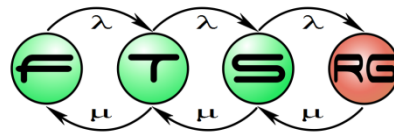


# Szöveges editorok készítése III.

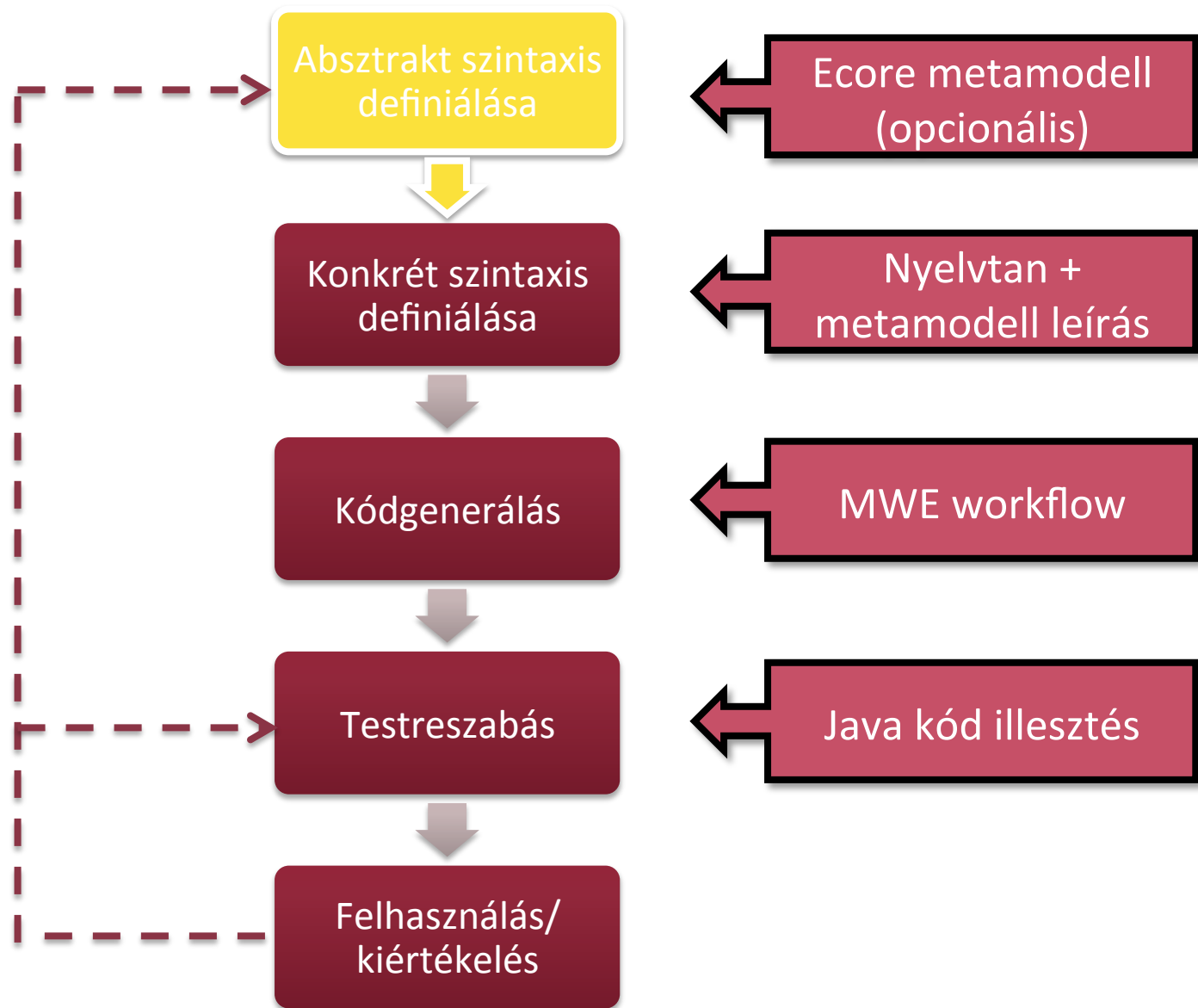
## Xtext – Fejlettebb lehetőségek



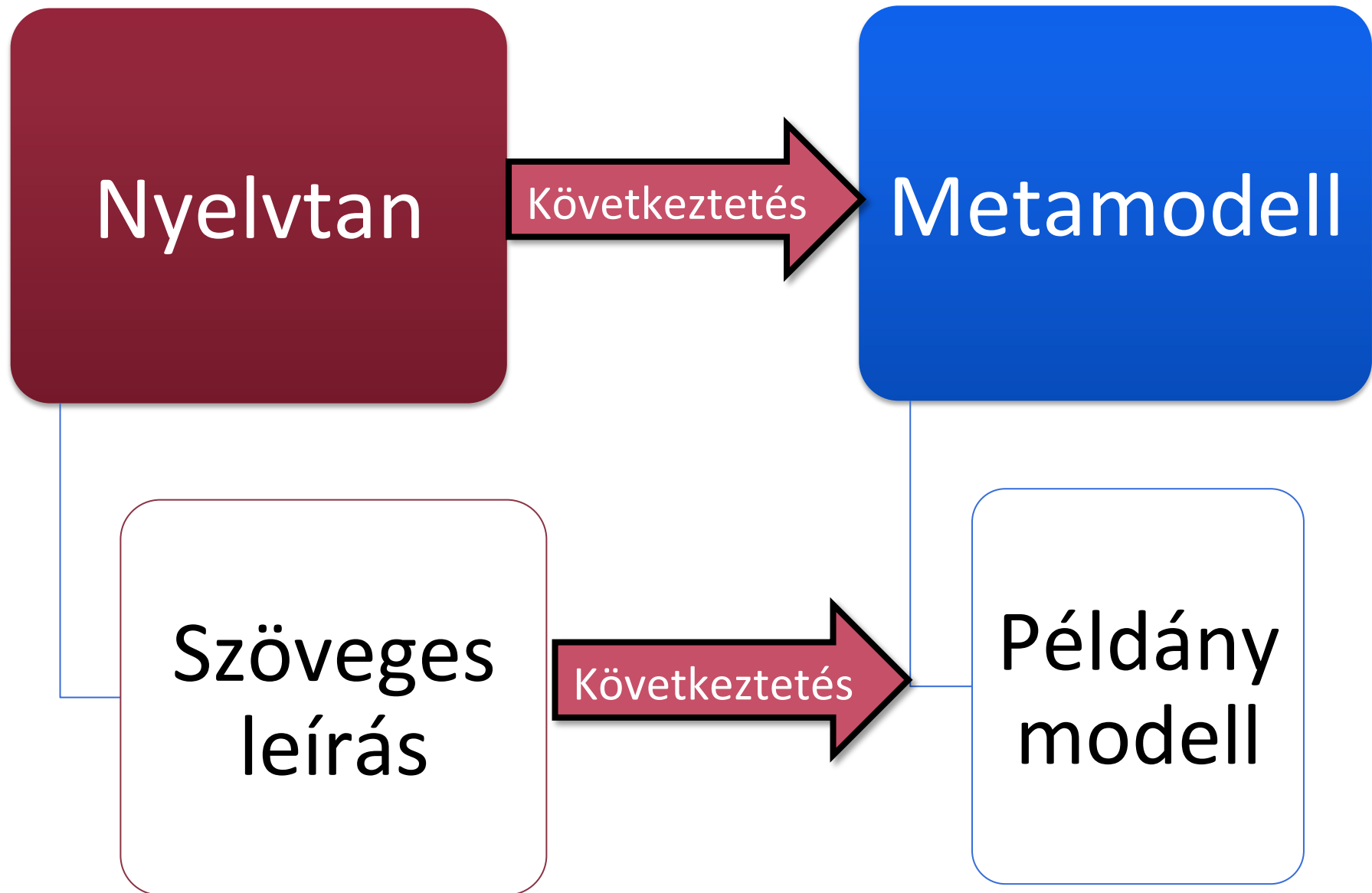
# Mi volt eddig?

- Xtext munkafolyamat
- Nyelvtanspecifikáció
- AST következtetés
- Laboron
  - Egyszerű testreszabás

# Felhasználás menete



# Nyelvek és modellek



# Testreszabás

# Testreszabás

- Szolgáltatásokkal bővíthető
  - **Scoping: linkfeloldás támogatása**
  - **Formatting: automatikus megjelenítés**
  - **Validáció: ellenőrzés**
  - Content Assist: precízebb segítség
  - Labeling: feliratozás (JFace Label Provider)
  - Outline: Outline view testreszabása
  - Quick Fix: javítások kezelése

# Scoping

- Feladat:
  - Változóknak van elérhetősége (scope)
  - Ez nyelvfüggő
  - Alapértelmezés: csak lokális referenciák elérhetőek!

# Scoping szolgáltatás

- Feladat
  - Referencia és hivatkozható elemek összekapcsolása
- Implementáció
  - Többféle megközelítés támogatott
    - Explicit import
      - EMF resource-ok kijelölése
    - Lokális scope
      - Fájlokon belüli scope megadására
    - **Deklaratív**
      - **Metódusnév alapján történő szűrés**
    - ...



# Deklaratív scoping

- AbstractDeclarativeScopeProvider osztály
  - Kétféle generikus metódus
    - `IScope scope_<RefDeclaringEClass>_<Reference>`  
`(ContextType> ctx, EReference ref)`
    - `IScope scope_<TypeToReturn>`  
`(<ContextType> ctx, EReference ref)`
  - Megfelelő metódusok felvehetőek
    - Hasonló az LPG visitorokhoz
    - De nem statikus összerendelés (ősosztályban nincsenek meg a megfelelő metódusok!)

# Deklaratív scoping

- Metódusok
  - Gyakorlatilag felsorolják az elérhető elemeket
  - Paraméterek:
    - Hivatkozás forrása (kontextus, ill. deklaráló elem)
    - Hivatkozás típusa (EClass)

# Declarative scoping: Példa

```
public IScope scope_Person_membership(final Person context,  
EReference reference) {  
    //Naive scoping  
    SocialNetwork network = (SocialNetwork) context.eContainer();  
    Iterable<Community> communities =  
        Iterables.filter(network.getEntities  
            (), Community.class);  
    ArrayList<Community> communityList = new  
        ArrayList<Community>();  
    for (Community community : communities) {  
        addChildren(communityList, community);  
    }  
    return scopeFor(communityList);  
}
```

# Google Collections

- Most már Google Guava
  - <http://code.google.com/p/guava-libraries/>
- Xtext
  - IP okokból a Google Collections használata
  - Korábbi, binárisan kompatibilis változat

# Validáció

- Feladat
  - További feltételek ellenőrzése a beolvasott modellen
  - Megközelítés
    - Tetszőleges EMF alapú eszköz működik (pl. OCL)
    - Xtext-specifikus Java-kódon alapuló validátor

# Java validátor

- Vázlat generál egyszerű mintával
  - Minden ellenőrzési metódushoz @Check annotáció
  - Metódus paramétere egy elem az AST-ből
- Hiba esetén az űsosztály error/warning metódusát kell meghívni

# Validátor példa

```
@Check
```

```
public void noNameCollision(Community entity) {  
    noNameCollision(entity, entity.eContainer().eContents(),  
SocialNetworkPackage.Literals.SOCIAL_ENTITY__NAME);  
}
```

```
private void noNameCollision(EObject eObject, List<EObject>  
siblings, EStructuralFeature nameFeature) {  
    String name = (String) eObject.eGet(nameFeature);  
    for (EObject sibling : siblings) {  
        if(name.equals(sibling.eGet(nameFeature)) && eObject !=  
sibling) {  
            error("Duplicate name", nameFeature.getFeatureID());  
        }  
    }  
}
```

# Formatter

- Cél:
  - Kód automatikus formázása
    - Nem látható karakterek illesztése
- Mikor kell
  - Felhasználó kéri
  - EMF modell dinamikus módosításakor



# Formatter

- Szabályok
  - Mire vonatkozik?
  - Mit hajt végre?
- Ez Java utasításokkal megadható

# Mire vonatkozik

- `after(token)`: token után
- `before(token)`: token előtt
- `around(token)`  $\Leftrightarrow$  `after(token)` és `before(token)`
- `between(token1, token2)`: token1-et követő token2
- `bounds(token1, token2)`  $\Leftrightarrow$  `after(token1)` és `before(token2)`
- `range(token1, token2)`: token1-től token2-ig

# Utasítások

- `setIndentationIncrement`
- `setIndentationDecrement`
- `setLinewrap`
- `setSpace`
- `setNoSpace`

# Formázás példa

```
protected void configureFormatting(FormattingConfig c) {  
    SocialNetworkGrammarAccess access =  
(SocialNetworkGrammarAccess) getGrammarAccess();  
  
    SocialNetworkElements sne = access.getSocialNetworkAccess();  
    c.setLinewrap(1, 1, 1).after(sne.getLeftCurlyBracketKeyword_2());  
    c.setLinewrap(1, 1, 1).before  
        (sne.getRightCurlyBracketKeyword_5());  
    c.setLinewrap(1, 1, 1).after(sne.getEntitiesAssignment_3());  
    c.setLinewrap(2, 2, 2).before  
        (sne.getAcquaintancesAcquaintanceParserRuleCall_4_0());  
    c.setIndentationIncrement().after  
        (sne.getLeftCurlyBracketKeyword_2());  
    c.setIndentationDecrement().before  
        (sne.getRightCurlyBracketKeyword_5());  
}
```

# Szolgáltatások regisztrációja

- Dependency injection
- Google Guice
  - <http://code.google.com/p/google-guice/>
- DI framework az új generációs Eclipse UI alapja is!

# Xbase

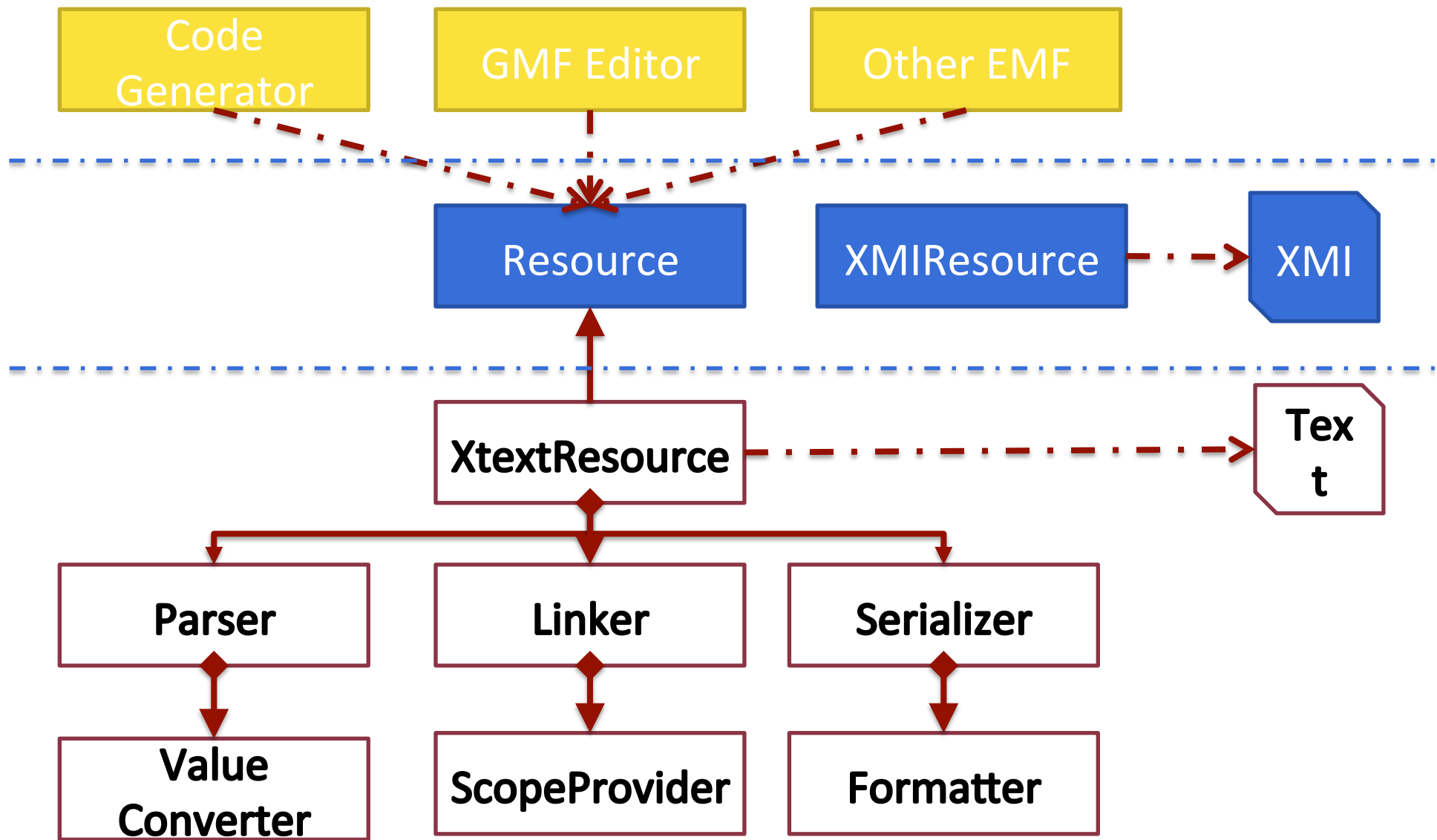
# Xbase

- Base nyelv koncepció
- Újrafelhasználható kifejezésnyelv

# Integráció EMF technológiákkal



# Integráció az EMF technológiákkal



# Integráció

- Xtext
  - Saját EMF resource
  - Ezen keresztül kapcsolódik
    - Nem kell foglalkozni vele, hogy ki sorosítja

# GMF editor

- Domain modell beállítható Xtext erőforrásra
  - Mentéskor automatikus szinkronizáció
- Oda kell figyelni a sorosíthatóságra!
  - GMF nem tud a nyelvtanról
  - Nem kell túl sok mindenre figyelni

# GMF és Xtext editor

The image shows a screenshot of an Xtext editor displaying a GMF diagram. The left pane shows the Xtext grammar for a social network, and the right pane shows the corresponding GMF diagram.

```
test.socialnetwork {
  SocialNetwork {
    Person Ujhelyi {
      male
      memberships BME, VVEC
    }
    Person Horvath {
      male
      memberships FTSRG
    }
    Community BME {
      Community FTSRG {
        Community test
      }
    }
    Person Test {
      female
      memberships test
    }
    Community VVEC
    Person Proba {
      male
    }
    Community Pr2
    Person valaki {
      male
    }
    Ujhelyi is friend of Horvath
    Test is married to Ujhelyi
  }
}
```

The GMF diagram (test.socialnetwork\_diagram) shows the following structure:

- Person Ujhelyi (male)
- Person Horvath (male)
- Person Test (female)
- Person Proba (male)
- Person valaki (male)
- Community BME (containing FTSRG, which contains test)
- Community VVEC

The diagram also includes a Palette on the right with the following elements:

- Community
- Person
- Acquaintance
- Membership

# Kódgenerálás

- Generator projekt
- Xtend alapú kódgenerátor
- Adatok
  - Be: EMF modell
  - Ki: Kód
- Vezérlés: MWE

# Összefoglalás

- Kiforrott technológia
  - Jó alapértelmezések
  - Kényelmes felhasználás
- Probléma
  - Korlátozott metamodell következtetés
  - Balelemzőt kell használni