



Modellalapú szoftverfejlesztés

XII. előadás

Modellalapú
fejlesztések

Dr. Semeráth Oszkár

Modellalapú fejlesztések

I. Fejlesztési fogalmak

II. Kritikus rendszerek fejlesztése

III. Funkciómodellezés

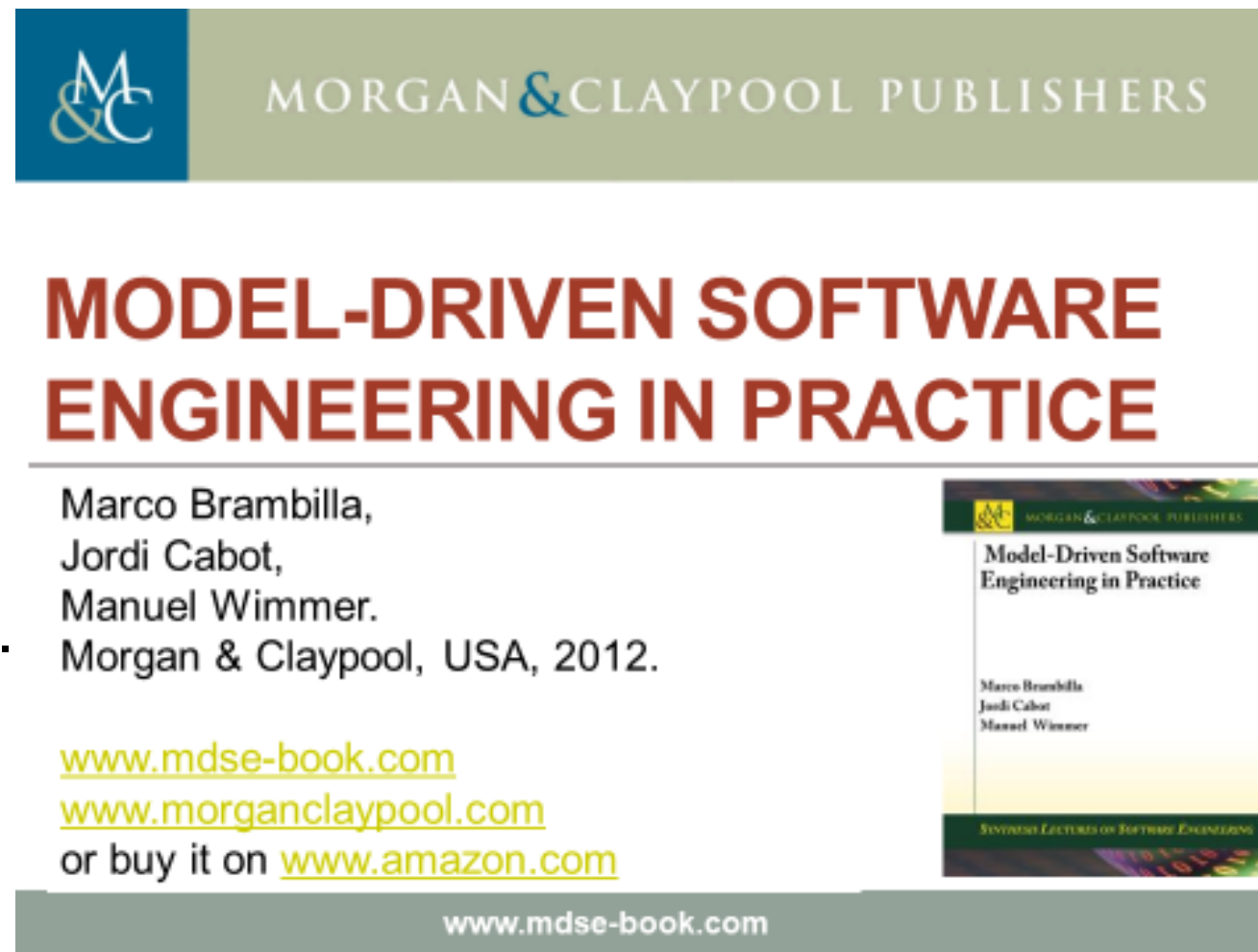
IV. Generatív programozás



Az MDSE hagyományos motivációi

Alapelvek és célkitűzések

- Melyek a modellalapú fejlesztés motivációi?
- Milyen kérdések merülnek fel?
- Ezt a rész fejezetet egy tankönyvből vettük át.
- A szerzők beleegyeztek abba, hogy az oktatásban használjuk a könyvüket.



www.mdse-book.com



Az MDSE hagyományos motivációi

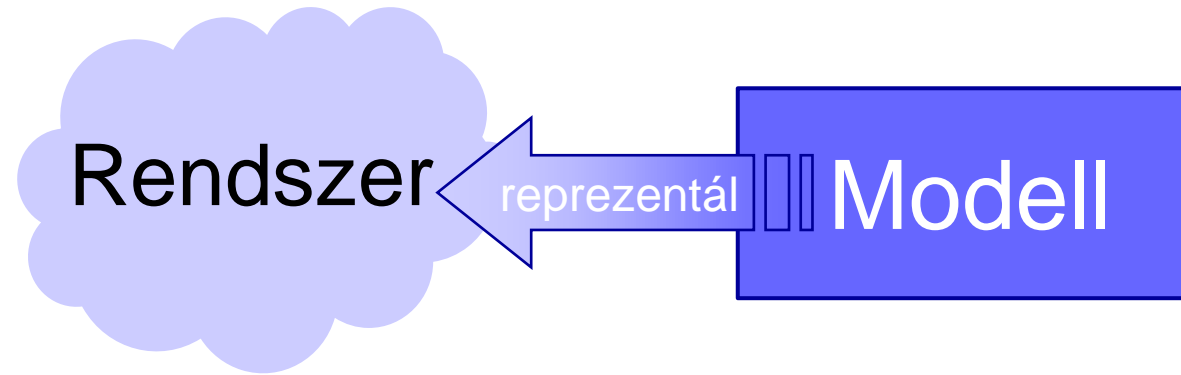
Alapelvek és célkitűzések

- **Absztrakció** a konkrét megvalósítási technológiáktól
 - Olyan modellező nyelveket igényel, amelyek nem tartalmazzák a megvalósítási technológiák specifikus fogalmait (pl. Java EJB).
 - A szoftverek jobb **hordozhatósága** új/változó technológiákhoz – model once, build everywhere
 - A különböző technológiák közötti **átjárhatóság** automatizálható (ún. Technológiai Hidak / Technology Bridges)
- **Automatizált kódgenerálás** absztrakt modellekből
 - pl. Java-API-k, XML-sémák stb. generálása UML-ből
 - Kifejező és pontos modelleket igényel
 - Fokozott **termelékenység** és **hatékonyság** (a modellek naprakészek maradnak)
- Az alkalmazás és az infrastruktúra **különálló fejlesztése**
 - Az alkalmazáskód és az infrastruktúrakód (pl. Application Framework) szétválasztása növeli az **újrafelhasználhatóságot**
 - **Rugalmas** fejlesztési ciklusok, valamint **különböző fejlesztési szerepek lehetségesek**



Modellek

Mi az a modell?



Leképezési Jellemző

A modell egy eredeti (=rendszer) alapján készül

Redukciós Jellemző

A modell az eredeti tulajdonságainak csak egy (releváns) részét tükrözi

Pragmatikus Jellemző

A modellnek használhatónak kell lennie az eredeti helyett valamilyen cél szempontjából

Célok:

- leíró célok
- előíró célok



MDSE Egyenlet

Modellek + Transzformációk = Szoftver



Modeling Languages

- **Szakterület-specifikus nyelvek (DSL):** olyan nyelvek, amelyeket kifejezetten egy adott szakterületre vagy kontextusra terveztek
- A DSL-eket nagymértékben használják az informatikában. Példák: HTML, Logo, VHDL, Mathematica, SQL
- **Általános célú modellező nyelvek** (GPML-ek, GML-ek vagy GPL-ek): olyan nyelvek, amelyek bármely ágazatban vagy területen alkalmazhatók (szoftver)modellezési célokra
- A tipikus példák: UML, Petri-hálók vagy állapotgépek



A modell típusai

- **Statikus modellek:** A rendszer statikus aspektusaira összpontosítanak a kezelt adatok, valamint a rendszer szerkezeti formája és felépítése szempontjából.
- **Dinamikus modellek:** A rendszer dinamikus viselkedését hangsúlyozzák a végrehajtás bemutatásával.
- **Runtime modellek:** A rendszer működés közbeni állapotát mutatják be.
- Gondoljunk csak az UML-re!

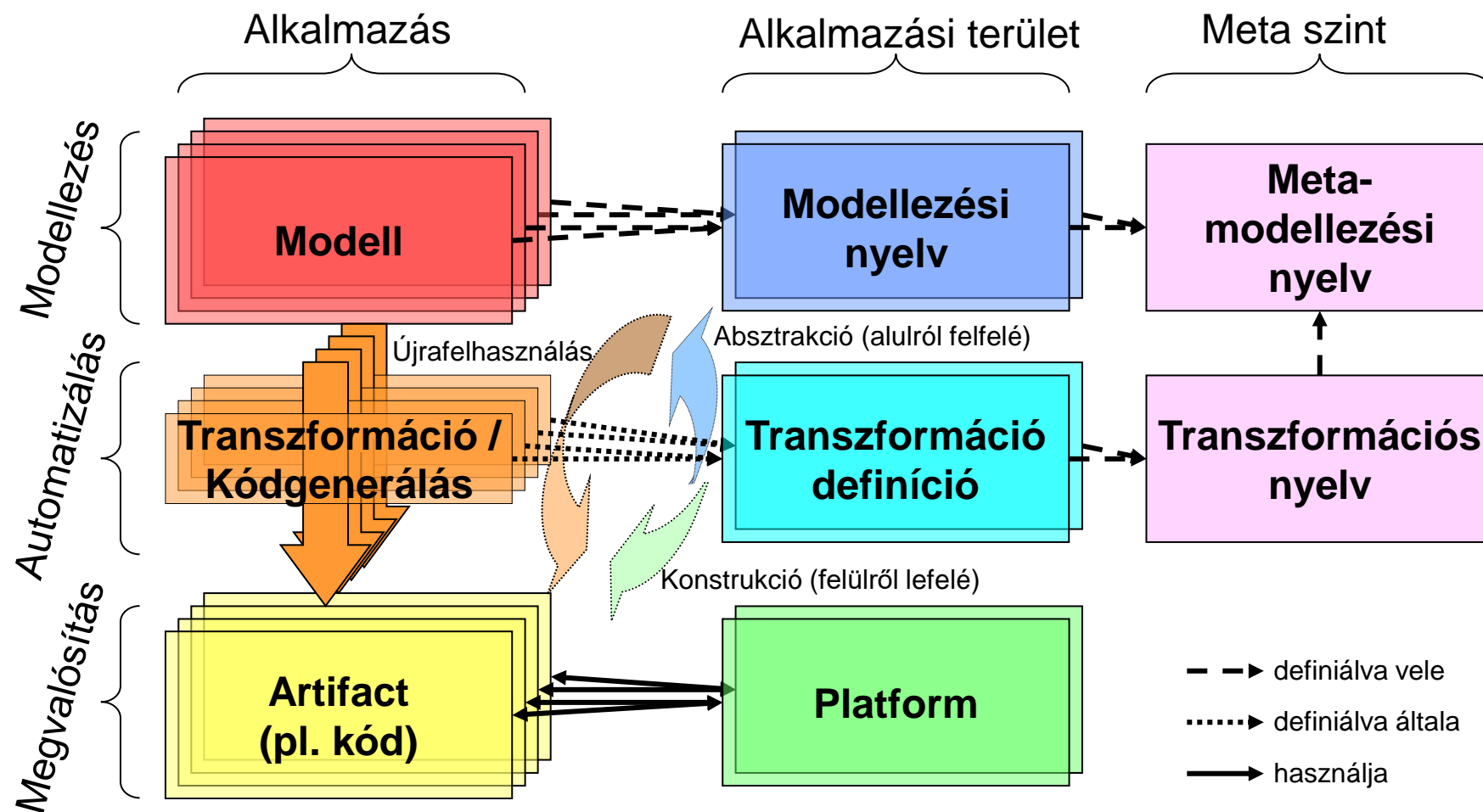
Felhasználás / Cél:

- Nyomonkövethetőségi Modellek
- Végrehajtáskövetési Modellek
- Elemzési Modellek
- Szimulációs Modellek



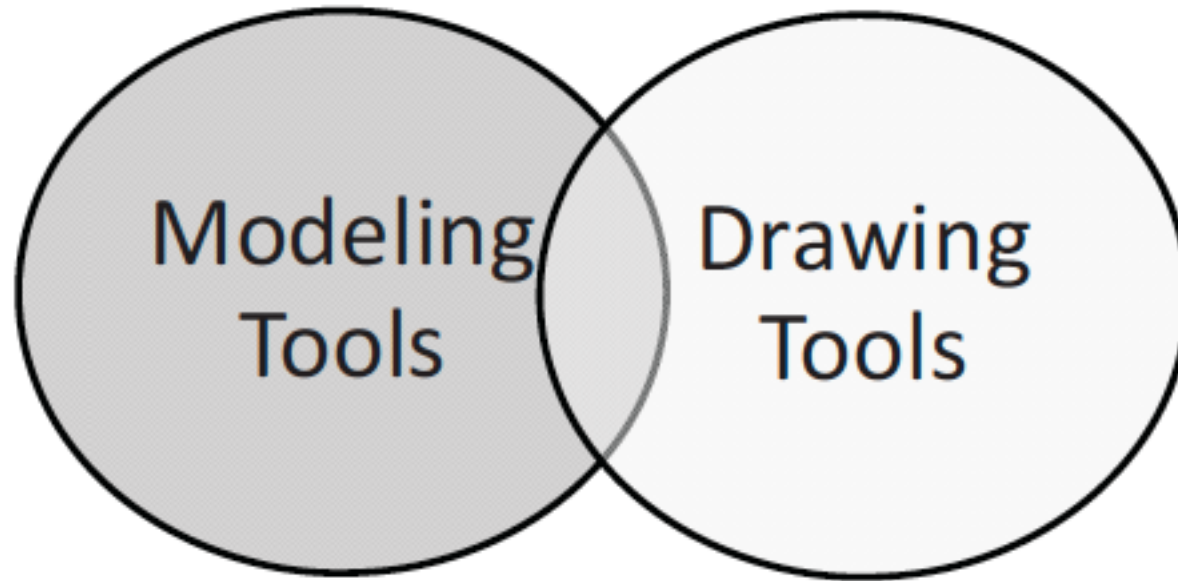
Fogalmak

Modellmérnöki alaparchitektúra



Eszköztámogatás

- Rajzolás vs. modellezés



▪ Módosított **fejlesztési folyamat**

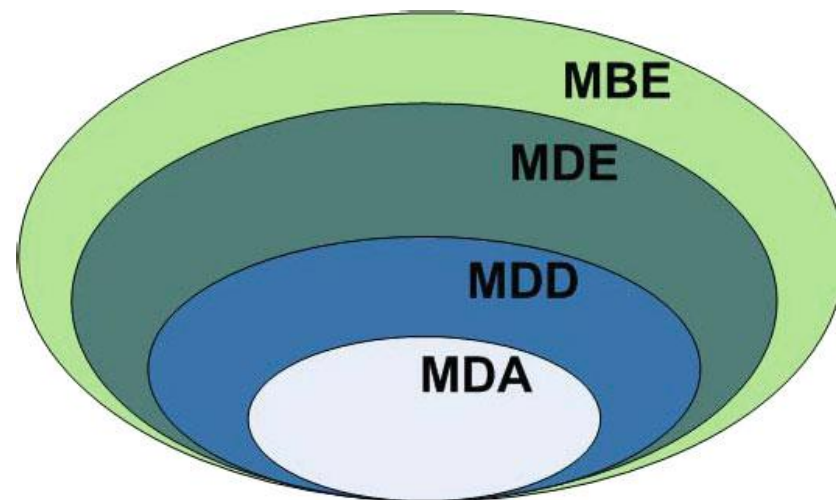
- A fejlesztés két szintje – alkalmazás és infrastruktúra
 - Az infrastruktúra fejlesztése magában foglalja a modellezési nyelvet, a platformot (pl. keretrendszer) és a transzformáció meghatározását
 - Az alkalmazásfejlesztés csak a modellezést foglalja magában - az infrastruktúra(k) hatékony újrafelhasználása
- Erősen egyszerűsített alkalmazásfejlesztés
 - Automatikus kódgenerálás helyettesíti a programozót
 - A kódszintű munka (megvalósítás, tesztelés, karbantartás) feleslegessé válik
 - *Milyen feltételek mellett reális ez ... vagy csak futurisztikus?*

▪ Új **fejlesztői eszközök**

- Nyelvi definíciós eszközök, különösen metamodellezési eszközök
- Szerkesztő és motor a modelltranszformációkhoz
- Testreszabható eszközök, például modellszerkesztők, tárolók, szimulációs, verifikációs és tesztelési eszközök



A rövidítések MD* dzsungel



- **A modellvezérelt fejlesztés (Model-Driven Development, MDD)** egy olyan fejlesztési paradigma, amely modelleket használ a fejlesztési folyamat elsődleges tárgyaként.
- **A modellvezérelt architektúra (Model-Driven Architecture, MDA)** az MDD sajátos elképzelése, amelyet az Object Management Group (OMG) javasolt.
- **A modellvezérelt tervezés (Model-Driven Engineering, MDE)** az MDD egy szuperhalmaza, mivel túlmutat a pusztán fejlesztésen.
- **A modellalapú tervezés** (vagy "modellalapú fejlesztés") **(Model-Based Engineering, MBE)** az MDE lazább változata, ahol a modellek nem "irányítják" a folyamatot



Az MDA megközelítés

Célok

- **Interoperabilitás** platformfüggetlen modellek révén
 - Az Object Management Group (**OMG**) szabványosítási kezdeményezése, amely OMG szabványokon, különösen az **UML**-en alapul
 - A CORBA megfelelője a modellezési szinten: különböző platformok közötti átjárhatóság
 - Különböző platformokra telepíthető alkalmazások → hordozhatóság, nincs probléma a technológiaváltással, a különböző platformok integrációjával stb.
- **Az alap architektúra módosításai**
 - A modellszint szegmentálása
 - **Platformfüggetlen** modellek (PIM): több (hasonló) platformra érvényesek
 - **Platformspezifikus** modellek (PSM): speciális módosítások egy adott platformra
 - Modell-modell transzformációt (PIM-PSM; vö. QVT) és modell-kód transzformációt (PSM-kód) igényel
 - A platformfejlesztést nem veszik figyelembe - általában az olyan ipari szabványok, mint a J2EE, .NET, CORBA, platformnak tekintendők

[www.omg.org/mda/]



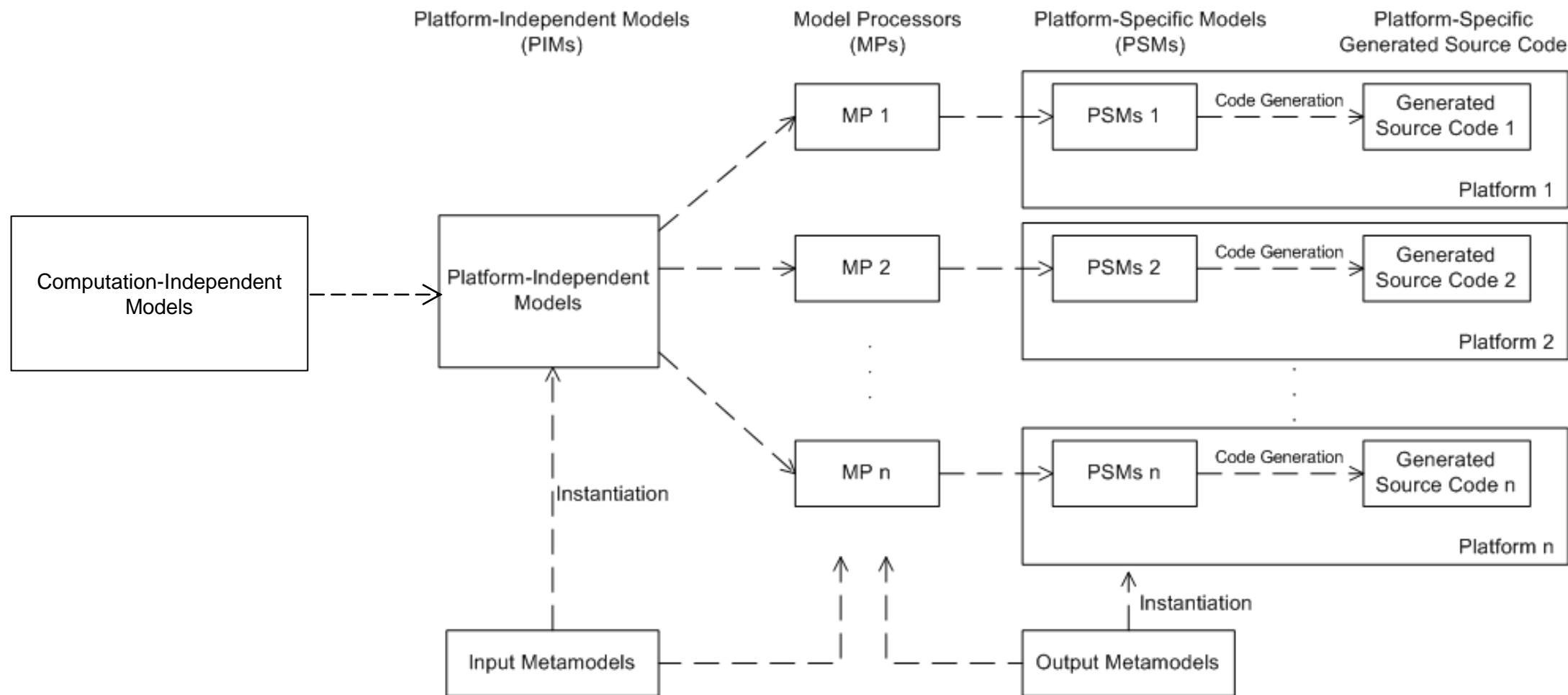
Modellezési Szintek

CIM, PIM, PSM

- **Számításfüggetlen (CIM):** a követelmények és igények leírása nagyon absztrakt szinten, a megvalósítás aspektusaira való hivatkozás nélkül (pl. felhasználói követelmények vagy üzleti célok leírása);
- **Platformfüggetlen (PIM):** a rendszerek viselkedésének meghatározása a tárolt adatok és a végrehajtott algoritmusok szempontjából, minden technikai vagy technológiai részlet nélkül;
- **Platformspezifikus (PSM):** minden technológiai szempont részletes meghatározása.



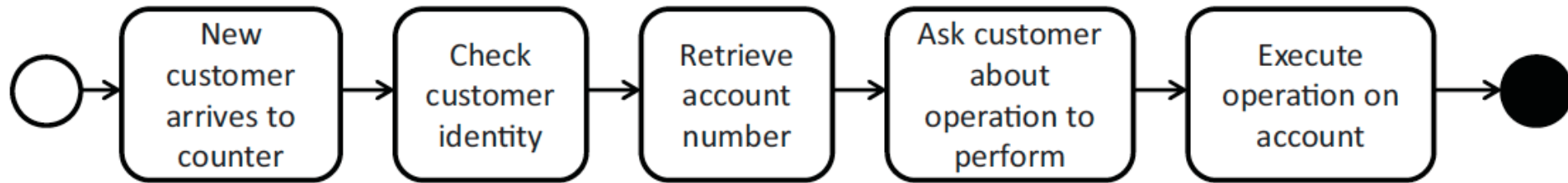
Hogyan képzeljük ezt el?



Modellezési Szintek

MDA Számításfüggetlen modell (CIM)

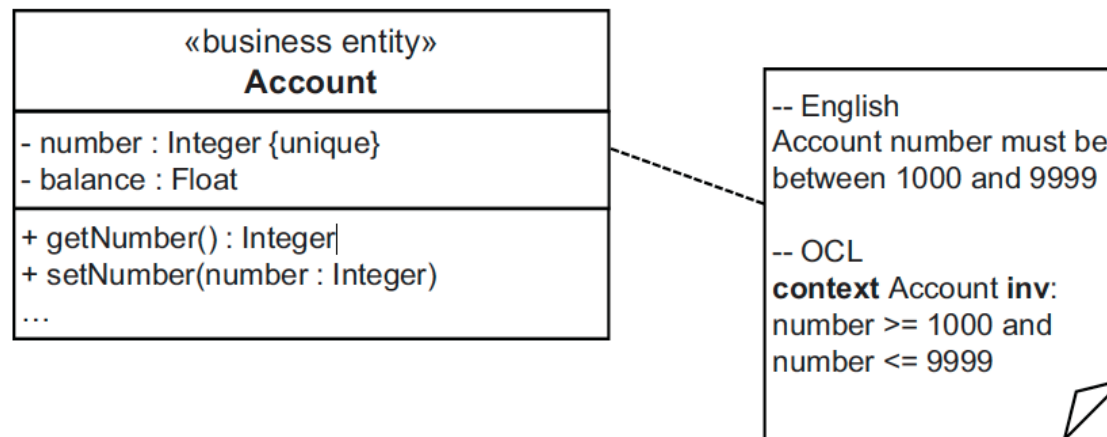
- PI. üzleti folyamat



Modellezési Szintek

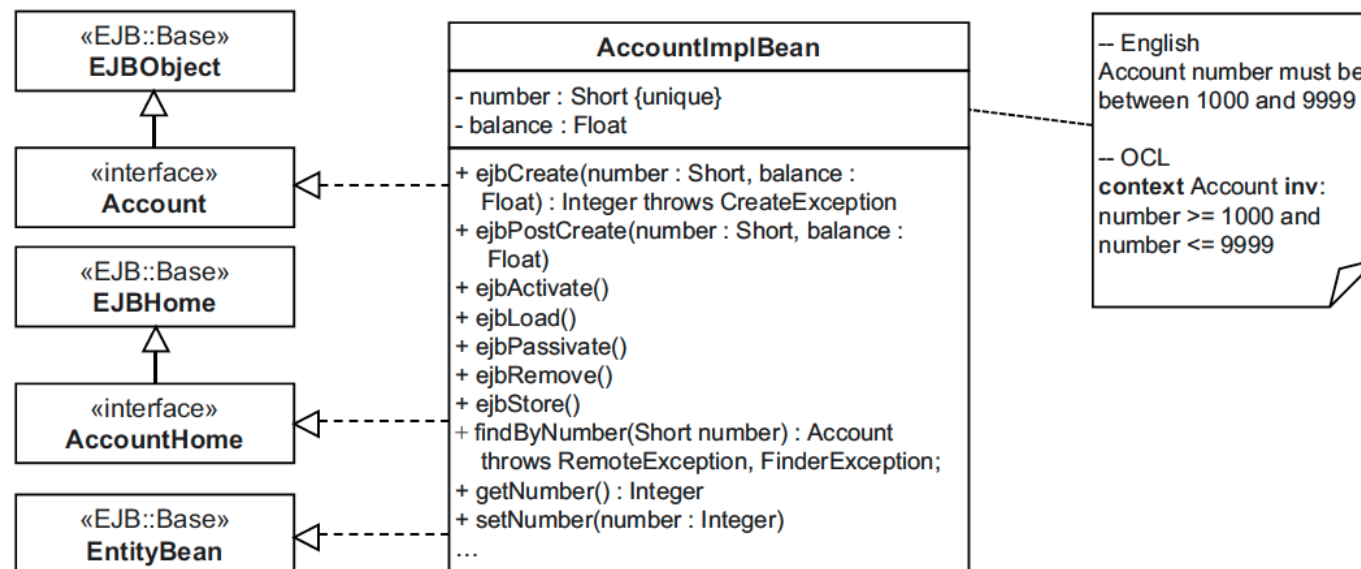
MDA Platformfüggetlen modell (PIM)

- egy rendszer szerkezetének és viselkedésének specifikációja, a technológiai részletektől elvonatkoztatva
- Az UML használatával (opcionális)
- A rendszer szerkezetének és viselkedésének absztrakciója a PIM segítségével egyszerűsíti a következőket:
 - A modell helyességének validálása
 - Implementációk létrehozása különböző platformokon
 - Eszköztámogatás a megvalósítás során



Modellezési Szintek

MDA Platformspecifikus modell (PSM)

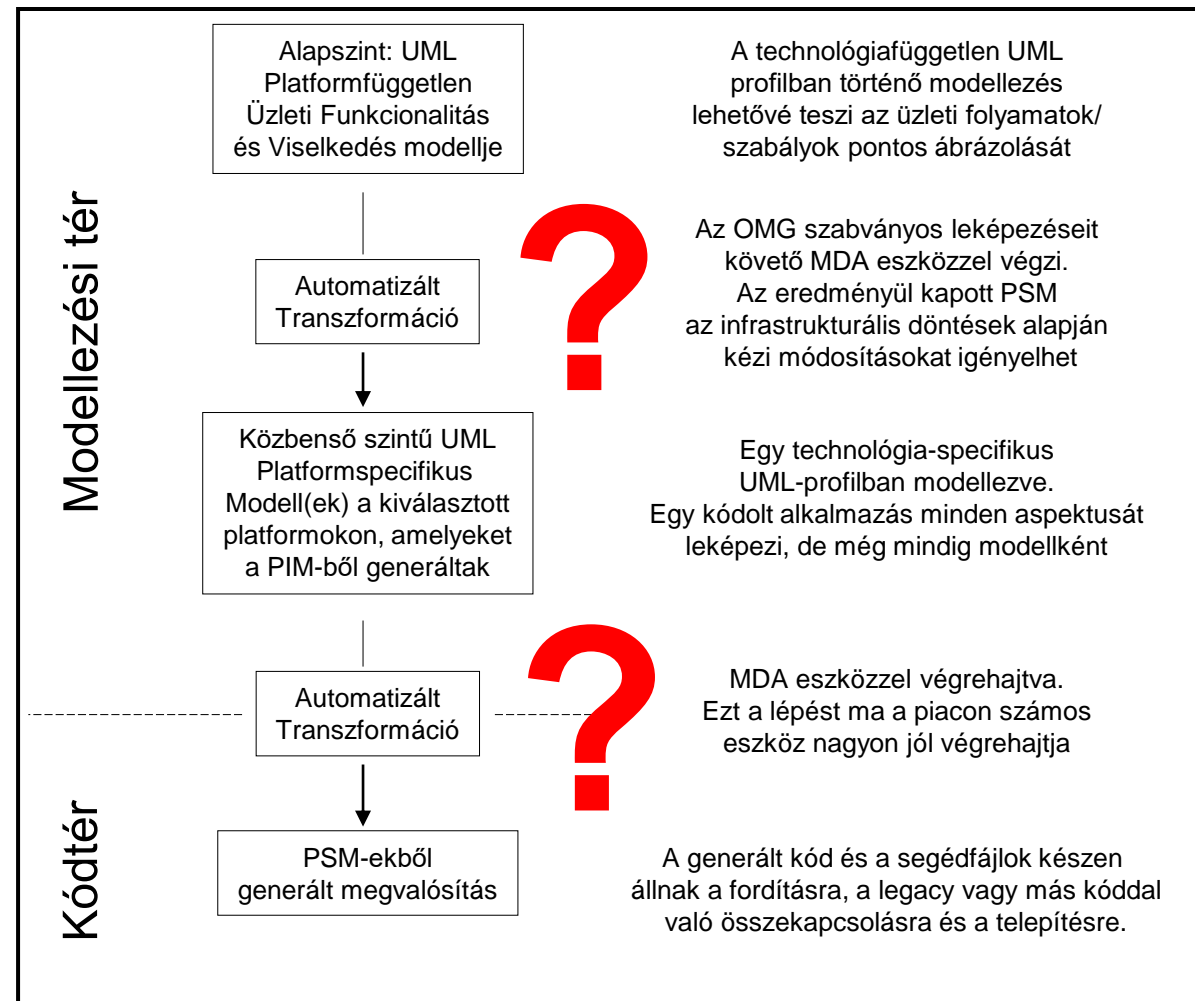


- Meghatározza, hogy a PIM-ben leírt funkcionalitás hogyan valósul meg egy adott platformon
- A kiválasztott platformra vonatkozó UML-profil használata, pl. EJB



Az MDA Megközelítés

MDA fejlesztési ciklus



Modellalapú fejlesztések

I. Fejlesztési fogalmak

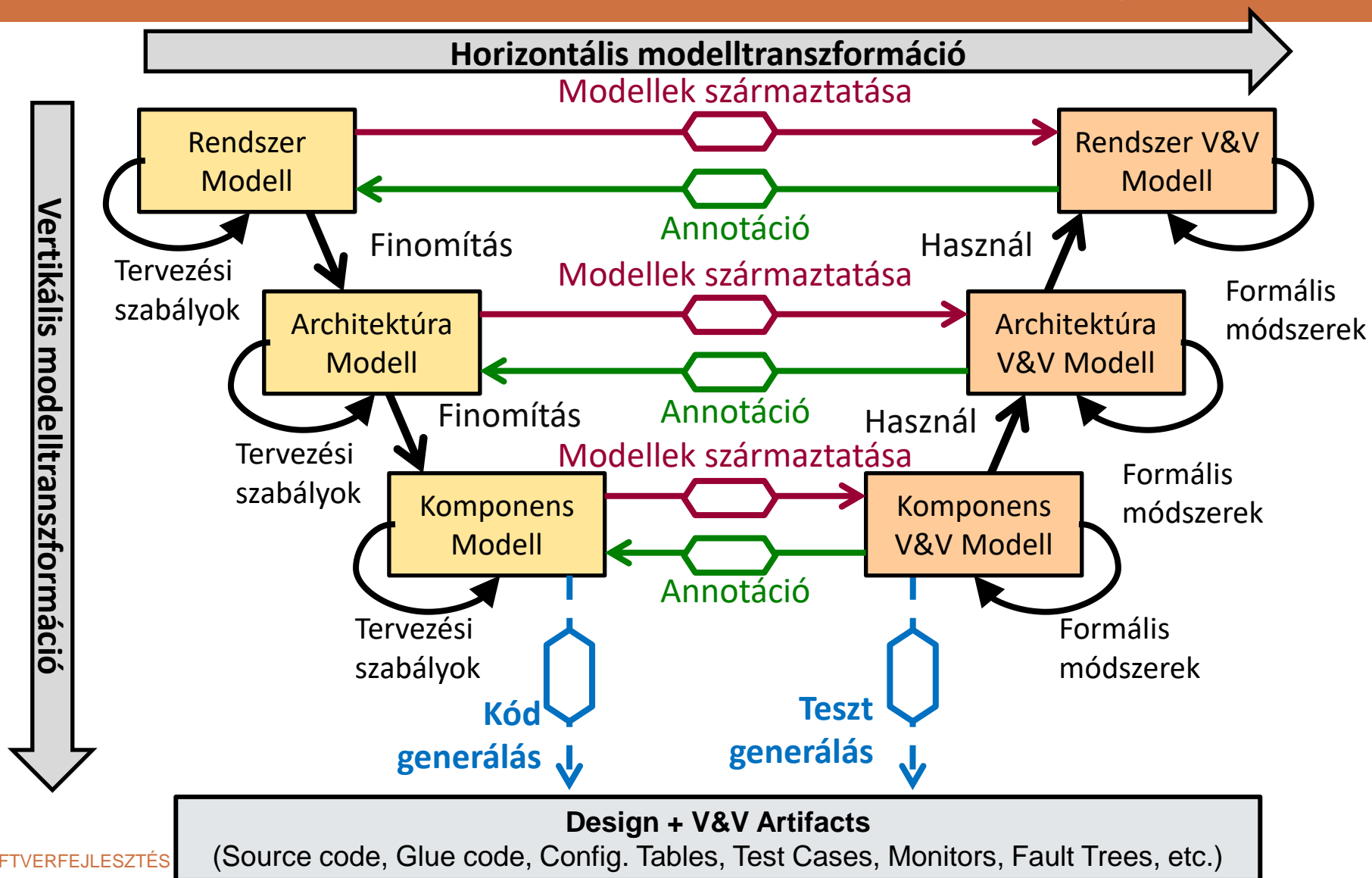
II. Kritikus rendszerek fejlesztése

III. Funkciómodellezés

IV. Generatív programozás



Modellek és Transzformációk kritikus rendszerek fejlesztésében



Development Process for Critical Systems

Egyedi Fejlesztési Folyamat (Hagyományos V-modell)



Kritikus rendszerek tervezése

- **tanúsítványozási folyamatot** igényel
- **alátámasztott bizonyítékok** kidolgozásához,
- hogy a **rendszer hibamentes**

Szoftvereszközök minősítése

- **tanúsítvány** megszerzése
- egy **szoftvereszköze**
- amelyet a **kritikus rendszerek tervezéséhez használnak**

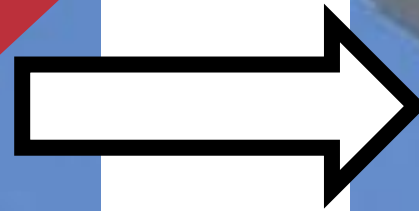
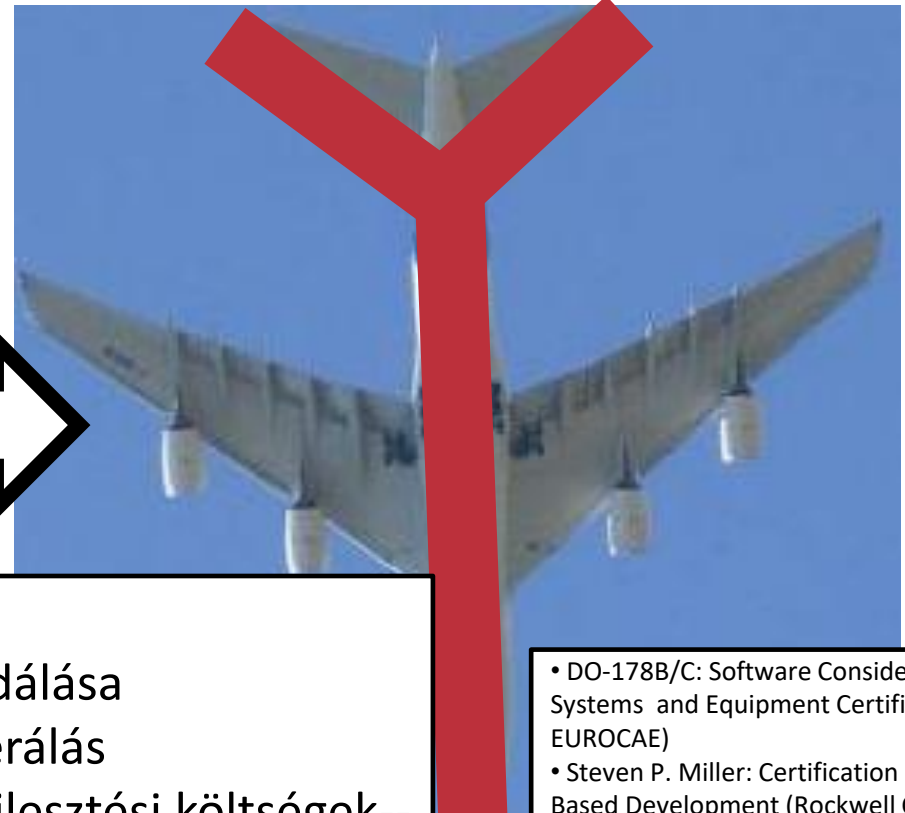
Innovatív Eszköz → Jobb Rendszer
Tanúsított Eszköz → Tanúsított Kimenet

Development Process for Critical Systems

Egyedi Fejlesztési Folyamat (Hagyományos V-modell)



Modellvezérelt Tervezés (Y-modell)



Az MDE fő gondolatai

- rendszermodellek korai validálása
- automatikus forráskód-generálás
- ➔ minőség++ eszközök ++ fejlesztési költségek--

- DO-178B/C: Software Considerations in Airborne Systems and Equipment Certification (RTCA, EUROCAE)
- Steven P. Miller: Certification Issues in Model Based Development (Rockwell Collins)

Modellalapú fejlesztések

I. Fejlesztési fogalmak

II. Kritikus rendszerek fejlesztése

III. Funkciómodellezés

IV. Generatív programozás



Funkciómodellezés – bevezetés

- Termékcsalád elemei közti különbségek
 - > Mobiltelefonok
 - Kijelző típusa
 - I/O interfészek
 - > Autógyártás
 - Ajtók száma
 - Motor típusa
- Szakterületi nyelv a különbségek összefogására: funkciómodellezés

Funkciómodellezés

- Funkciómodell (feature model)

- > *implementációtól független, tömör leírása a különböző szakterületi változatoknak*
- > a konkrét termékpéldányok közti különbségek
- > termékcsalád konfigurációs lehetőségei

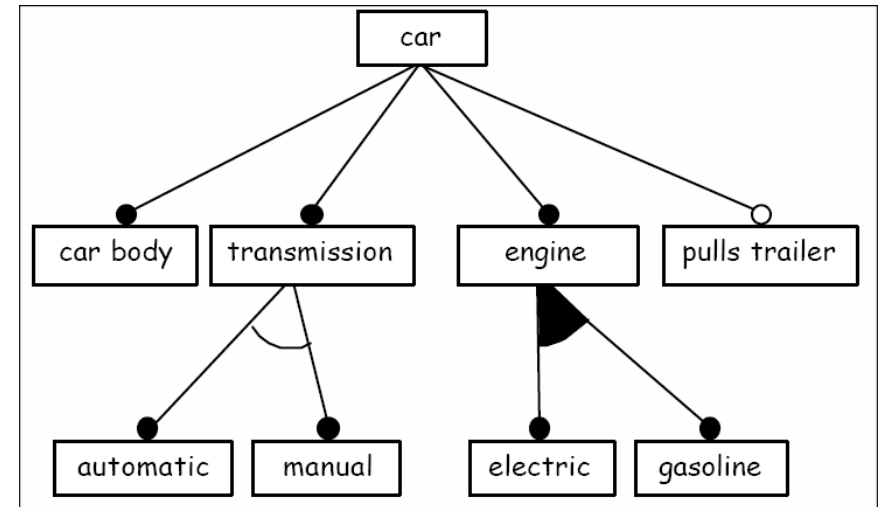
- Kulcs: újrahasznosítás

- Segít elkerülni:

- > Fontos funkció / variáció kimaradjon
- > Feleslegesen vegyünk fel funkciót / variációt

Funkciómodellezés

- Modellelemek
 - > Csomópontok
 - > Irányított élek
 - > Jelölés az éleknél
- Gyökérelem: fogalom (concept)
- Funkciók (feature node)
- Konfiguráció: funkciók részhalmaza
- A konfigurációnak be kell tartania bizonyos szabályokat!

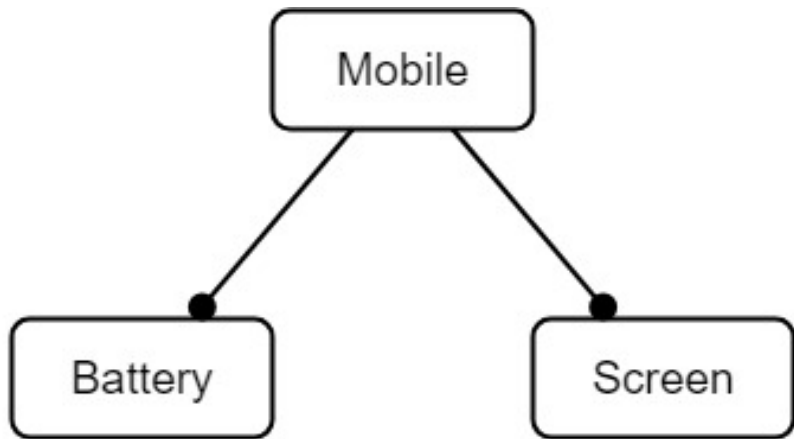


Funkciómodellezés

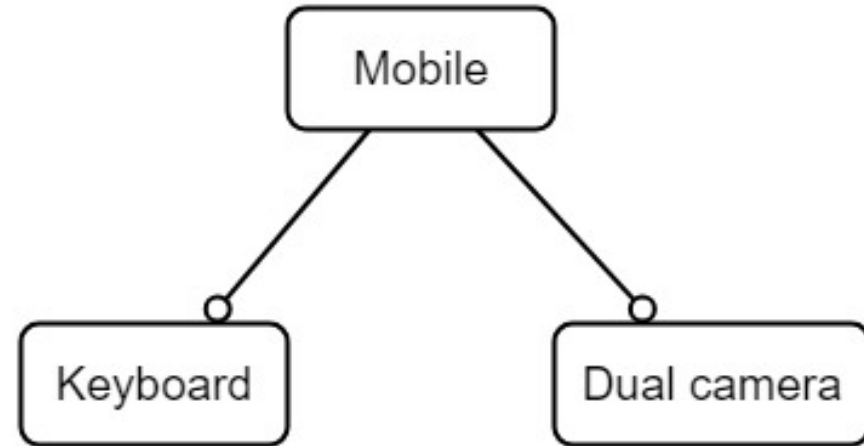
- Központban a funkció (feature)
 - > *Tudás egy része, a fogalom építőeleme*
 - > Segít megtalálni az azonosságokat, különbségeket termékek, termékcsoporthoz között
 - > Hasznos ha több változat van ugyanabból a termékből
- <https://modeling-languages.com/analysis-of-feature-models/>

Funkciómodellezés

■ Kötelező funkció

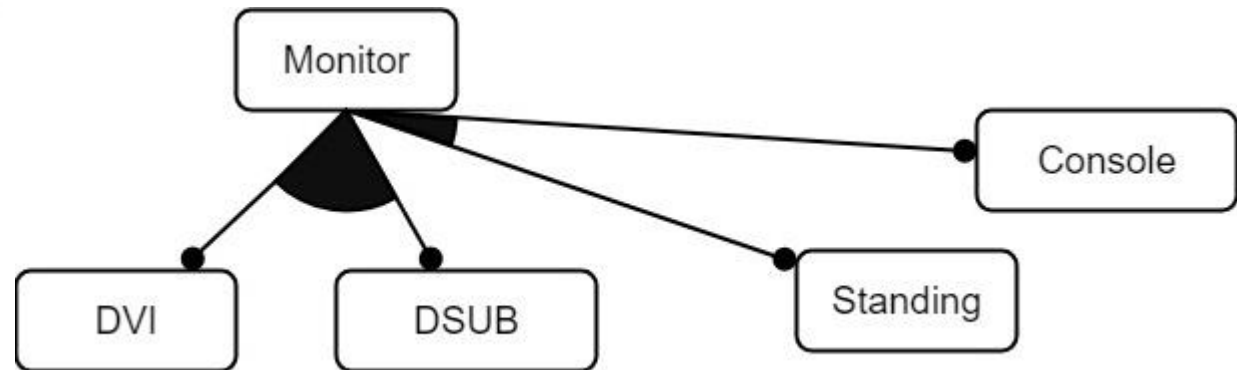
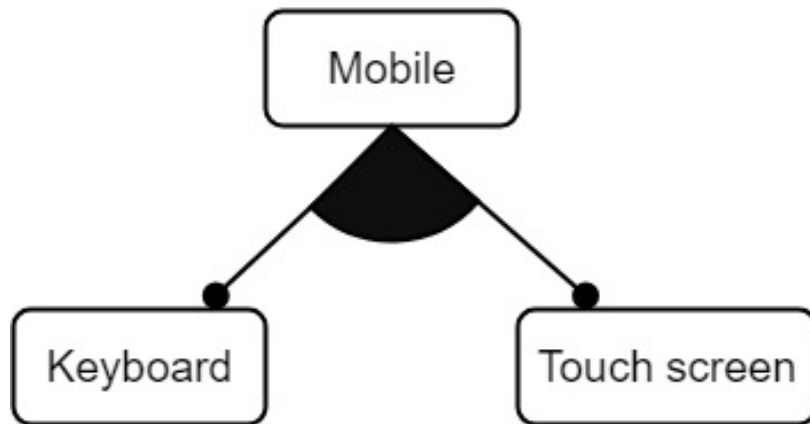


- Opcionális funkció



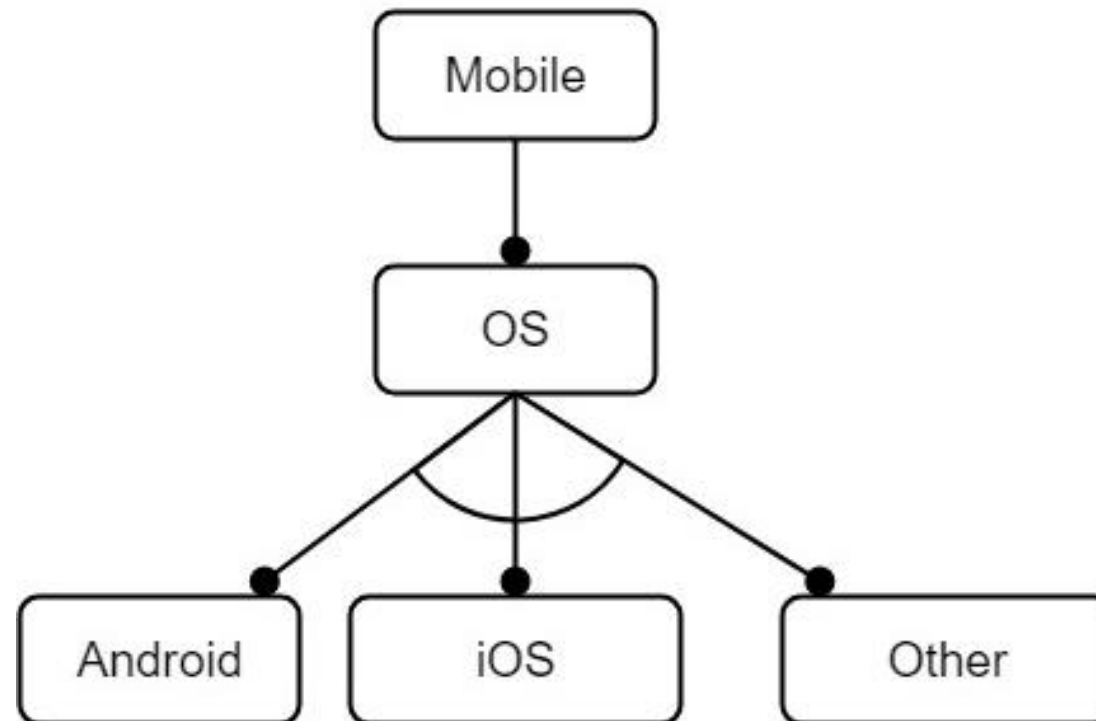
Funkciómodellezés

- „Vagy” kapcsolat (legalább 1)



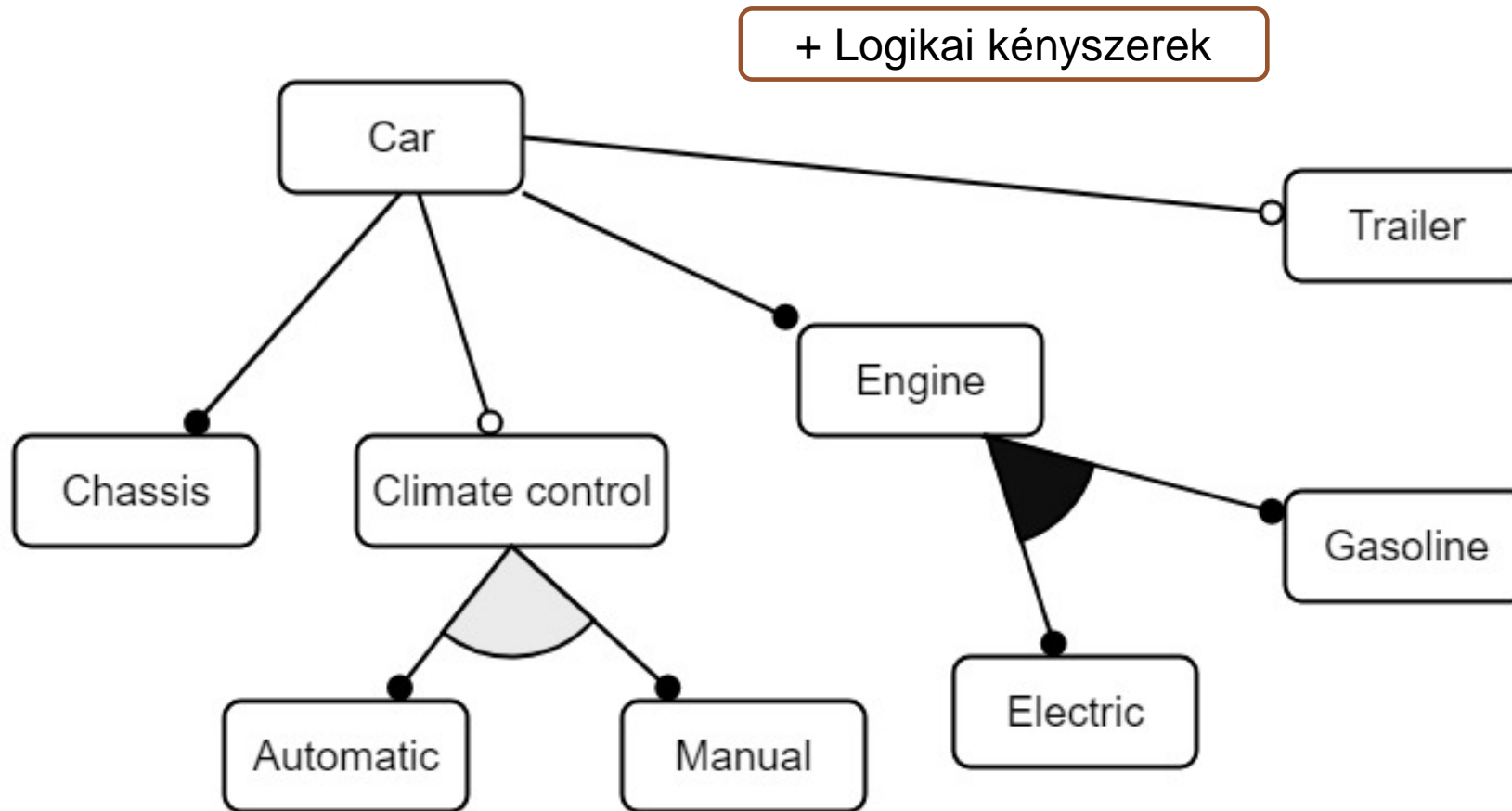
Funkciómodellezés

- Alternatív funkciók (valamelyik elem)



Funkciómodellezés

■ Példa: autó

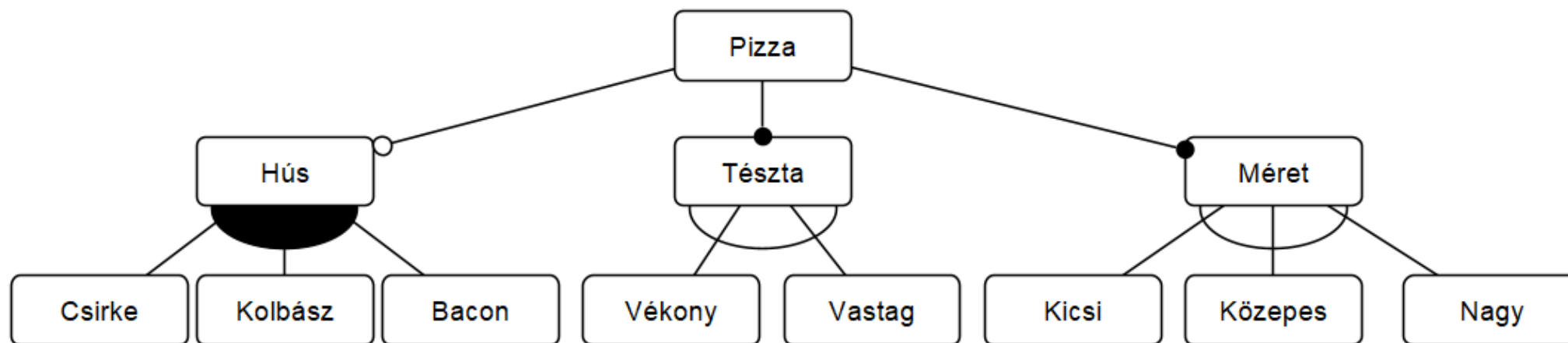


Funkciómodellezés - Példa

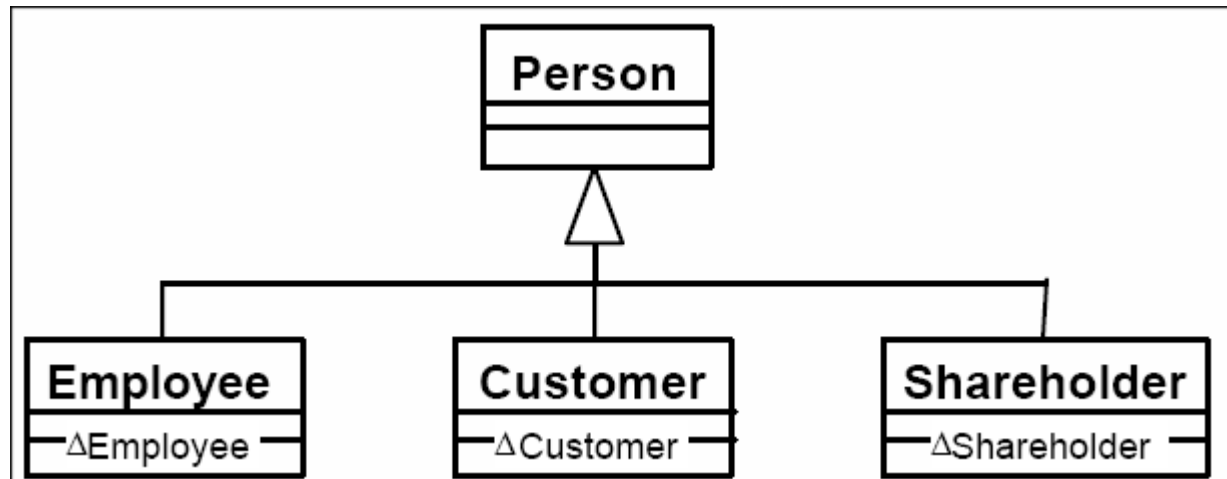
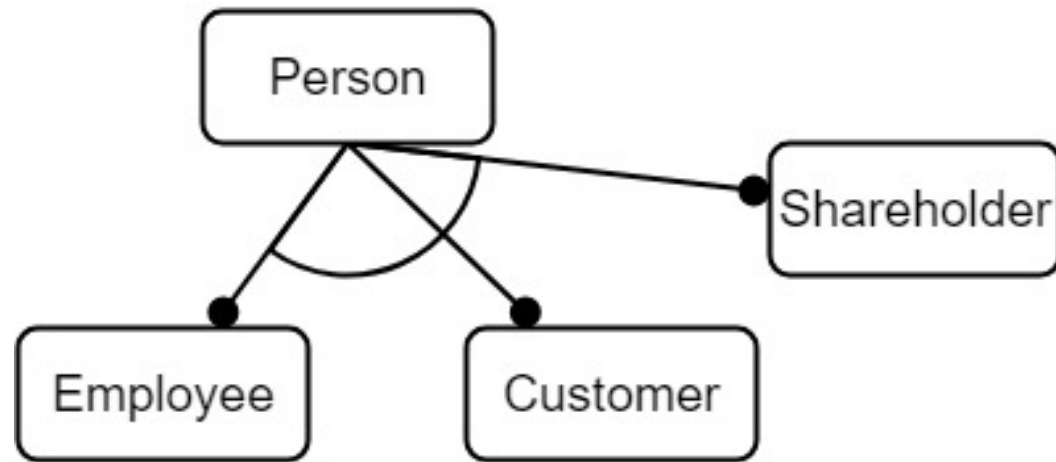
- Készítsen funkciómodellt a következő feladathoz: pizza elkészítése. A modellekben többek között legyen lehetőség megadni húsokat (csirke, kolbász, bacon), további feltéteket (paradicsom, hagyma, paprika), tészta típusokat (hagyományos, light), méretet (kicsi, közepes, nagy). A modell tartalmazzon opcionális-, kötelező és kizáró (OR) funkciót. Szövegesen indokolja röviden a modell felépítését!

Funkciómodellezés – Példa - Megoldás

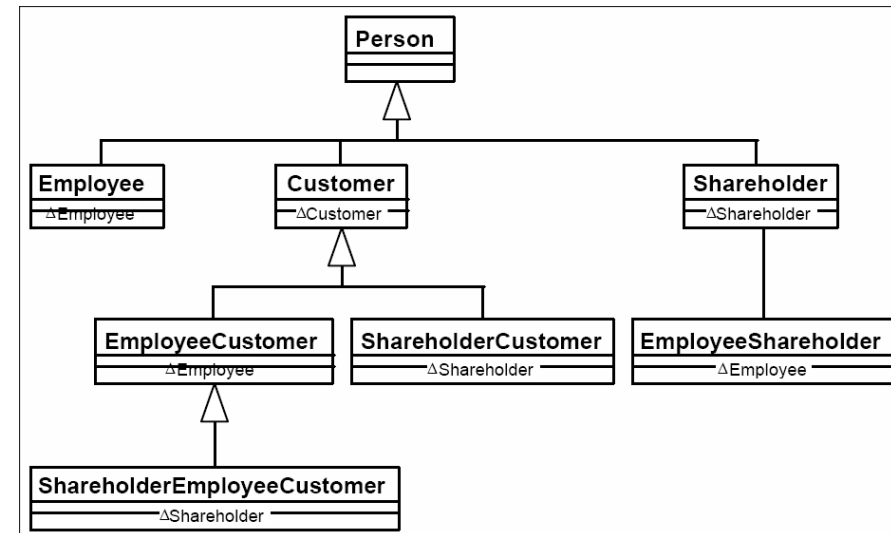
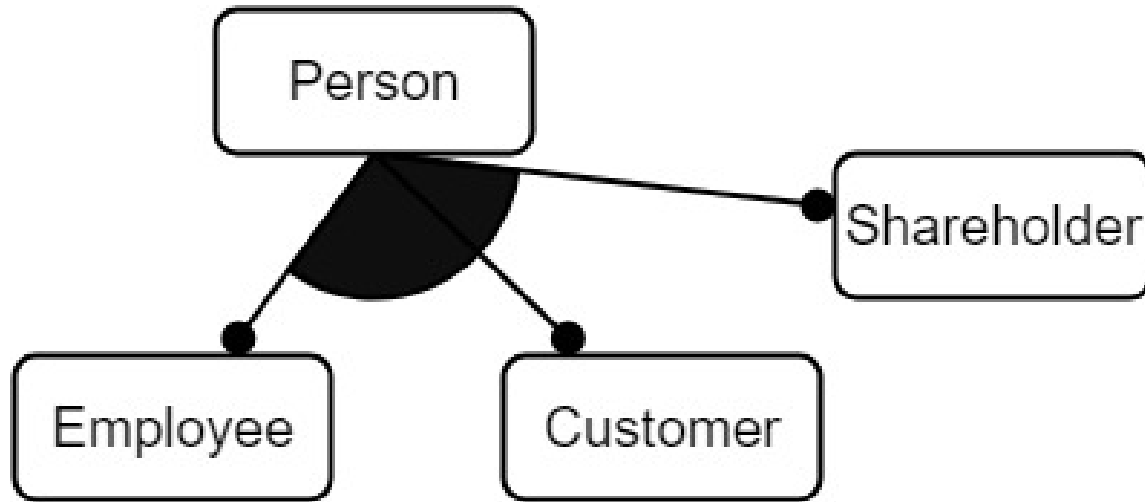
- Készítsen funkciómodellt a következő feladathoz: pizza elkészítése. A modellekben többek között legyen lehetőség megadni húsokat (csirke, kolbász, bacon), további feltéteket (paradicsom, hagyma, paprika), tészta típusokat (hagyományos, light), méretet (kicsi, közepes, nagy). A modell tartalmazzon opcionális-, kötelező és kizáró (OR) funkciót. Szövegesen indokolja röviden a modell felépítését!



Funkciómodellezés - kódgenerálás



Funkciómodellezés - kódgenerálás



Funkciómodellezés a gyakorlatban

- Cél alkalmazás
 - > A funkciómodellnek megfelelő termékeket tartalmazó webes katalógus
 - > Funkciómodell szerint megvalósított keresés a katalógusban
- Generálás a modell alapján
 - > Web alkalmazás
 - > Adatbázis tábladefiníciók








Funkciómodellezés a gyakorlatban

82%

FEATURE MODEL

- Trek Bikes
 - Category
 - 1.1
 - bike-path
 - cruiser
 - kids
 - mountain-full-suspension
 - mountain-hardtail
 - road
 - track
 - triathlon
 - urban
 - Rider Category
 - Price - US\$
 - 1.1
 - 1,000 or under
 - 1,001-2,000
 - 2,001-3,000
 - 3,001-5,000
 - 5,001-10,000
 - Frameset
 - Wheels
 - Drivetrain
 - Components
 - Saddle
 - 1.1
 - Seat Post
 - 1.1
 - Alloy
 - Bontrager-Alloy-Infinite-Angle...
 - Bontrager-Approved

PRODUCT CATALOG (9 products)

1. KDR 7.2 FX-34 cm Price (US\$): \$439.99 View this model	
2. Wasabi 24"-24" Price (US\$): \$329.99 View this model	
3. MT 240-24" Price (US\$): \$549.99 View this model	
4. Drift 20"-20" Price (US\$): \$269.99 View this model	
5. MT 60-20" Price (US\$): \$309.99 View this model	
6. Mystic 20-20" Price (US\$): \$219.99 View this model	
7. Float-16" Price (US\$): \$219.99 View this model	

- Funkciók lemodellezése
- Konfiguráció kiválasztása
- Kódgenerálás
- Feladat megoldása
- #Konfiguráció > #Termék

Marcilio Mendonca, Andrzej Wąsowski, and Krzysztof Czarnecki.
2009. **SAT-based analysis of feature models is easy.**
In *Proceedings of the 13th International Software Product Line
Conference (SPLC '09)*. Carnegie Mellon University, USA, 231–240.

Modellalapú fejlesztések

I. Fejlesztési fogalmak

II. Kritikus rendszerek fejlesztése

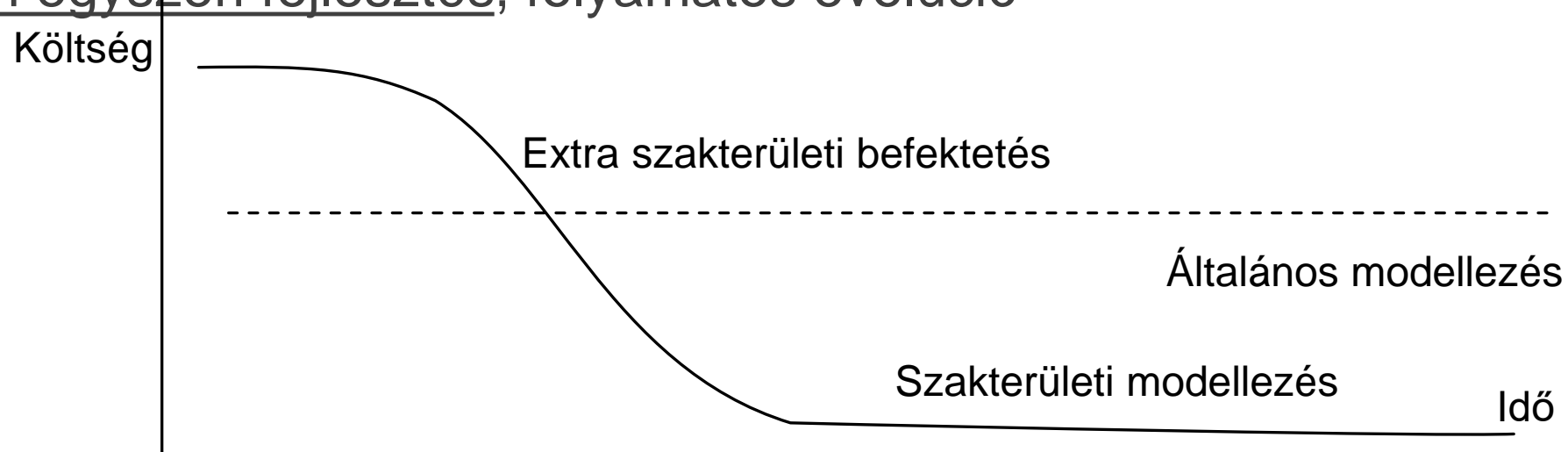
III. Funkciómodellezés

IV. Generatív programozás



Generatív programozás

- Programozási módszertan, alapja az automatikus forráskód-generálás
- Párhuzamba vonható a komponens-alapú szoftverfejlesztéssel és a termékcsalád tervezéssel
- Újrahasznosítható termék
- Nem egyszeri fejlesztés, folyamatos evolúció



Generatív programozás

- Generatív paradigma
 - > Működés: modellezőnyelv+generátorok
 - > Többszöri használatnál éri meg
- Kódgenerálás
 - > Nincsenek univerzális DSL fordítók (mint C fordítók)
 - > Gyakran a DSL-t és a generátort ugyanott fejlesztik
 - Gyors fejlesztés, finomhangolási lehetőség
 - Hibalehetőségek

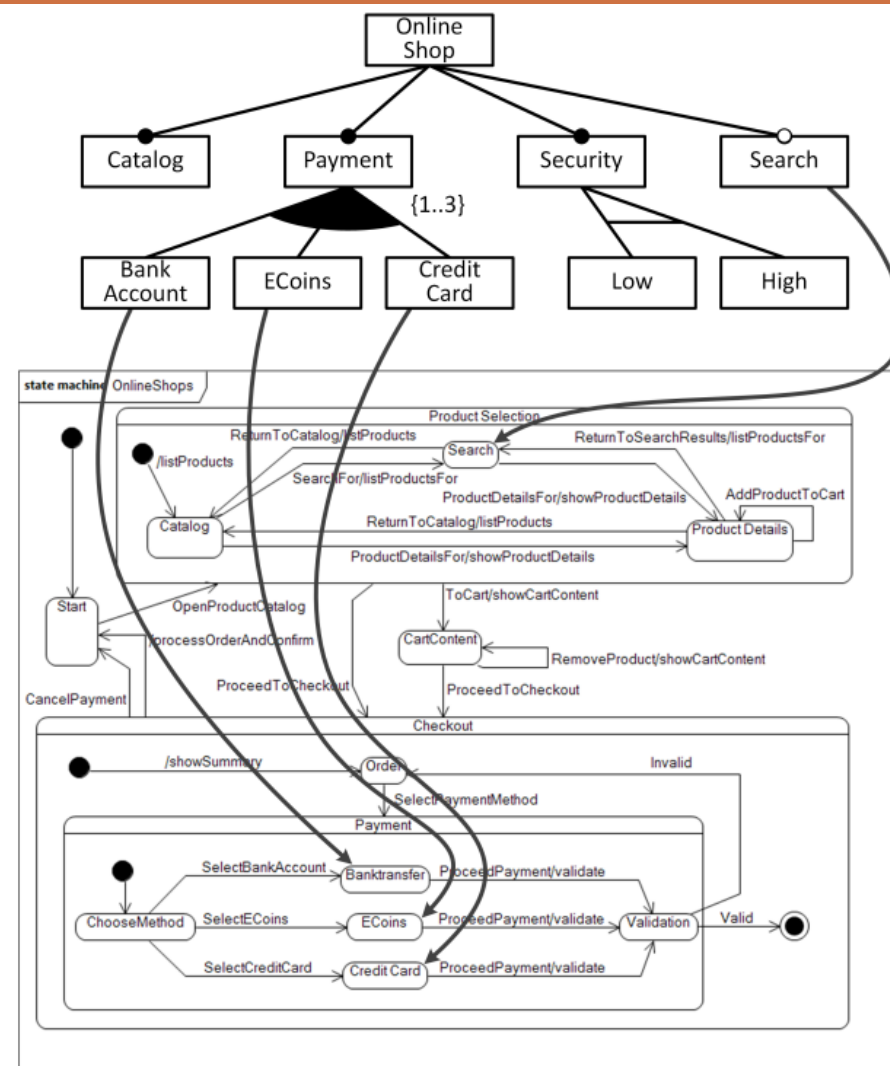
Alkalmazás generálása

- Tipikus modellfeldolgozás: alkalmazás generálása
 - > Amire szükségünk van
 - > Szakterületi nyelv modellje
 - > Generátor
 - > Keretrendszer (pl. osztálykönyvtár)
- Kihívások:
 - > Túl részletes/általános nyelv → kicsi absztrakciós szint ugrás → kis előny a generátorból
 - > szakterület nem illeszkedik → komplex generator
 - validálás + hozzáadott infó miatt
 - Jel: fejlesztők úgy építik a modellt, hogy a generátor elfogadja

Funkciók leképezése modellelemekre

- Mit tegyünk, ha nem kód, hanem modell kell?
- Funkciók leképezése modellelemekre
- 1 kombináció = 1 modell
- Σ konfiguráció = 150% modell
- A 150%-os modell nem feltétlenül szabályos modell

Stephan Weißleder, Hartmut Lackner: Top-Down and Bottom-Up Approach for Model-Based Testing of Product Lines





Köszönöm a figyelmet!