

# ***Bazy danych***



***Andrzej Łachwa, UJ, 2015***

***12/14***

Po wykonaniu modelu danych przechodzimy do budowy modeli procesów.

Narzędzia modelowania wzajemnie się uzupełniają, a każde z nich ukazuje wybrany aspekt systemu. Aby analiza była spójna modelowanie danych i modelowanie procesów powinno być wykonywane jednocześnie.

W analizie procesów system rozkładamy stopniowo na części w taki sposób, aby każda część dała się opisać na jednej stronie (minispecyfikacje), a także aby każda część była funkcjonalnym elementem całości.

Narzędziem modelowania są diagramy przepływu danych (DPD). Diagramy te rysujemy na różnych poziomach w taki sposób, by od spraw ogólnych przechodzić do coraz bardziej szczegółowych. Analizę tę prowadzimy tak głęboko, jak trzeba, by zrozumieć funkcjonowanie systemu rzeczywistego.

DPD jest modelem działania systemu rzeczywistego: jest mapą na której widzimy rzeczywiste przepływy danych, rzeczywiste procesy, magazyny danych i terminatory. System może działać, jeżeli każdy proces produkuje poprawne dane wyjściowe na podstawie danych wejściowych, które otrzymuje z przepływów danych bądź pobiera z magazynów danych.

Terminatory to obiekty znajdujące się na zewnątrz naszego systemu: firmy, osoby, systemy etc. Terminatory wysyłają dane do systemu i/lub otrzymują dane przetworzone przez system. System przenosi dane (przepływy), przetwarza je (procesy) i przechowuje (magazyny).

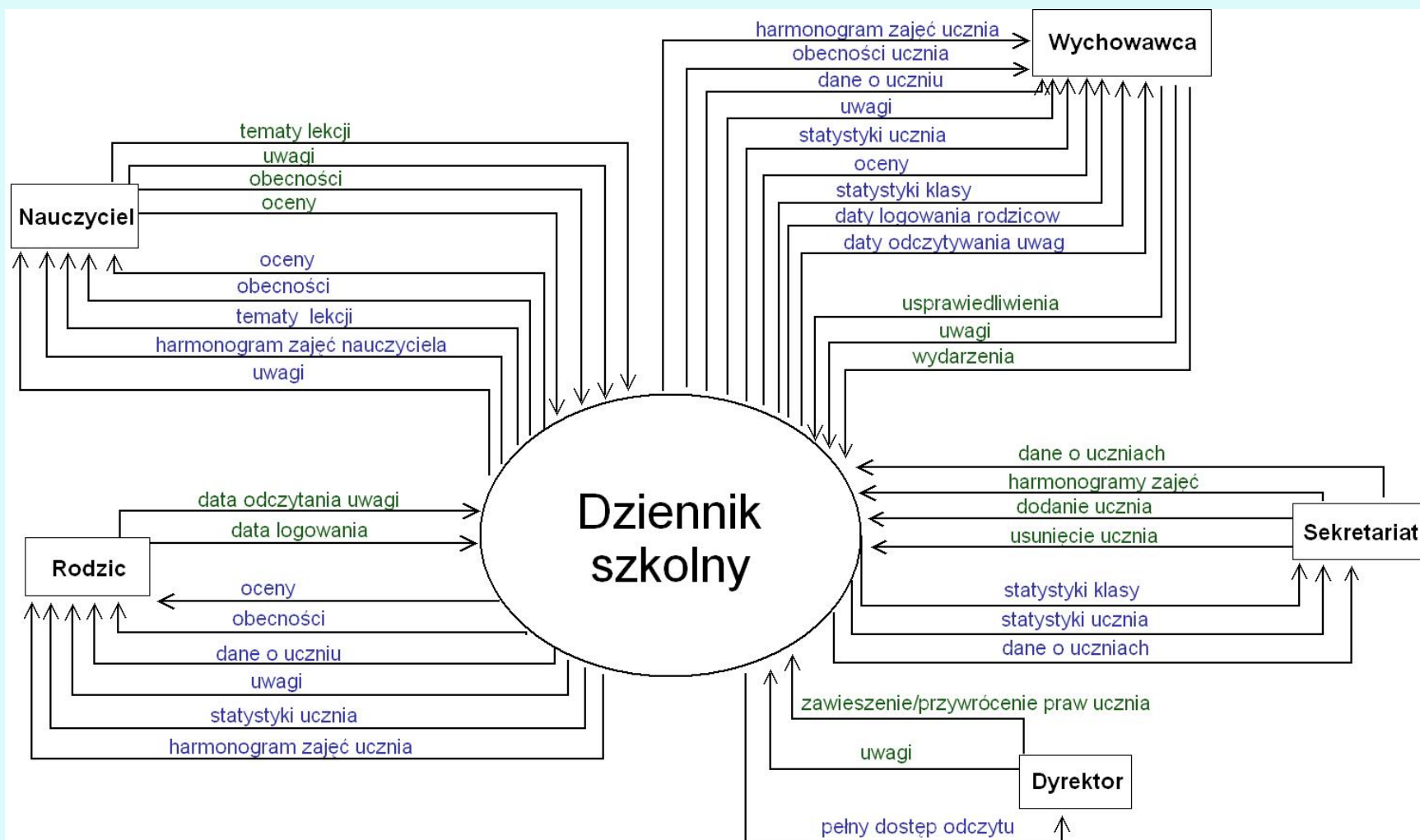
Diagram kontekstowy to diagram najwyższego poziomu, najbardziej ogólny. Przedstawia on wszystkie procesy realizowane przez analizowany system w postaci jednego głównego procesu. Diagram ten służy do zakreślenia granic naszej analizy i ustalenia właściwego kontekstu analizowanego systemu.

Procesy są opisane przy pomocy rozwijających je diagramów niższych poziomów, a procesy najniższego poziomu – w minispecyfikacjach.

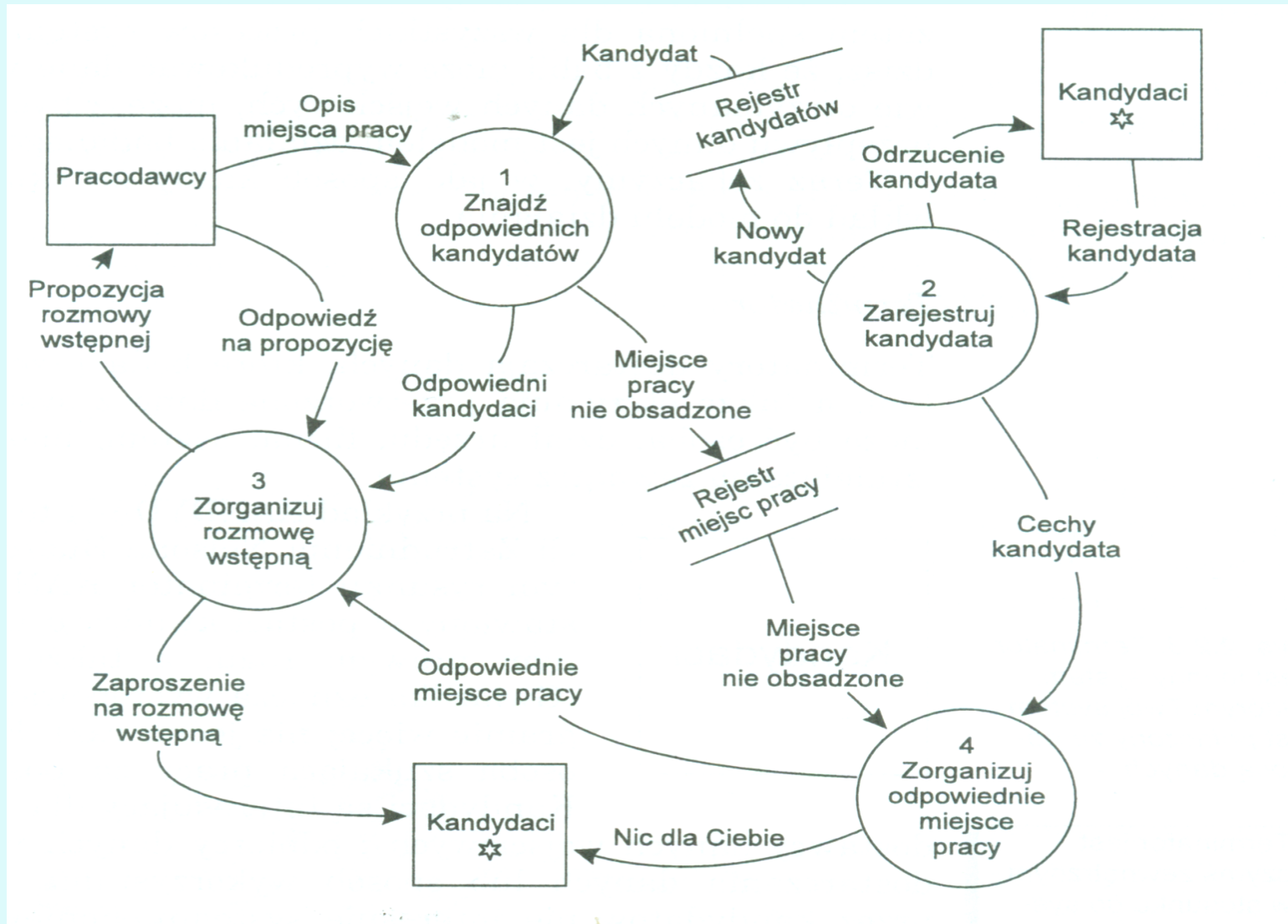
Procesy otrzymują dane z przepływów wchodzących i z magazynów, a produkują dane wyjściowe (przepływy wychodzące do innych procesów, do magazynów i do terminatorów). Procesy nie interesują się źródłami danych ani tym, dokąd wysyłają dane.

Procesy pozostają w uśpieniu do czasu otrzymania danych. Wtedy je przetwarzają, wysyłają i wchodzą w stan uśpienia.

# Przykład diagramu kontekstowego



# Przykład DPD pierwszego poziomu



## TEST CRUD

C – create, R – reference, U – update, D – delete

W pierwszym etapie sprawdza się czy każda encja i każdy związek modelu danych mają wszystkie potrzebne procesy CRUD. W etapie drugim, czy każdy atrybut jest tworzony i wykorzystywany, a jeśli potrzeba – uaktualniany lub usuwany.

Brak jakiegoś procesu CRUD przy encji, związku czy atrybucie oznacza, że pominięto zdarzenia lub że dane są redundantne.

# Diagramy UML



UML (*Unified Modeling Language*) to *Ujednolicony Język Modelowania*, język do analizy i modelowania świata obiektów w analizie obiektowej i programowaniu obiekowym. Stosuje się zarówno do opisywania istniejącej rzeczywistości jak i „rzeczywistości” dopiero projektowanej.

Jest używany także do modelowania procesów biznesowych, reprezentowania struktur organizacyjnych, modelowania zagadnień inżynierii systemów i innych.

Jest mocno związany z obiekowymi metodami tworzenia oprogramowania.

W wersji 2.0 mamy 17 rodzajów diagramów (13 – głównych i 4 – abstrakcyjne).

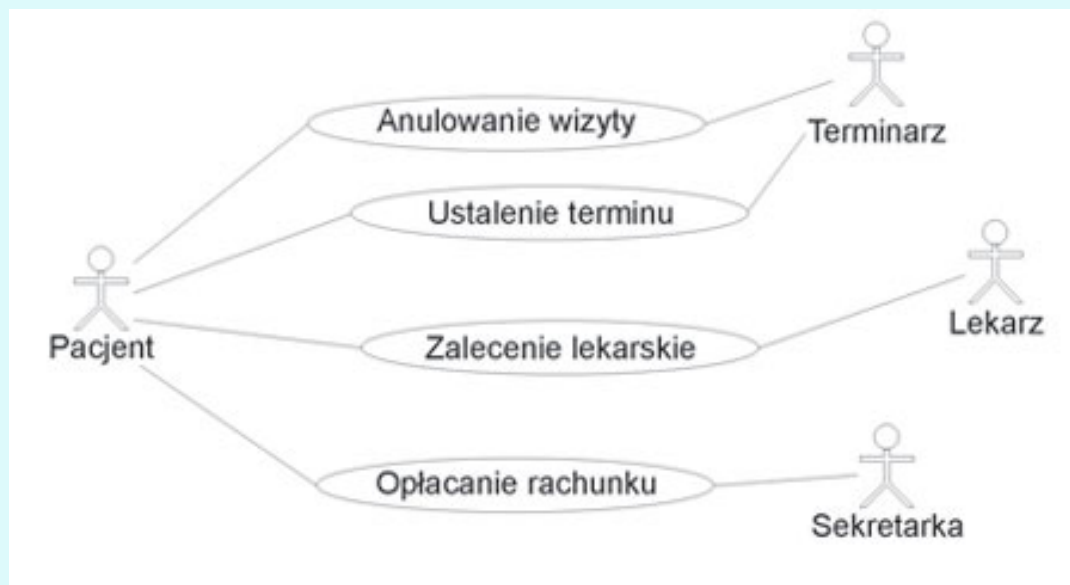
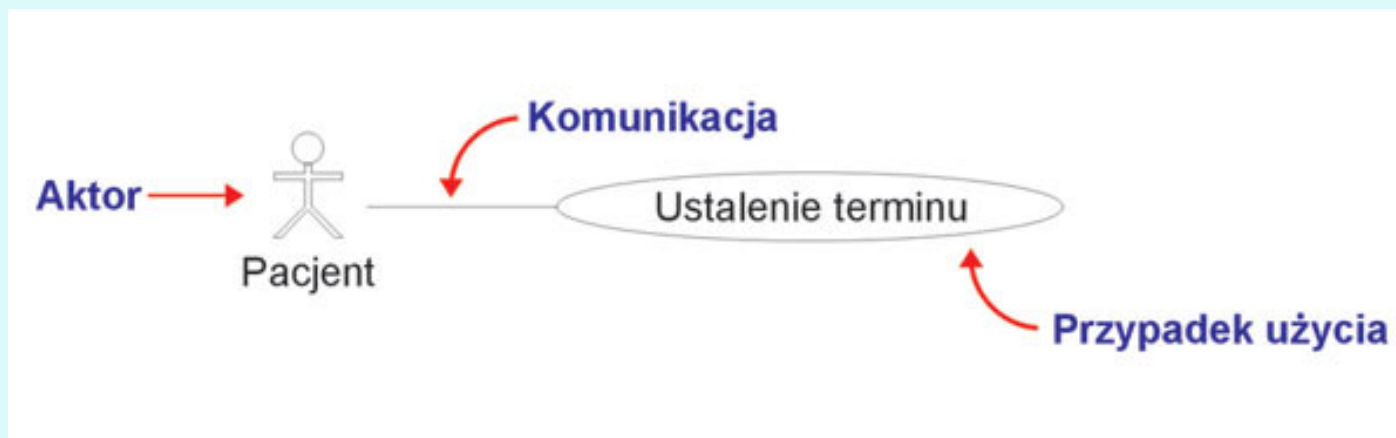
## DIAGRAMY GŁÓWNE – diagramy struktury:

- diagram klas
- diagram obiektów
- diagram pakietów
- diagram struktur połączonych
- 2 diagramy wdrożeniowe (komponentów i rozlokowania)

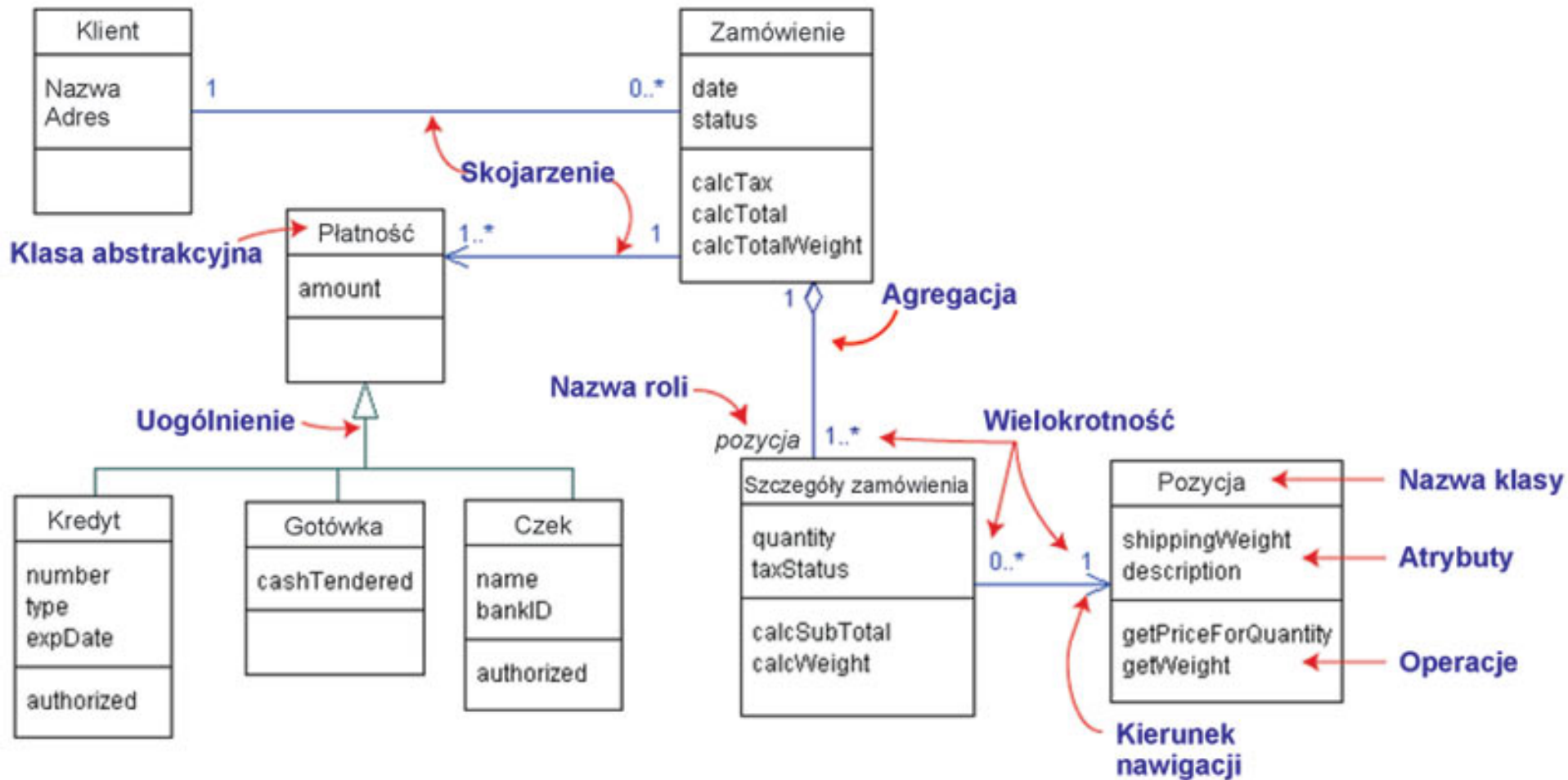
## DIAGRAMY GŁÓWNE – diagramy dynamiki:

- diagram przypadków użycia
- diagram aktywności
- diagram maszyny stanowej
- 4 diagramy interakcji (sekwencji, komunikacji, harmonogramowania i sterowania interakcją)

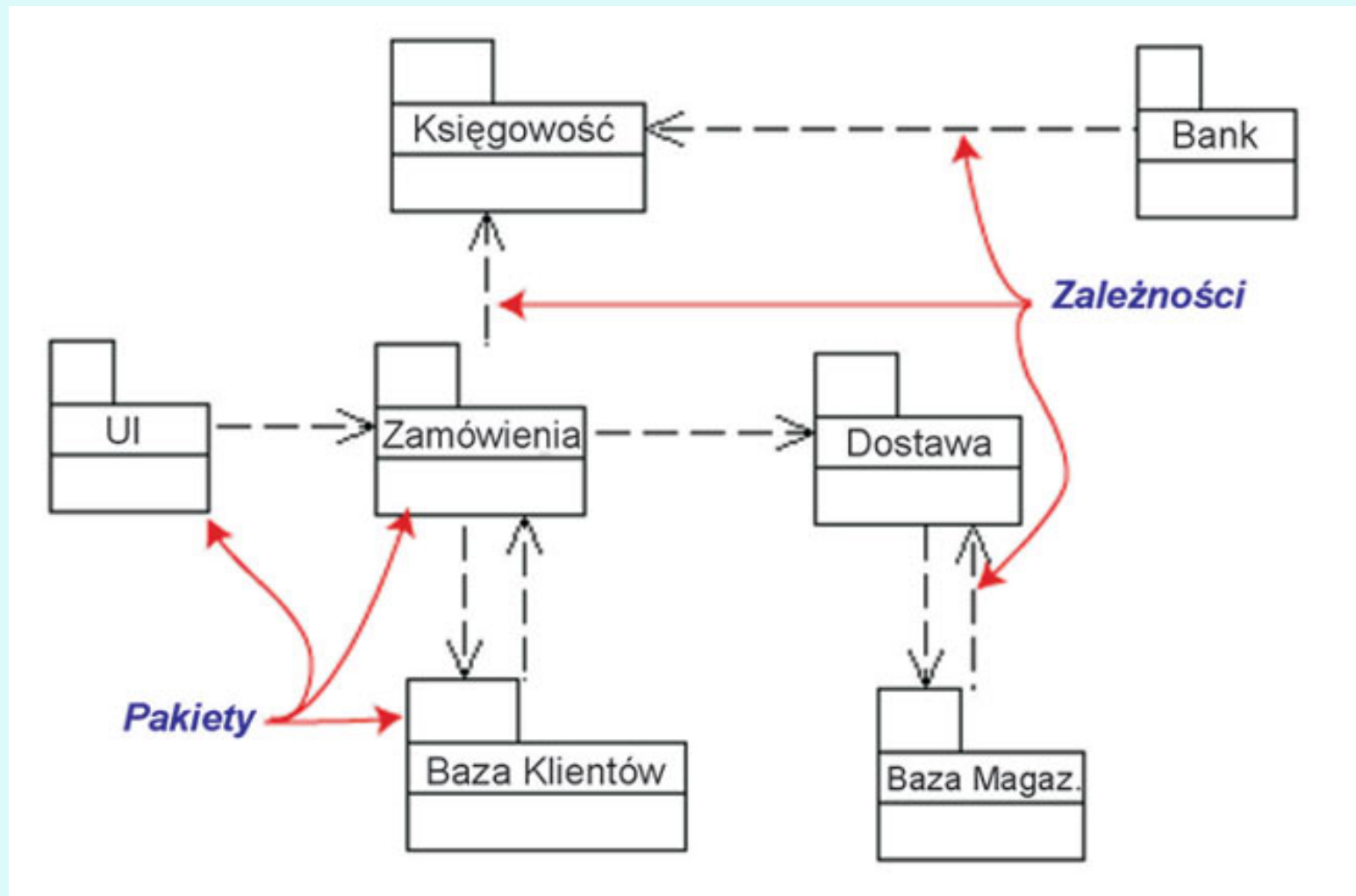
# Przykłady diagramów przypadków użycia



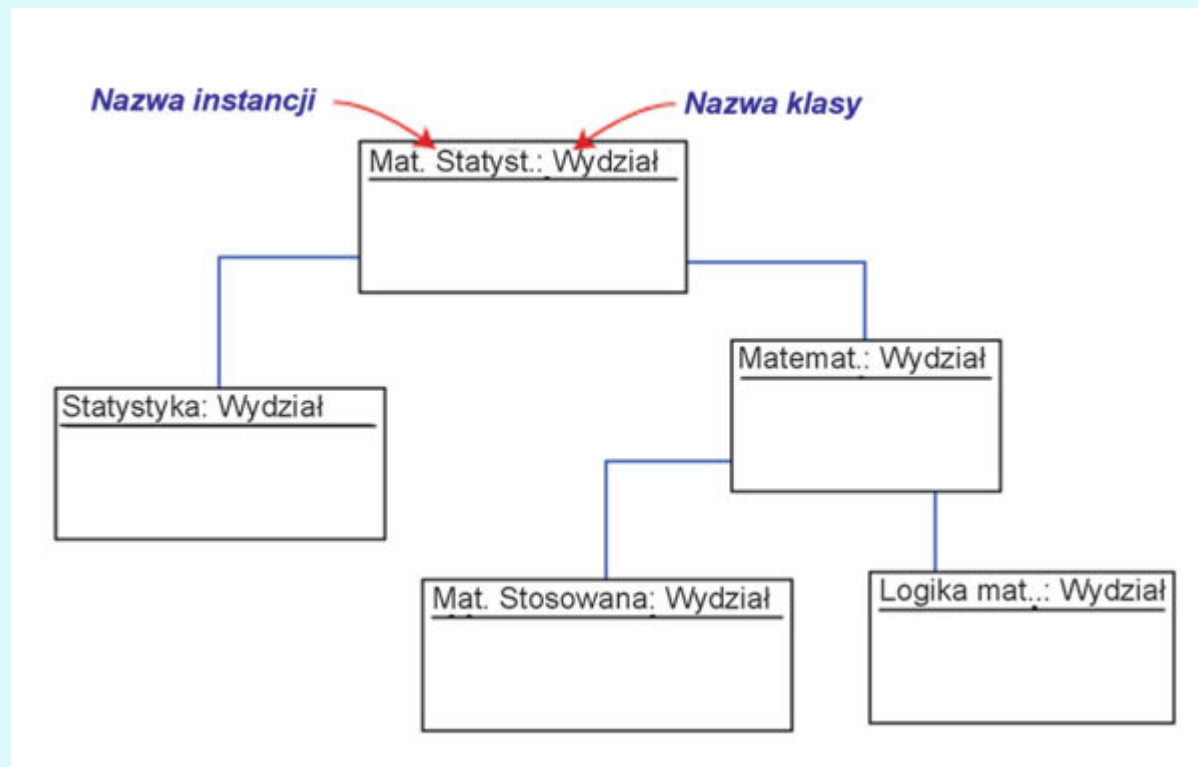
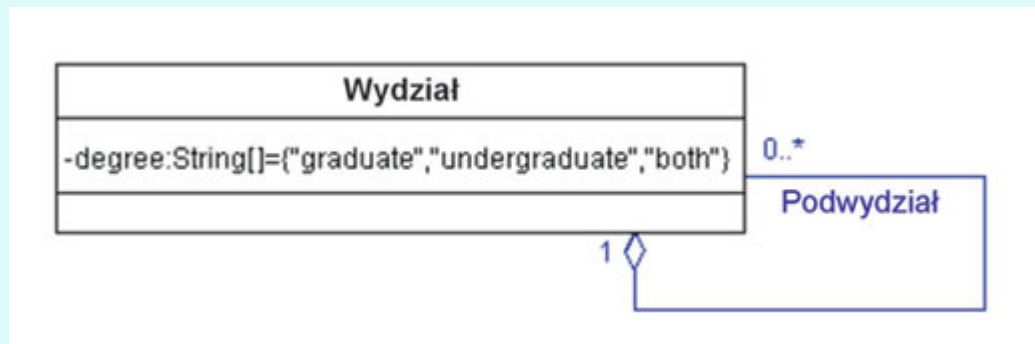
# Przykład diagramu klas



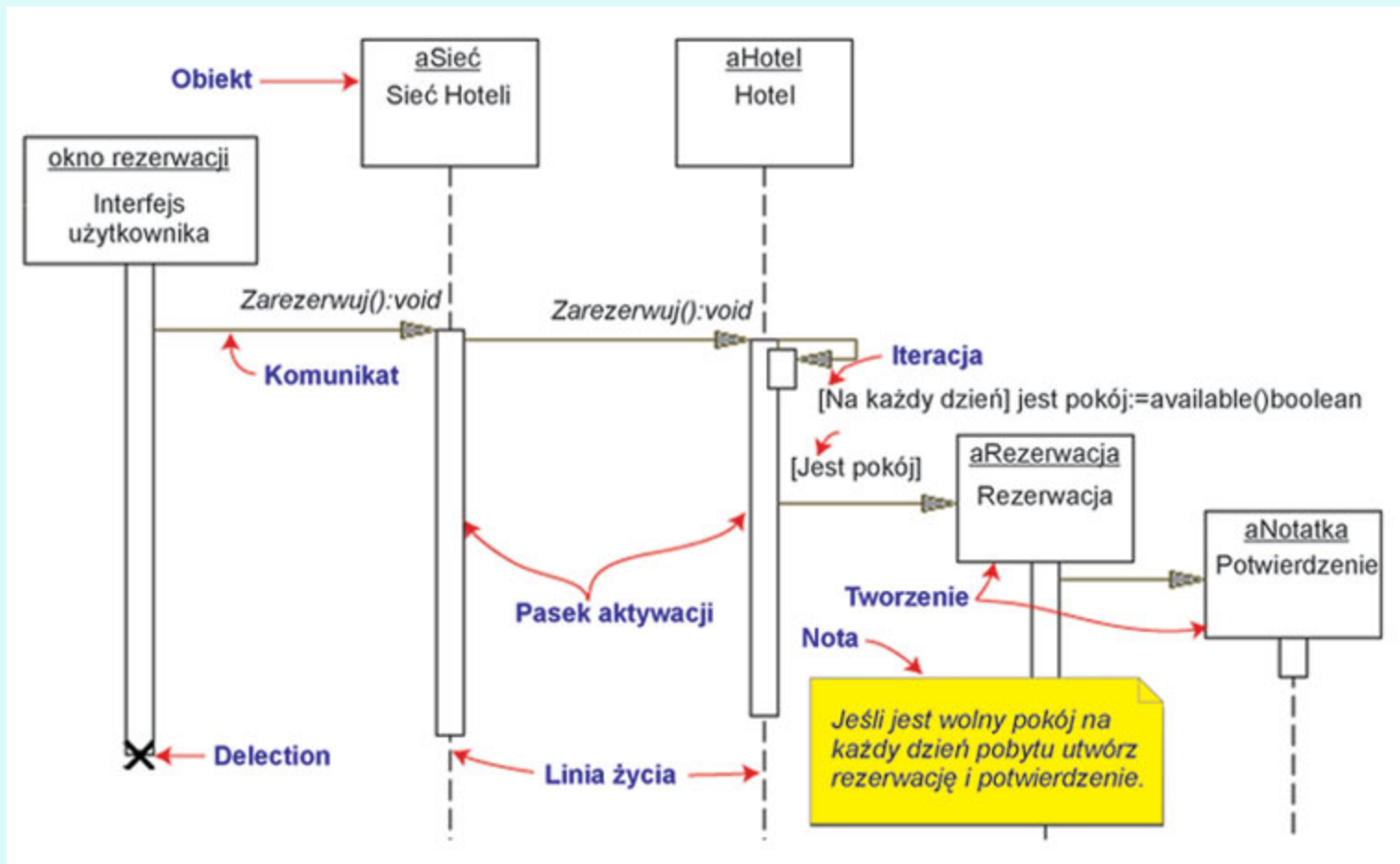
## Przykład diagramu pakietów



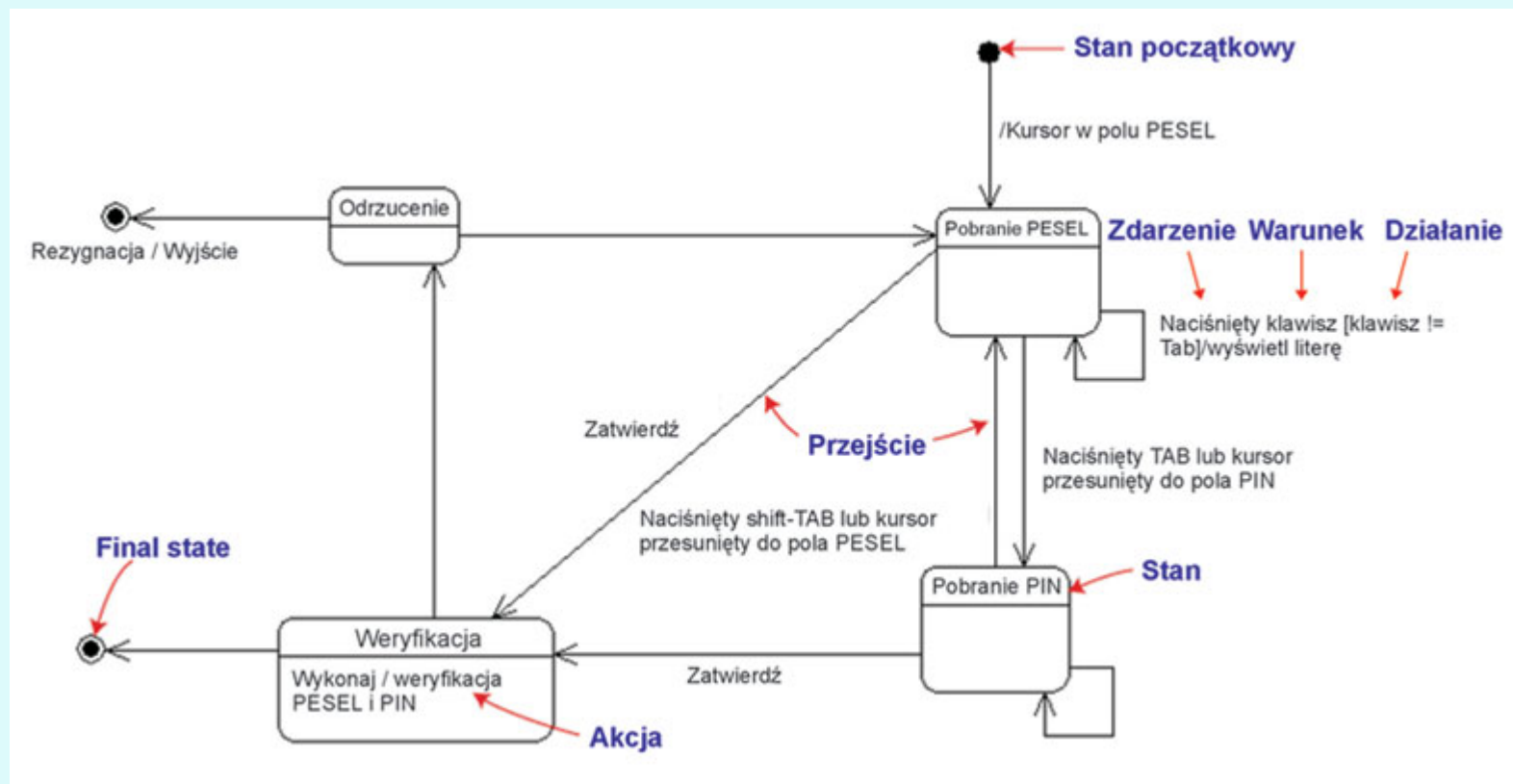
# Przykład diagramu klas i diagramu obiektów



# Przykład diagramu sekwencji

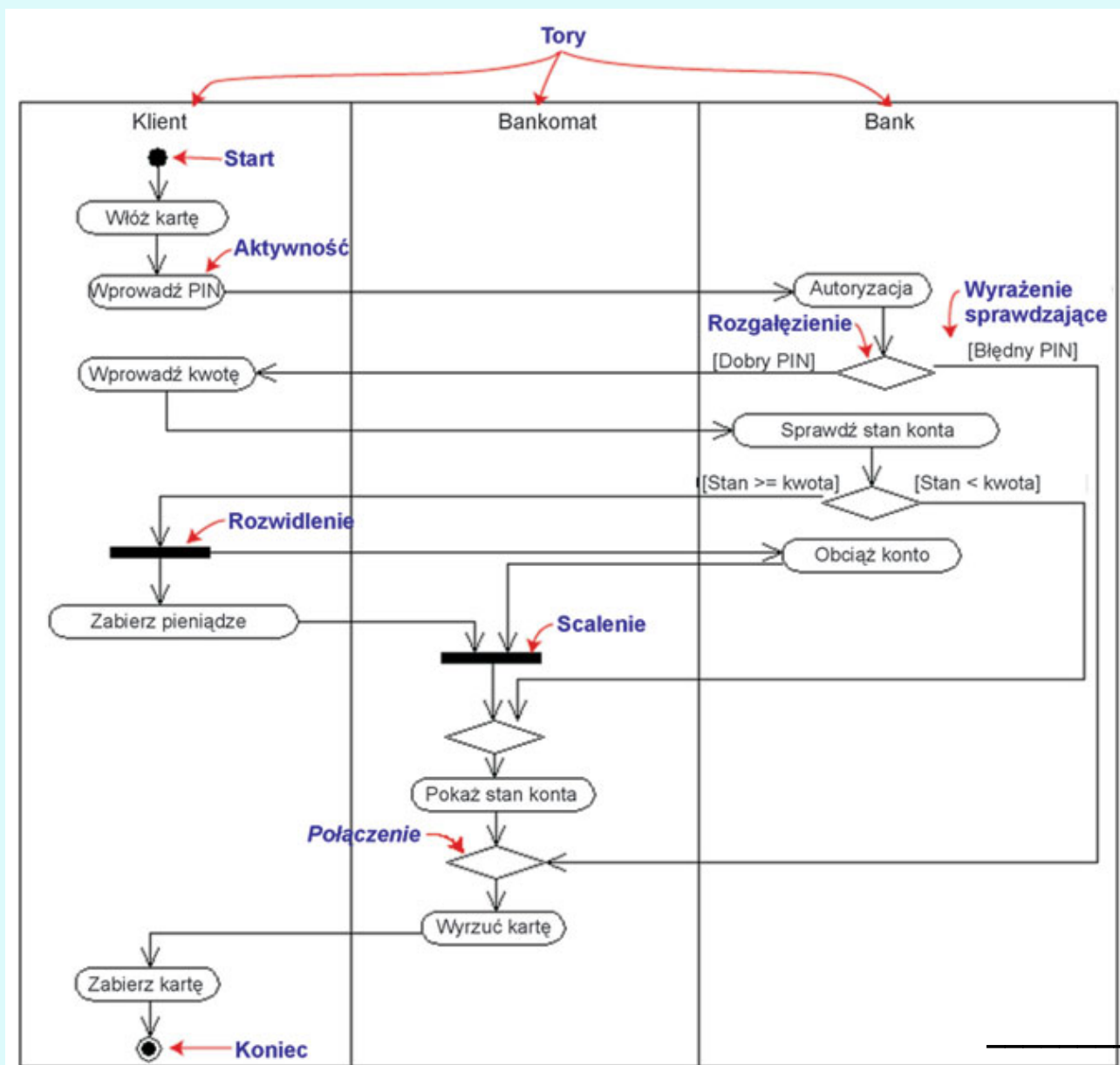


# Przykład diagramu stanów

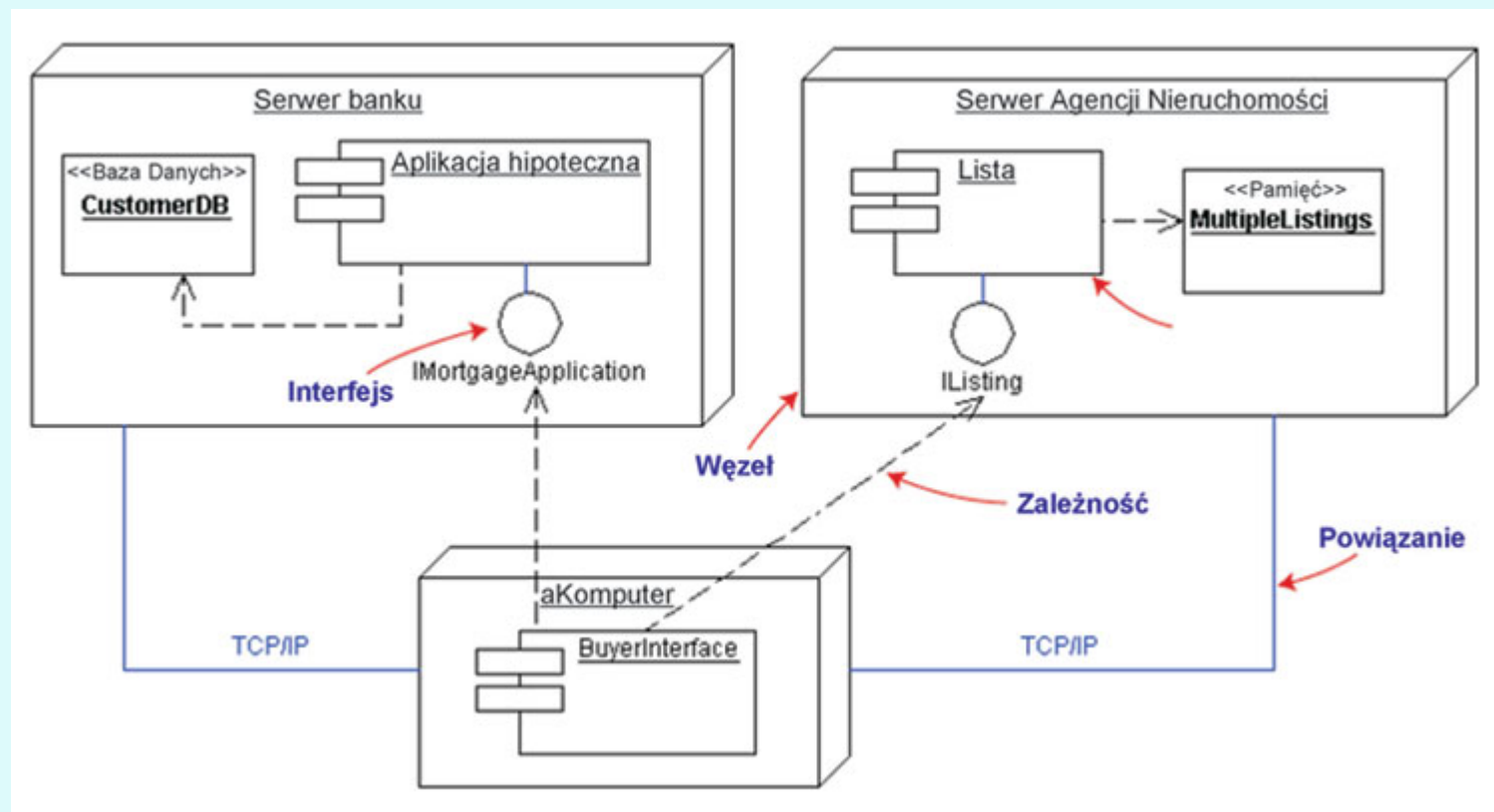




# Przykład diagramu aktywności (przepływu)



# Przykład diagramu wdrożenia



## Wybrane narzędzia do modelowania UML

- *Borland Together ControlCenter*
- *Enterprise Architect*
- *IBM Rational Rose*
- *Microsoft Visio*
- *Sybase PowerDesigner*
- *Visual Paradigm for UML*
- *StartUML*
- *Dia*
- *MagicDraw*