

Inżynieria oprogramowania

Projektowanie systemów informatycznych cz. 2



Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
www.mimuw.edu.pl/~dabrowski

Abstrakcja i dekompozycja

■ Przypomnienie

- „Managing complexity in software development”
 - czyli
- Jak radzić sobie z dużymi / złożonymi systemami?

■ Podejście

- Abstrakcja
 - uprość i uogólnij
- Dekompozycja
 - podziel i zwycięż



Abstrakcja

- Wnioskowanie na temat problemu
 - upraszcza problem
 - nie rozwiązuje problemu
- Dobra abstrakcja
 - ukrywa/ignoruje niepotrzebne szczegóły
 - upraszcza analizę
 - znajduj analogie pomiędzy różnymi bytami

Dekompozycja

- Rozwiązywanie dużych problemów
 - metoda **dziel i zwyciężaj**
- Dobra dekompozycja
 - każdy podproblem jest podobnej wielkości
 - podproblemy można rozwiązywać niezależnie
 - z rozwiązań podproblemów można uzyskać rozwiązanie całości

Abstrakcja i dekompozycja: zalety i wady

■ Zalety

- można przypisać podzadania różnym osobom
- podzadania można rozwiązywać równolegle
- łatwo utrzymywać system w przyszłości
- można ukryć nieistotne szczegóły

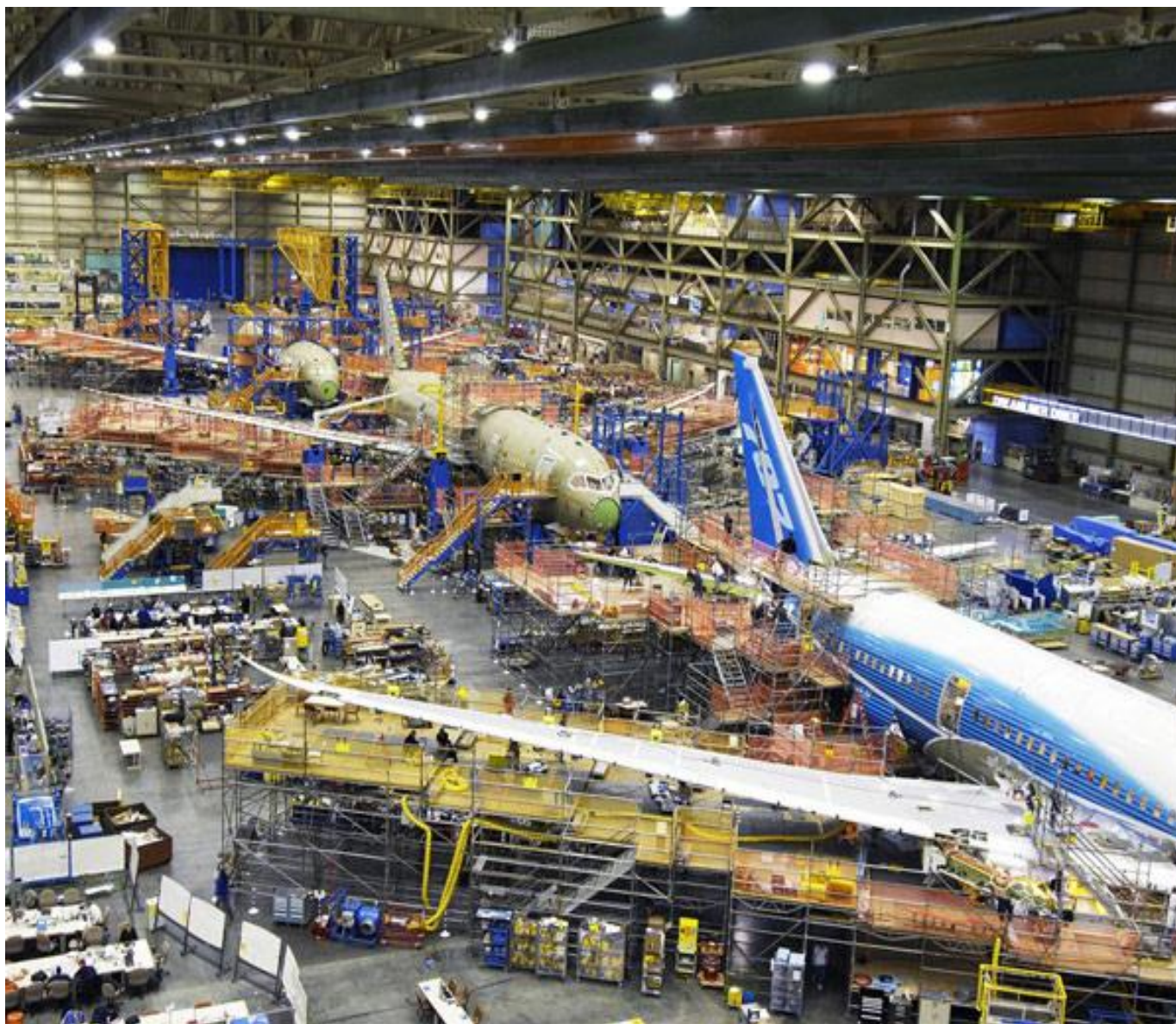
■ Wady

- scalanie podrozwiązań może stanowić problem
- problem źle zrozumiany nie da się dobrze zdekomponować
- ukrywanie szczegółów nie rozwiązuje problemu

Podjęcie obiektowe

- Wspiera ideę abstrakcji i dekompozycji
- Zaleta modeli wizualnych
 - Języki programowania nie są na wystarczająco wysokim poziomie abstrakcji, by dało się w nich projektować duże systemy
- UML (Unified Modelling Language)
 - Notacja graficzna
 - Oparta na pojedynczym metamodelu
 - Głównie zalecana przy projektowaniu obiektowym
- Historia UML
 - pojawił się 1997
 - ujednolicenie wielu notacji graficznych
 - lata 1980' i wczesne lata 1990'
 - standard (dość) otwarty
 - kontrolowany przez Object Management Group (OMG)
 - otwarte konsorcjum firm
 - utrzymuje między innymi standardy CORBA
 - Common Object Request Broker Architecture
- Mity na temat UML
 - „No Silver Bullet”
 - „Death by UML Fever”

Podjęcie obiektywne: świat rzeczywisty



Podjęcie obiektowe: abstrakcja i dekompozycja

dziedzina problemu



Plane

tailNumber

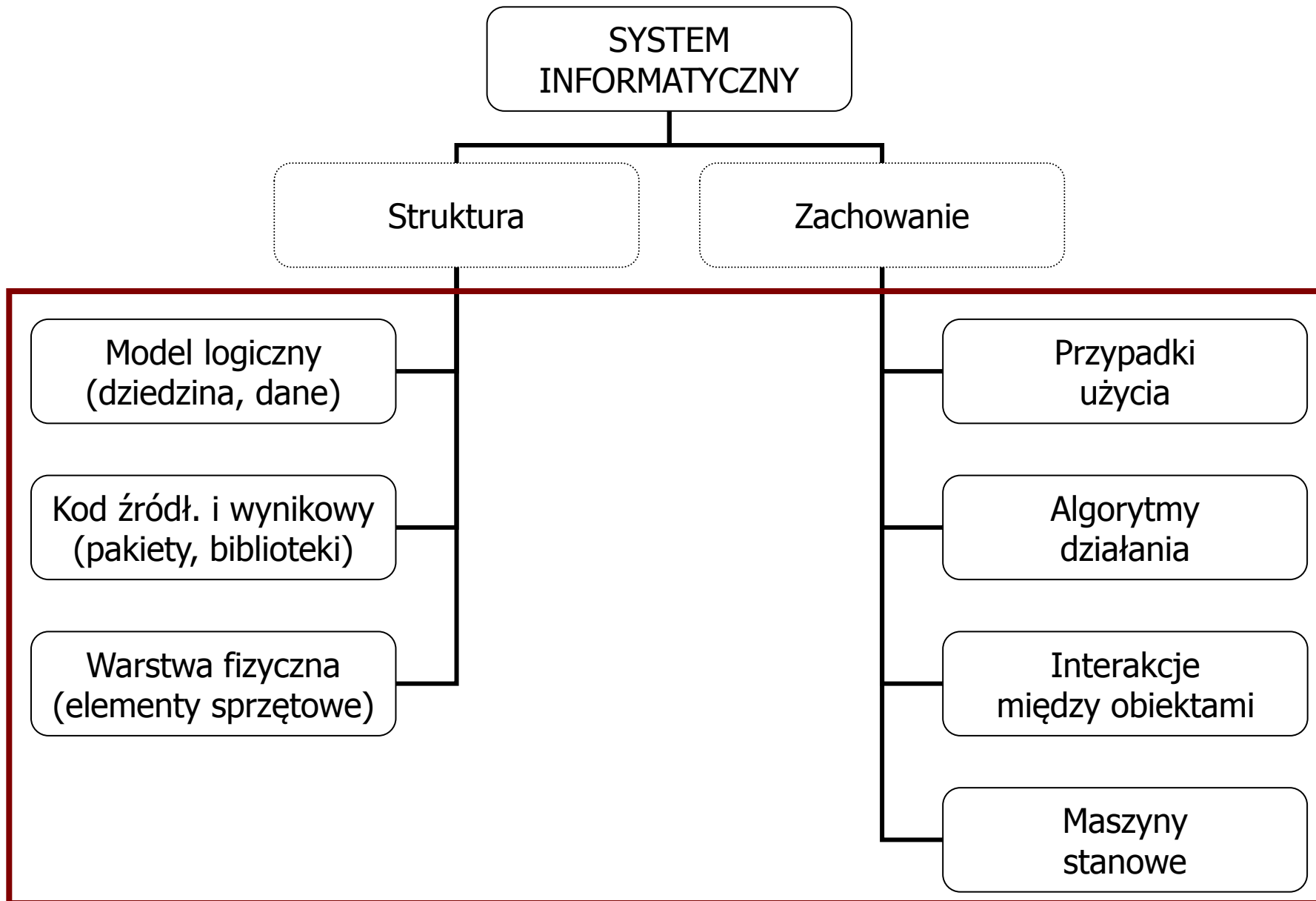
identyfikacja
pojęć z dziedziny
problemu

```
public class Plane
{
    private String tailNumber;

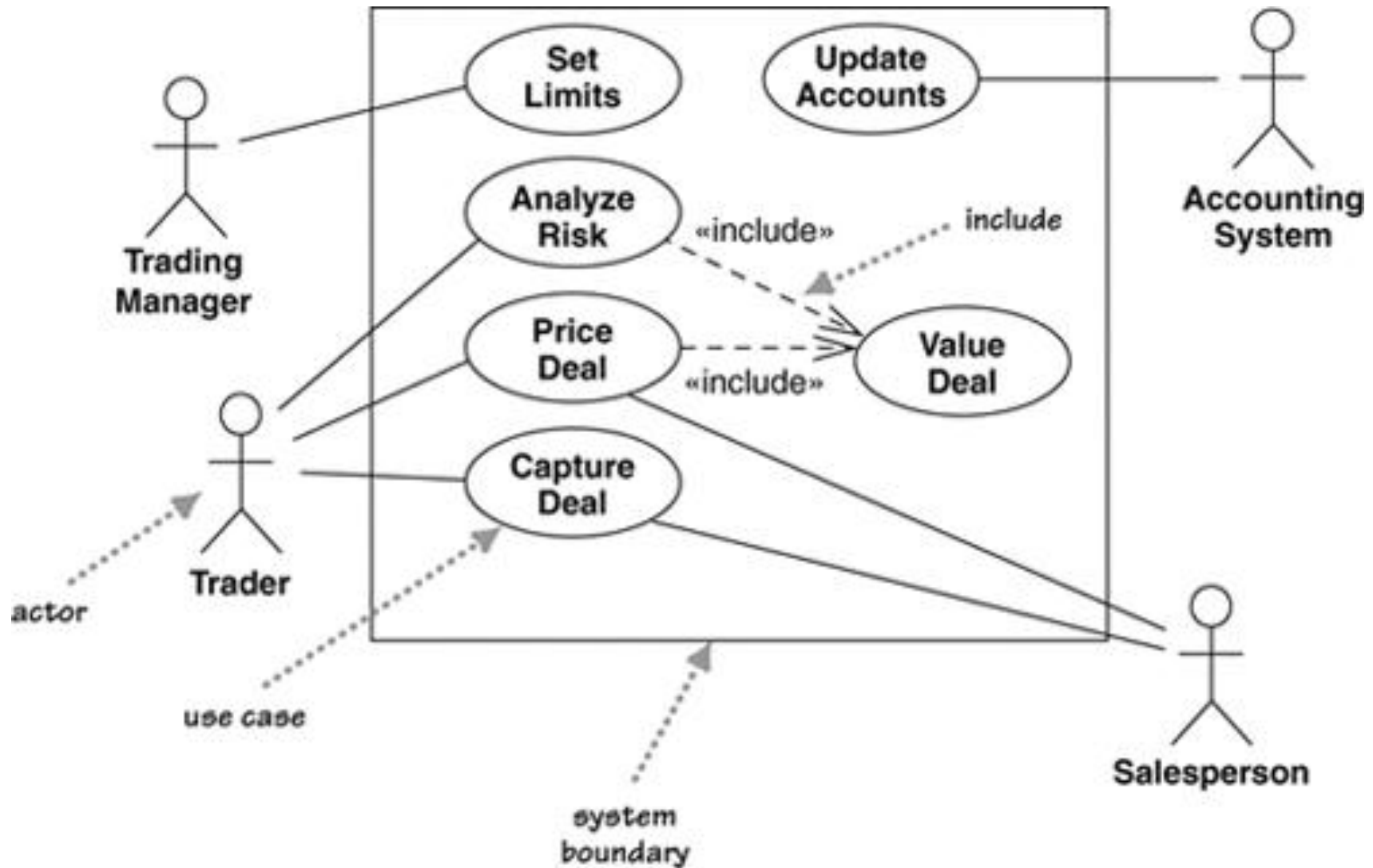
    public List getFlightHistory() {...}
}
```

reprezentacja
pojęć w
języku programowania

Jeden SYSTEM – różne PERSPEKTYWY



Przypadki użycia



Przypadki użycia

Buy a Product

Main Success Scenario:

1. Customer browses catalog and selects items to buy
2. Customer goes to check out
3. Customer fills in shipping information (address; next-day or 3-day delivery)
4. System presents full pricing information, including shipping
5. Customer fills in credit card information
6. System authorizes purchase
7. System confirms sale immediately
8. System sends confirming e-mail to customer

Extensions:

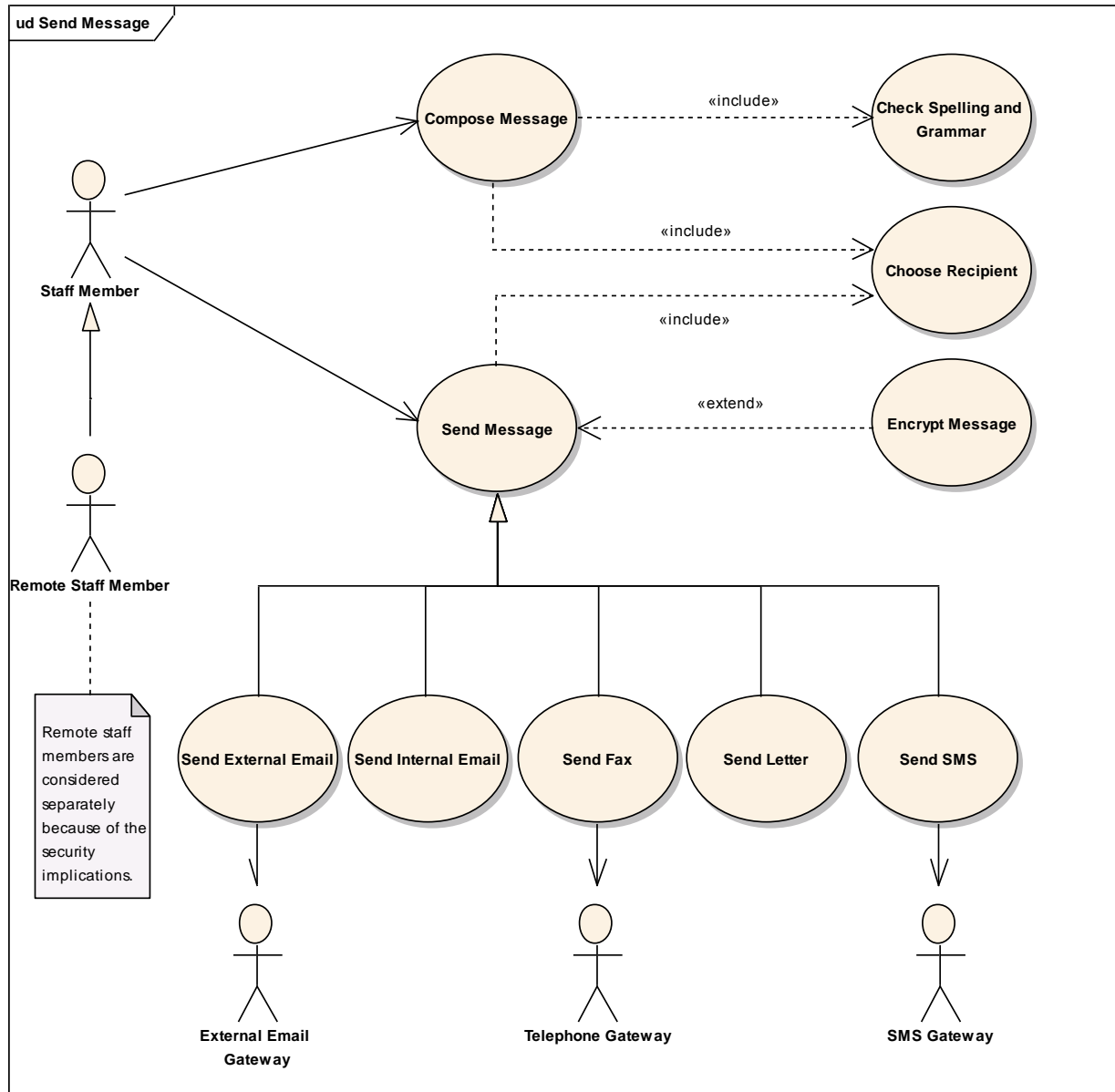
3a: Customer is regular customer

- .1: System displays current shipping, pricing, and billing information
- .2: Customer may accept or override these defaults, returns to MSS at step 6

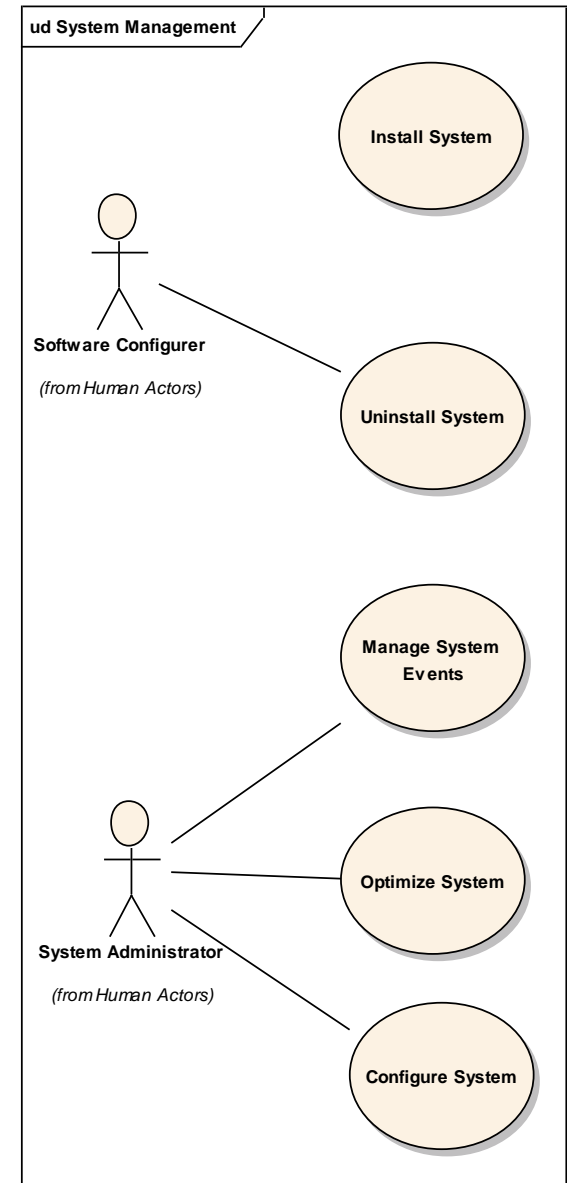
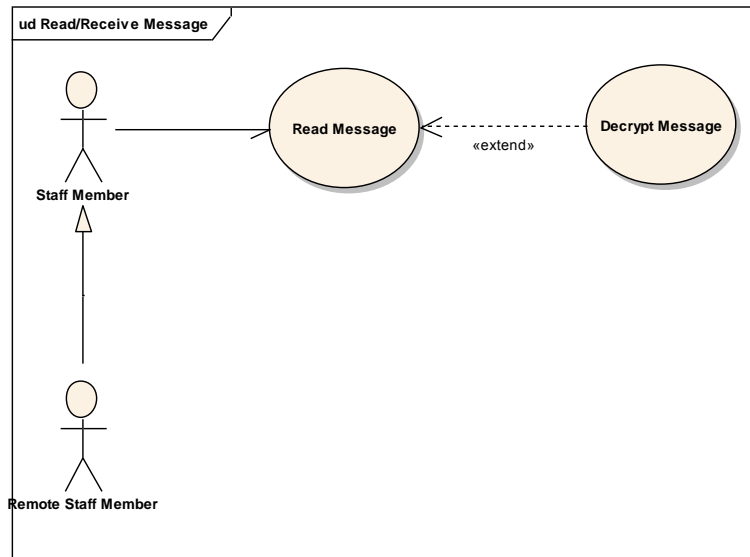
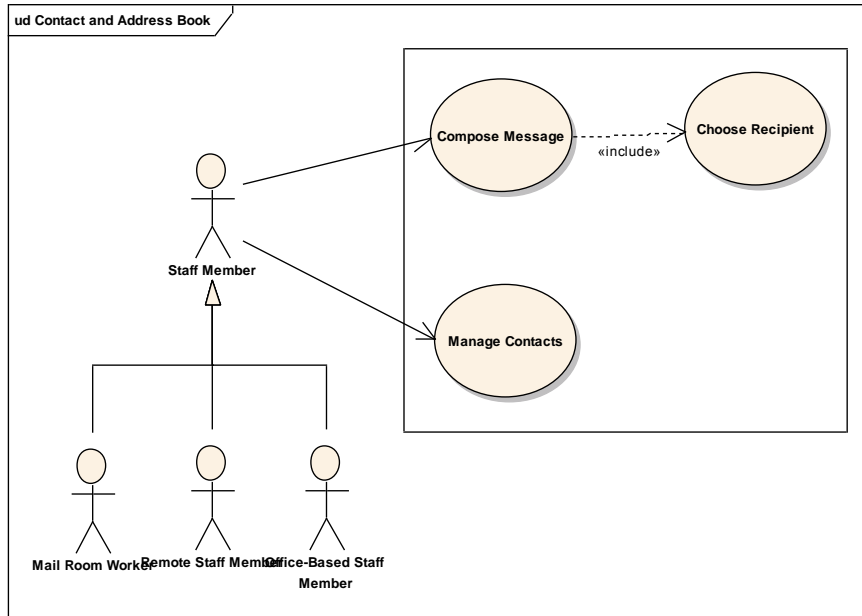
6a: System fails to authorize credit purchase

- .1: Customer may reenter credit card information or may cancel

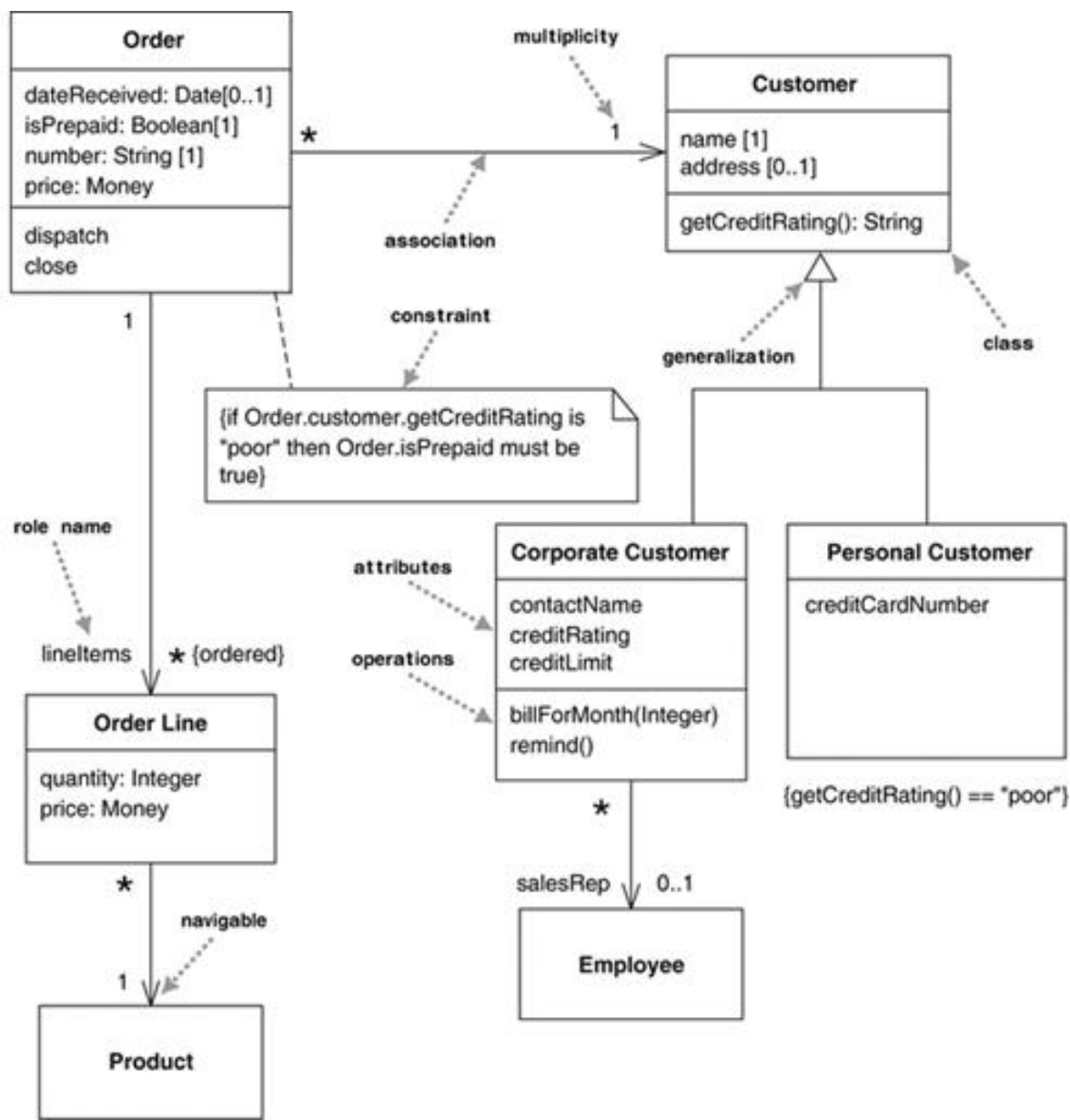
Przypadki użycia



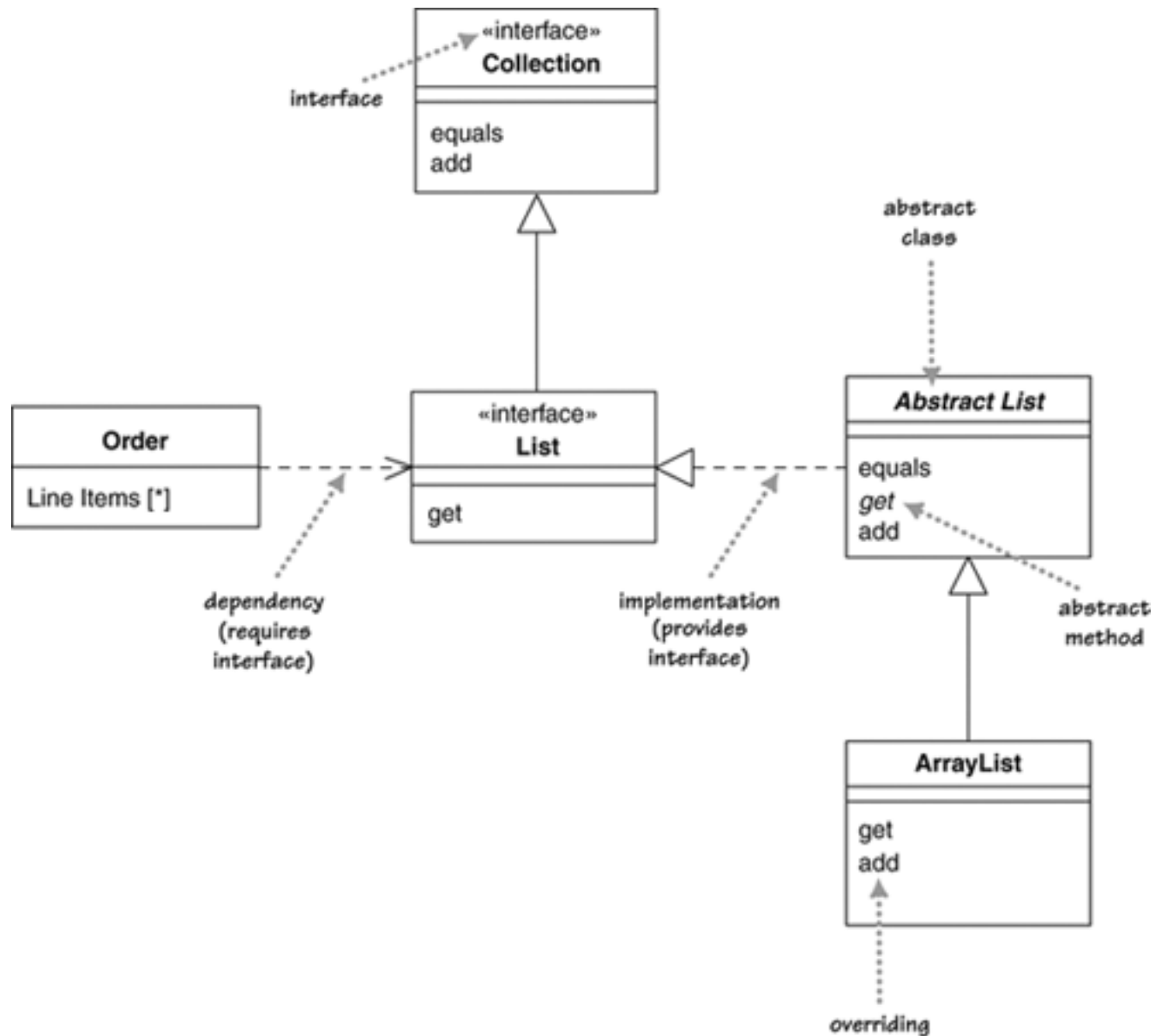
Przypadki użycia



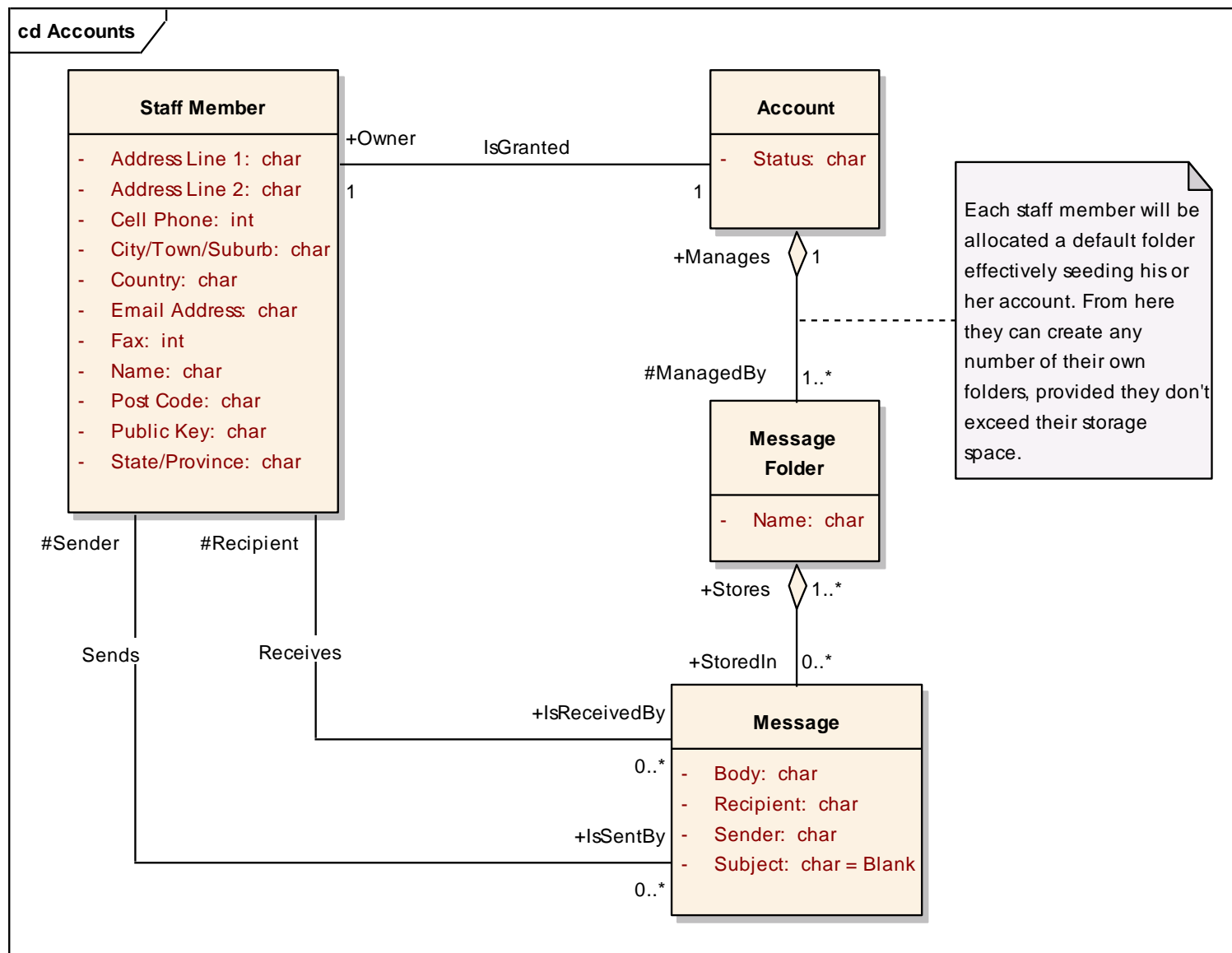
Dziedzina / klasy / interfejsy / dane



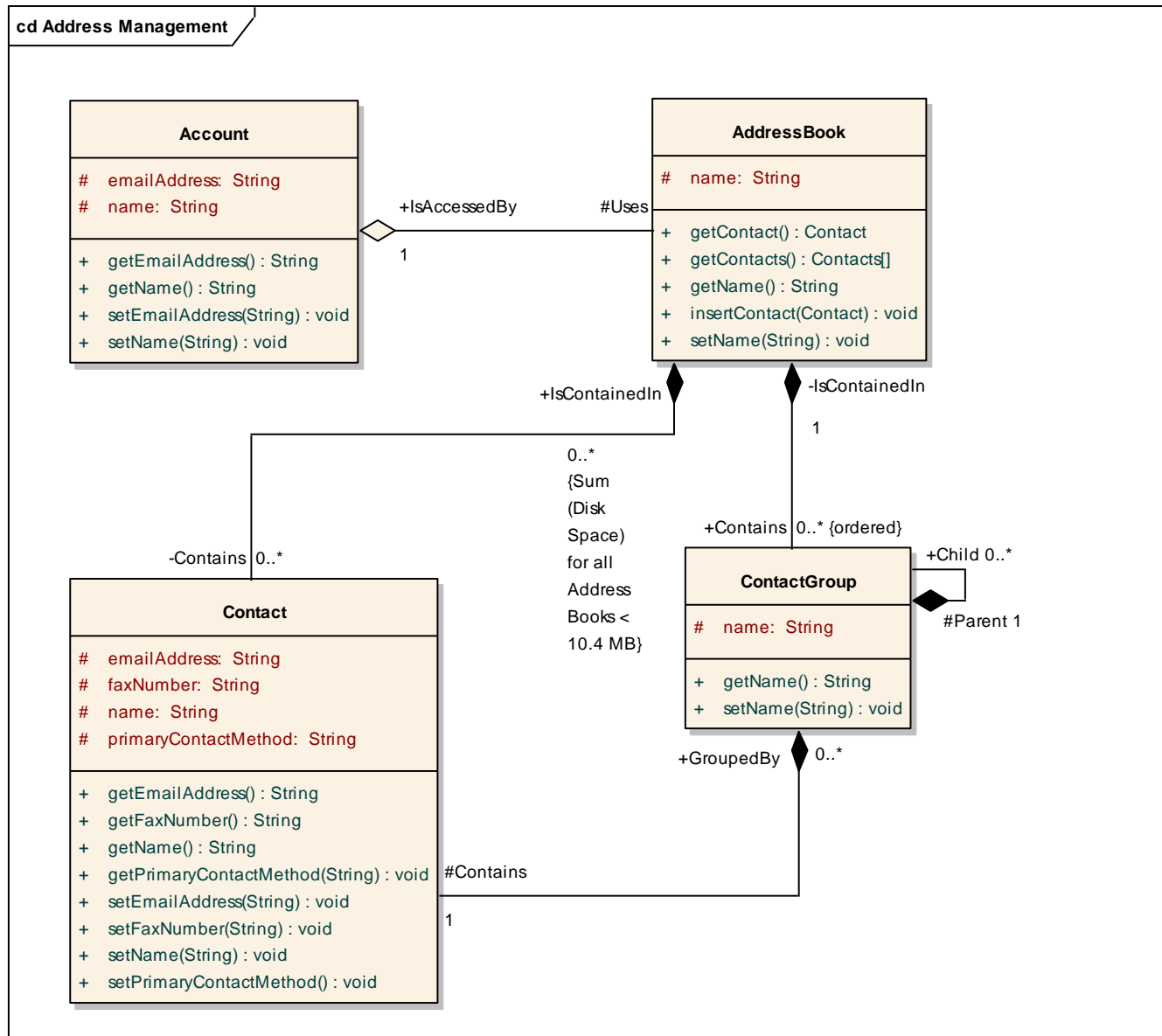
Dziedzina / klasy / interfejsy / dane



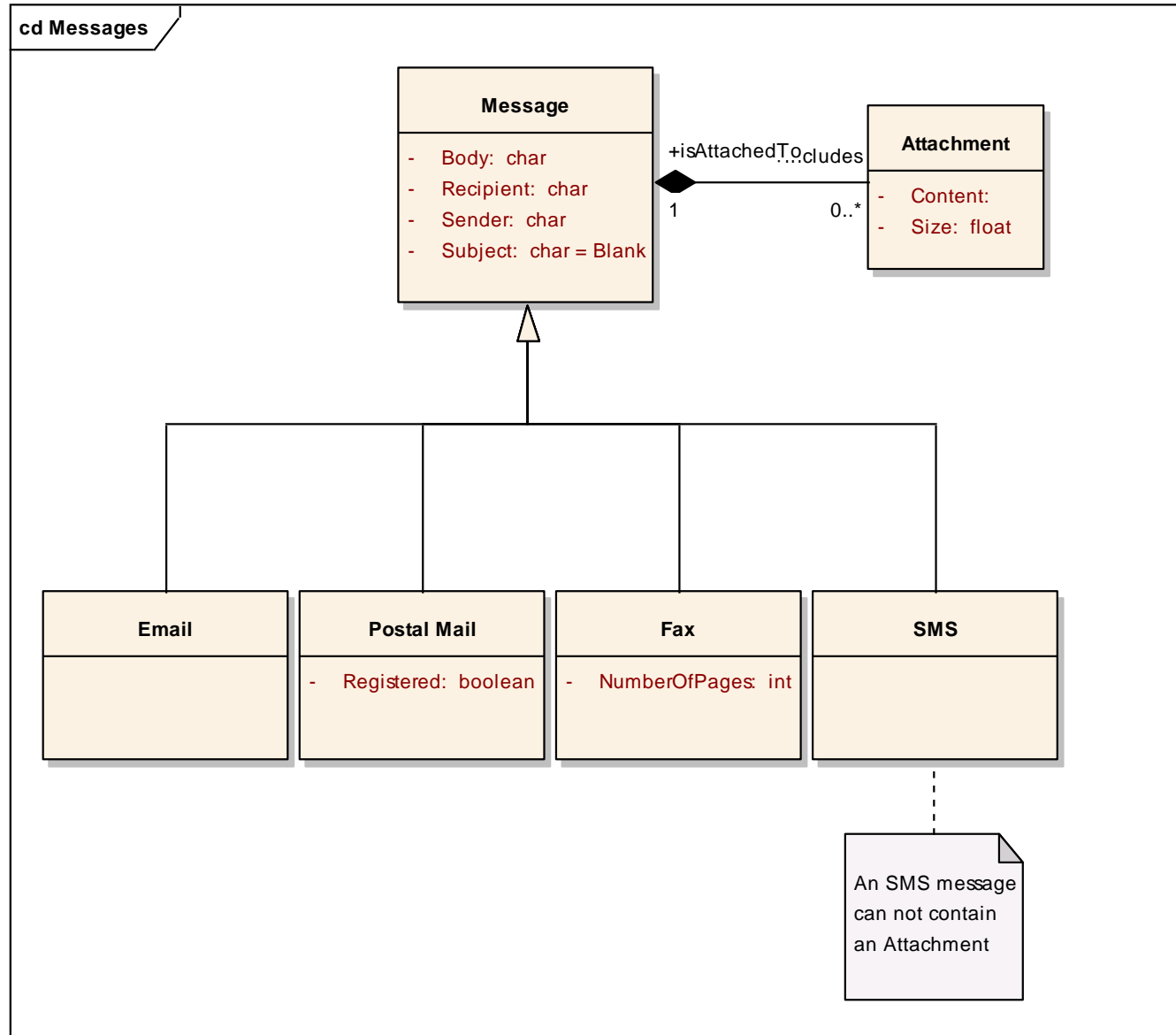
Dziedzina / klasy / interfejsy / dane



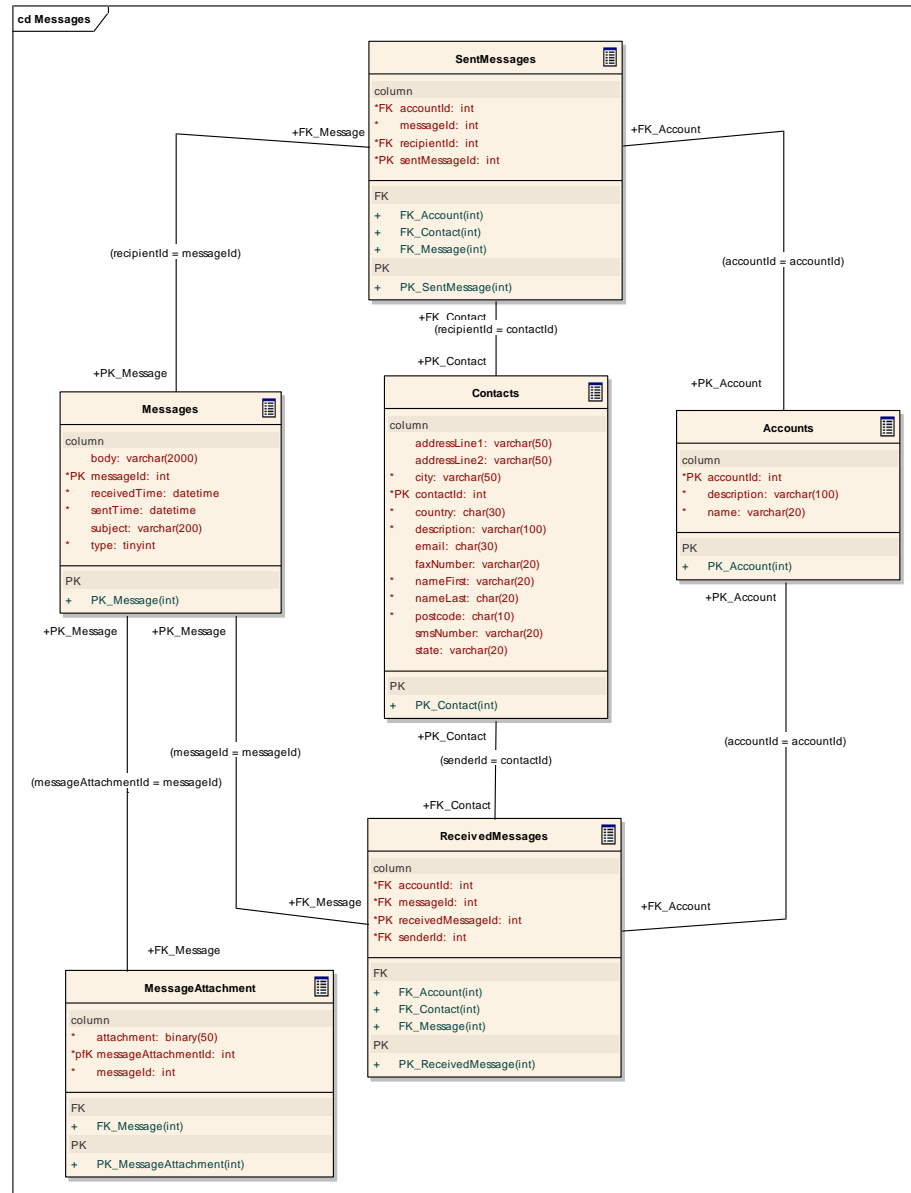
Dziedzina / klasy / interfejsy / dane



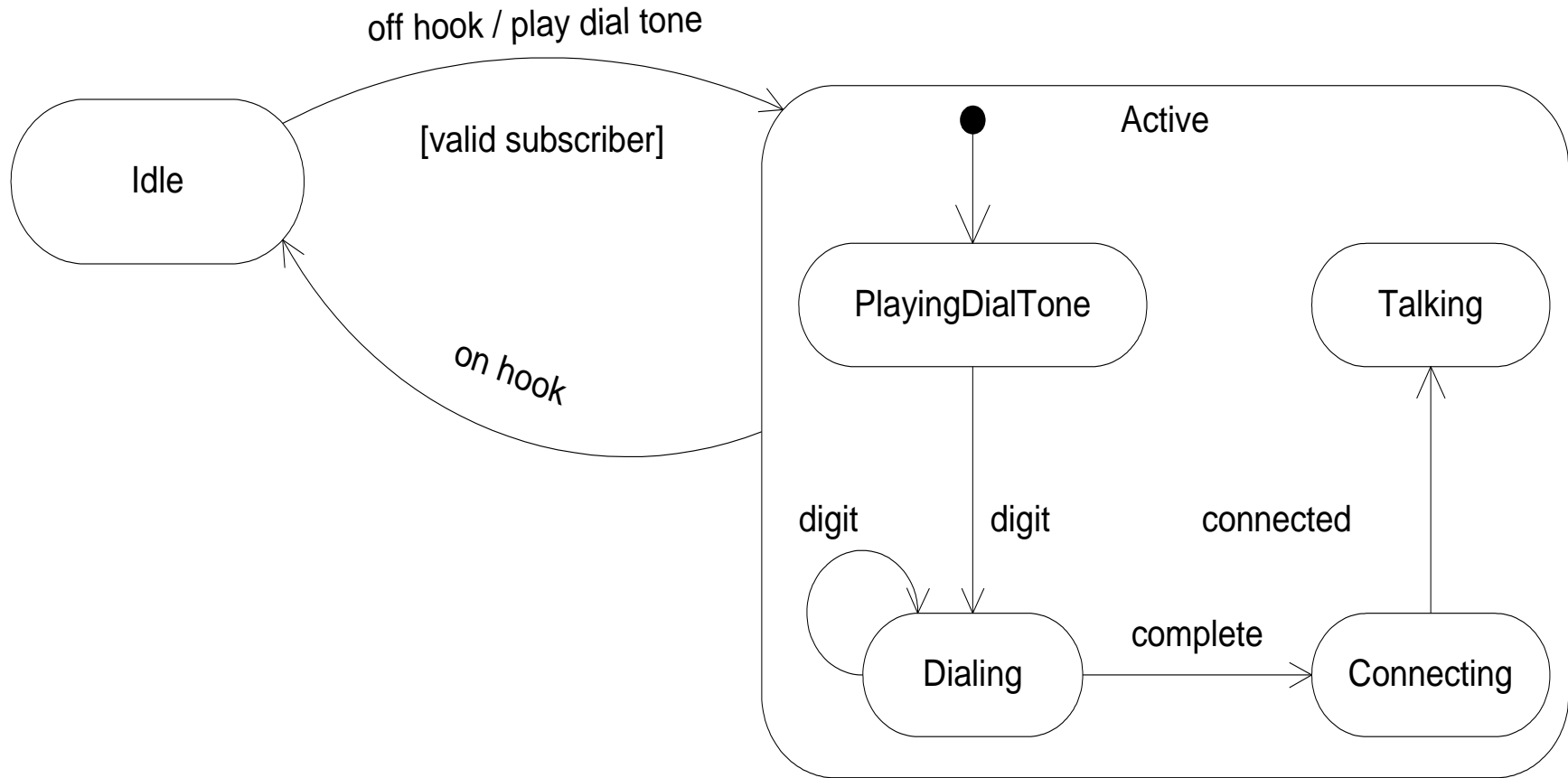
Dziedzina / klasy / interfejsy / dane



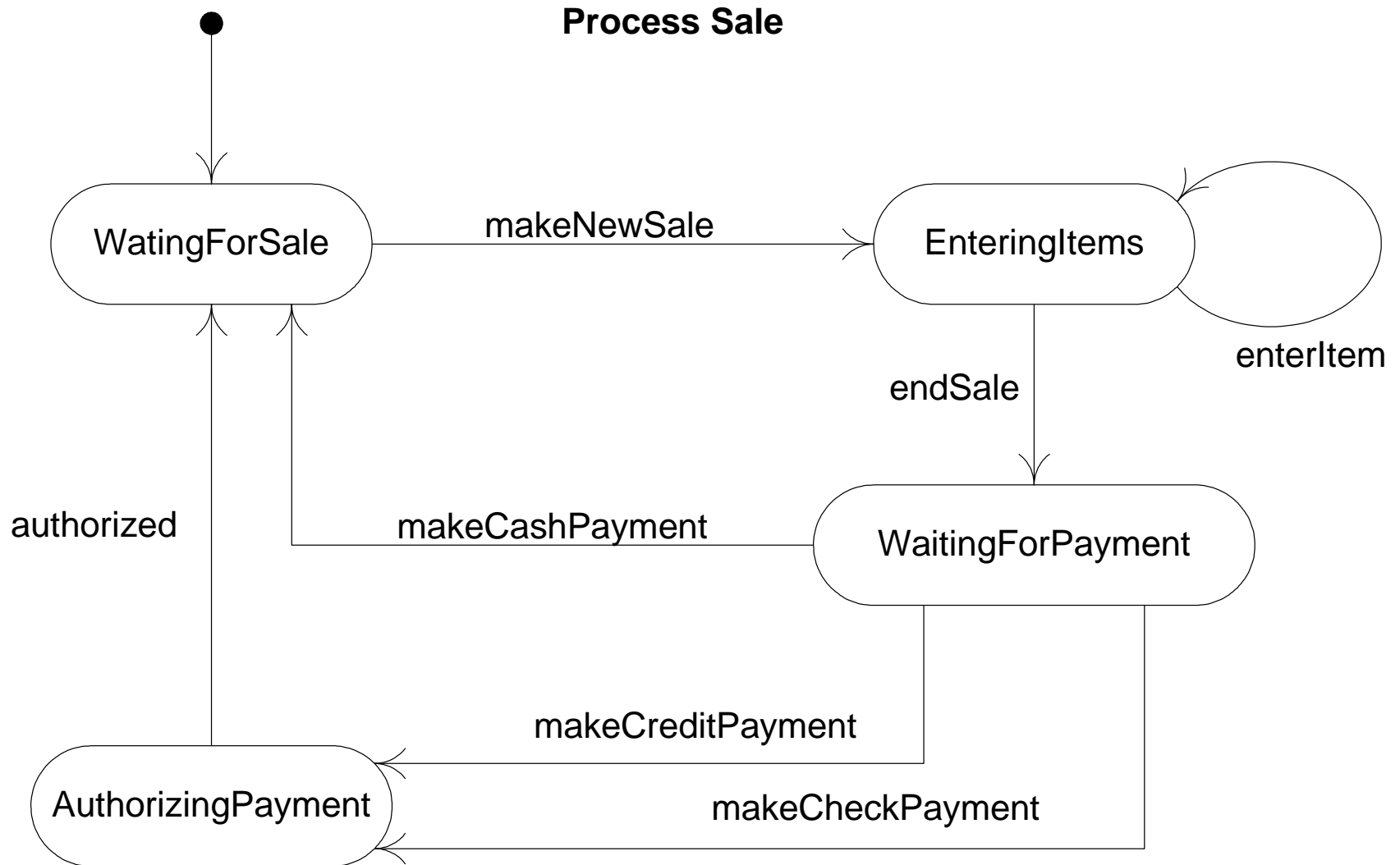
Dziedzina / klasy / interfejsy / dane



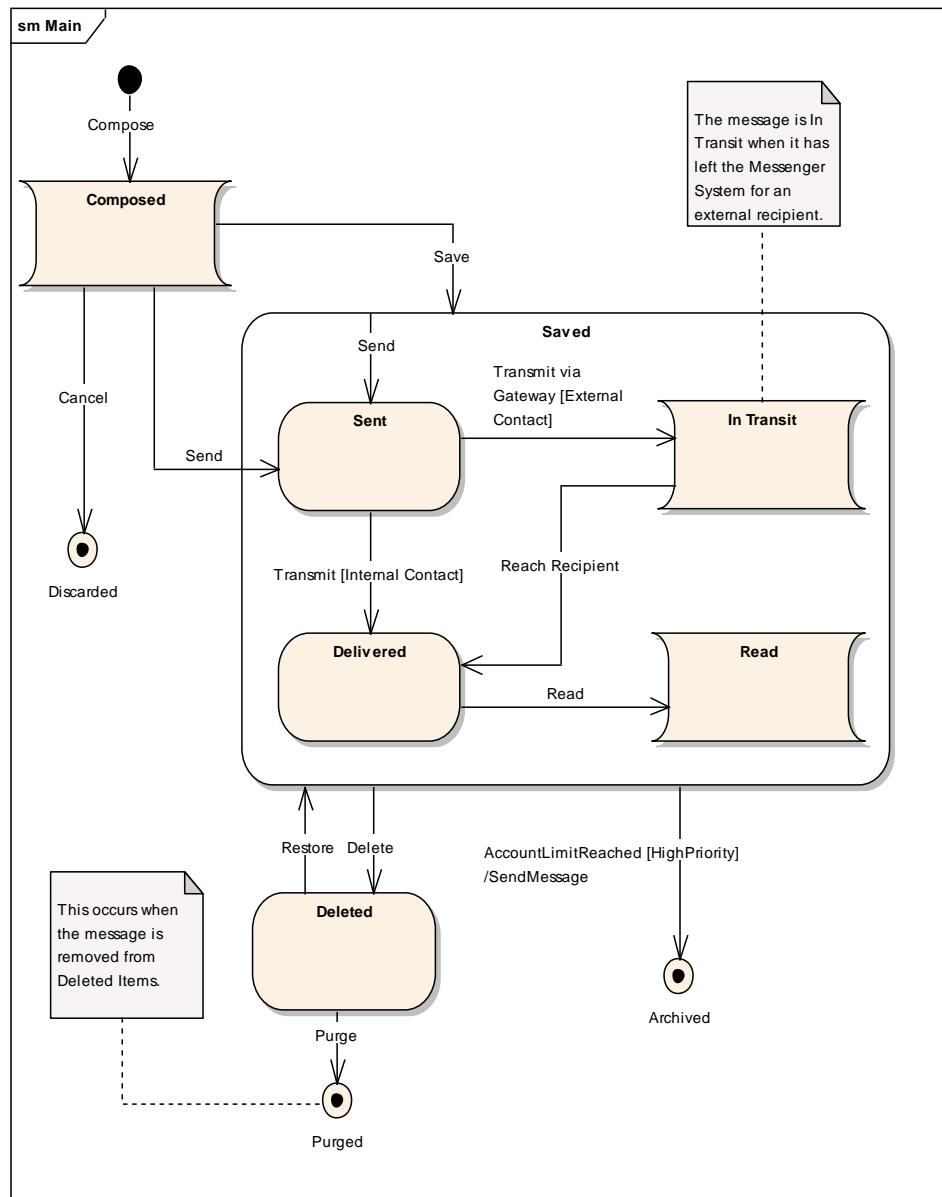
Maszyny stanowe



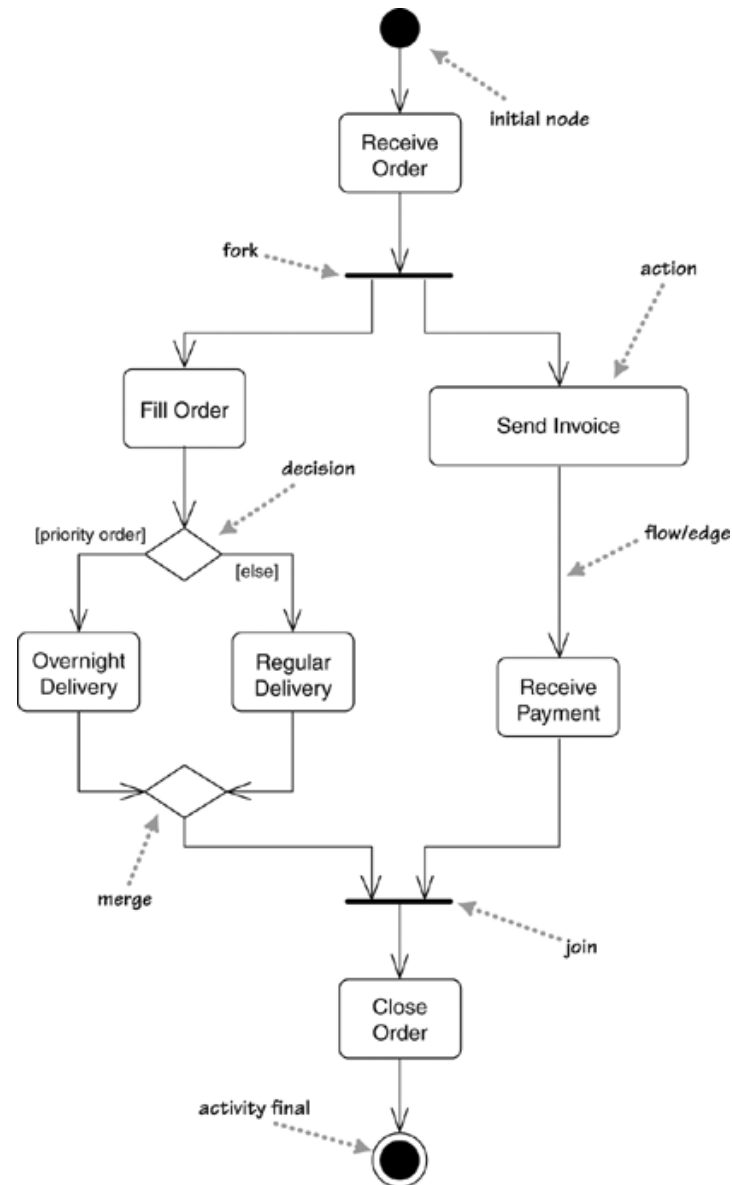
Maszyny stanowe



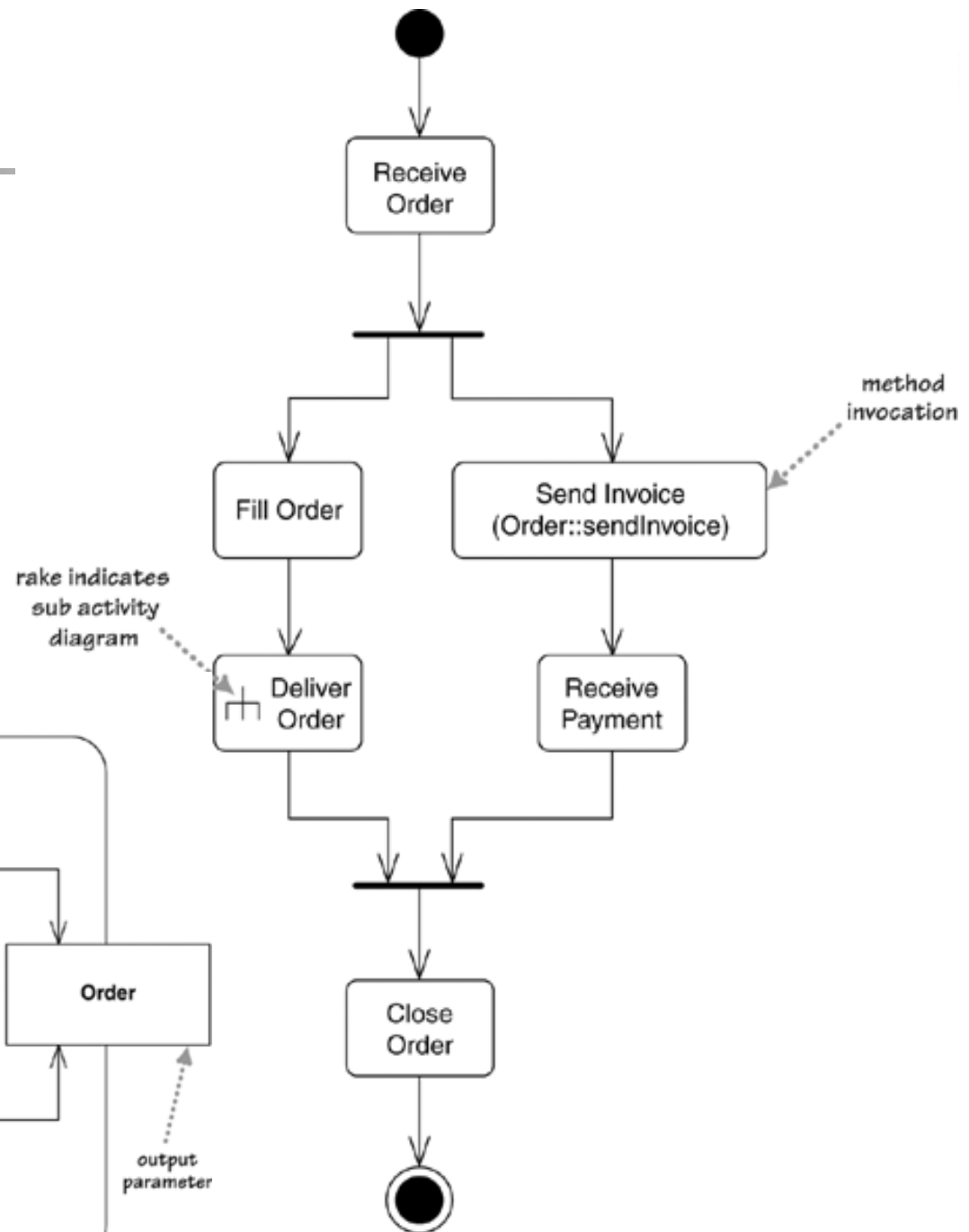
Maszyny stanowe

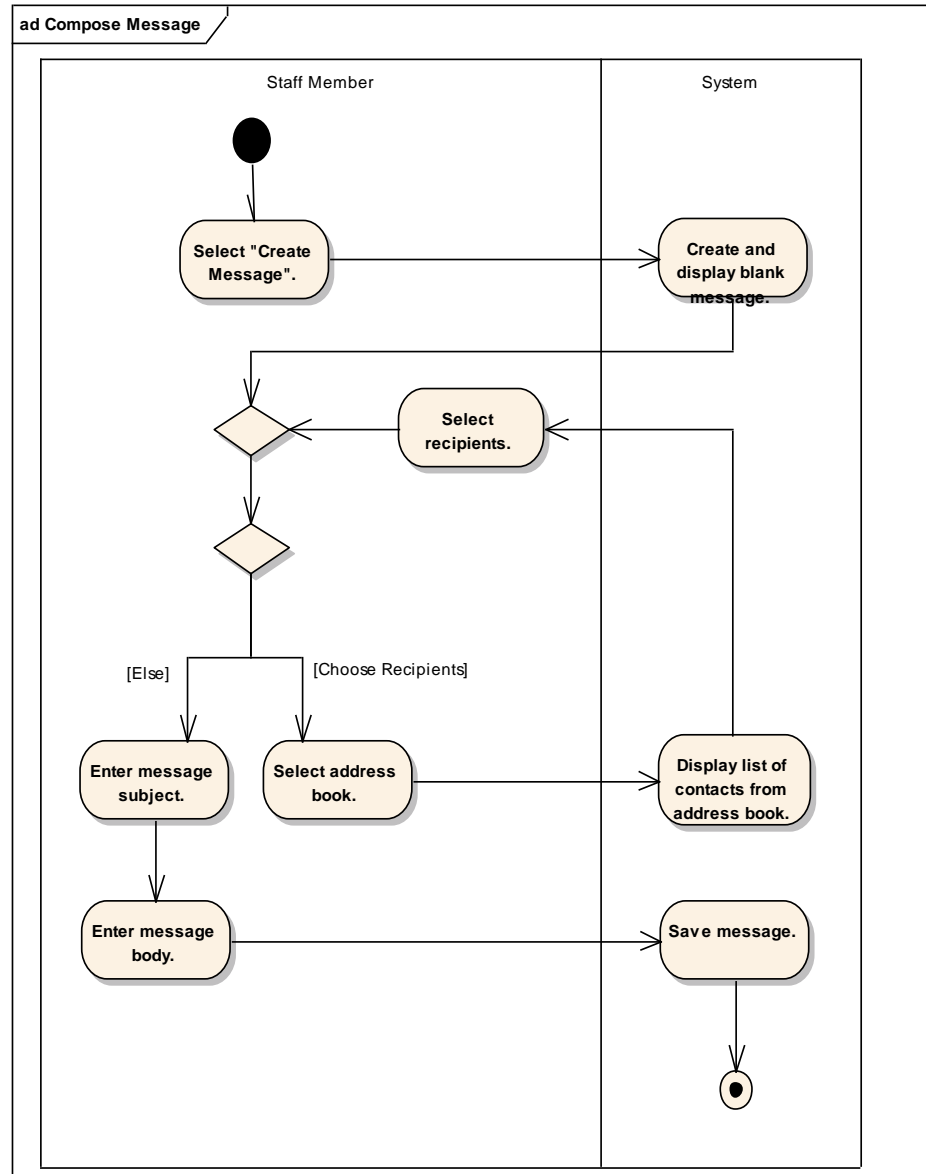


Aktywności / algorytmy

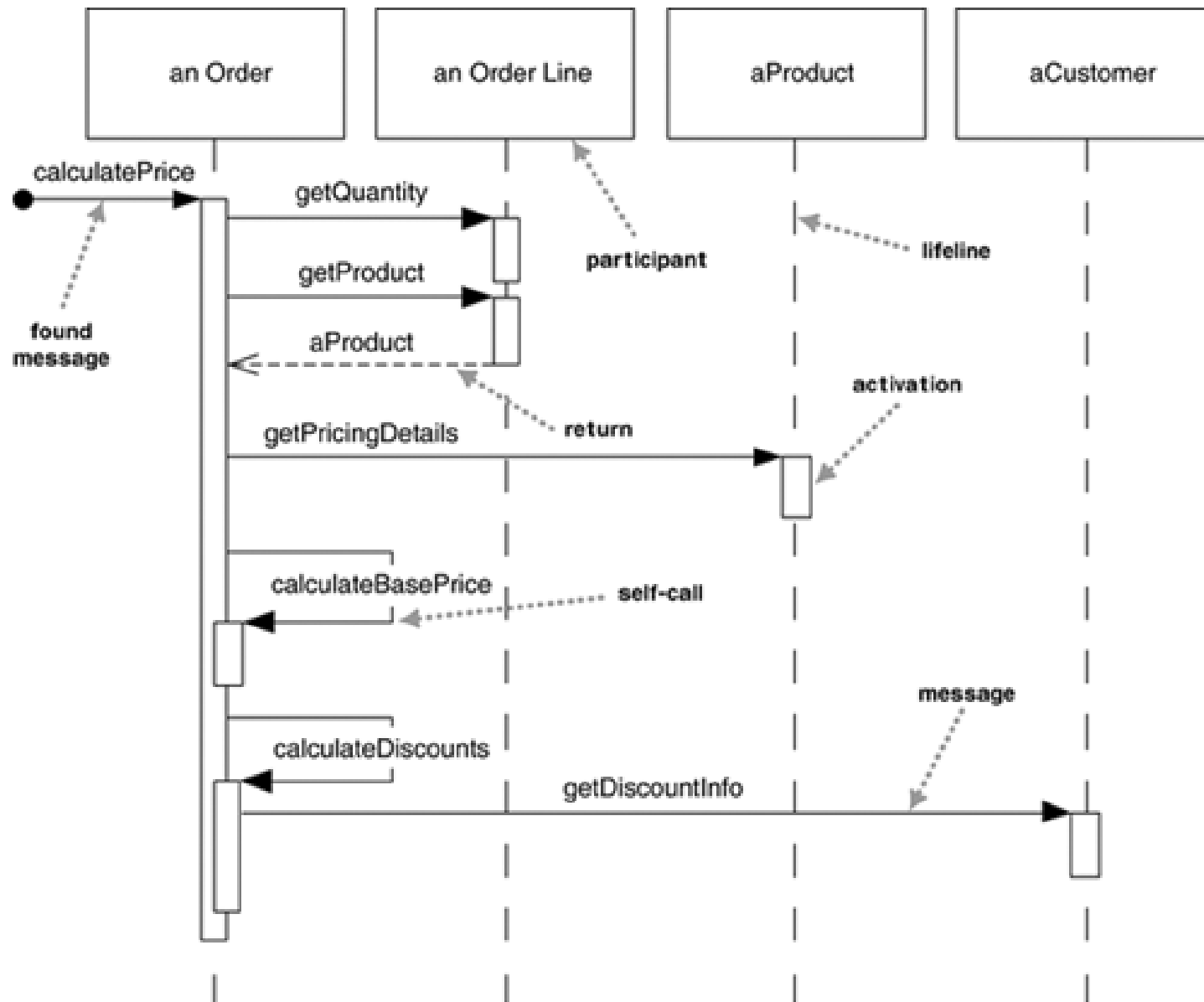


Aktywności / algorytmy

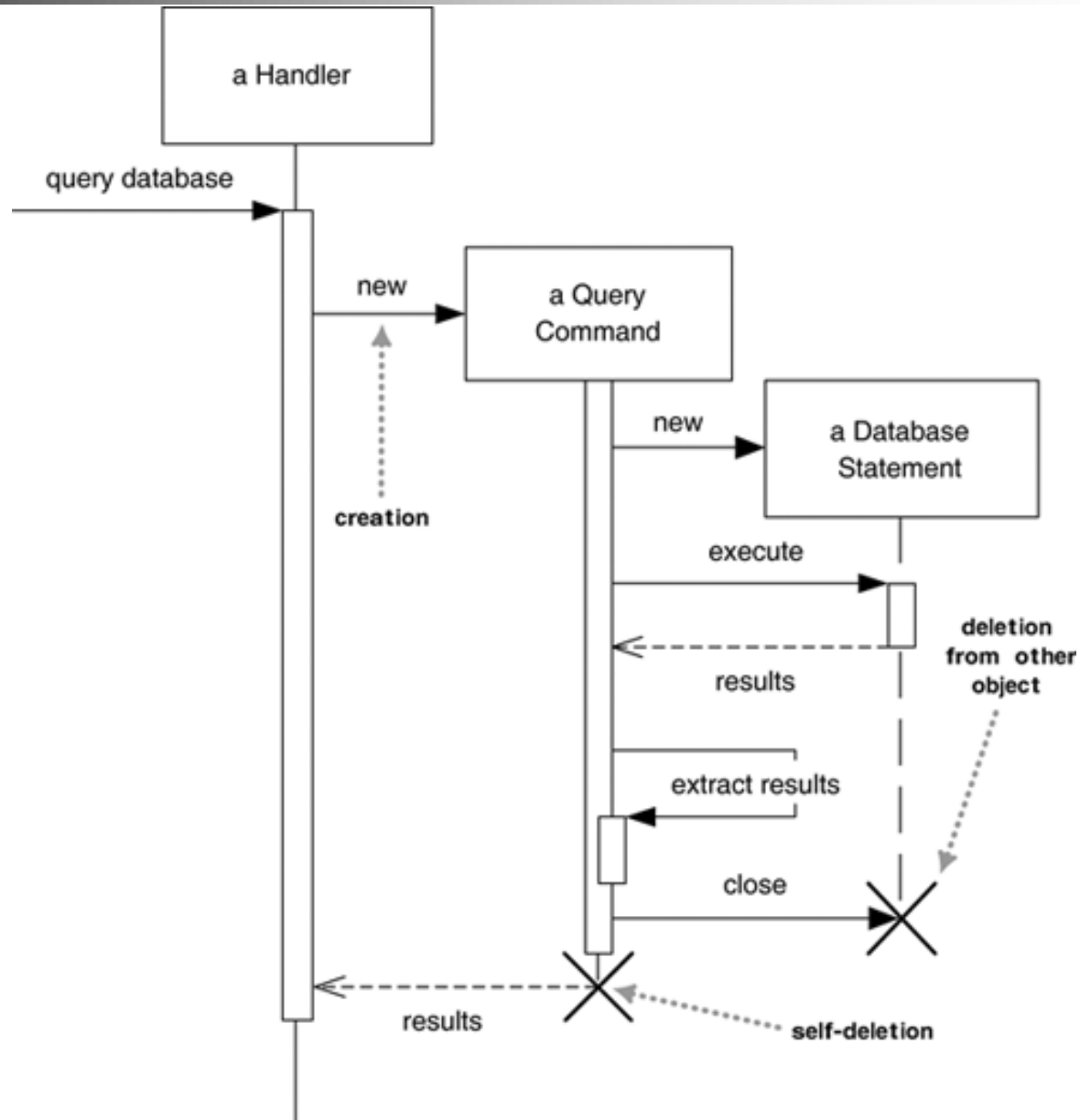




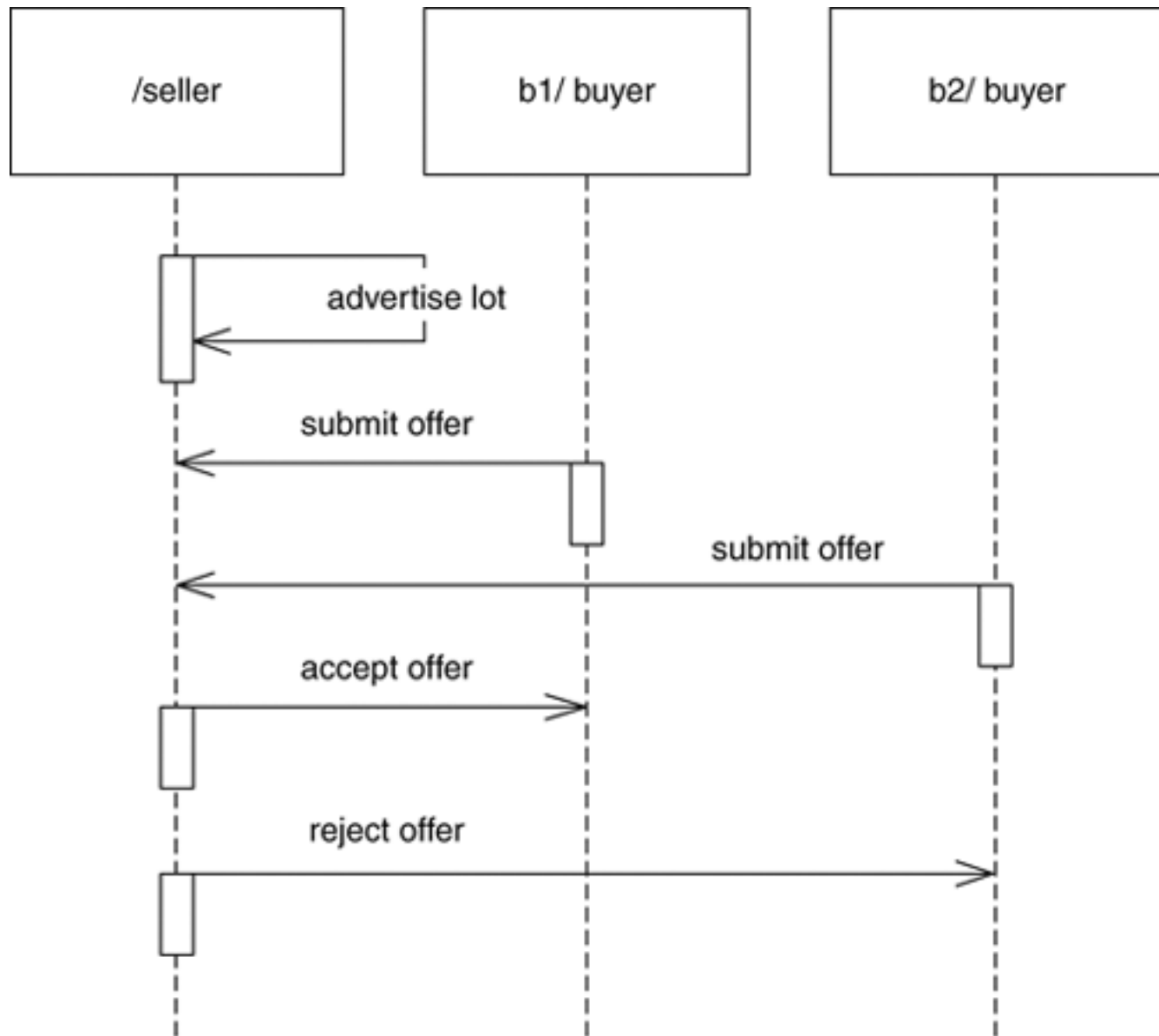
Interakcje



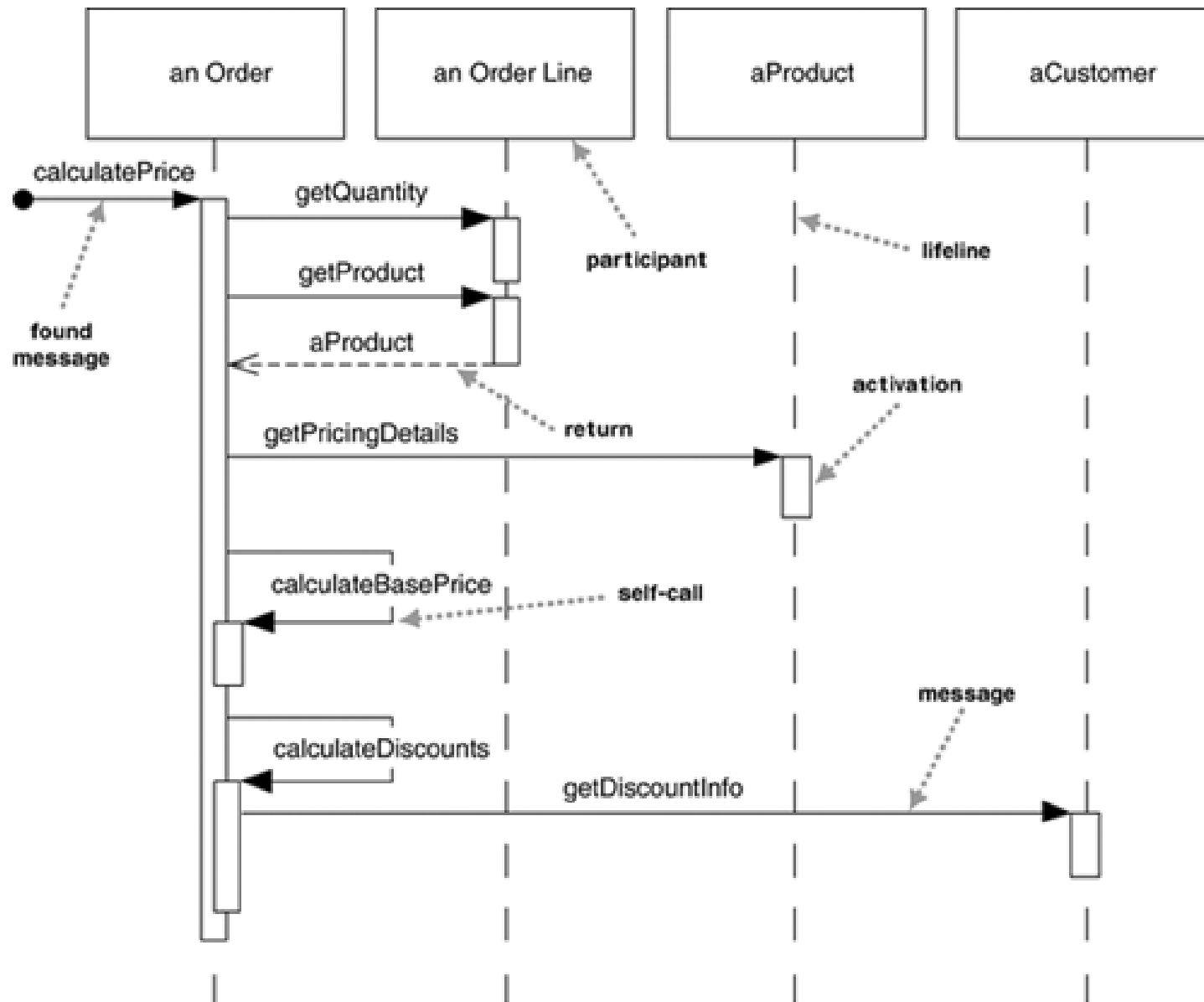
Interakcje



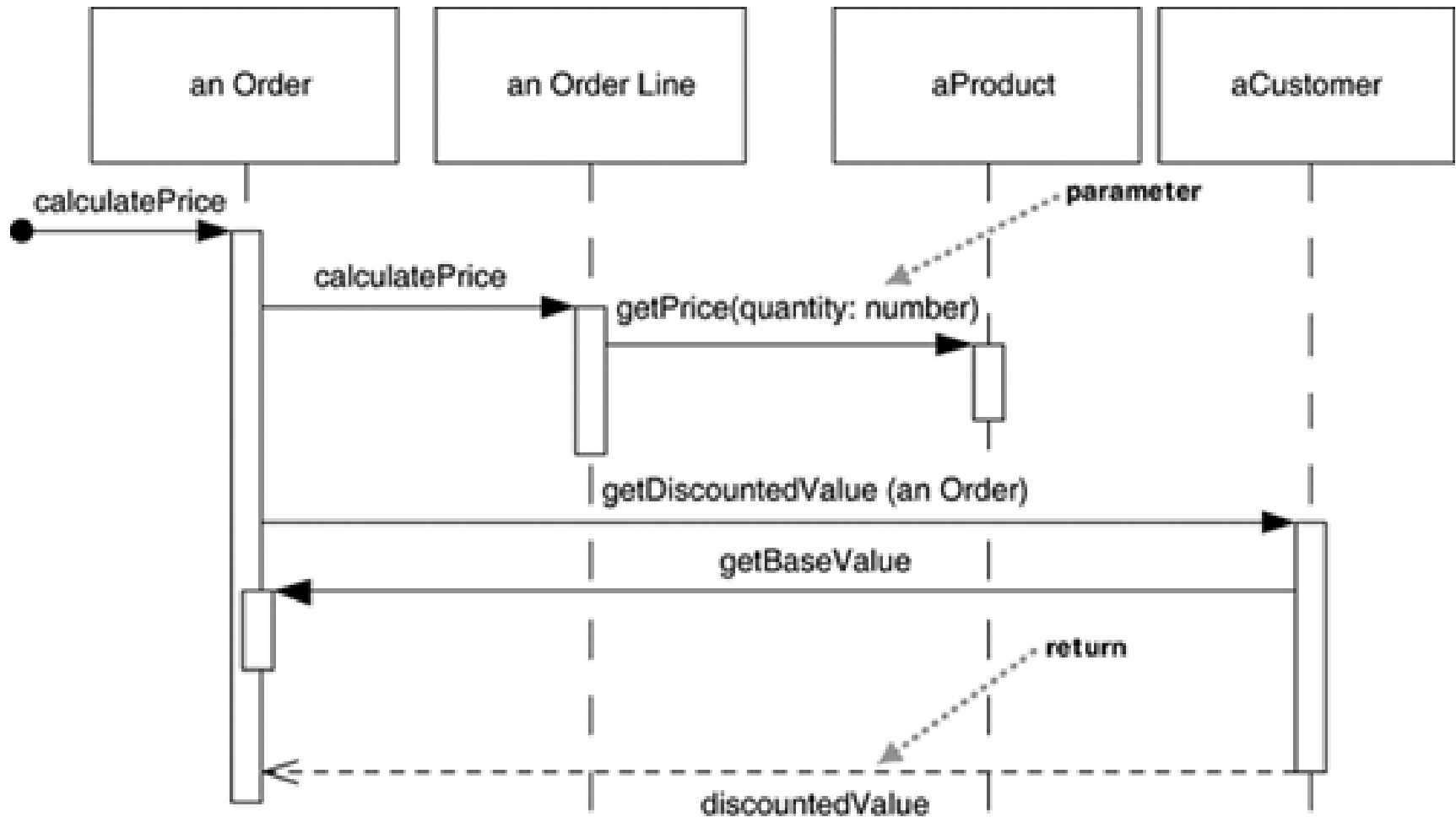
Interakcje



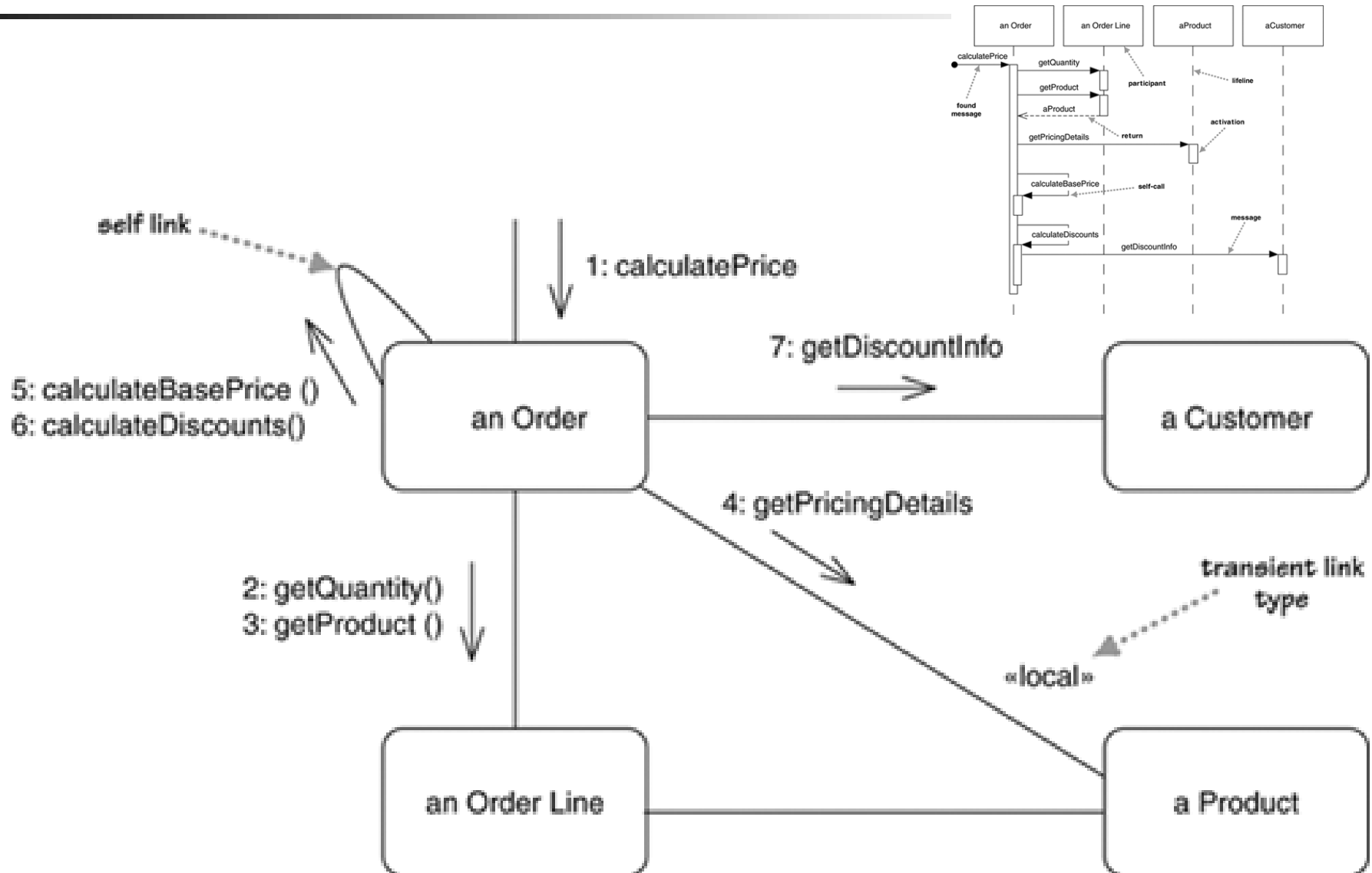
Interakcje – centralizacja



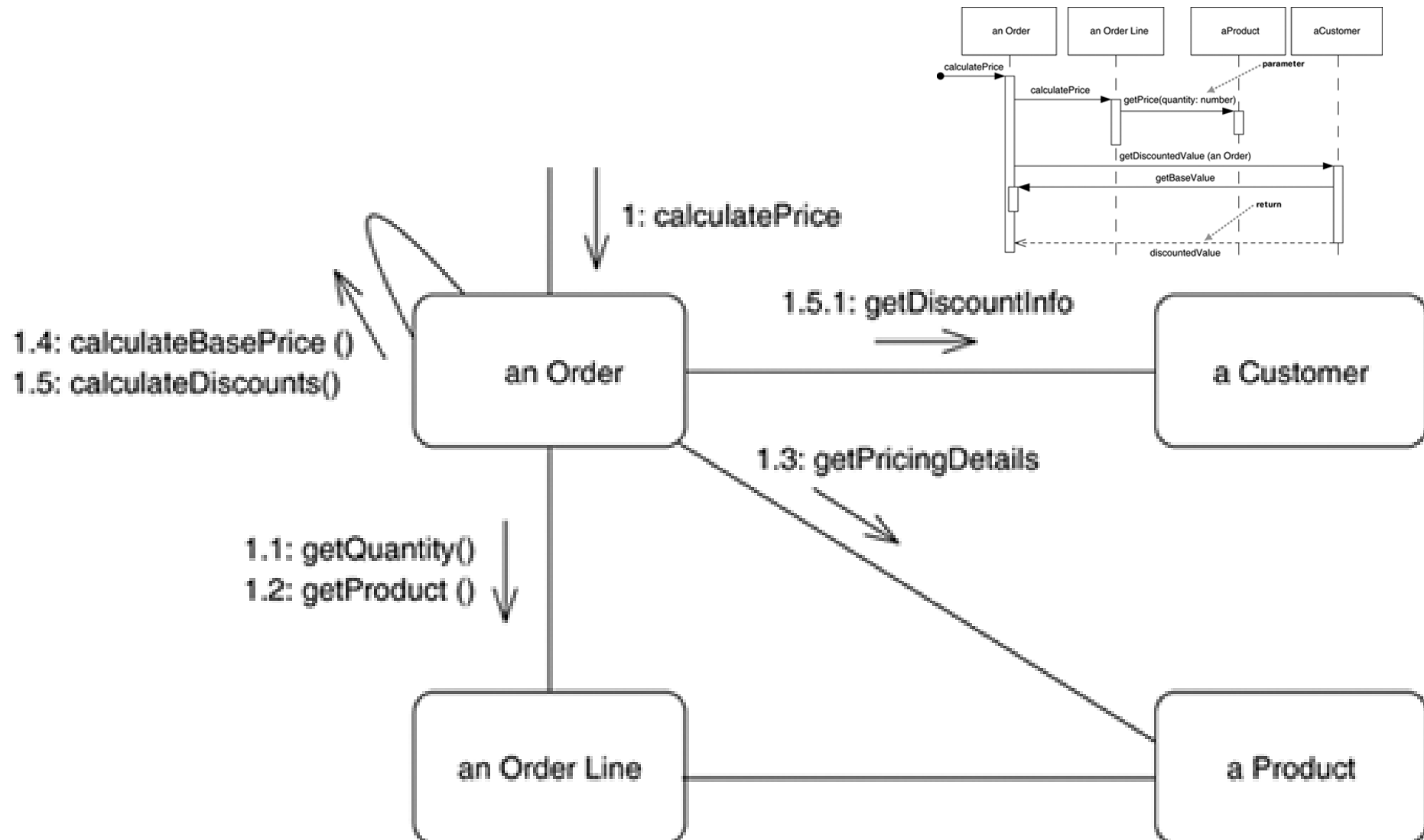
Interakcje – delegowanie



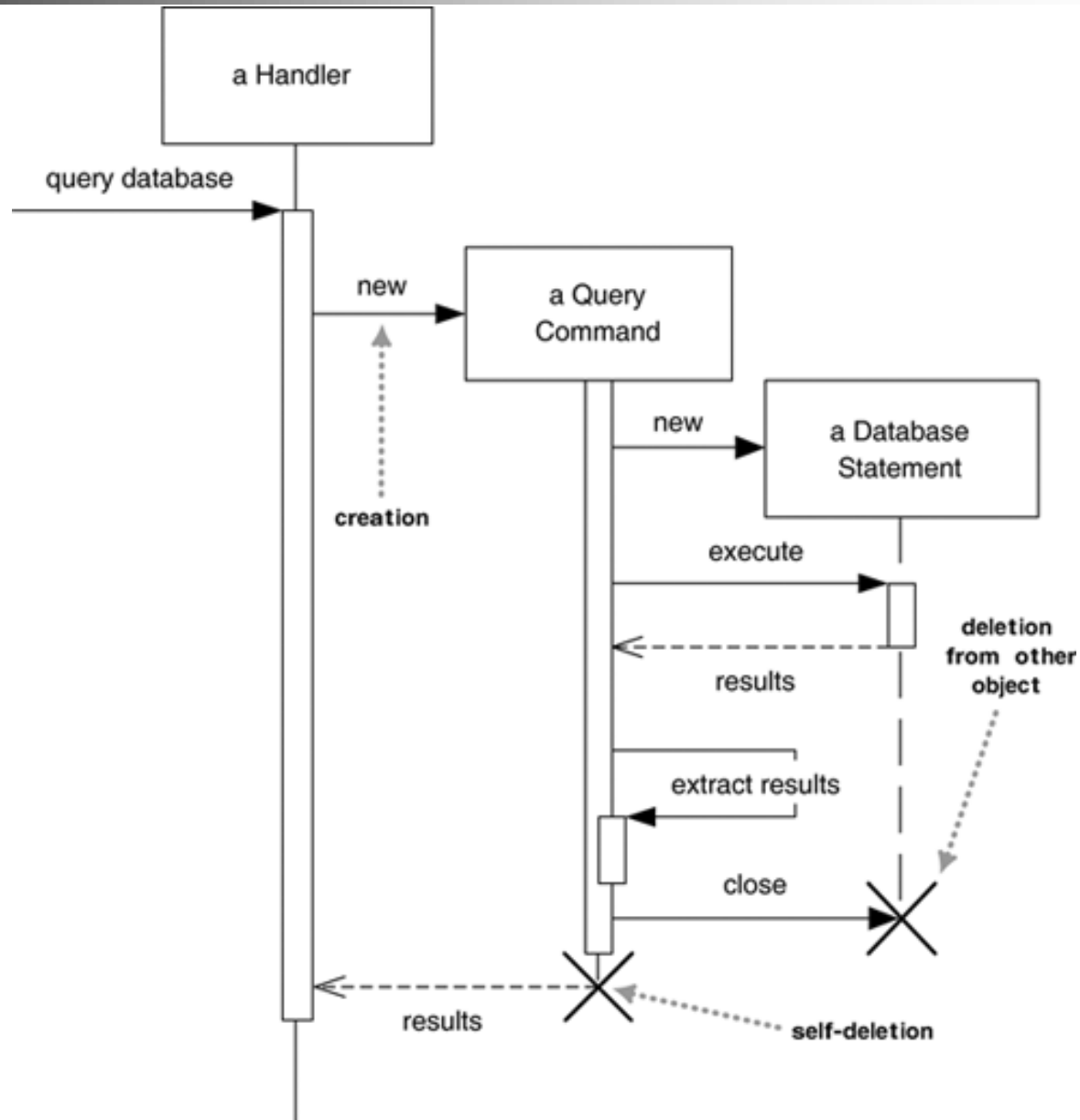
Interakcje – centralizacja



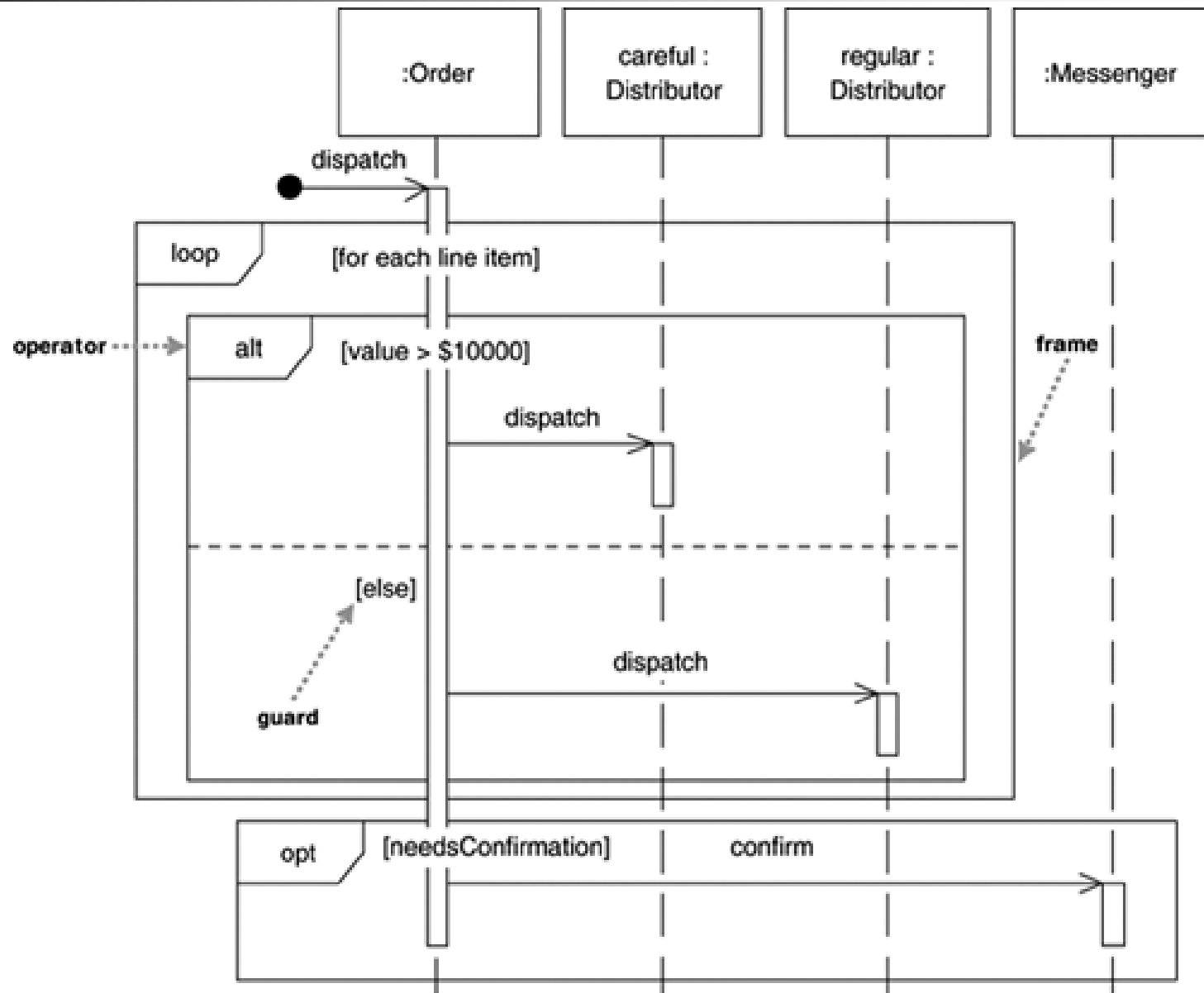
Interakcje – delegowanie



Interakcje – konstruktory i destruktory



Interakcje – pętle i warunki

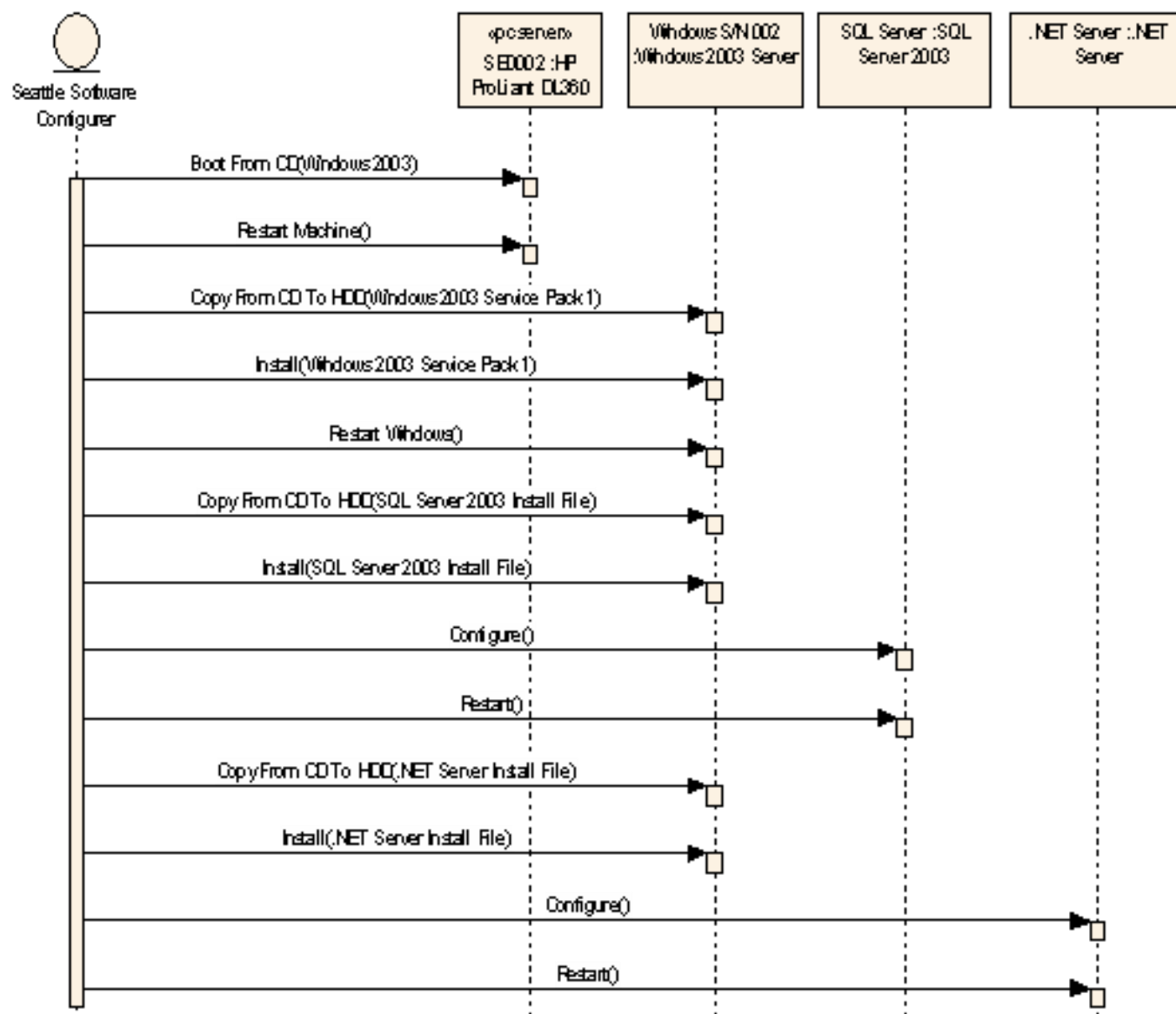




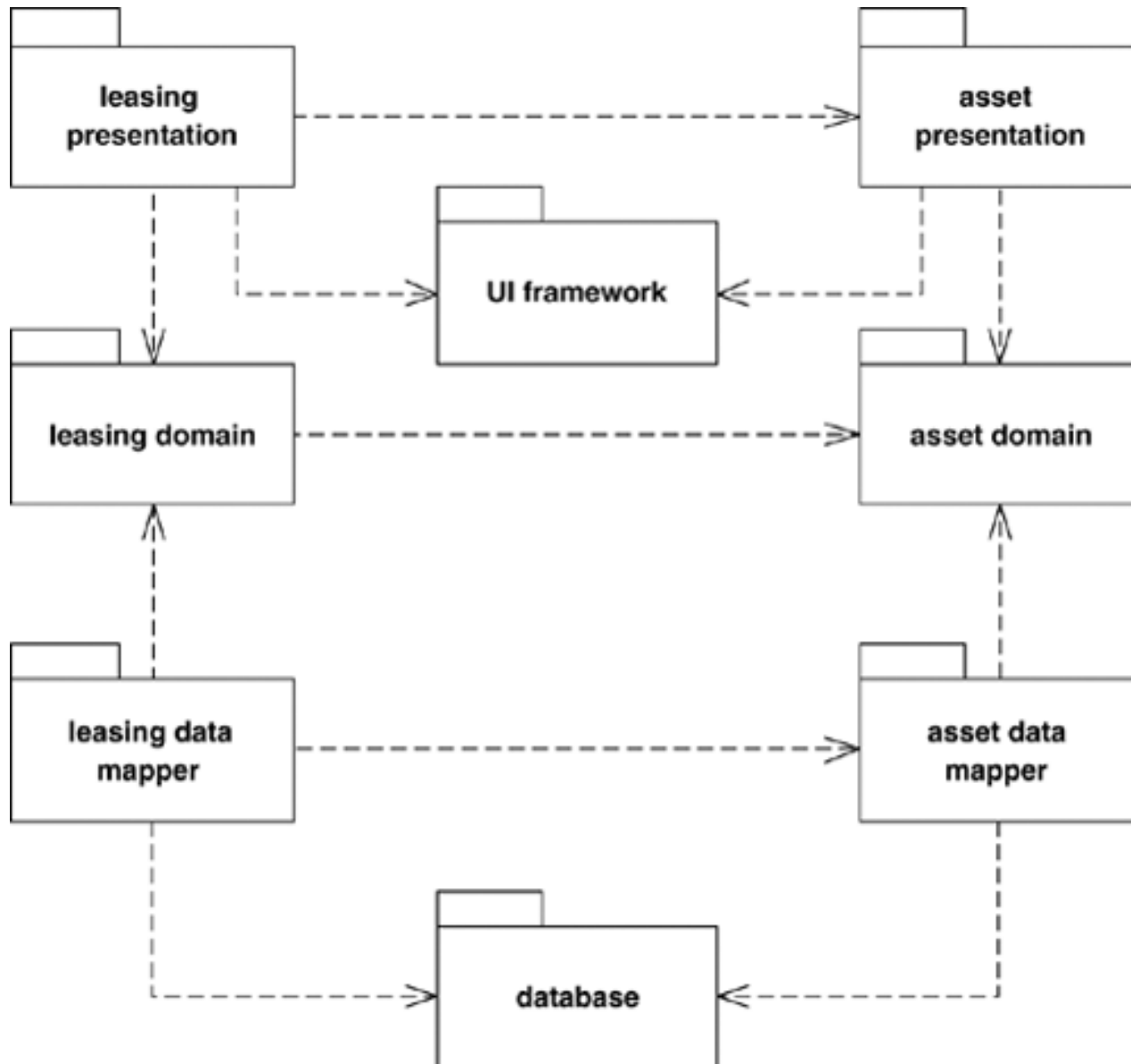
Interakcje – pętle i warunki

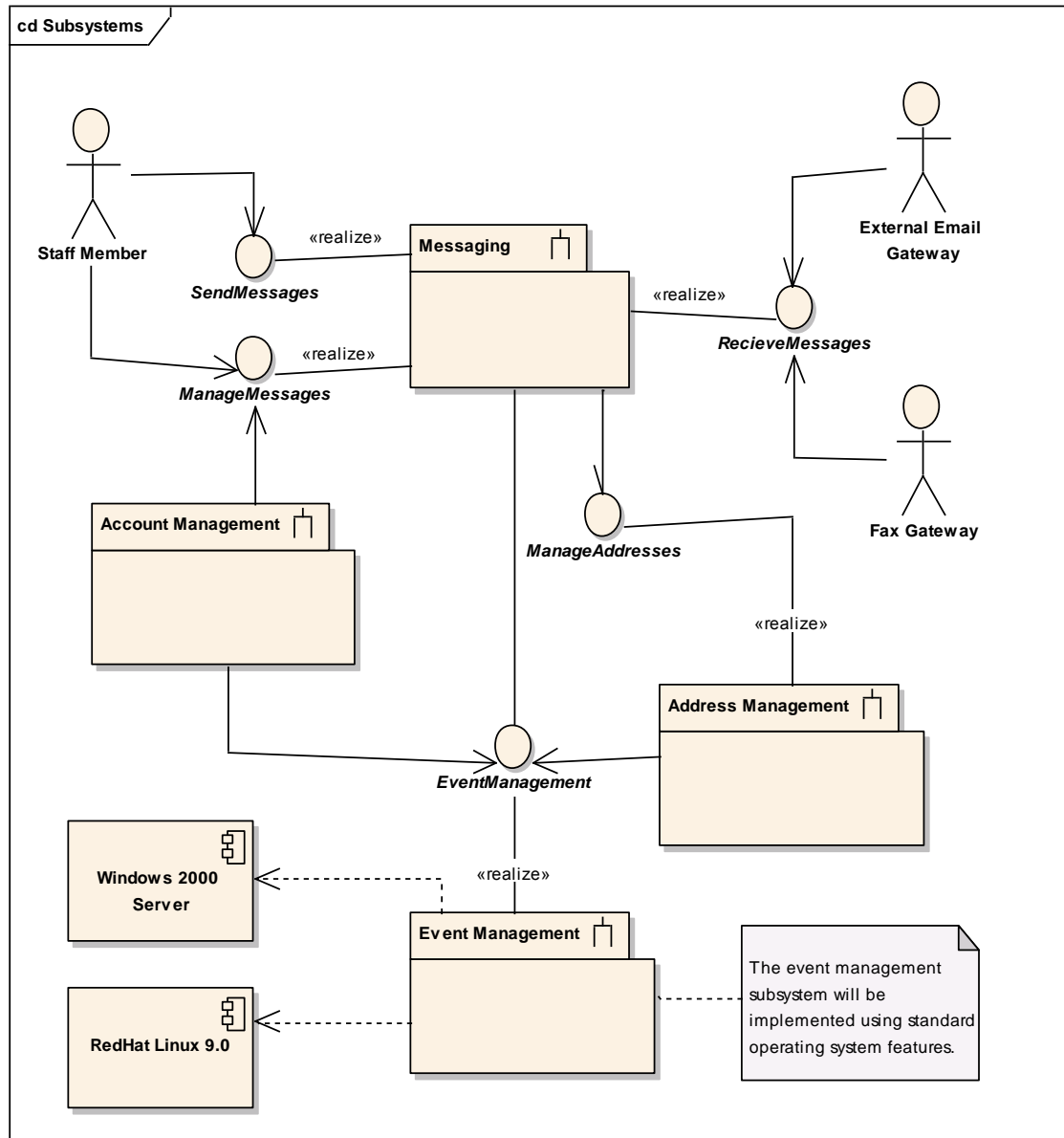
- procedure dispatch
 - foreach (lineitem)
 - if (product.value > \$10K)
 - careful.dispatch
 - else
 - regular.dispatch
 - end if
 - end for
 - if (needsConfirmation)
 - messenger.confirm
- end procedure

Interakcje

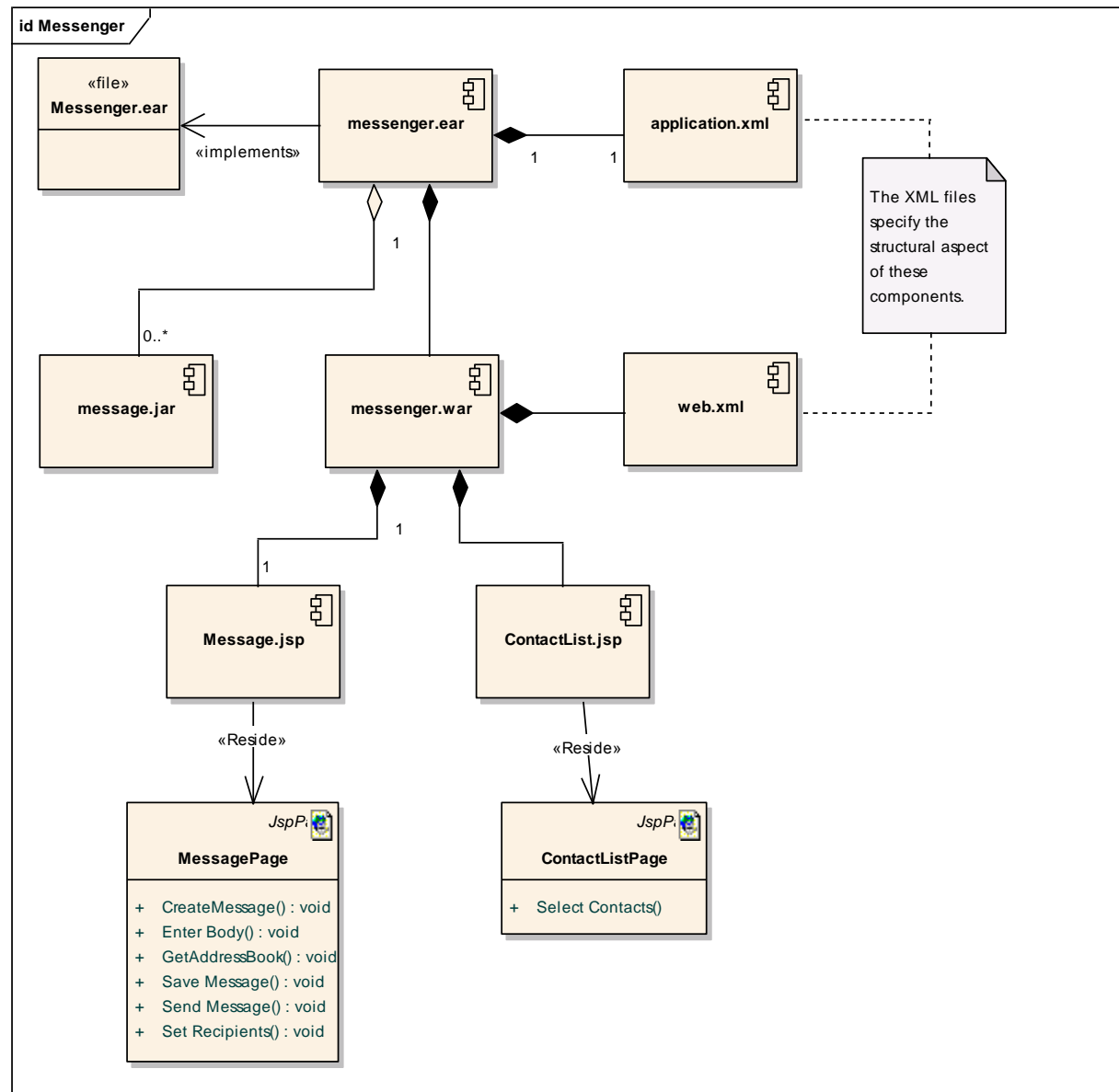


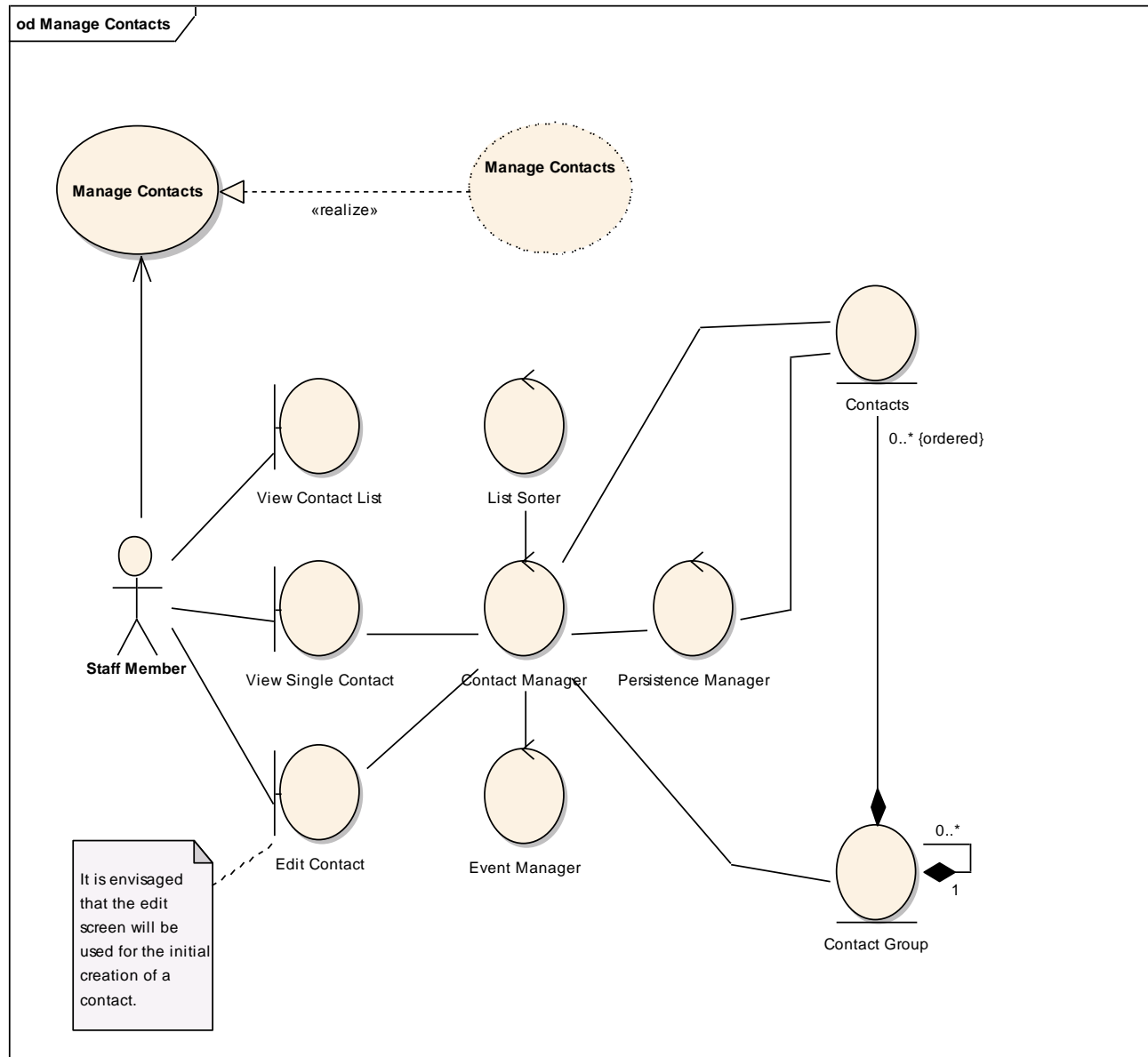
Moduły, pakiety, interfejsy



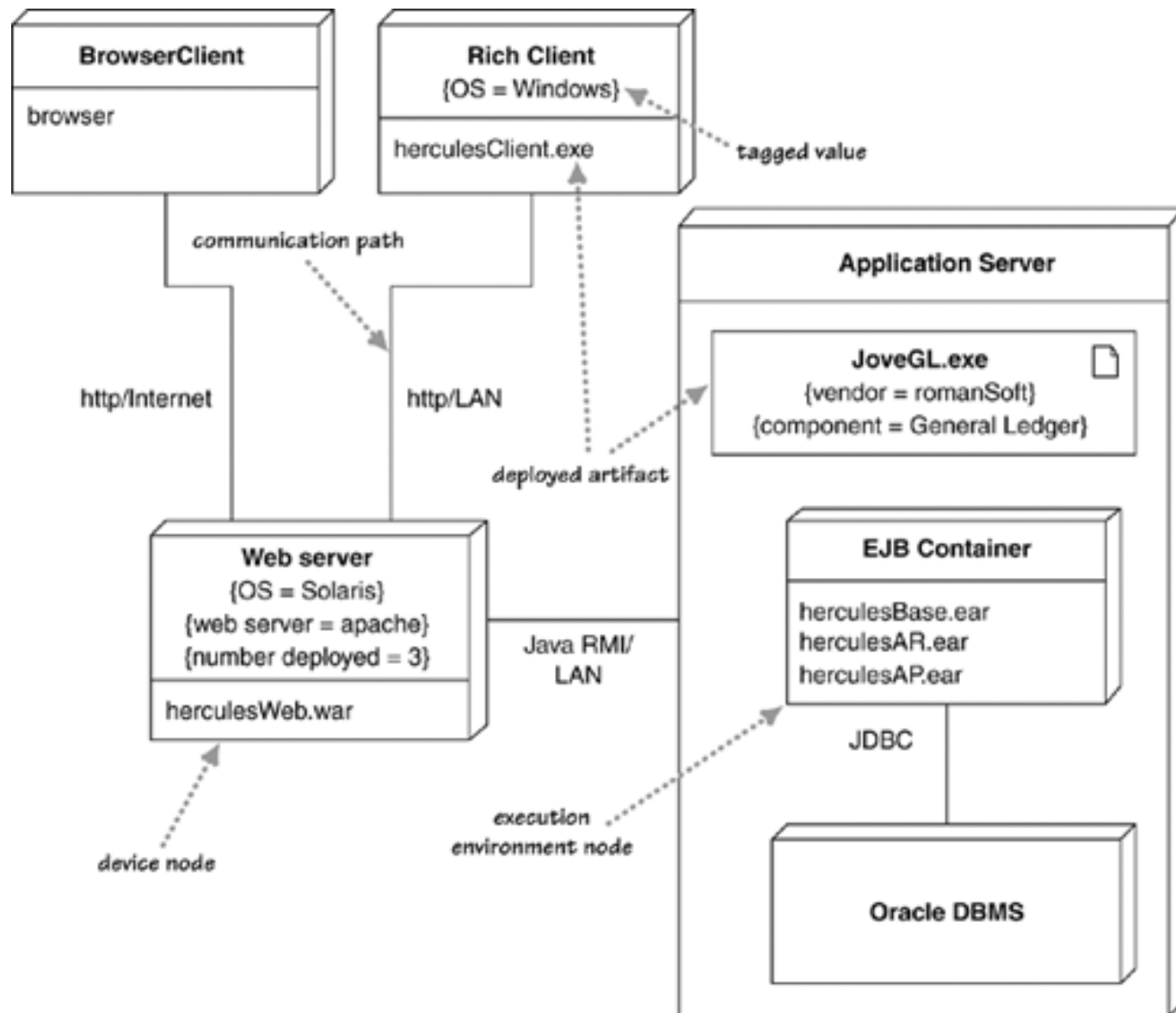


Moduły, pakiety, interfejsy

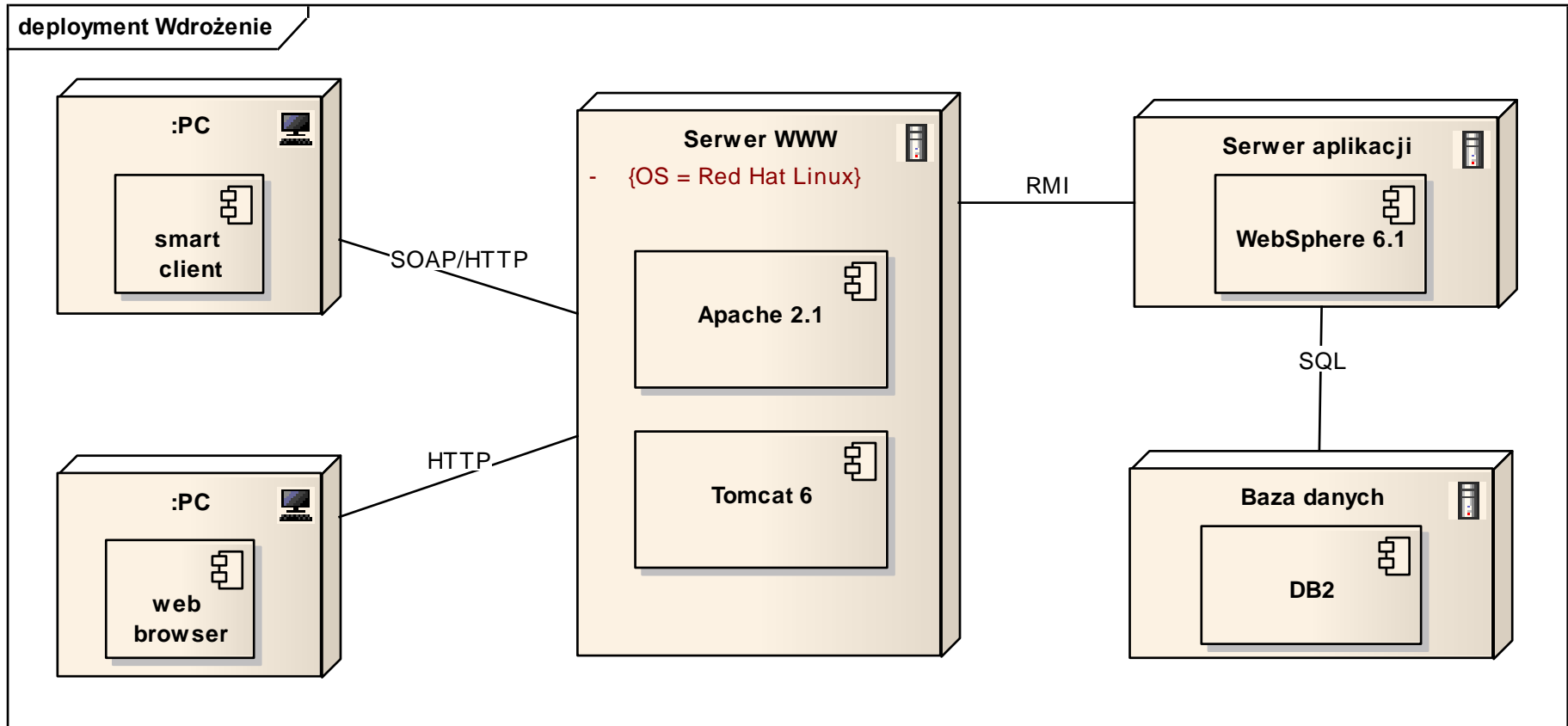


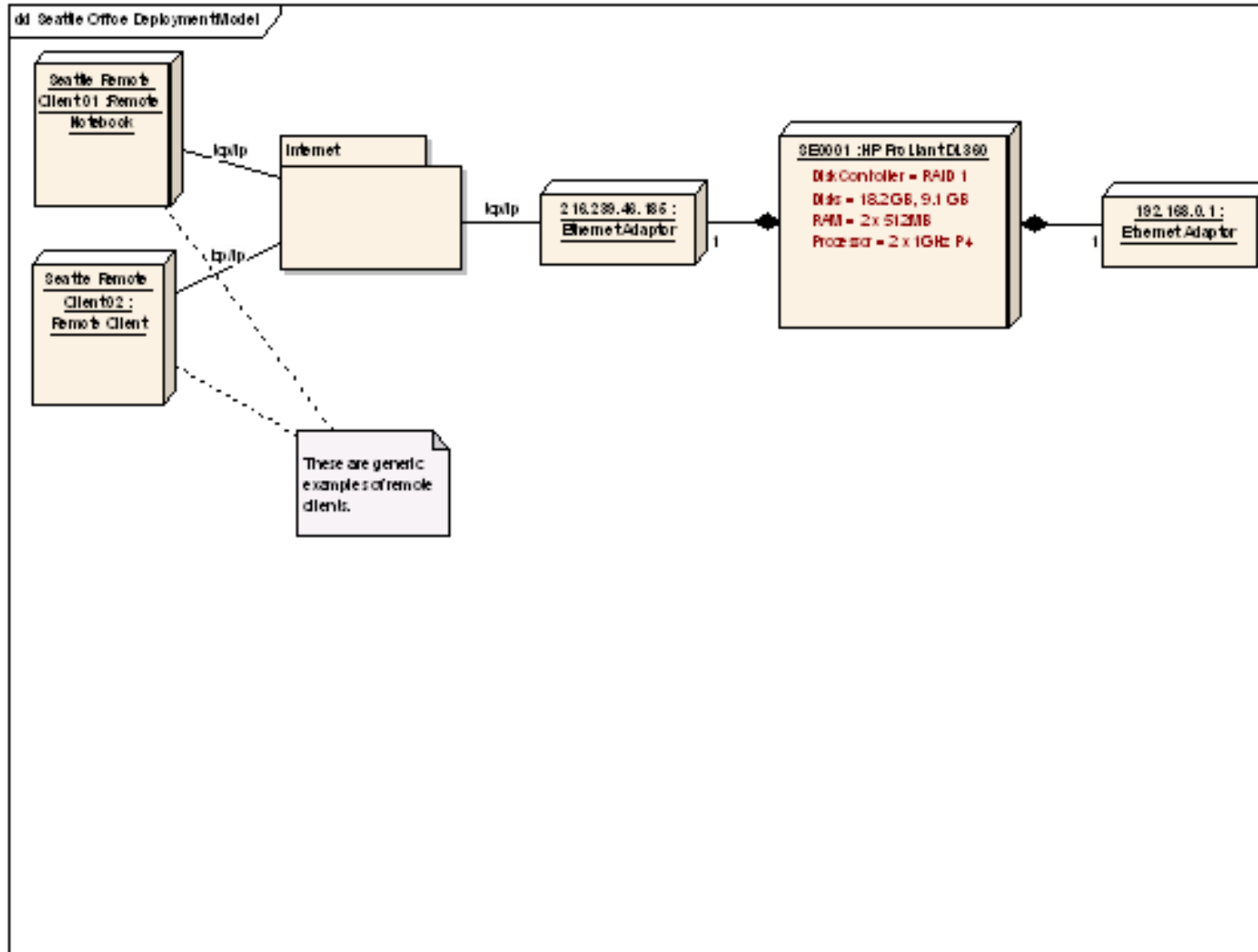


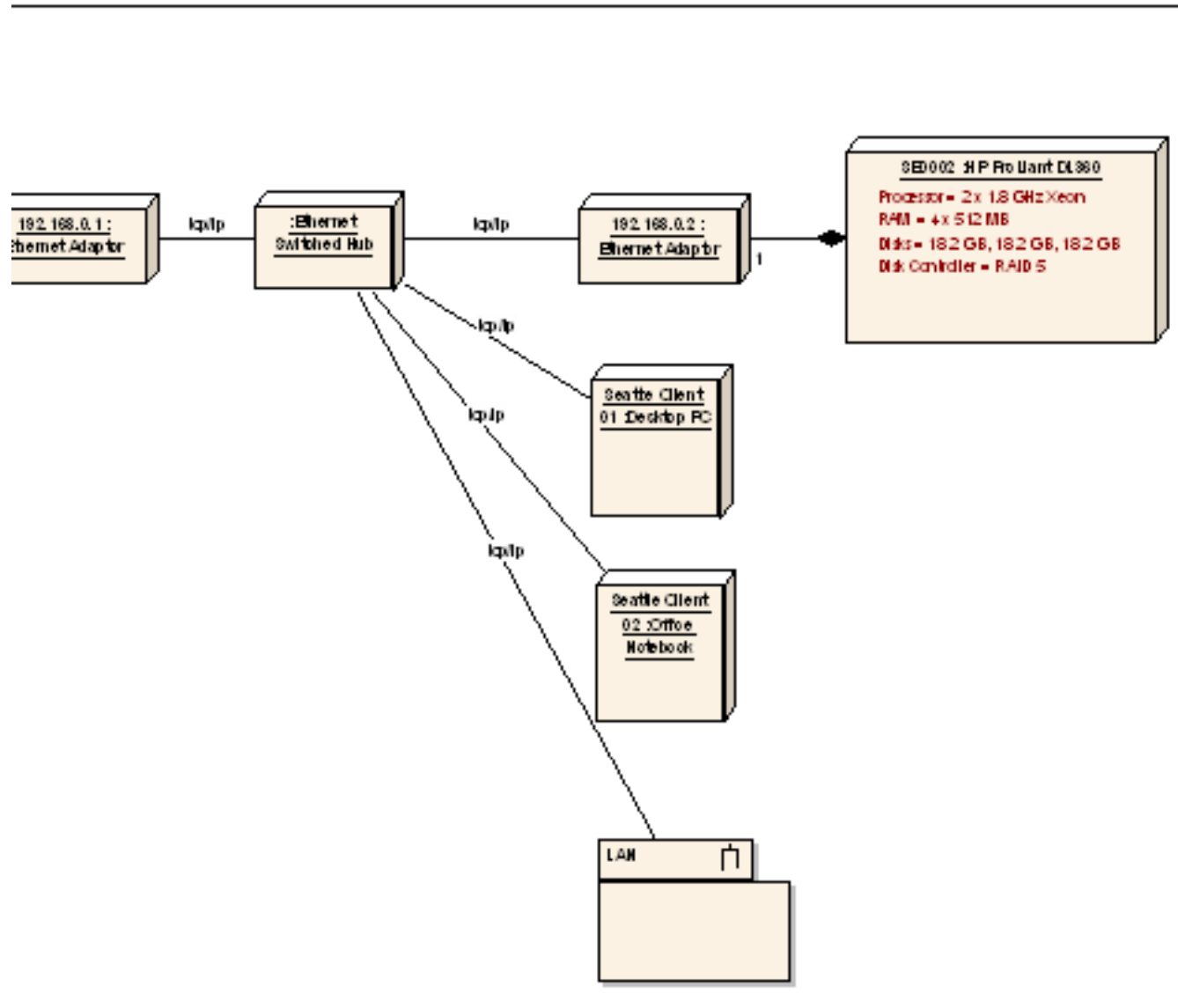
Infrastruktura, wdrożenie

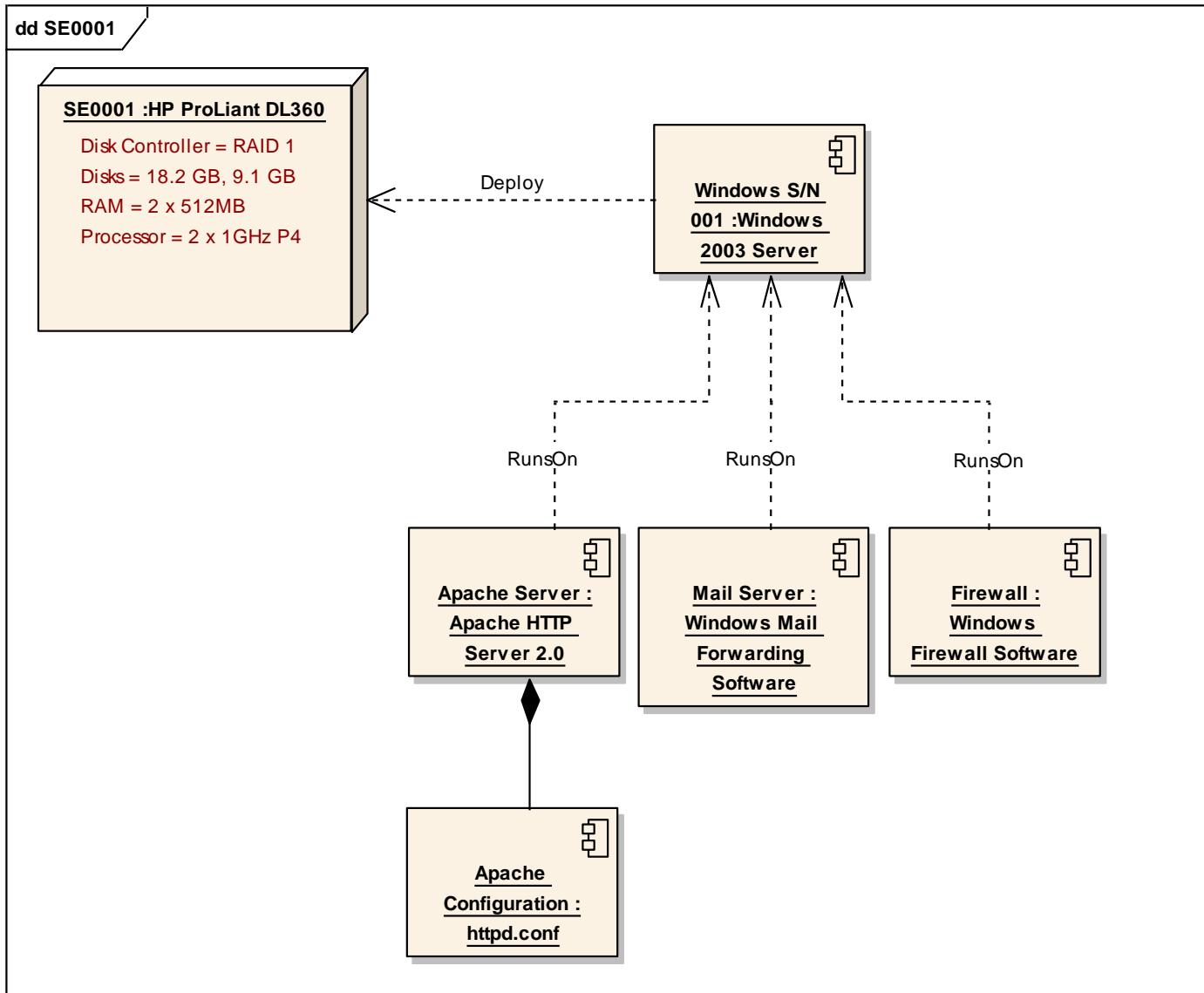


Infrastruktura, wdrożenie









Podsumowanie – warstwy

GUI windows
reports
speech interface
HTML, XML, XSLT, JSP, Javascript, ...

UI
(AKA **Presentation**, View)

handles presentation layer requests
workflow
session state
window/page transitions
consolidation/transformation of disparate data for presentation

Application
(AKA Workflow, Process, Mediation, App Controller)

handles application layer requests
implementation of domain rules
domain services (*POS*, *Inventory*)
- services may be used by just one application, but there is also the possibility of multi-application services

Domain
(AKA Business, Application Logic, Model)

very general low-level business services
used in many business domains
CurrencyConverter

Business Infrastructure
(AKA Low-level Business Services)

(relatively) high-level technical services
and frameworks
Persistence, *Security*

Technical Services
(AKA Technical Infrastructure, High-level Technical Services)

low-level technical services, utilities,
and frameworks
data structures, *threads*, *math*,
file, *DB*, and *network I/O*

Foundation
(AKA Core Services, Base Services, Low-level Technical Services/Infrastructure)

more
app
specific
↑
dependency
↓

width implies range of applicability →

Przez analogię: Projekt domu – koncepcja



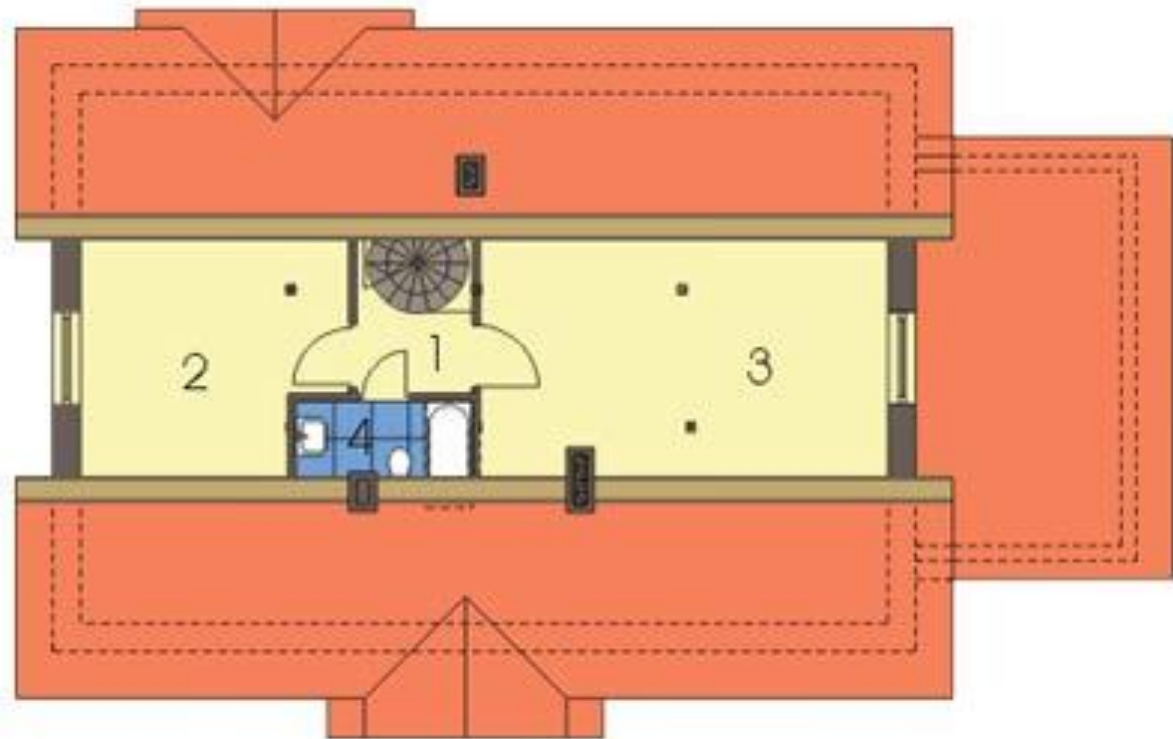
Projekt domu – perspektywa 1 (wizualizacja wnętrza)



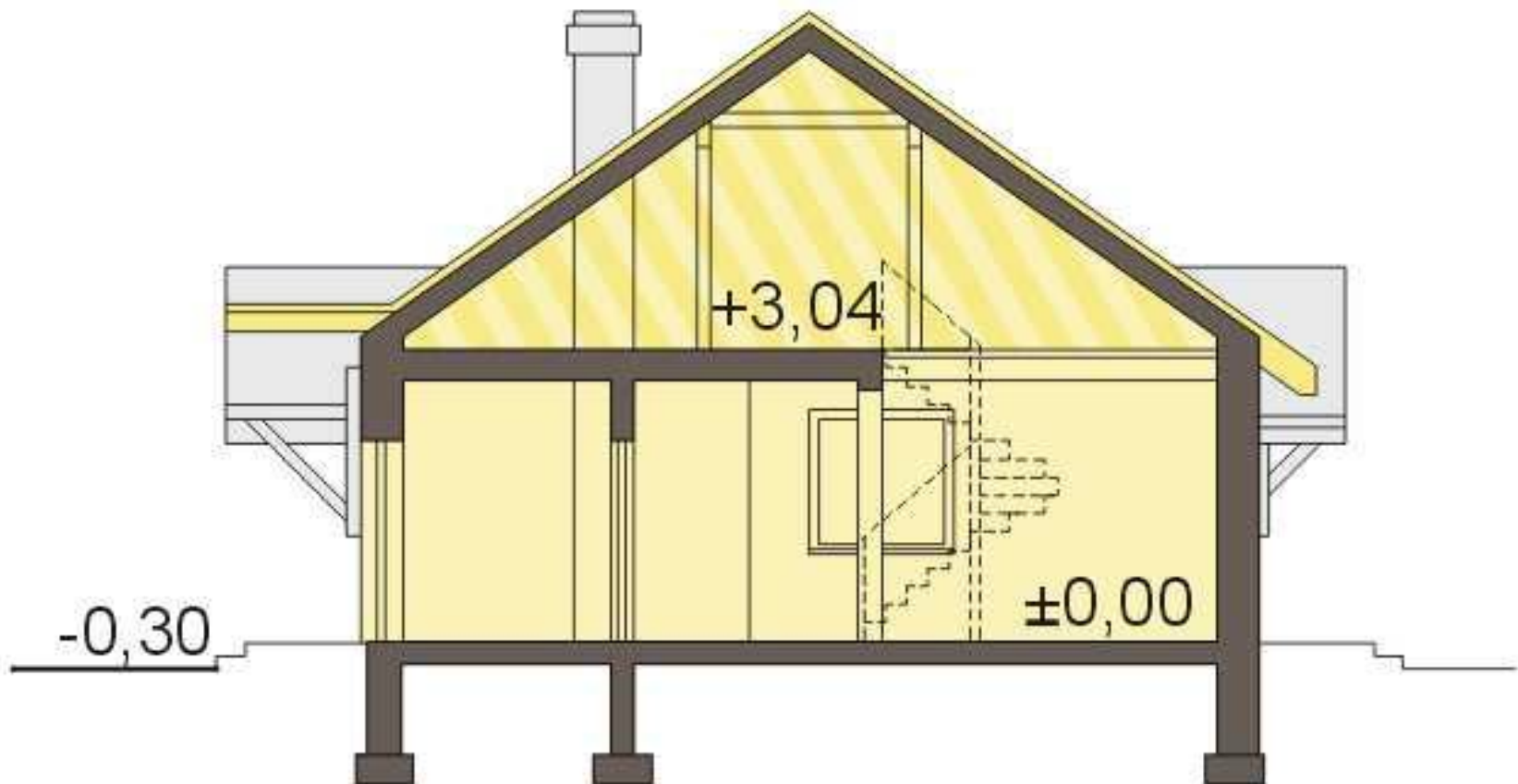
Projekt domu – perspektywa 2 (rzut pionowy)



Projekt domu – perspektywa 2 (rzut pionowy)



Projekt domu – perspektywa 3 (przekrój pionowy)



Projekt domu – perspektywa 4 (elewacje)



Projekt domu – projekt, kosztorys

Technologia

Fundamenty - beton
 Ściany zewnętrzne - Protherm
 Ściany nośne - Porothersm
 Strop - drewniany/Teriva
 Więźba dachowa - drewniana
 Dach - dachówka ceramiczna, cementowa
 blachodachówka.

Dane techniczne (Wersja B (strop drewniany))

Powierzchnia użytkowa	98.2 m ²
Powierzchnia zabudowy	152.6 m ²
<i>Minimalne wymiary działki</i>	
Wersja B (strop drewniany)	25.4 × 17.4 m
Wersja A (strop drewniany)	21.8 × 17.4 m
Wersja C (strop Teriva)	21.8 × 17.4 m
Wersja D (strop Teriva)	25.4 × 17.4 m
Wersja E (strop drewniany)	28.4 × 17.4 m
Wersja F (strop Teriva)	28.4 × 17.4 m
Powierzchnia dachu	197 m ²
Kąt nachylenia dachu	35 °
Wysokość budynku	6.8 m
Kubatura	324 m ³
Wysokość pomieszczeń	2.7 m

Cennik

wersja projektu	garaż	cena projektu	koszt budowy stanu surowego	koszt budowy z wykończeniem
Wersja B (strop drewniany) → Zamów	Tak	1100 zł netto	156.5 tys. zł netto	196.2 tys. zł netto
Wersja A (strop drewniany) → Zamów	Nie	1050 zł netto	135.9 tys. zł netto	171.3 tys. zł netto

AUTORSKA PRACOWNIA PROJEKTOWA
 "HORYZONT"
 ul. A. Młodnickiego 38/1
 50-305 Wrocław

KOSZTORYS BUDOWLANY

NAZWA INWESTYCJI : DOM JEDNORÓDZINNY WOLNOSTOJĄCY "GIGIA"
 DLA ZADOSPOKOJENIA PODDASZA
 BRANŻA : BUDOWLANA
 SPORZĄDZIŁ KALKULACJE : mgr inż. Bartosz Stręski
 DATA OPRACOWANIA : maj 2003 (aktualizacja cen I kwartał 2005)

Stosunek robocizny : 7.80 zł
 Paszport cen : 2 kw. 05

NARZUTY
 Koszt pośrednie (Pp) : 04.00 N.R. S
 Koszt zysku (Kz) : 7.00 S.M.
 Zysk (Z) : 12.00 S.R.Kp(R), S.Kp(S)
 Ogółem wartość kosztorysowa robót : 196 171.00 zł

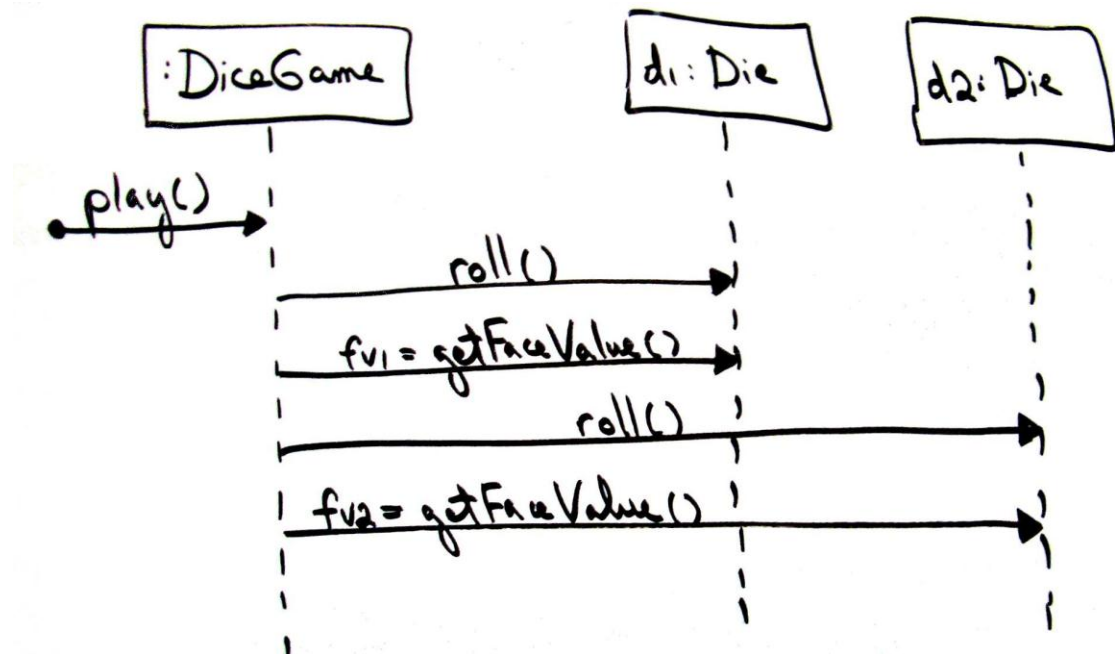
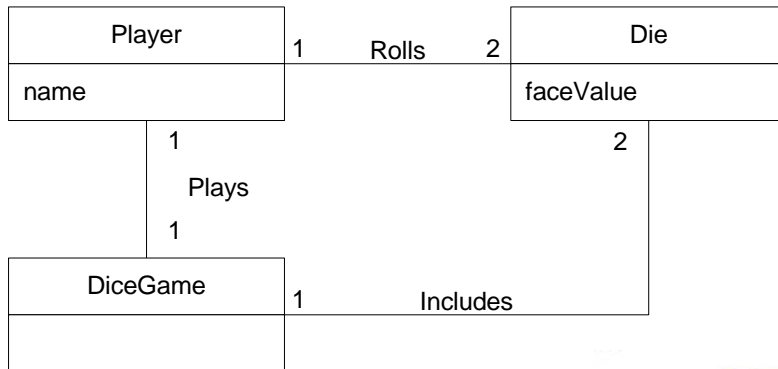
Słownie: sto dziewięćdziesiąt sześć tysięcy sto siedemdziesiąt jeden i 00/100 zł

Uwaga o uproszczeniu kosztorysu:
 KOSZTORYS NINIEJSZY JEST WYCENĄ SPORZĄDZONĄ DLA OKREŚLENIA SZCZEGÓŁOWYCH WARTOŚCI ROBÓT BUDOWLANYCH, OPRACOWANĄ W OPARCIU O PROJEKT BUDOWLANY, PRZY ZAŁOŻENIU PRZECIĘTNYCH WARUNKÓW WYKONANIA ROBÓT I WYBRANYCH ROZWIĄZAŃ TECHNICZNYCH OPISANYCH W OPISIE CHARAKTERYSTYCE OBIEKTU. ILOŚCI OBRABOWE JAK RÓWNIEŻ ZESTAWIENIA MATERIAŁÓW SĄ ILOŚCIAMI PRZYBLIŻONYMI I UŚREDNIONYMI I MOGĄ RÓŻNIC SIĘ OD ILOŚCI RZECZYWISTYCH W ZALŻNIKOŚCI OD ZASTOSOWANYCH ROZWIĄZAŃ MATERIAŁOWYCH ORAZ PRZYJĘTYCH TECHNOLOGII WYKONANIA ROBÓT. PRZED ZAMÓWIENIEM MATERIAŁÓW ILOŚCI OKREŚLONE W ZESTAWIENIU MATERIAŁÓW NALEŻY KAZDORAZOWO ZWERYFIKOWAĆ NA BUDOWIE. KOSZTORYS NALEŻY ROZPATRYWAĆ JĄCZNIE Z DOKUMENTACJĄ PROJEKTOWĄ.

WYKONAWCA :

INWESTOR :

UML – to tylko notacja!



Podsumowanie

- „Obraz jest wart tysiąca słów”
 - Ale sam model może nie wystarczyć
- Przykład:
 - Przypadki użycia a model przypadków użycia
 - Wymagania нефunkcjonalne
 - Tabele
 - Macierze zależności
- Diagramy nie należące do standardu UML
 - Jak najbardziej!
 - Naszym celem czytelność / komunikatywność
 - (a nie zachowywanie „standardu”)
- Raz jeszcze: UML to tylko notacja!
 - Ze jej znajomości notacji nie wynika jakość modelu!