

1. Gyakorlat: EMF

Simon Balázs, BME IIT, 2013.

1 Feladat

Készítsünk EMF metamodellt, amely alkalmas context diagramok (CD) és data-flow diagramok (DFD) leírására!

Lépések:

1. Olvassuk el a use-case metamodel elkészítésére vonatkozó EMF tutorial-t (2. fejezet)!
2. Készítsük el a tutorial segítségével a CD és DFD diagramok metamodeljét! A 3. fejezet a CD és DFD szemantikáját írja le. Olyan metamodelt kell alkotni, amely alkalmas a megadott szemantikájú CD és DFD diagramok reprezentálására. Grafikus jelölést még nem kell készíteni, ezt majd egy későbbi gyakorlaton fogjuk megvalósítani.
3. A 2. fejezetben leírt tutorial legutolsó lépése egy önálló Eclipse indítását mutatja be. Ebben az utolsó lépésben készítsünk a fahierarchia szerkesztőben egy CD-t és egy DFD-t! Minden elemből (CD, DFD, terminator, process, store, data-flow) készítsünk legalább egy elemet!
4. Vizsgáljuk meg az EMF generátor által készített projekteket és fájlokat. Különös tekintettel nézzük át a kiinduló projekt forráskódjait, amelyben az ecore interfészek és implementációk találhatók!

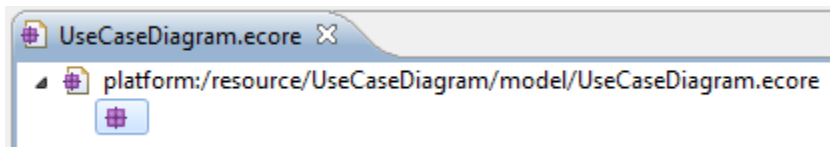
2 EMF Tutorial: Use-case metamodel

2.1 Feladat

Készítsünk use-case metamodelt EMF-ben!

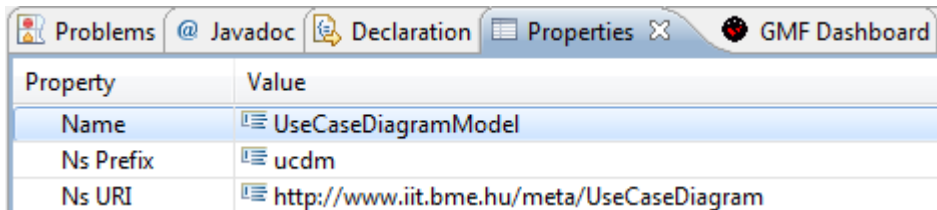
2.2 Megoldás

1. **File > New > Project...**
 - 1.1. **Eclipse Modeling Framework / Empty EMF Project**
 - 1.2. Project name: **UseCaseDiagram**
 - 1.3. **Finish**
2. **model** könyvtáron jobbgomb, majd: **New > Other**
3. **Eclipse Modeling Framework / Ecore Model**
 - 3.1. File name: **UseCaseDiagram.ecore**
 - 3.2. Model object: **EPackage**
 - 3.3. XML Encoding: **UTF-8**
 - 3.4. **Finish**
4. Nyissuk ki a keletkező gyökérelemet és válasszuk ki az alatta lévő üres csúcsot:



Ha nem ez a nézet jelenne meg, akkor kattintsunk jobbgommbal a **UseCaseDiagram.ecore** elemen, és válasszuk az **Open With > Sample Ecore Model Editor** menüpontot!

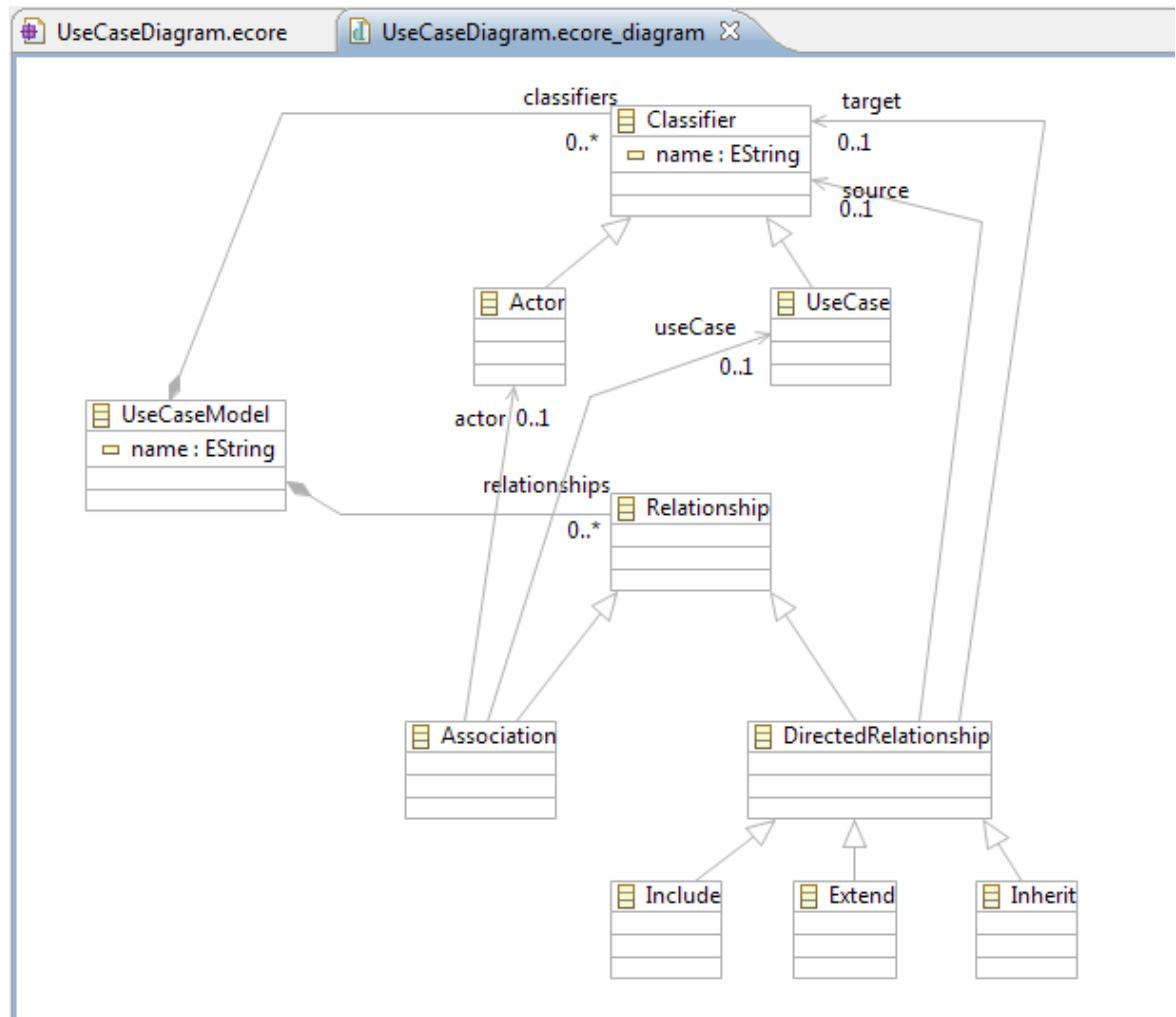
5. A **Properties** ablakban adjuk meg a következő adatokat:



Property	Value
Name	UseCaseDiagramModel
Ns Prefix	ucdm
Ns URI	http://www.iit.bme.hu/meta/UseCaseDiagram

6. Mentsük el a **UseCaseDiagram.ecore** fájlt!
7. Kattintsunk rajta jobbgommbal, majd: **Initialize ecore_diagram diagram file** (ha nincs ilyen menüpont, telepítsük a GMF SDK plugint)
 - 7.1. File name: **UseCaseDiagram.ecore_diagram**
 - 7.2. Select diagram root element: **UseCaseDiagramModel**
 - 7.3. **Finish**
8. Rajzoljuk meg a következő diagramot:

8.1. Diagram:



8.2. Absztrakt osztályok (Properties ablak / Abstract=true):

Classifier, Relationship, DirectedRelationship

8.3. A **0..*** multiplicitáshoz a **Properties** ablakban az **Upper Bound** értéket **-1**-re kell állítani.

9. Kattintsunk jobbgombbal a **model** könyvtáron, majd válasszuk a **New > Other... / Eclipse Modeling Framework / EMF Generator Model-t**

9.1. File name: **UseCaseDiagram.genmodel**

9.2. Model importers: **Ecore model**

9.3. Model URIs: **Browse Workspace... / UseCaseDiagram / model / UseCaseDiagram.ecore**

9.4. Root packages: **UseCaseDiagramModel**

9.5. **Finish**

10. Nyissuk meg a **UseCaseDiagram.genmodel** fájlt, majd válasszuk ki a gyökérelmet

(**UseCaseDiagram**) és kattintsunk ezen jobbgombbal: **Generate All**

11. Ennek hatására elkészülnek a következő projektek:

11.1. **UseCaseDiagram.edit**

11.2. **UseCaseDiagram.editor**

11.3. **UseCaseDiagram.tests**

12. Kattintsunk jobbgombbal a **UseCaseDiagram.editor** projekten, majd válasszuk a **Run As > Eclipse Application** menüpontot!

12.1. A megjelenő Eclipse-ben készítsünk egy üres EMF projektet!

12.2. A projektben pedig a **model** könyvtáron jobbgombbal kattintva hozzunk létre a **New > Other... / Example EMF Model Creation Wizards / UseCaseDiagramModel Model-t**

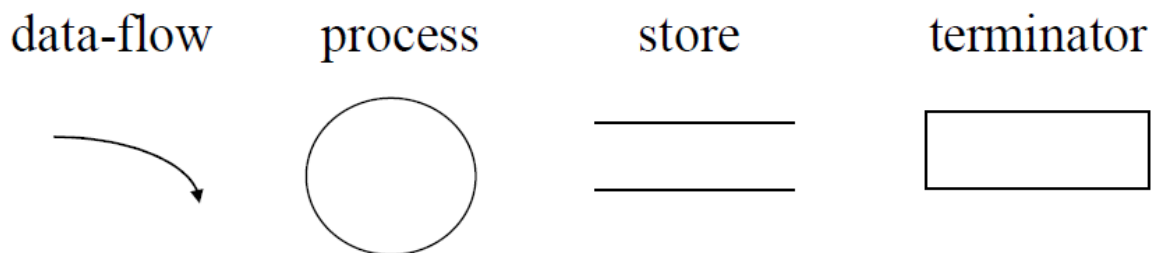
12.3. Model object-nek válasszuk a **UseCaseModel-t**!

12.4. Itt egy fahierarchiában hozhatjuk létre a metamodelnek megfelelő UseCase modellt.

Később a GMF segítségével saját grafikus szerkesztőt készítünk majd.

3 Context Diagram & Data-Flow Diagram

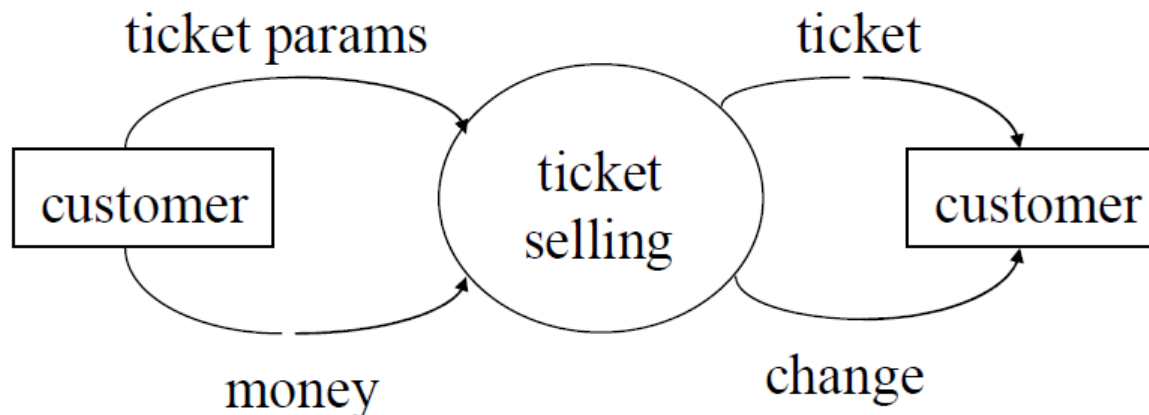
A CD és DFD a következő elemekből épülnek fel:



Ezek jelentése:

- Data-flow: adatáramlás iránya. A nyílra rá van írva az adat neve.
- Process: folyamat, amely adatokat dolgoz fel. A gombócba bele van írva a folyamat neve.
- Store: tár, ami adatokat tárol. A két párhuzamos vonal közé van írva a tároló neve.
- Terminator: a külvilággal való kapcsolatot jelzi. A téglalapba bele van írva a külső entitás neve.

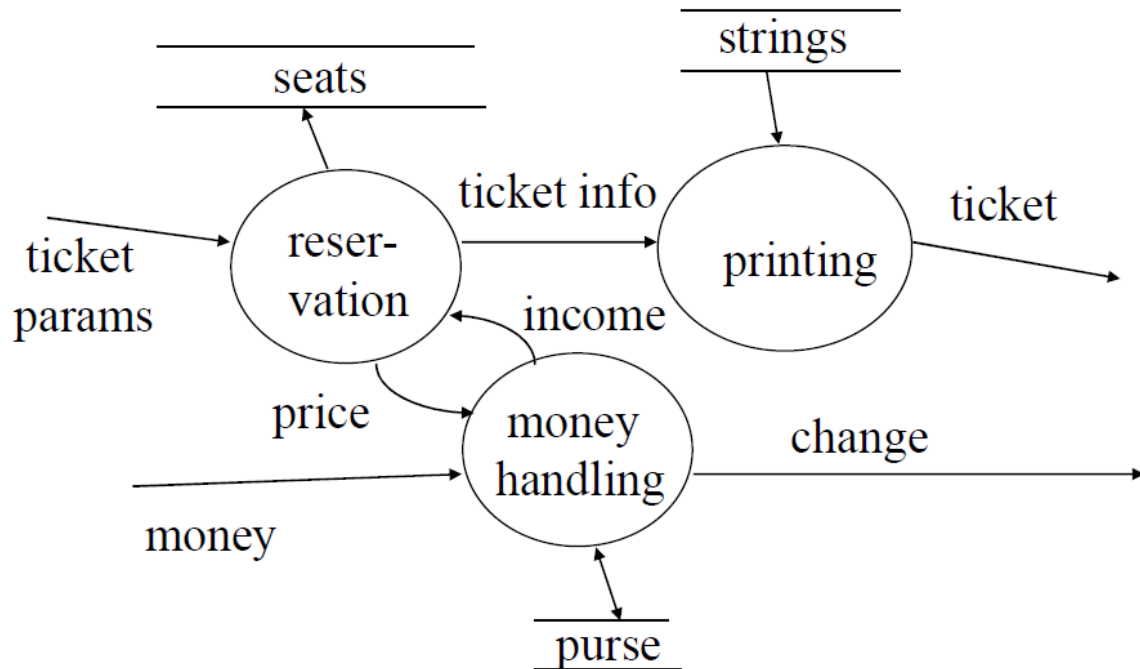
3.1 Példa CD-re



Szabályok:

- CD-n lehet csak terminator
- A CD a legfelső szint, itt csak egyetlen process lehet.

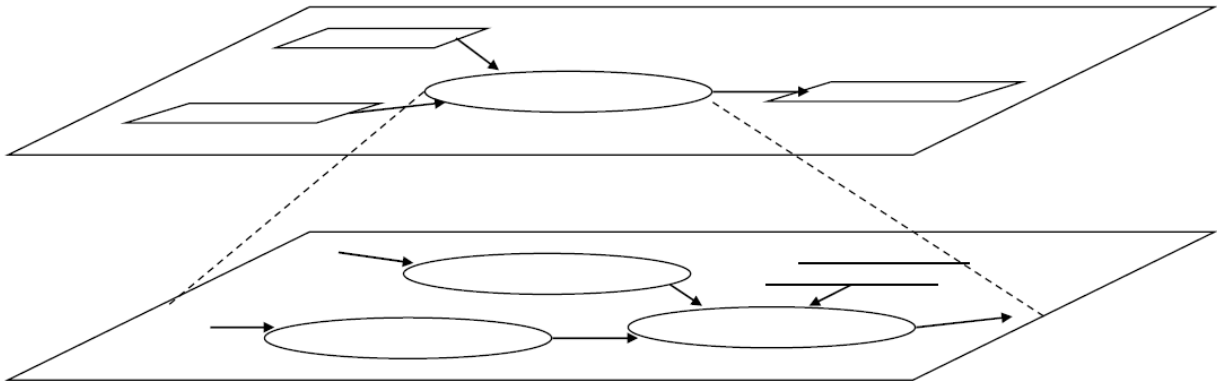
3.2 Példa DFD-re



Szabályok:

- A DFD egy CD process-ét vagy egy magasabb szintű DFD valamelyik process-ét fejt ki részletesebben
- A szabad végű data-flow-k feliratának és irányának meg kell egyeznie az eggyel magasabb szintről kifejtett process be- és kimenő data-flow-ival
- Egy data-flow legalább egyik végének process-hez kell kötődnie
- Két store között közvetlenül nem lehet data-flow, csak process-en keresztül közvetve
- DFD-n nincs terminator
- DFD-n több process is lehet
- Egy store több szinten is előfordulhat

A CD és DFD viszonyát az alábbi ábra szemlélteti:



Az alsó DFD tetszőleges egy-egy process-e további szinteken kifejthető. A kifejtést addig kell végezni, amíg elemi funkcionalitáshoz nem jutunk.