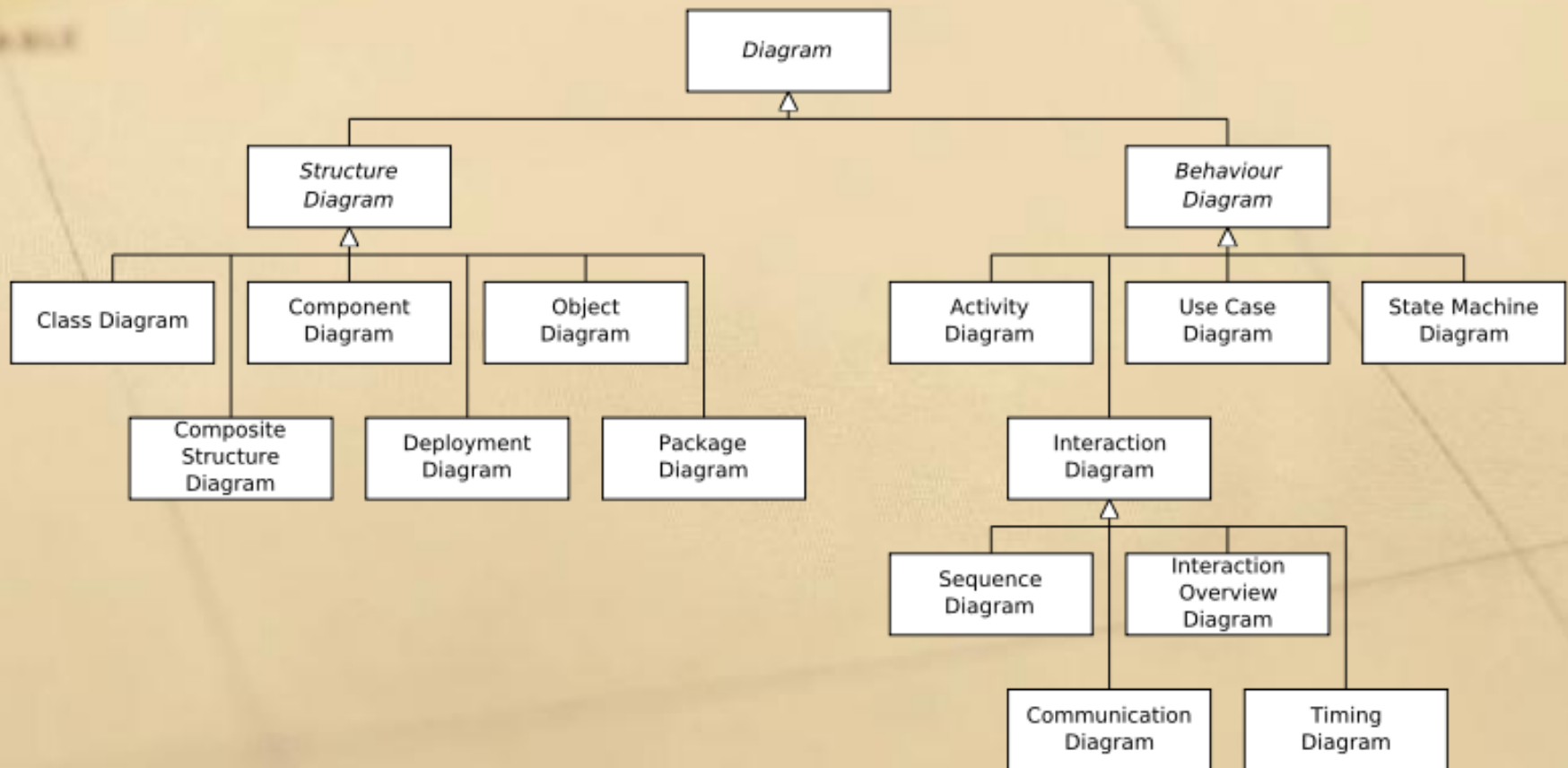


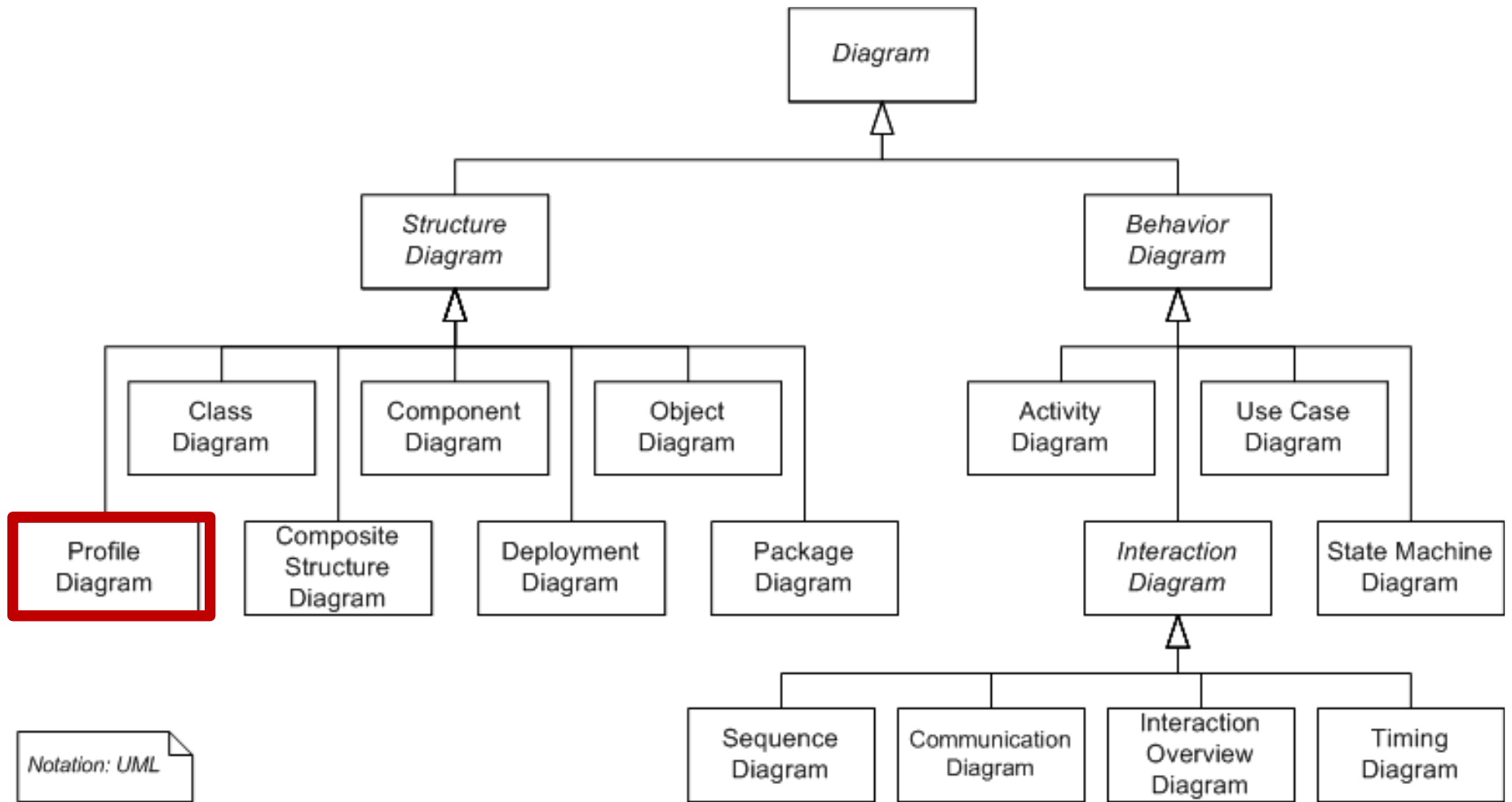
Modelowanie i analiza systemów informatycznych

Zbigniew Huzar

UML – diagramy



UML – diagramy



Rozszerzenia UML

- UML – problem „dopasowania” do:
 - dziedziny zastosowań
 - metodyki modelowania
 - platformy implementacji

Rozszerzenia UML

- Profil UML – zbiór stereotypów – dodatkowych elementów modelowania
 - nie naruszają struktury opisu (meta-modelu) języka
 - modyfikują (zwykle uściślają) semantykę istniejących elementów modelowania

Rozszerzenia UML

- Standardowe profile UML
 - UML Profile for Enterprise Distributed Object Computing Specification
 - UML Profile for Schedulability, Performance, and Time Specification
 - UML Profile for Business Modeling
 - UML Profile for Software Development Processes
 -

Rozszerzenia UML

- Mechanizmy rozszerzeń:
 - stereotypy
 - metki
 - ograniczenia
- Ograniczenie
 - wymaganie semantyczne, warunek nałożony na element modelowania
 - wyrażony w języku naturalnym lub sformalizowanym

Rozszerzenia UML

- Metka
 - Para: nazwa-wartość
 - przeznaczona do umieszczania w modelach dodatkowej informacji związanej z zarządzaniem modelowania
 - metka – specyficzny atrybut przypisywany elementowi modelowania, np.

autor = Kowalski

dataOpracowania = 21.02.2009

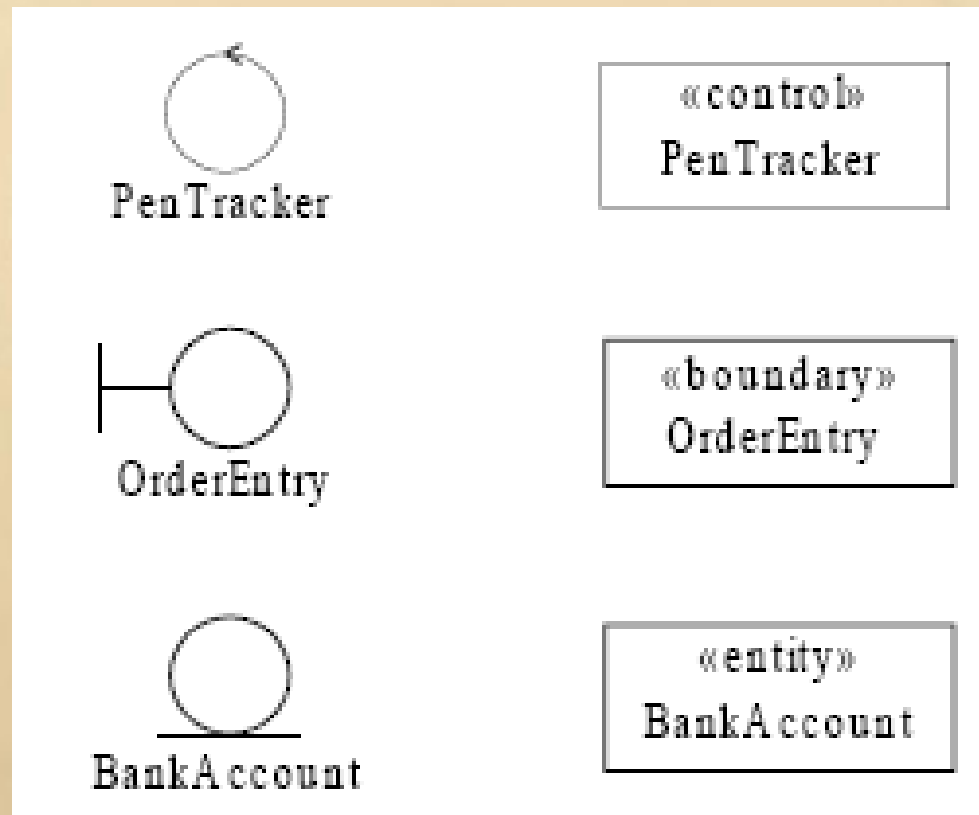
statusDokumentu = zaakceptowany

Rozszerzenia UML

- Stereotyp
 - wskazanie bazowego elementu modelowania UML
 - ustalenie nazwy (opcjonalnie ikony)
 - opcjonalnie dołączenie metek i nałożenie ograniczeń
 - (nieformalny) opis semantyki w języku naturalnym

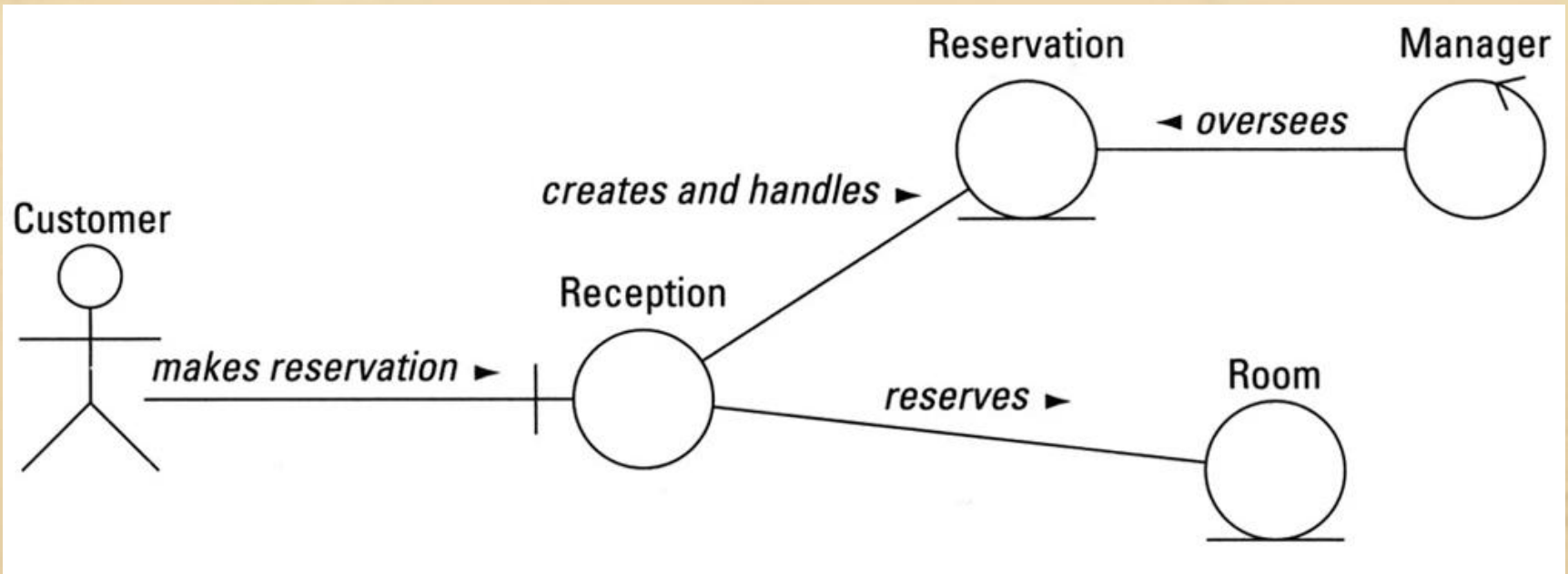
Rozszerzenia UML

- Stereotypy - przykład

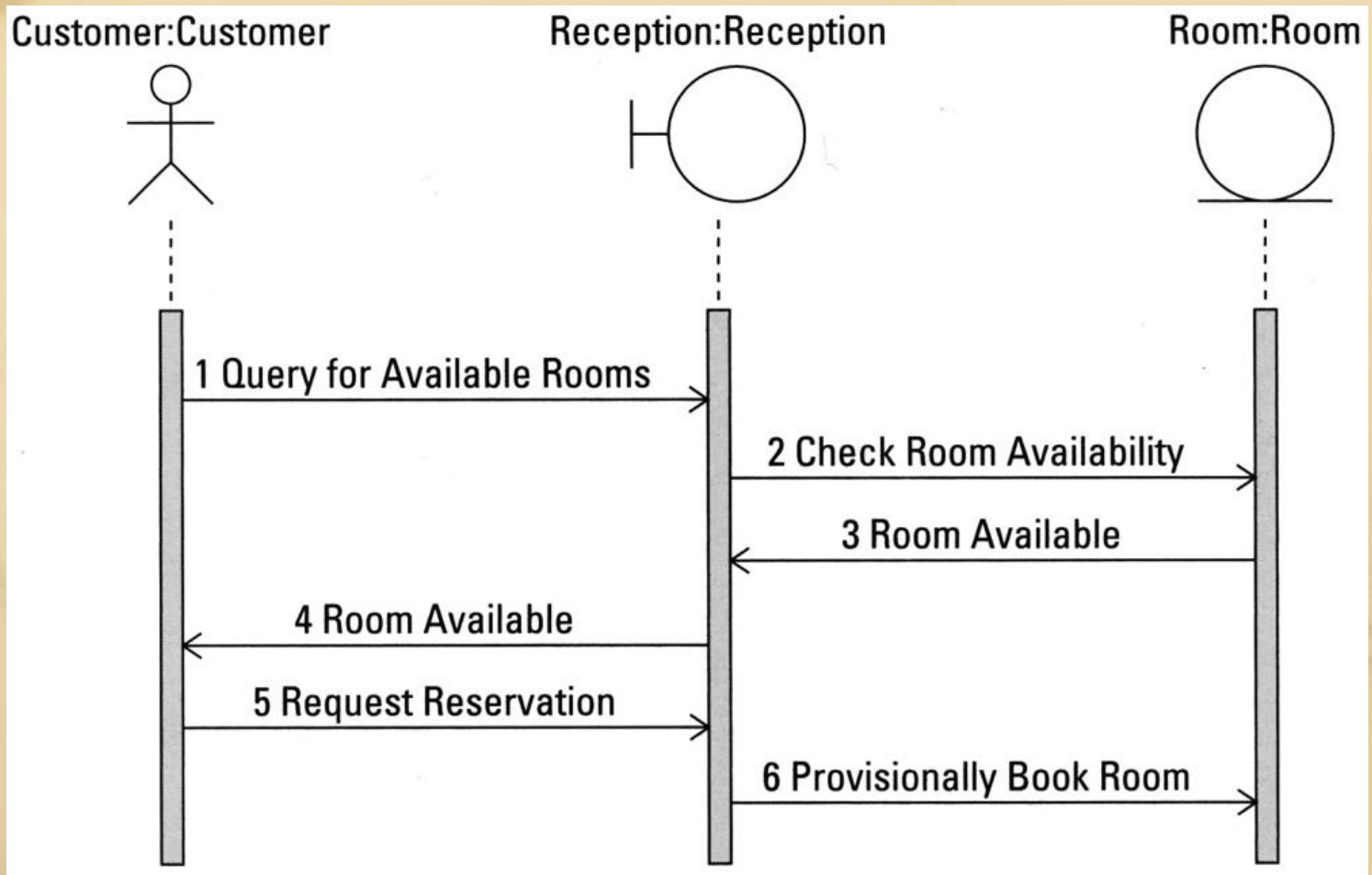


UML Profile
for Software
Development
Processes

Rozszerzenia UML



Rozszerzenia UML



Rozszerzenia UML

- Uwagi
 - problem interoperatywności powodowany odejściem od standardowej semantyki UML
 - równoważenie korzyści i kosztów definiowania i stosowania profili
 - główna motywacja dla wprowadzania profili
 - dziedzina zastosowań (systemy rozproszone, systemy wbudowane, ...)
 - środowisko projektowo-wytwórcze (Java, C#,...)

Definicja UML

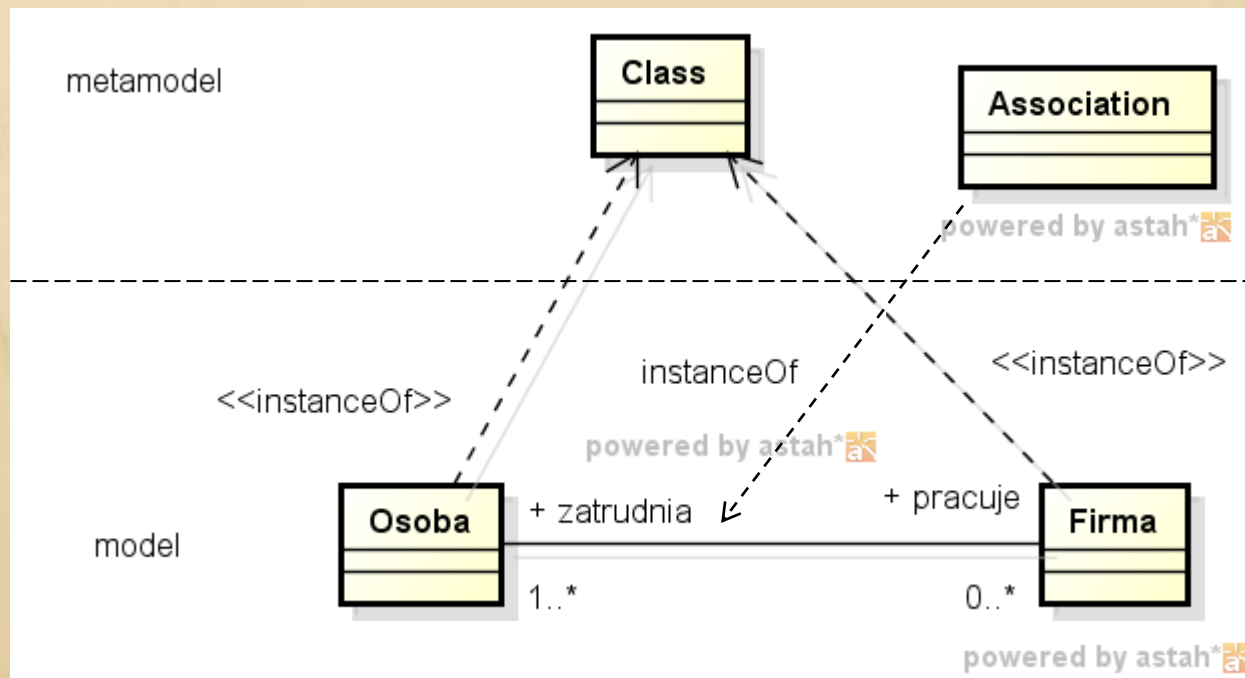
- Składnia
 - bezkontekstowa – zdefiniowana w podzbiorze UML (MOF – Meta-Object Facility)
 - kontekstowa – zdefiniowana w OCL
- Semantyka – język naturalny (plain English)
- Model w UML
 - zbiór diagramów
 - interpretacja diagramów odniesiona do:
 - dziedziny aplikacji
 - platformy implementacji

Metamodelowanie

- Model – wyrażenie, zestaw diagramów zapisanych w danym języku
- Metamodel – model definiujący język, w którym jest wyrażony model
- Model jest instancją metamodelu
- Model składa się z elementów modelowania; metamodel – z metaelementów modelowania

Metamodelowanie

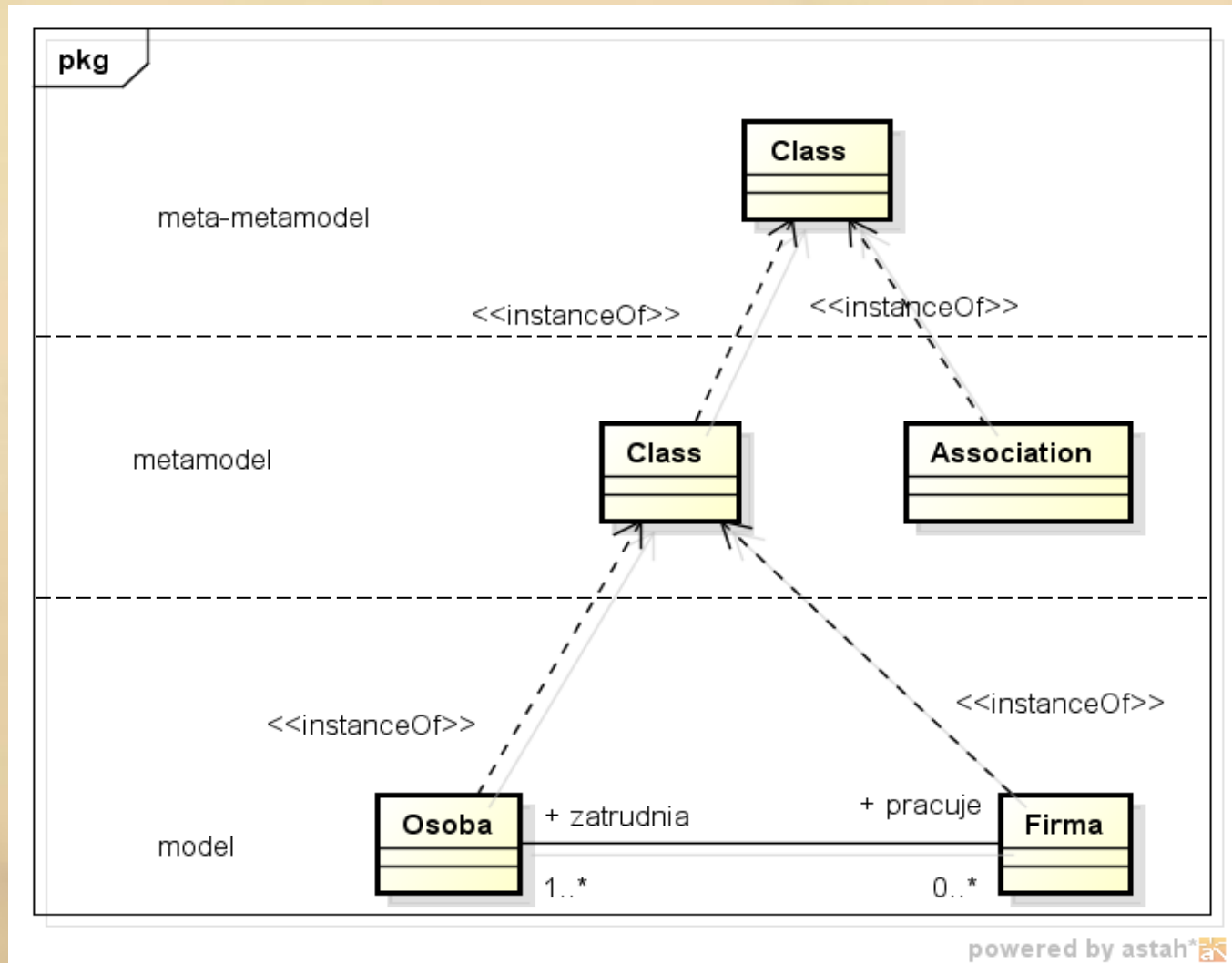
- Przykład



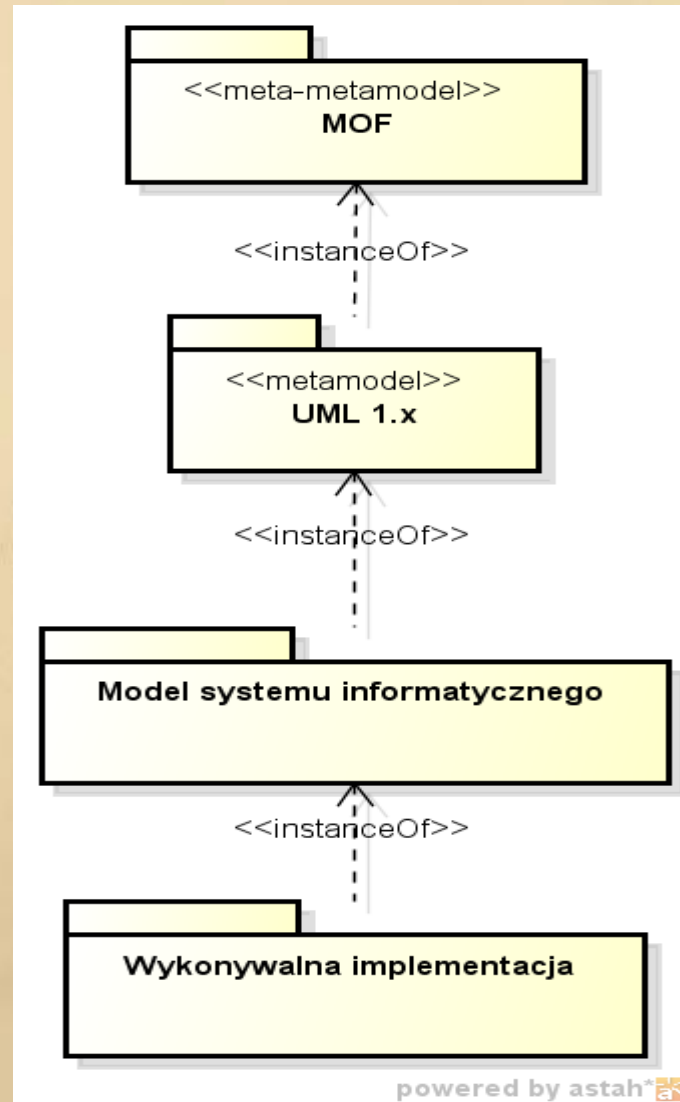
Metamodelowanie

- Jak definiować metamodel?
- MOF (Meta Object Facility) – standard OMG definiujący język do definiowania języków modelowania – poziom meta-metamodelu
- Inne języki modelowania oparte na MOF
 - UML
 - SysML (System Modeling Language)
 - SPEM (Software Process Engineering Metamodel)
 - CWM (Common Warehouse Metamodel)

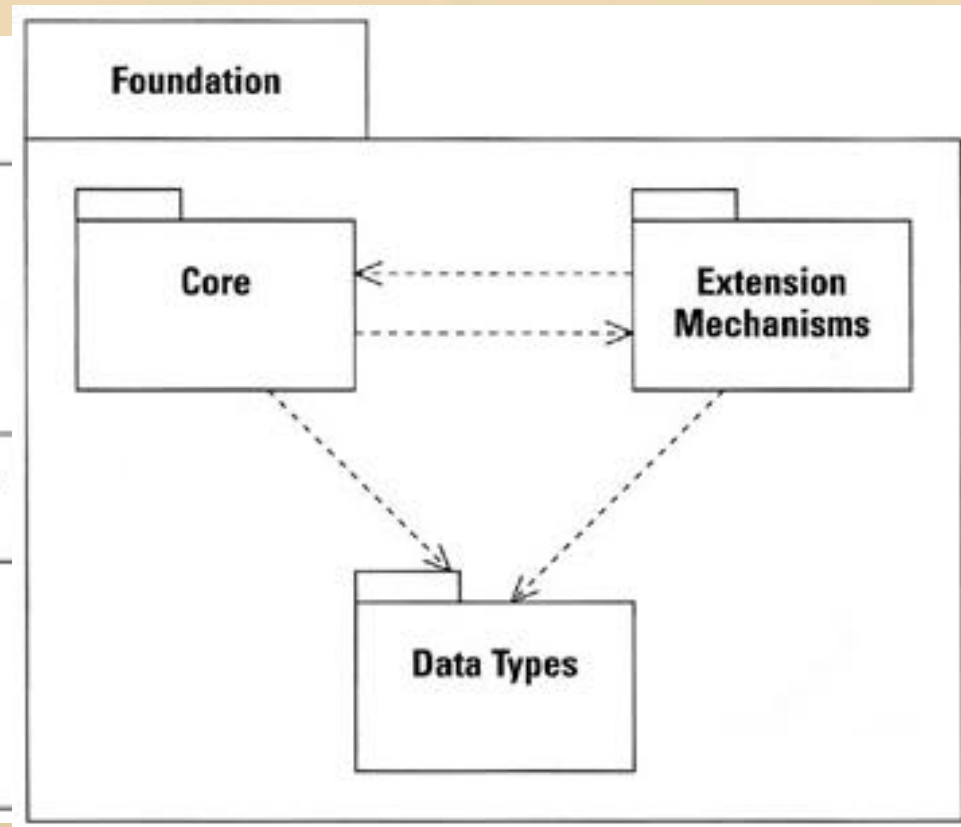
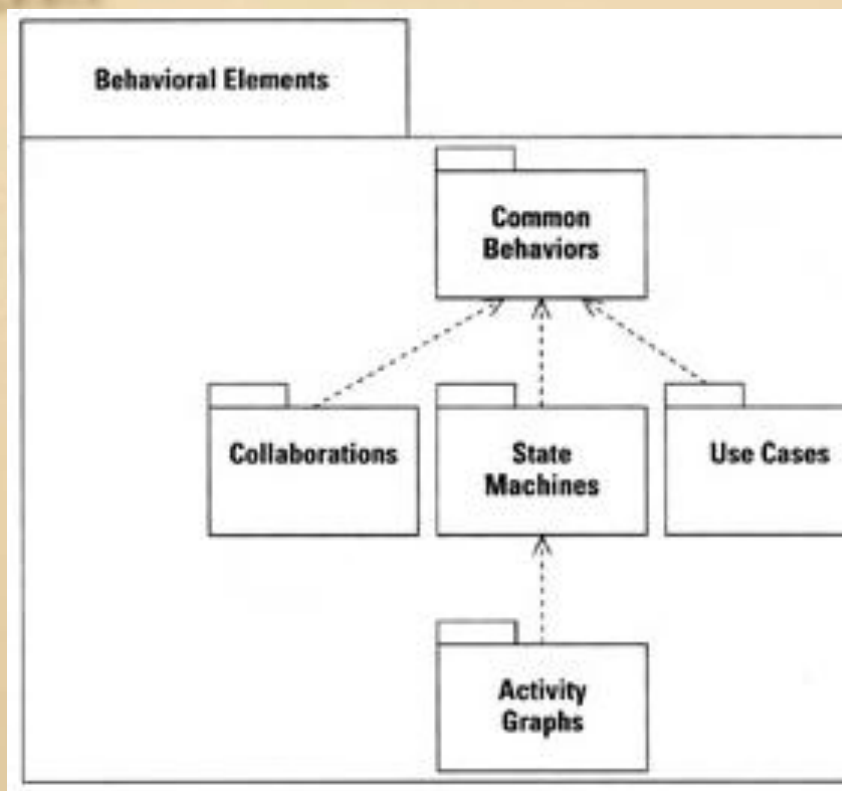
Meta-metamodelowanie



Czteropoziomowa architektura – UML 1.x



Metamodel UML 1.x



Metamodel UML 2.x

- MDA (Model Driven Architecture) – podejście do wytwarzania oprogramowania oparte na modelowaniu:

CIM → PIM → PSM → Code

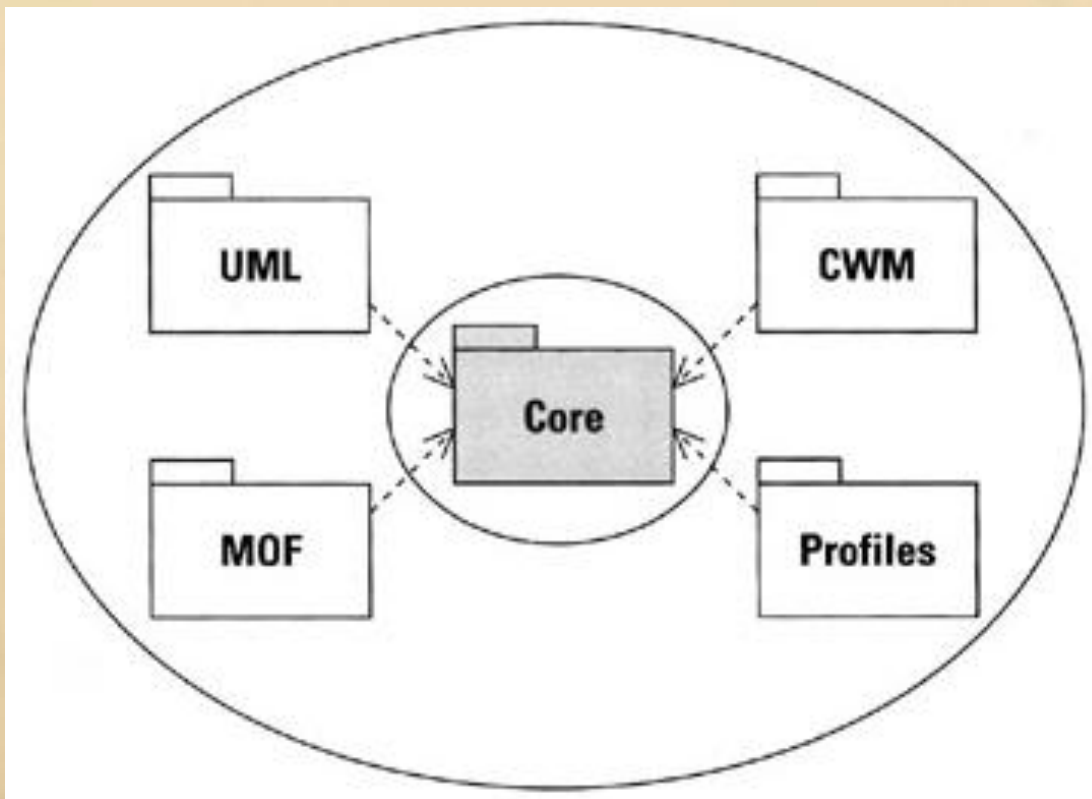
- Dwie specyfikacje
 - OMG Unified Modeling Language (OMG UML), Infrastructure
 - UML 2.0 Superstructure Specification
- Infrastructure - metamodel dla MDA

UML Infrastructure

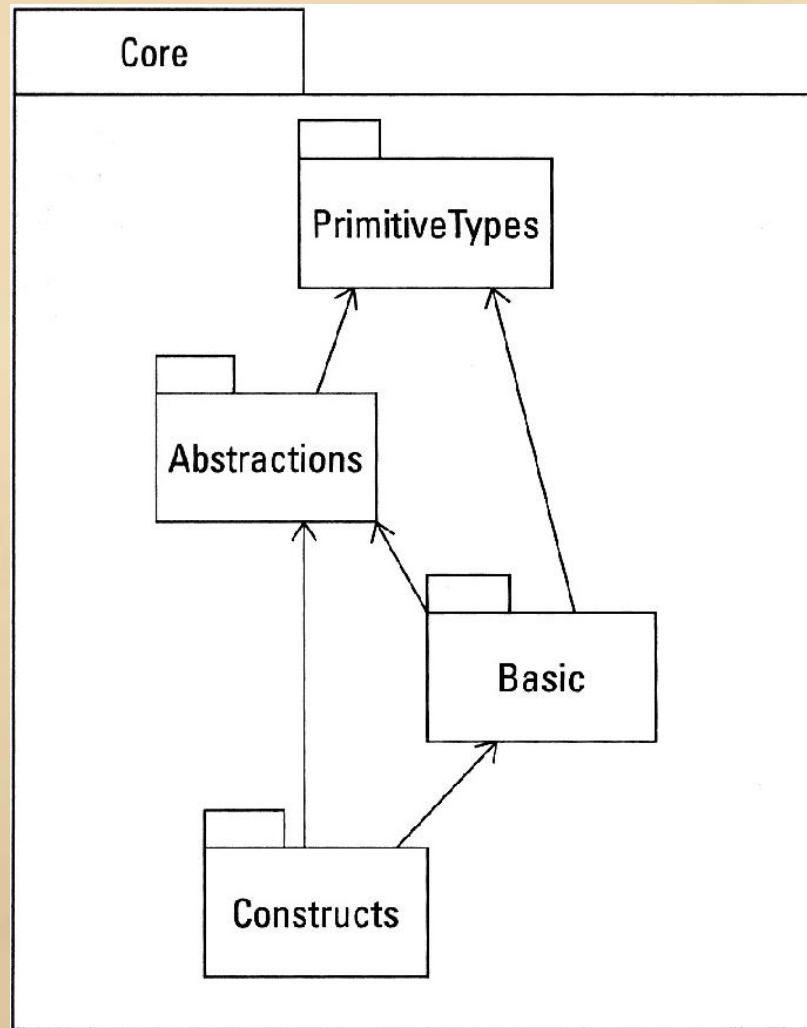
- Infrastructure – poziom najwyższy modelowania
- Oparte na Infrastructure:
 - MOF
 - UML
 - CWM (Common Warehouse Metamodel)
- UML jest nadal również wyprowadzany z MOF

UML Infrastructure

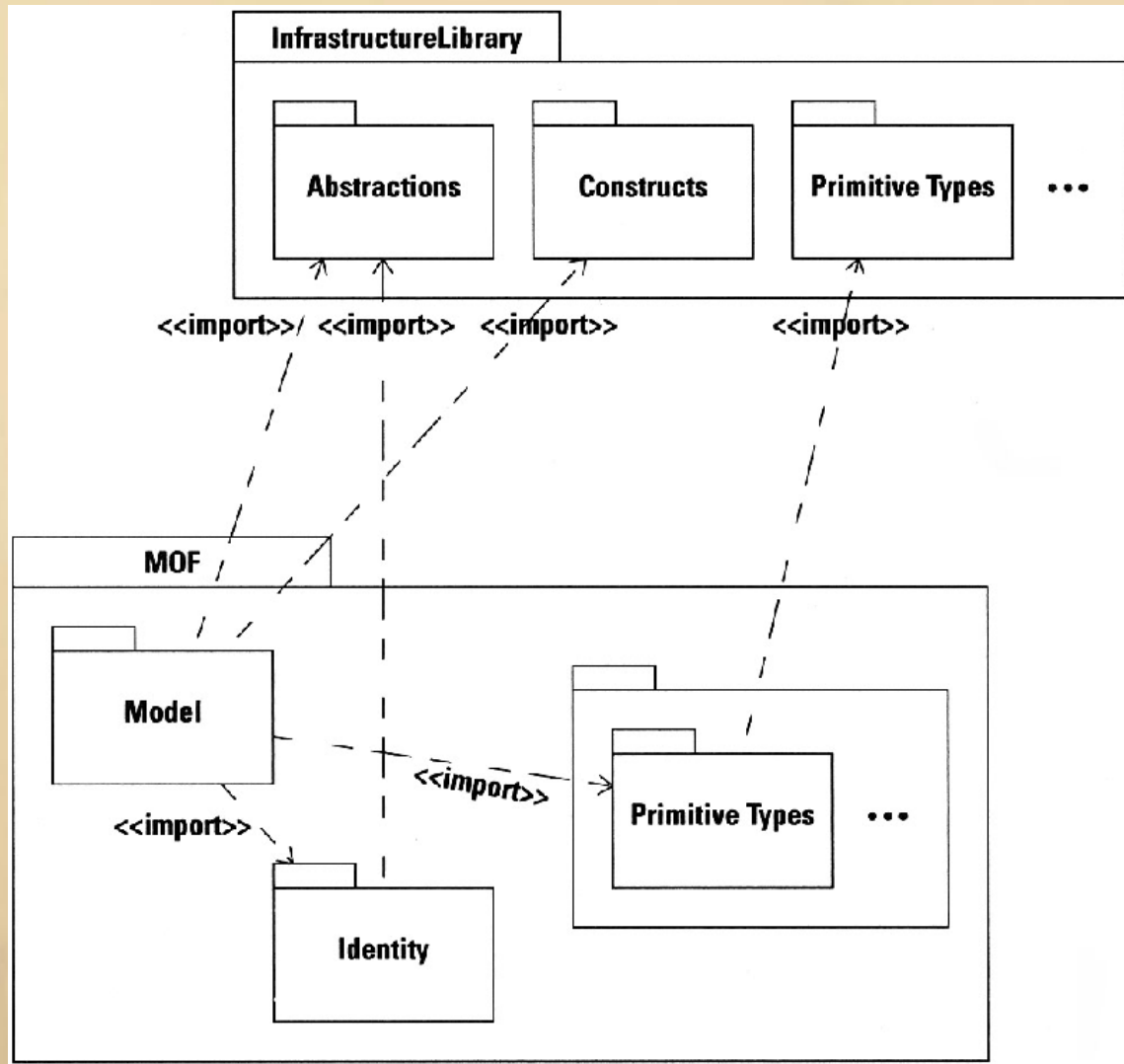
- Definicja jądra metamodelu dla MDA



UML Infrastructure

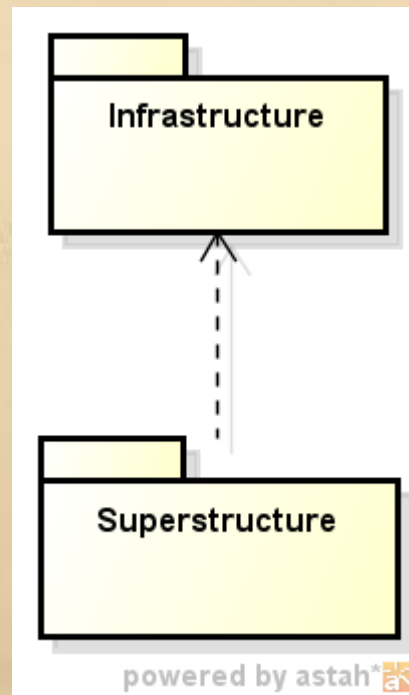


UML Infrastructure

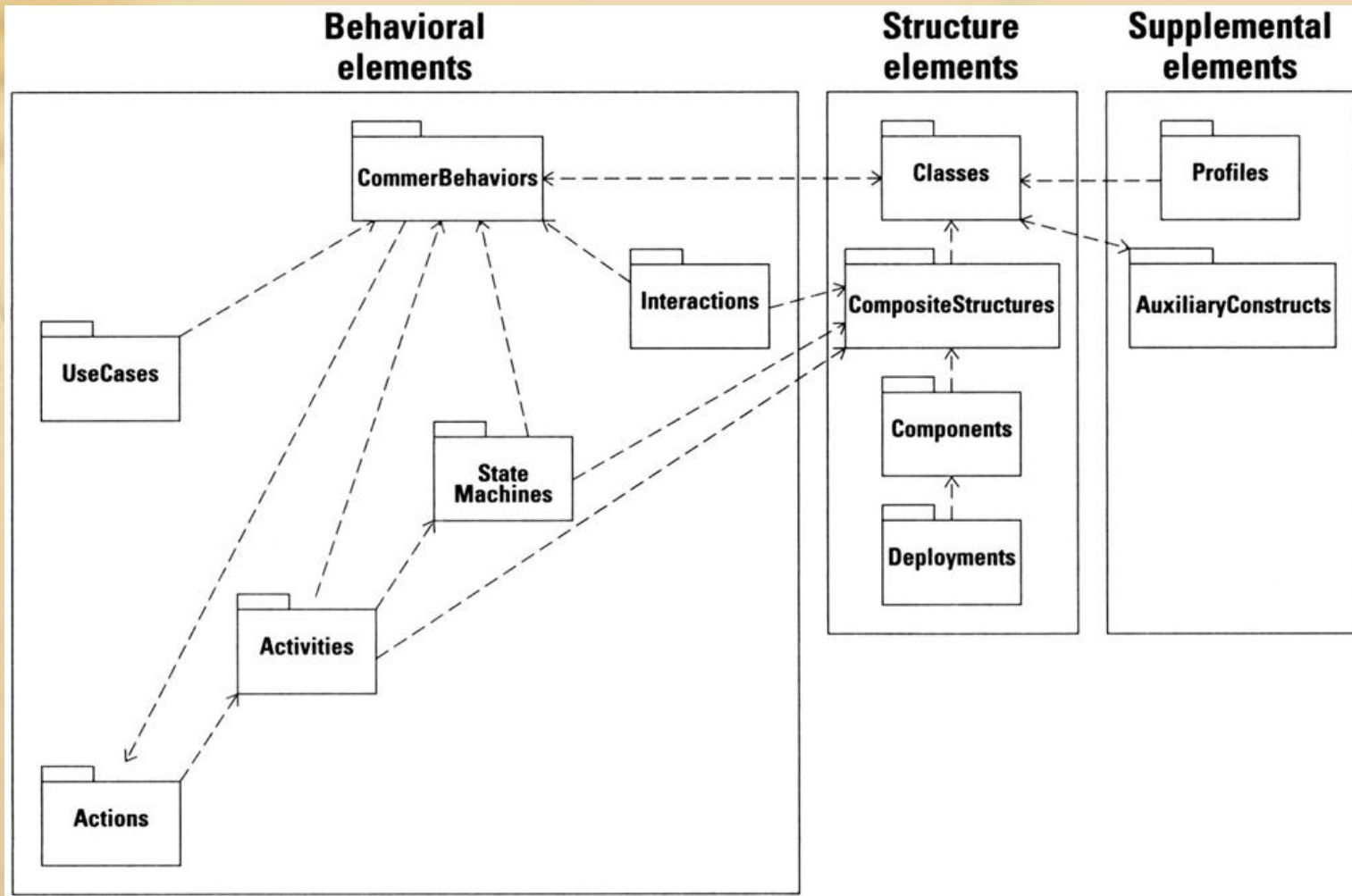


UML Superstructure

- Rozszerza i dopasowuje pojęcia Infrastructure do definicji metamodelu UML

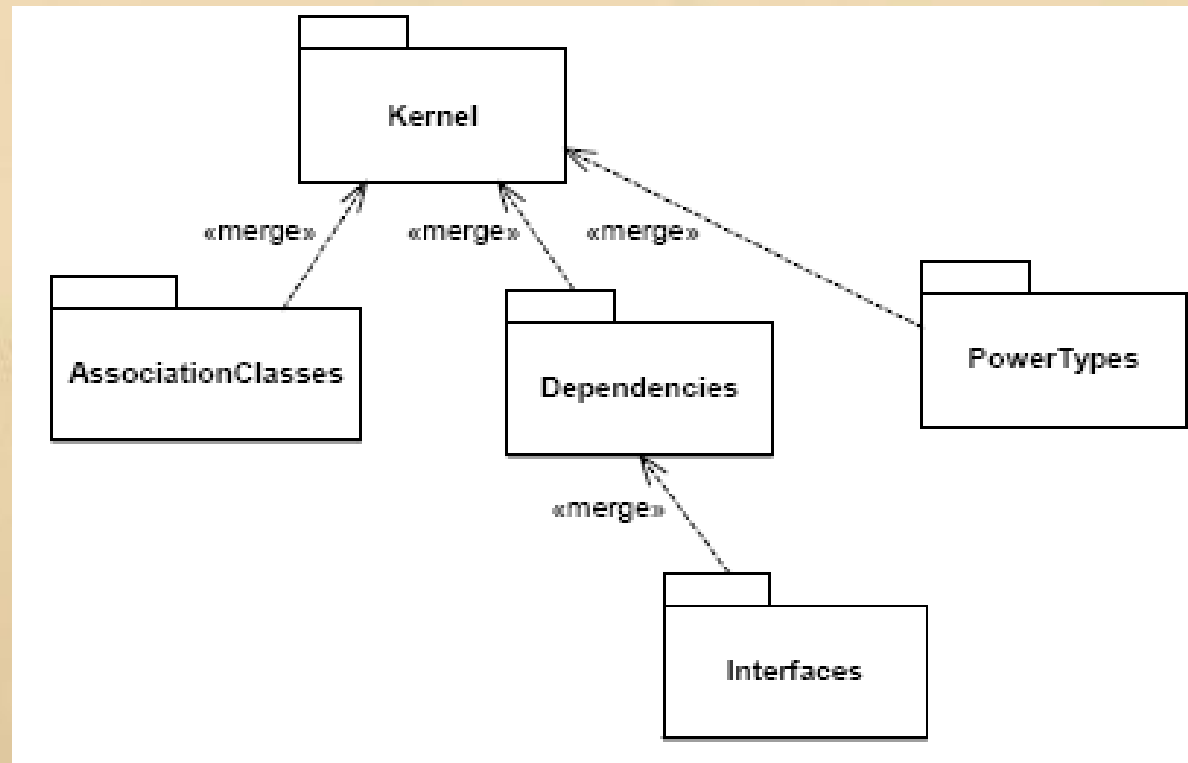


UML Superstructure



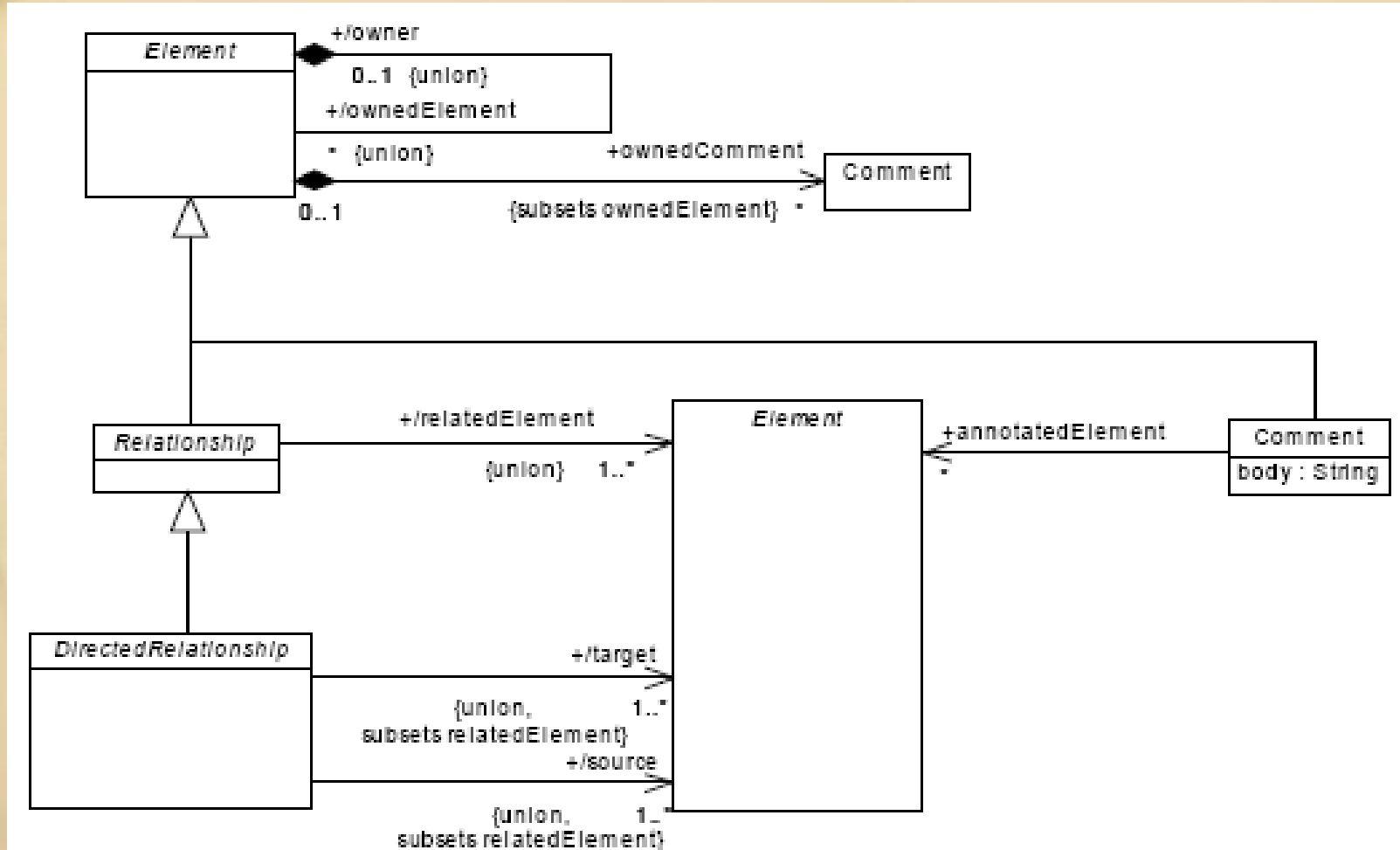
UML Superstructure

Zawartość
podpakietu
Classes

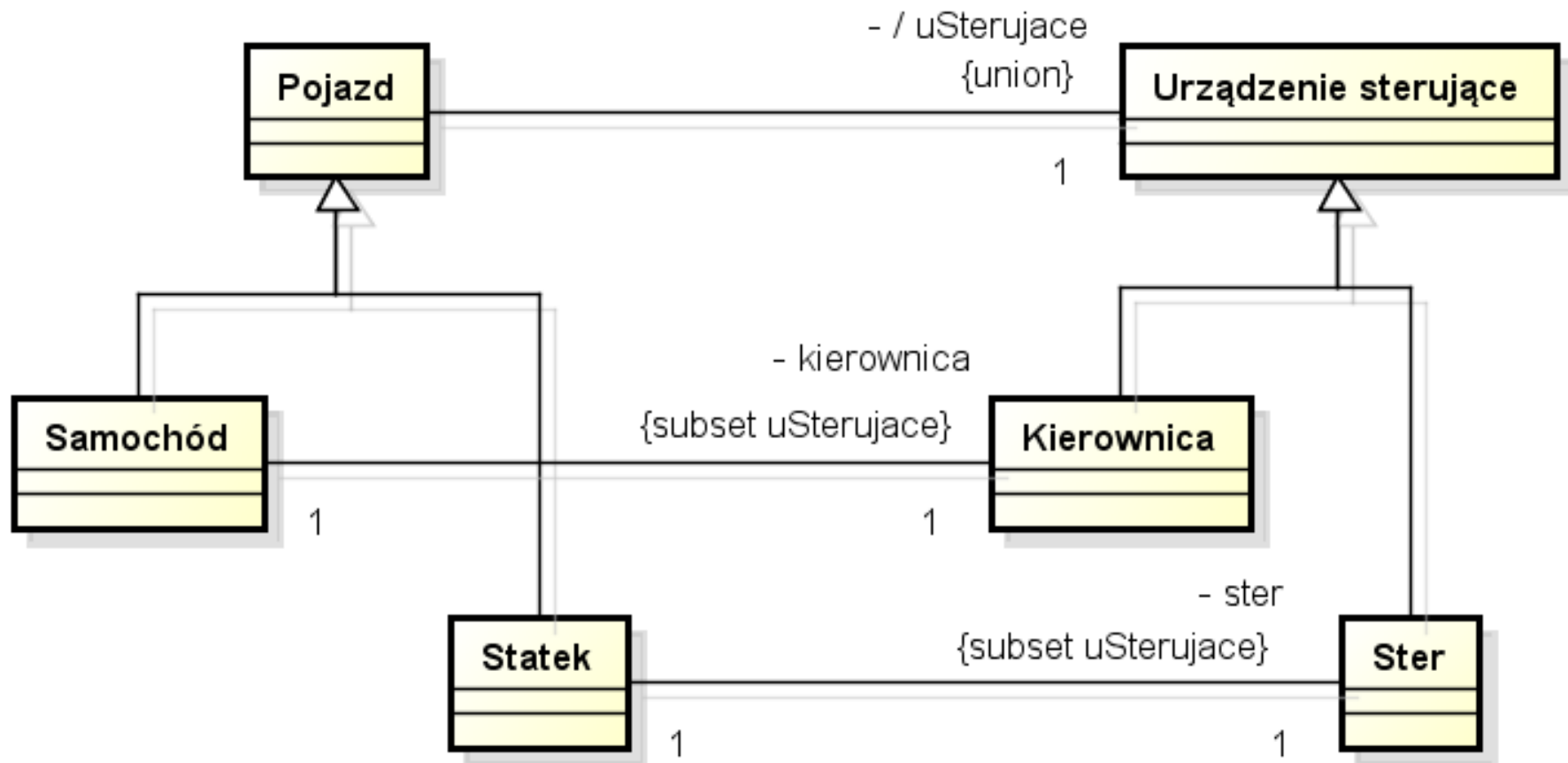


UML 2.0 Infrastructure

Główny
diagram
podpakietu
Kernel



Własności asocjacji union-subset



powered by astah*

UML 2.0 Infrastructure

- Opis Elementu

7.2.3 Element (from Kernel)

An element is a constituent of a model. As such, it has the capability of owning other elements.

Description

Element is an abstract metaclass with no superclass. It is used as the common superclass for all metaclasses in the infrastructure library. Element has a derived composition association to itself to support the general capability for elements to own other elements.

Attributes

No additional attributes.

UML Infrastructure

- Opis Elementu

Associations

- `ownedComment: Comment[*]` The Comments owned by this element. Subsets *Element::ownedElement*.
- `/ ownedElement: Element[*]` The Elements owned by this element. This is a derived union.
- `/ owner: Element [0..1]` The Element that owns this element. This is a derived union.

UML Infrastructure

- Opis Elementu

Constraints

- [1] An element may not directly or indirectly own itself.
`not self.allOwnedElements()->includes(self)`
- [2] Elements that must be owned must have an owner.
`self.mustBeOwned() implies owner->notEmpty()`

UML 2.0 Infrastructure

- Opis Elementu

Additional Operations

- [1] The query `allOwnedElements()` gives all of the direct and indirect owned elements of an element.

```
Element::allOwnedElements(): Set(Element);  
allOwnedElements = ownedElement->union(ownedElement->collect(e | e.allOwnedElements()))
```

- [2] The query `mustBeOwned()` indicates whether elements of this type must have an owner. Subclasses of `Element` that do not require an owner must override this operation.

```
Element::mustBeOwned() : Boolean;  
mustBeOwned = true
```

UML Infrastructure

- Opis Elementu

Semantics

Subclasses of Element provide semantics appropriate to the concept they represent. The comments for an Element add no semantics but may represent information useful to the reader of the model.

Notation

There is no general notation for an Element. The specific subclasses of Element define their own notation.

UML Infrastructure

- Maszyny stanów

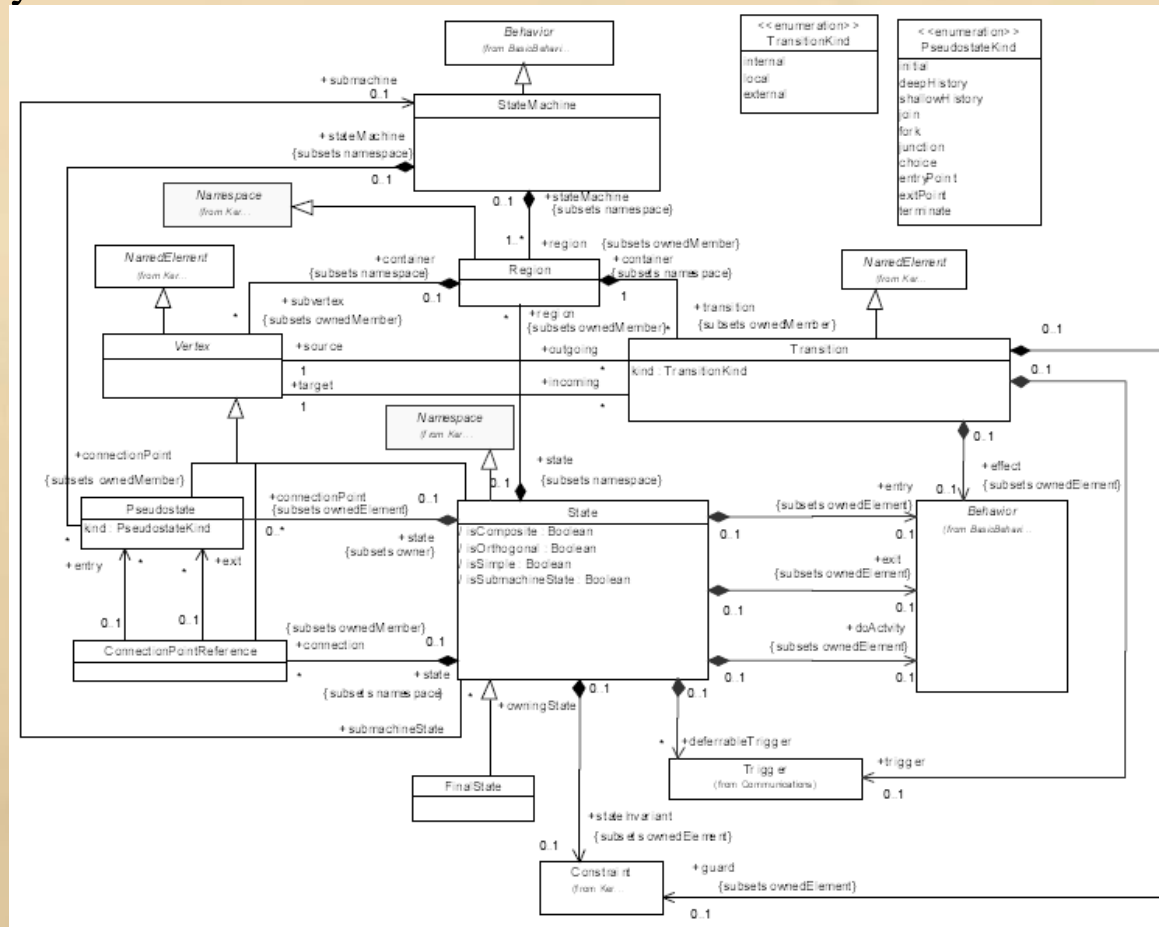
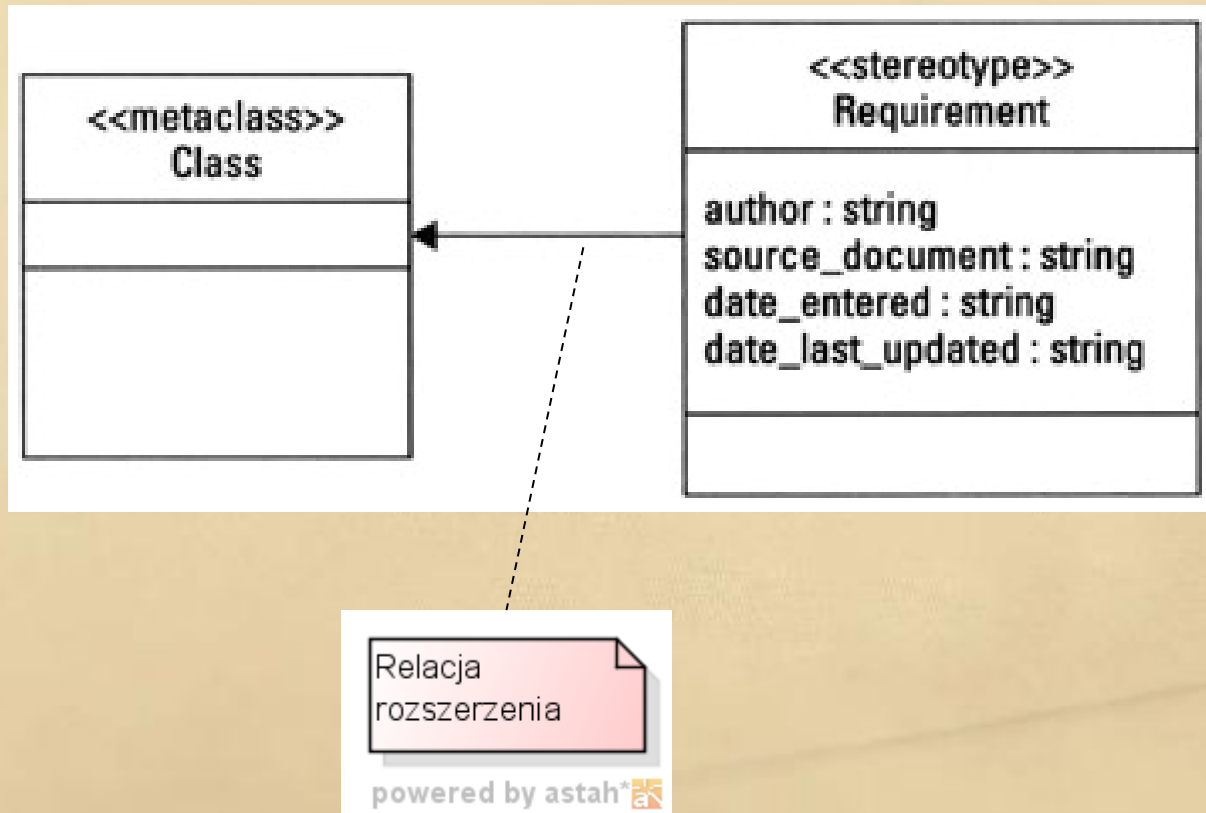


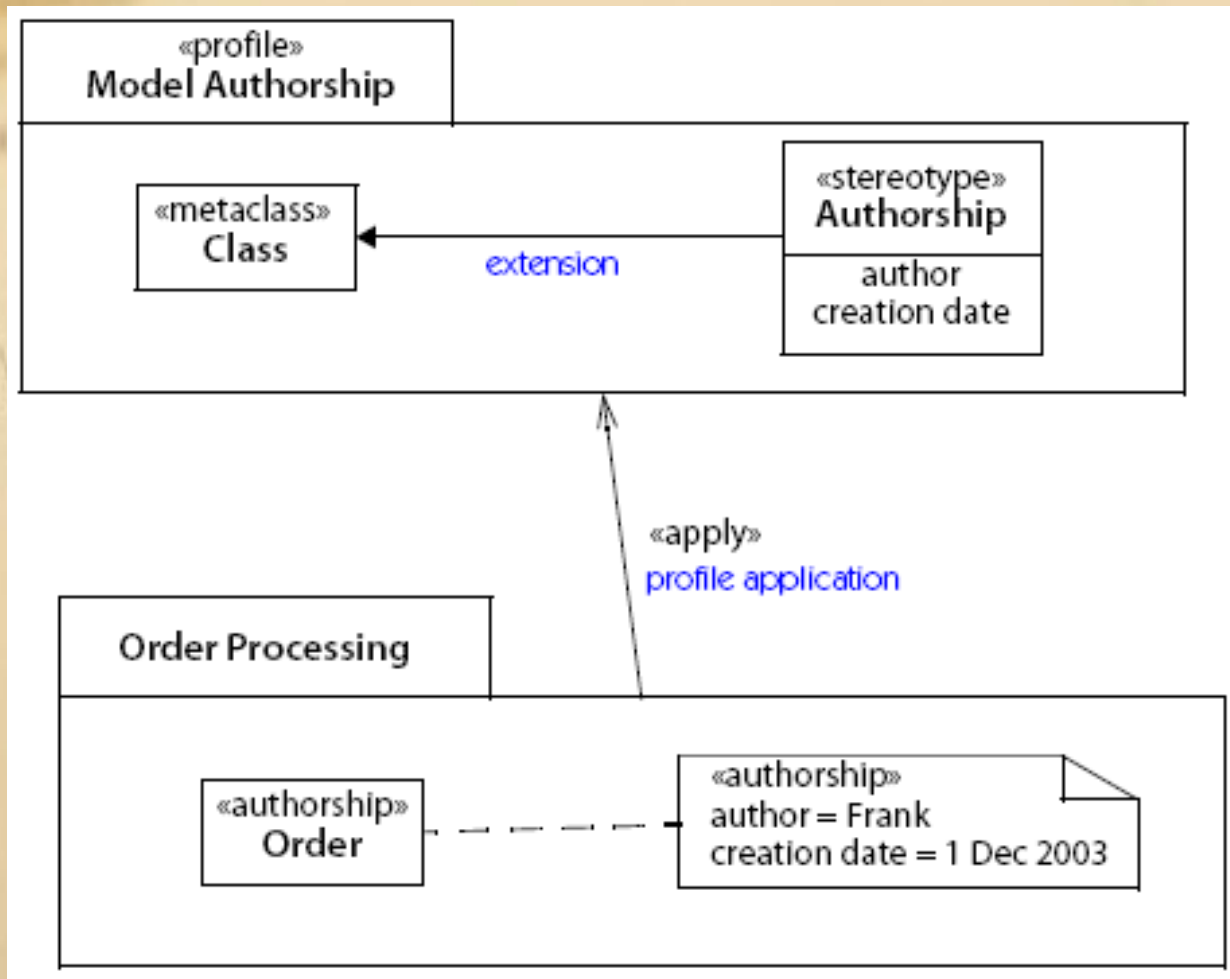
Diagram profilu

- Rozszerzenia UML na poziomie metamodelu
 - przez wprowadzenie nowej metaklasy
 - przez modyfikację metaklasy
- Diagram profilu – zbiór stereotypów
- Definicja stereotypu
 - wskazanie elementu metamodelu
 - ustalenie nazwy (opcjonalnie ikony)
 - dołączenie metek i nałożenie ograniczeń
 - opis semantyki w języku naturalnym

Definicja stereotypu – przykład

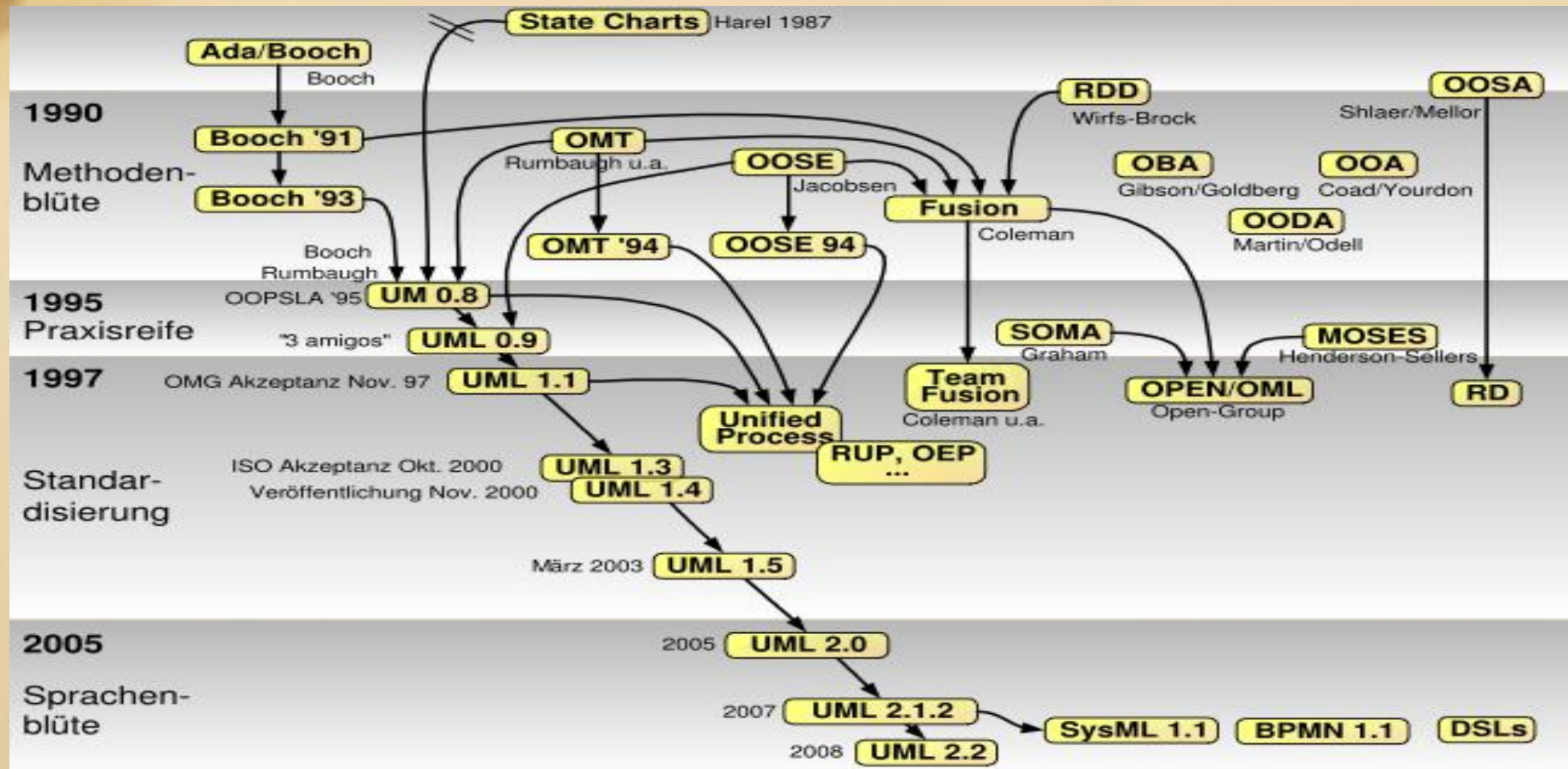


Przykład profilu – zastosowanie



Source:
Rumbaugh J., Jacobson I., Booch G.,
*The Unified Modeling Language –
Reference Manual*. Second edition,
Addison-Wesley, 2005

UML – historia

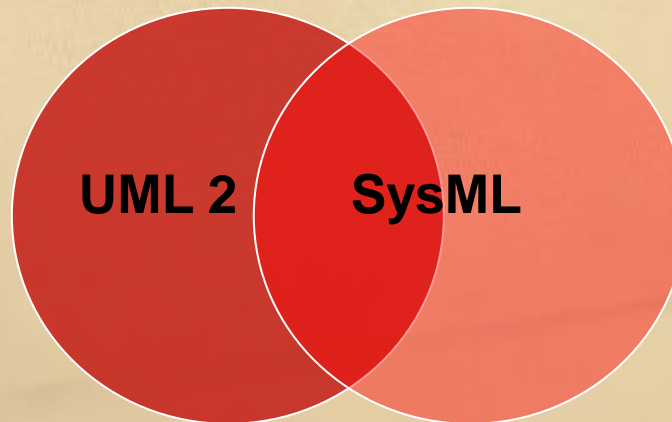


Inne języki modelowania

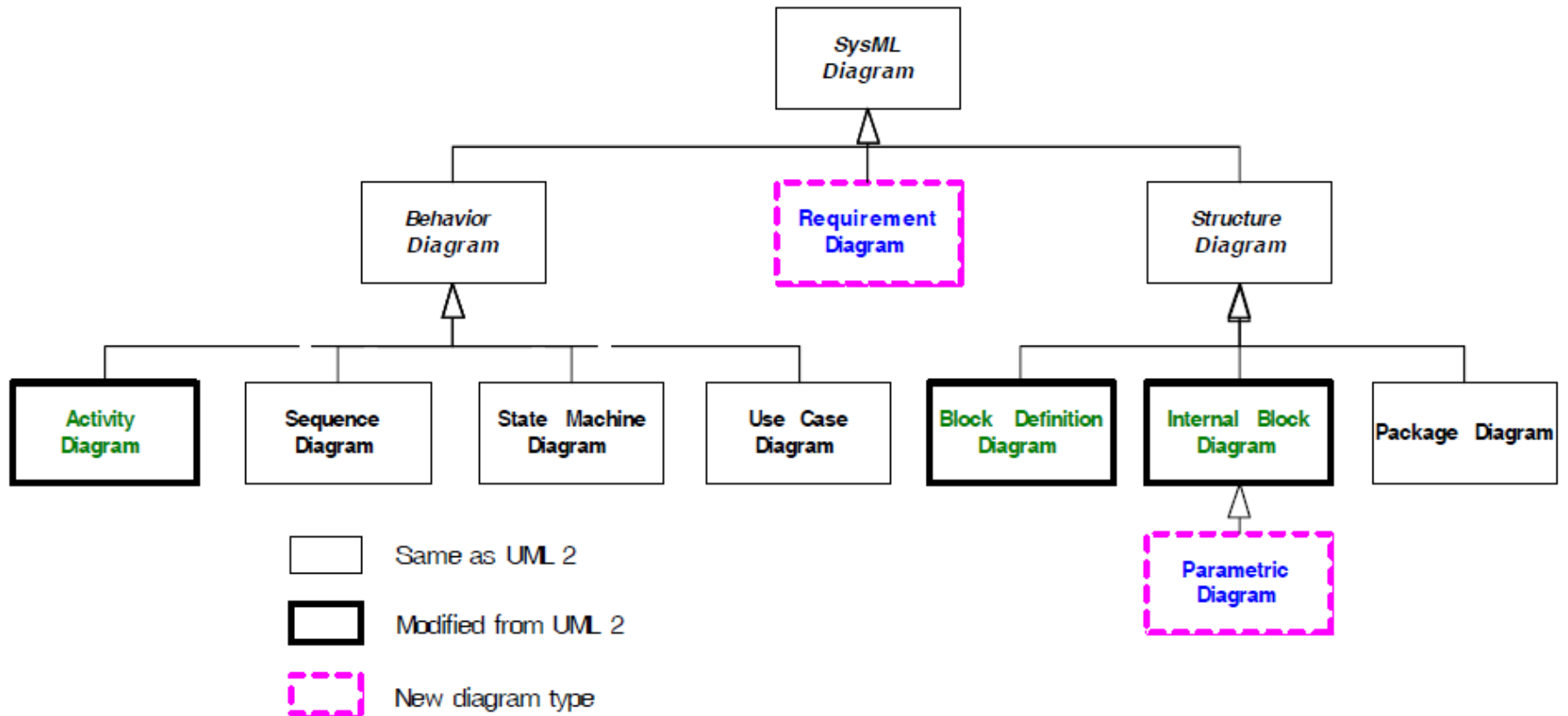
- SysML (Systems Modeling Language)
- BPMN (Business Process Modeling Notation)
- DSLs (Domain Specific Languages)

SysML

- Rozwinięcie profilu
UML Profile for Schedulability, Performance, and
Time Specification
- Porównanie zbiorów elementów modelowania



SysML – diagramy



SysML – diagram wymagań

- Wymagania mogą specyfikować
 - funkcje, które system ma realizować
 - wymagania wydajnościowe, które system ma spełniać
- Mogą odnosić się wskazanych elementów z różnych faz wytwarzania oprogramowania
- Typ elementu może być sprawdzany pod kątem spełnialności wymagań
- Mogą mieć charakter ilościowy

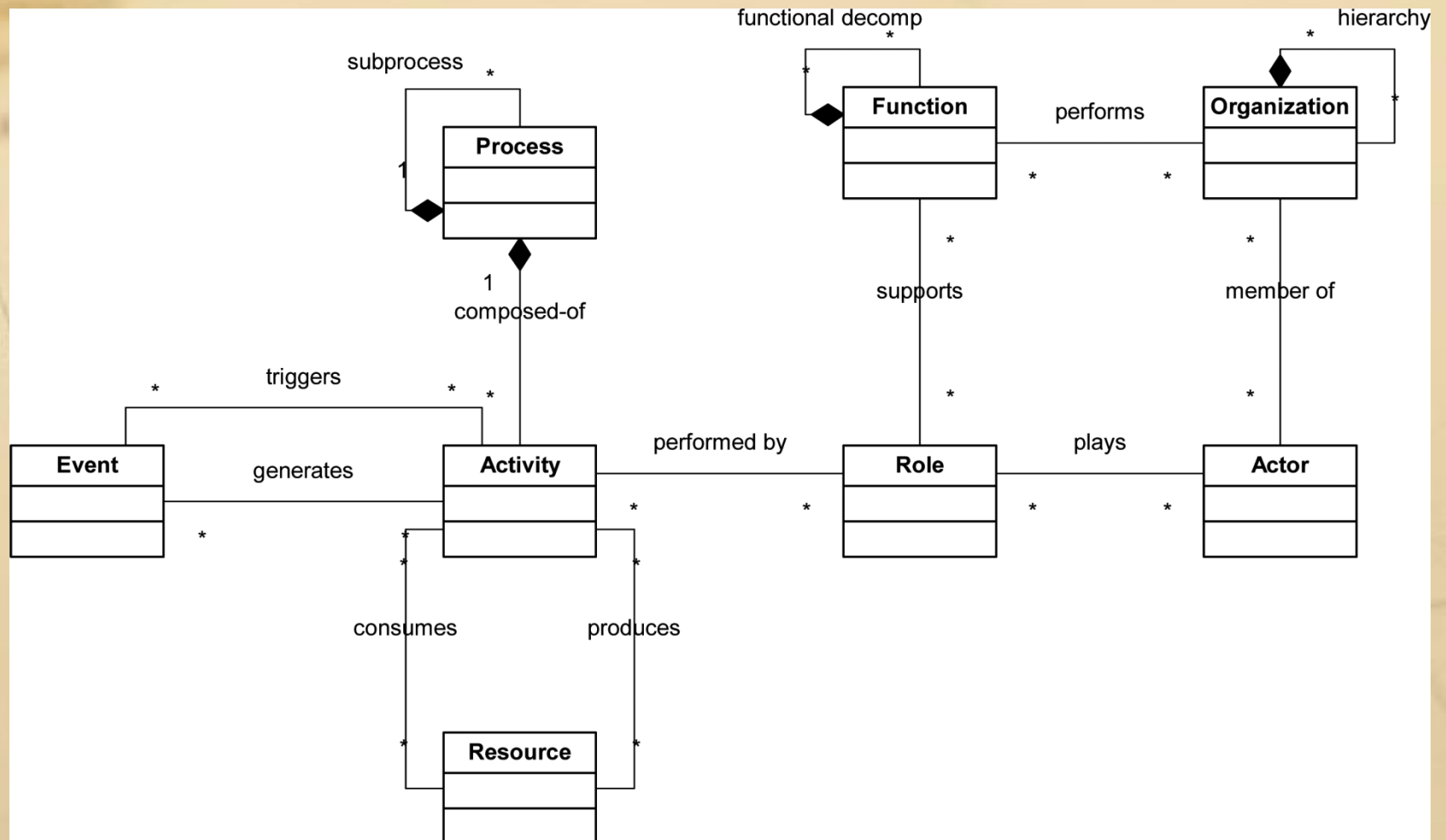
SysML – niedoskonałości

- Brak kompleksowych metod weryfikacji i walidacji
- Brak wsparcia dla testowania
- Definiowanie zachowania wyłącznie w terminach operacyjnych

BPMN

- Zastosowanie – modelowanie dziedzinowe
- Notacja graficzna specyfikująca procesy dziedzinowe (biznesowe)
- Notacja zrozumiała dla analityków dziedzinowych
- Notacja wzorowana na diagramach aktywności UML
- Proces biznesowy, choreografia

BPMN – metamodel procesu



BPMN

- Powiązanie z językiem BPEL (Web Services Business Process Execution Language)
- BPEL – oparty na notacji XML język definiowania akcji w obrębie procesów biznesowych z serwisami webowymi

DSL

- Języki programowania i języki modelowania
- Specyficzne dziedziny zastosowań
 - firmy ubezpieczeniowe
 - gry zręcznościowe
 - symulacja pola walki
 - usługi bankowe
 - biling
- Wykorzystanie MOF jako meta-metamodelu

Modelowanie i analiza systemów informatycznych

- Koniec wykładu 10