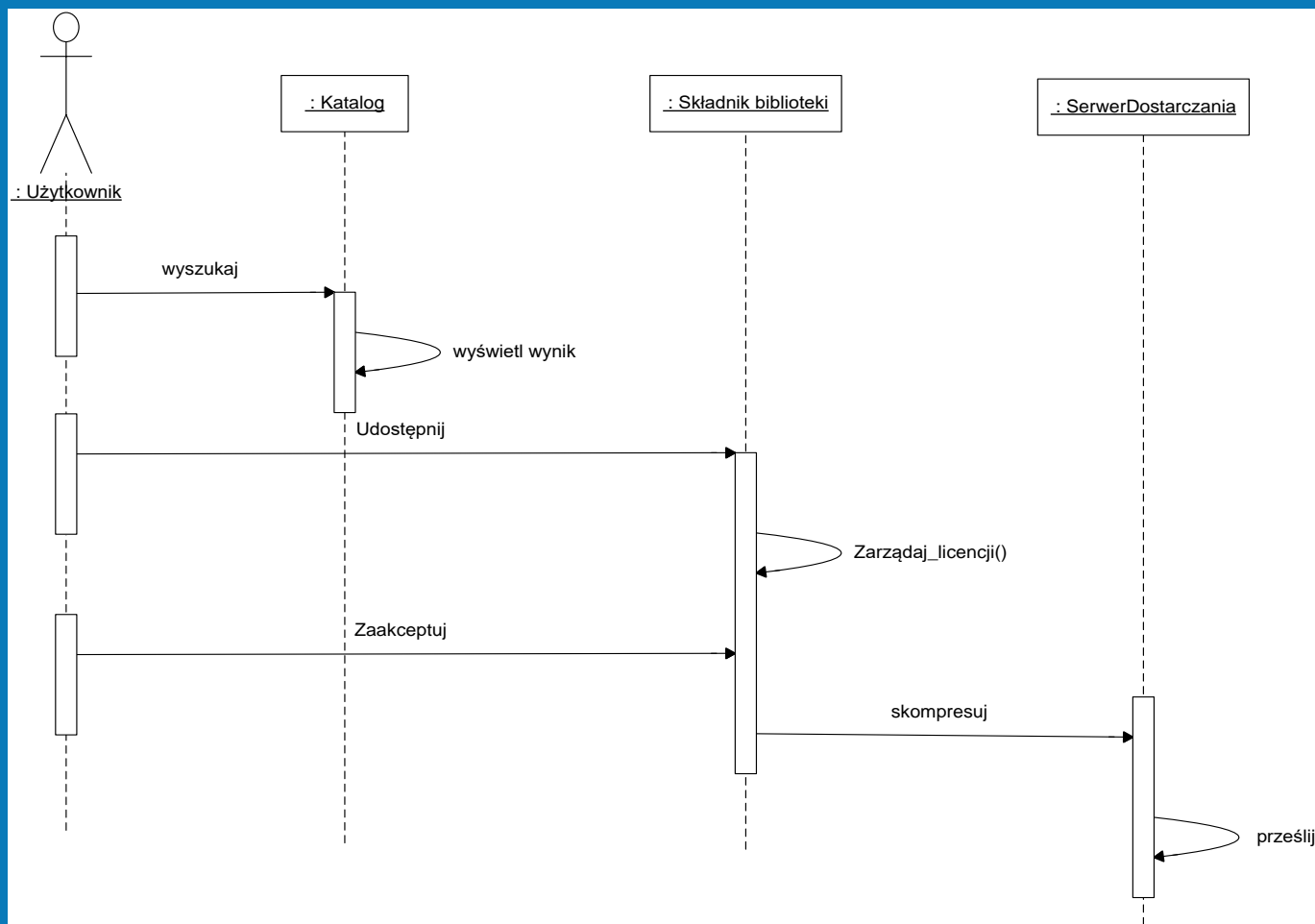


Modelowanie zachowania obiektów



- Głównym zadaniem pomocniczego **modelu dynamicznego** (zachowania) w UML jest wypełnienie diagramu klas metodami wynikającymi z analizy zachowania systemu w trakcie wykonywania zadań, gdzie zadaniem może być np. realizacja przypadku użycia czy też jednego konkretnego scenariusza danego przypadku użycia.
- W UML do modelowania zachowania wykorzystuje się głównie diagramy sekwencji i współpracy wspomagane diagramami stanów.

Pobranie składownika elektronicznego



Słabości modeli



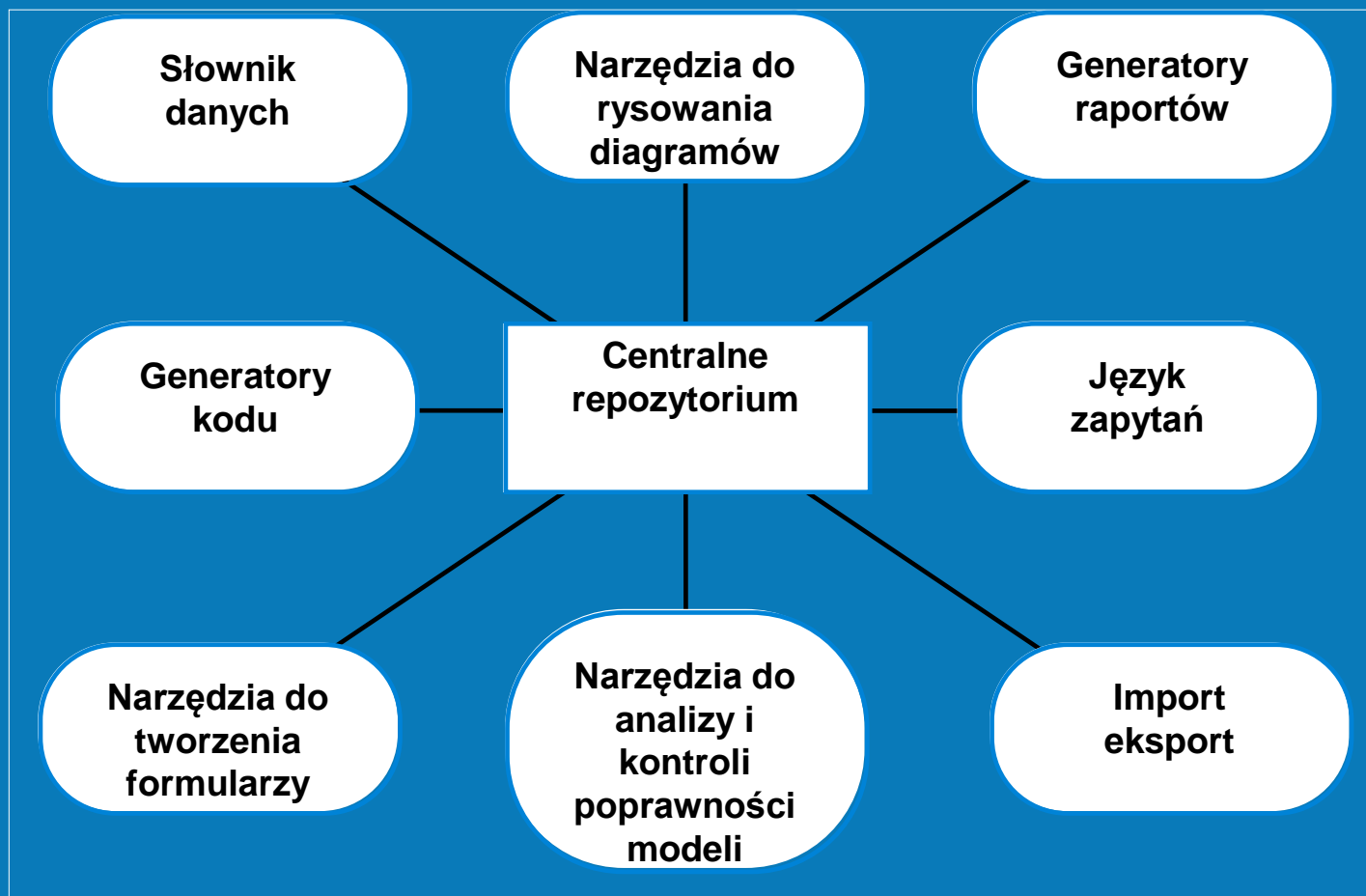
- Nie modelują wymagań niefunkcjonalnych.
- Zazwyczaj nie zawierają informacji o tym czy dana metoda rozwiązania jest odpowiednia do problemu.
- Mogą produkować nadmierną liczbę dokumentacji.
- Czasami modele są zbyt szczegółowe przez co trudne do zrozumienia przez użytkowników.

Narzędzia CASE



- Spójny zestaw narzędzi zaprojektowany w celu wsparcia określonej aktywności np. analizy, projektowania czy testowania.
- Narzędzia do analizy i projektowania wspomagają modelowanie systemu tak w procesie inżynierii wymagań jak i w procesie projektowania systemu.
- Narzędzia takie mogą wspomagać specyficzne metodyki lub udostępniać możliwość tworzenia wielu różnych typów modeli.

Struktura narzędzi CASE do analizy i projektowania



Do poczytania



- Sommerville I.: *Inżynieria Oprogramowania*, rozdział 7.
- Dąbrowski, Subieta: *Podstawy Inżynierii Oprogramowania*, rozdział 4.
- O UML'u książek jest dużo...
 - Tak na początek,...
 - Booch G., Rumbaugh J., Jacobson I.: *UML przewodnik użytkownika*, WNT, 2002.

Internet



- UML
 - www.uml.org
 - <http://www-306.ibm.com/software/rational/uml/>