

# Az UML2 és a modell-vezérelt alkalmazásfejlesztés

Papp Ágnes, [agi@delfin.unideb.hu](mailto:agi@delfin.unideb.hu)

Debreceni Egyetem EFK

# Tartalom

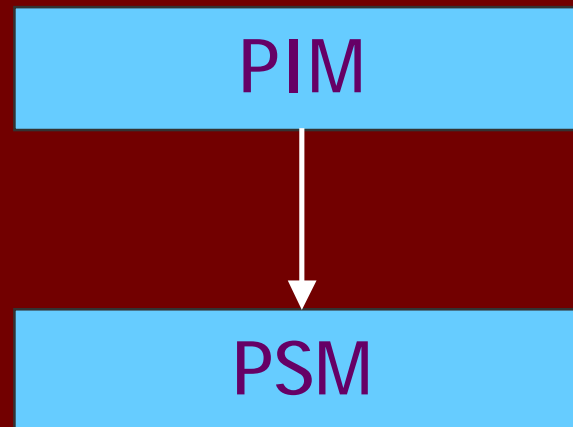
- n A Modell-vezérelt Architektúra
- n UML 1.x
- n UML 2.0

# A Modell-vezérelt Architektúra

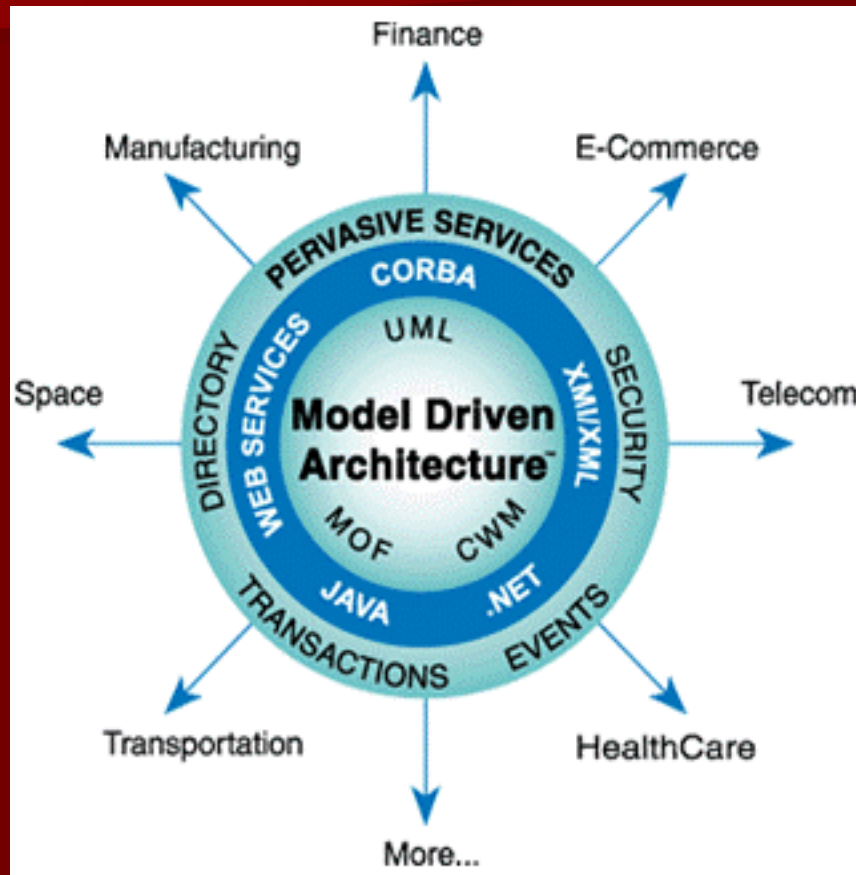
## n MDA specifikáció

- Egy platform független modell (PIM)
- Egy vagy több platform-függő modell (PSM)

## n Modelltranszformáció



# MDA együttműködése



n Szabványokkal:  
UML, MOF, XMI, CWM

n Platformokkal:  
CORBA, J2EE, .NET,  
XML/SOAP

# MDA előnyök

- n Modellezés magas szinten

- n Modelltranszformációk

  - Modellnézetek

  - Modellezési szintek

- n Modellverifikáció, tesztelés

- n Automatikus kódgenerálás

Cél: a megoldás közelítése a probléma megfogalmazásához

# Az UML 1.x

- n Modellező nyelv
- n 1997-ben vált szabvánnyá, több módosítás
- n Az UML specifikáció tartalmazza:
  - Jelölés: a szintaxis, a jelölésrendszer elemei
  - Szemantika: az egyes jelölések jelentése
- n Diagramok: követelmények, statikus struktúra, viselkedés és implementáció környezetének ábrázolása

# Az UML általános gyengeségei

- n A specifikáció nem formális
  - különböző módon implementált eszközök
  - diagramok konzisztenciájának ellenőrzése
- n Metamodellezési koncepció nem érvényesül következetesen
- n Túl sok különböző típusú diagram és elem
  - melyik kombináció a legalkalmasabb
  - a diagramok közötti kapcsolat nem definiált

# Az UML általános gyengeségei

- n Nincs lehetőség a modellek hierarchikus struktúrálására
  - nagyobb rendszerek megértése
  - komponensek és alrendszerek ugyan vannak
  - nincs szintenkénti kompozíció
- n Explicit hibakezelés hiánya



# Az UML2

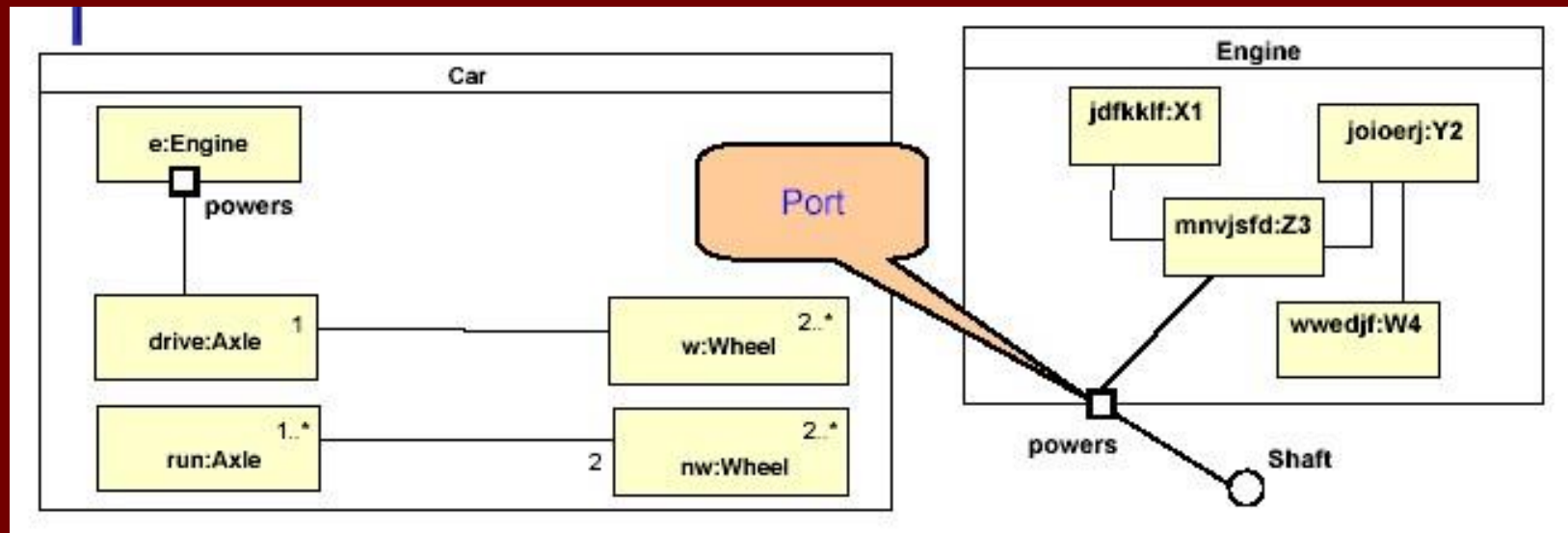
- n Az UML 2.0 véglegesítés alatt
- n Négy különböző dokumentum:
  - UML2 Infrastructure
  - UML2 Superstructure
  - UML2 Object Control Language (OCL)
  - UML2 Diagram Interchange (XMI)
- n Architektúra, skálázhatóság

# Az UML2 jellemzői

- n A kompones-alapú szoftverfejlesztés támogatása
- n Szoftver-architektúra modellezésének támogatása
- n Több lehetőség a szimulációval és kódgenerálással történő fejlesztésre
- n Végrehajtható modellek és a dinamikus viselkedés támogatása
- n Diagramok cseréjének specifikálása az eszközök között
- n Skálázhatóság
- n Kiterjesztési mechanizmus – UML Profile-ok

# A szerkezet modellezése

- n Arhitektúramodellezési koncepció támogatása
- n Class, Component, Collaboration
- n Part, Connector, Port



# Szerkezeti diagramok áttekintése

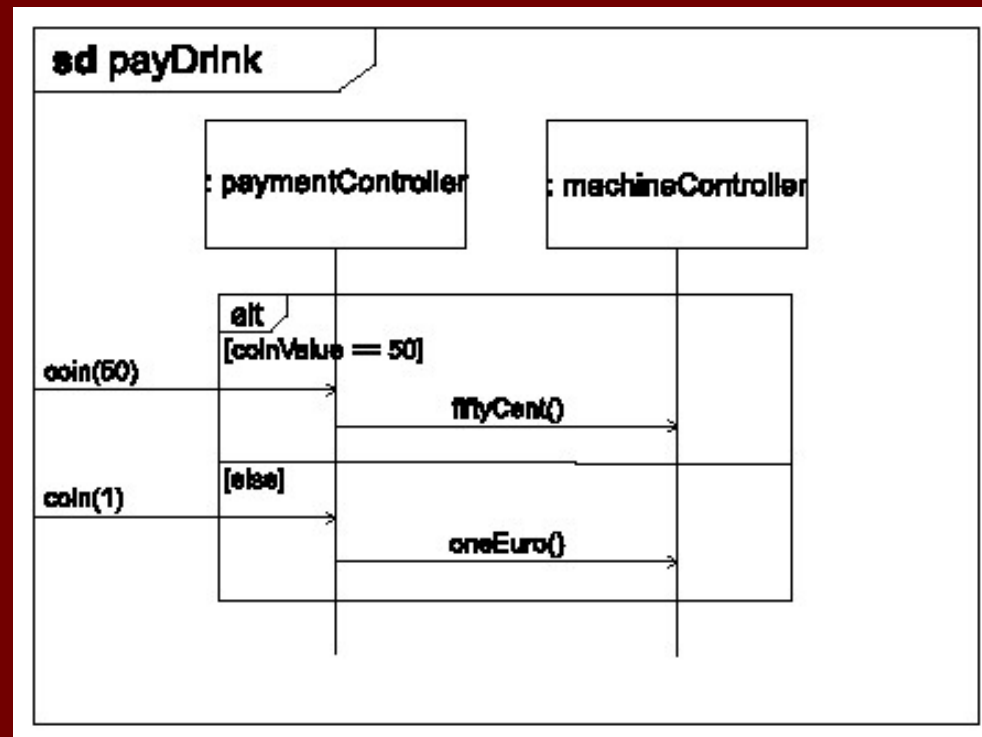
- n Osztály diagram (Class Diagram),  
Objektum diagram (Object Diagram):
- n Telepítési diagram (Deployment Diagram)
- n Csomag diagram (Package Diagram),  
Komponens diagram (Component  
Diagram)
- n Composite Structure Diagram

# A viselkedés modellezése

- n Szekvencia diagram (Sequence Diagram)
- n Kommunikáció diagram (Communication Diagram)
- n Aktivitás diagram (Activity Diagram)
- n Interakció áttekintés (Interaction Overview Diagram)
- n Állapotgép diagram (Statemachine Diagram)
- n Idő diagram (Timing Diagram)
- n Használati eset diagram (Use Case Diagram)

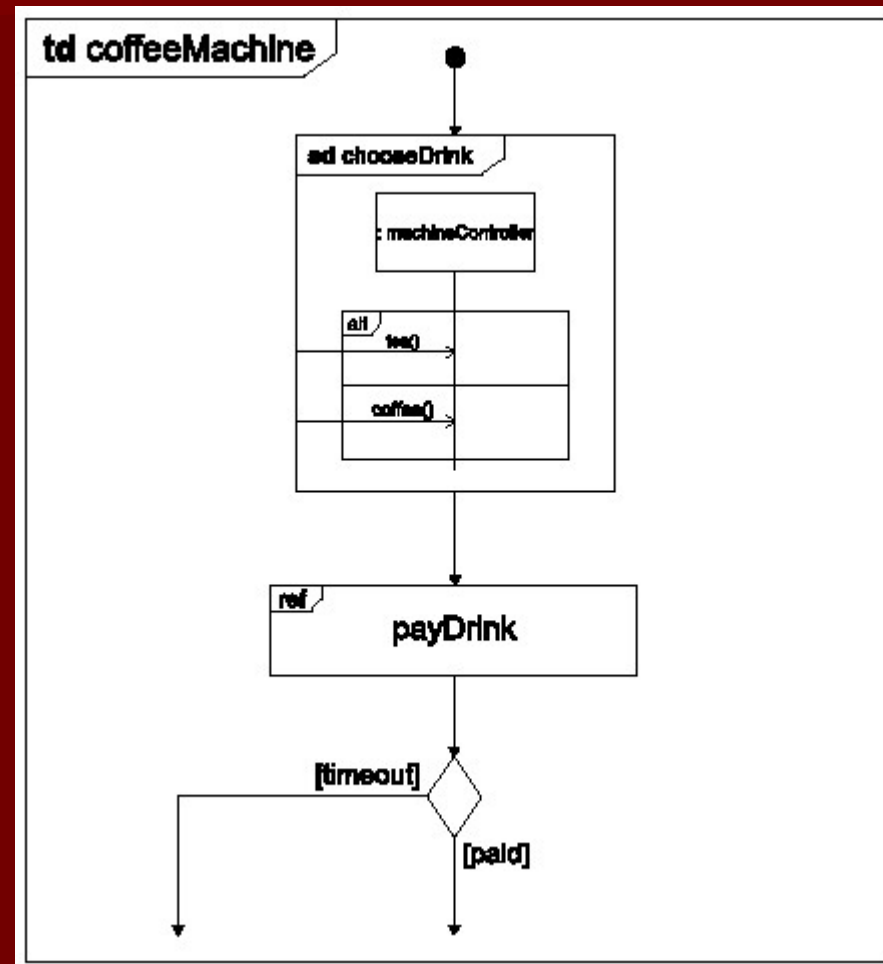
# Szekvencia diagramok

- n Interaction Fragment
- n Lifeline Decomposition



# Interakció diagramok

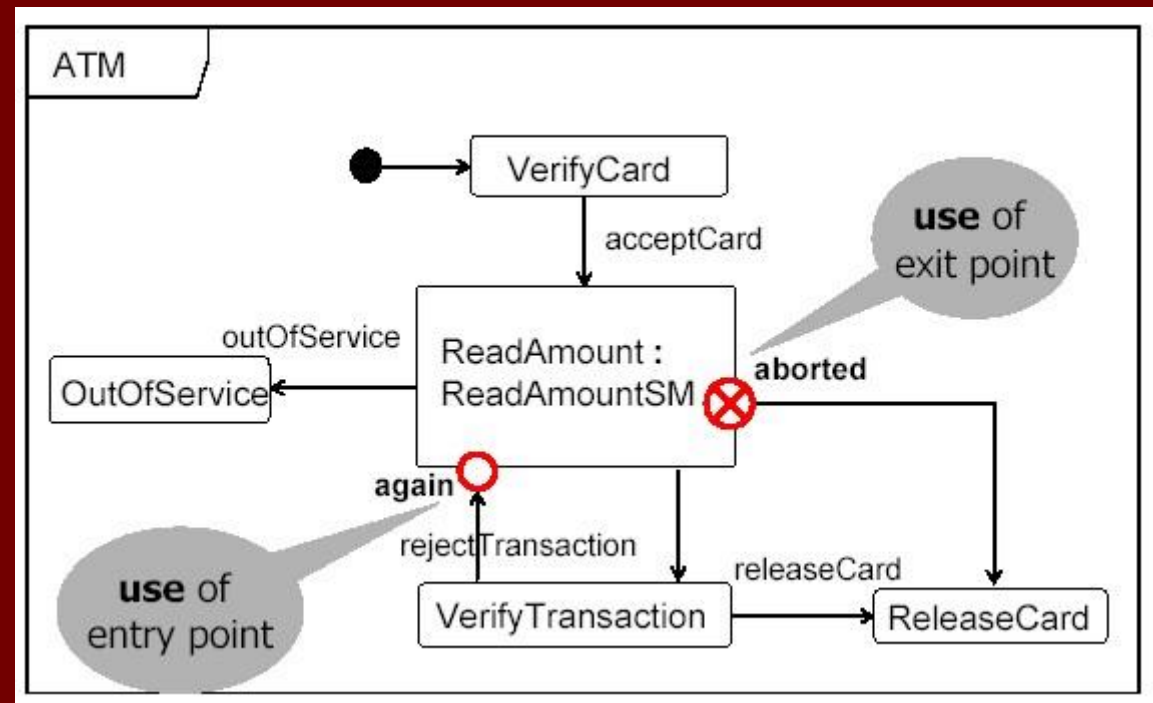
- n Aktivitás diagram
- n Interakció áttekintés diagram



# Állapotgépek

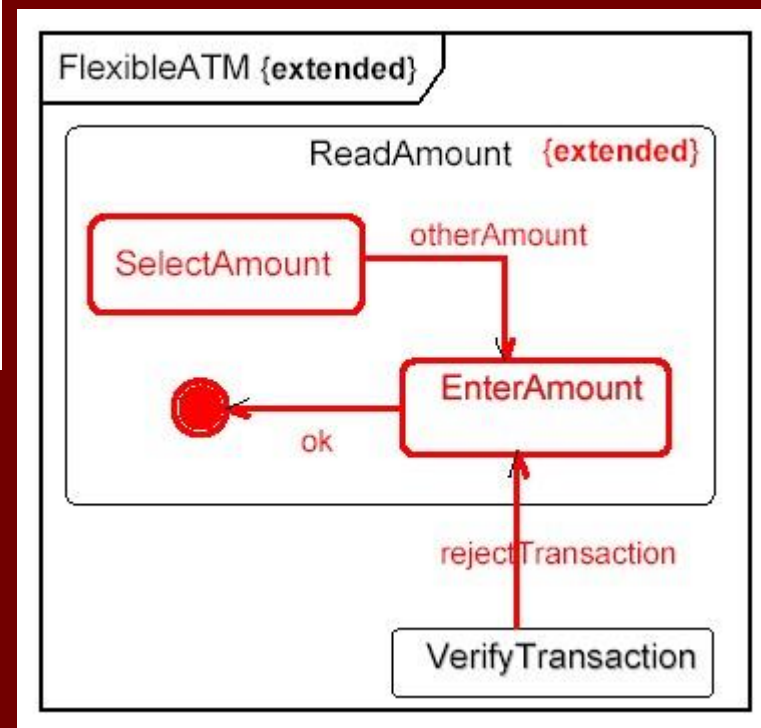
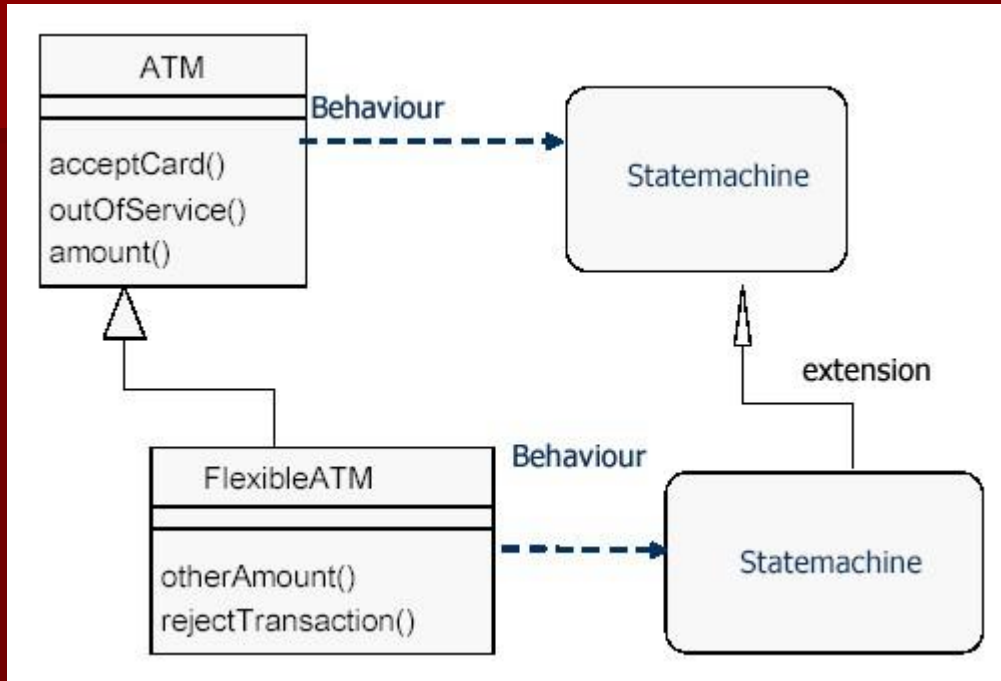
- n Viselkedési állapotgépek
  - viselkedés bezárása
  - viselkedés öröklődése
- n Protokoll állapotgépek

A viselkedés bezárása :



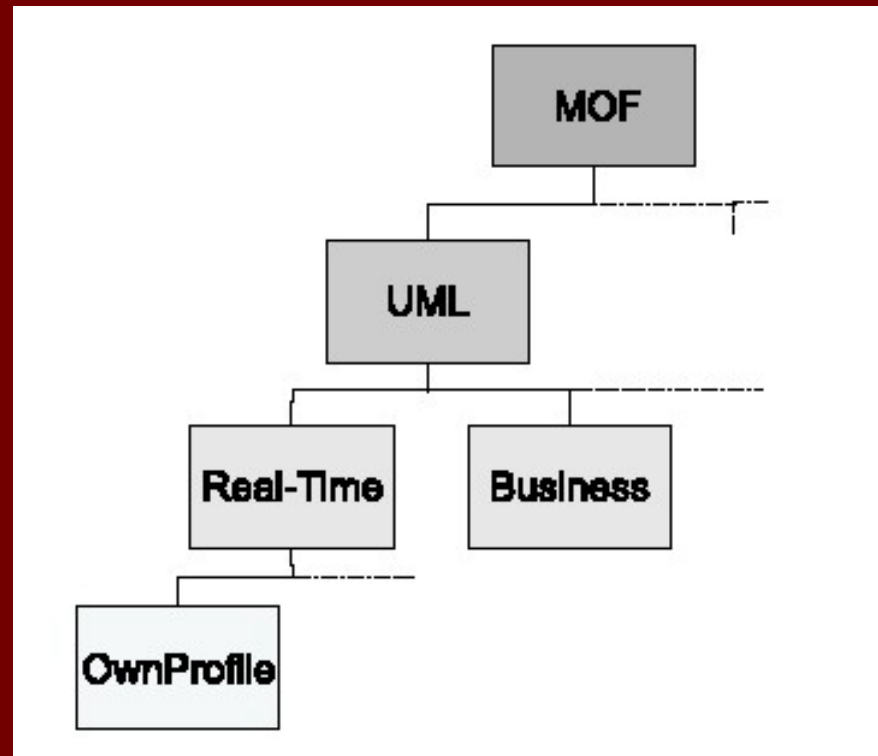


# A viselkedés öröklődése :



# UML Profile-ok

- n Kiterjesztési mechanizmust biztosítanak
- n Alkalmazási területre, platformra



# UML2 hátrányok és előnyök

- n A specifikáció nem formális
- n Túl sok diagram és elem
- n Diagramok közötti kapcsolat nem definiált
- n Modellek hierarchikus dekompozíciója
- n Komponens-alapú fejlesztés támogatása
- n Továbbfejlesztett diagramok
- n Profile-ok

# MDA - folytatódik a trend

