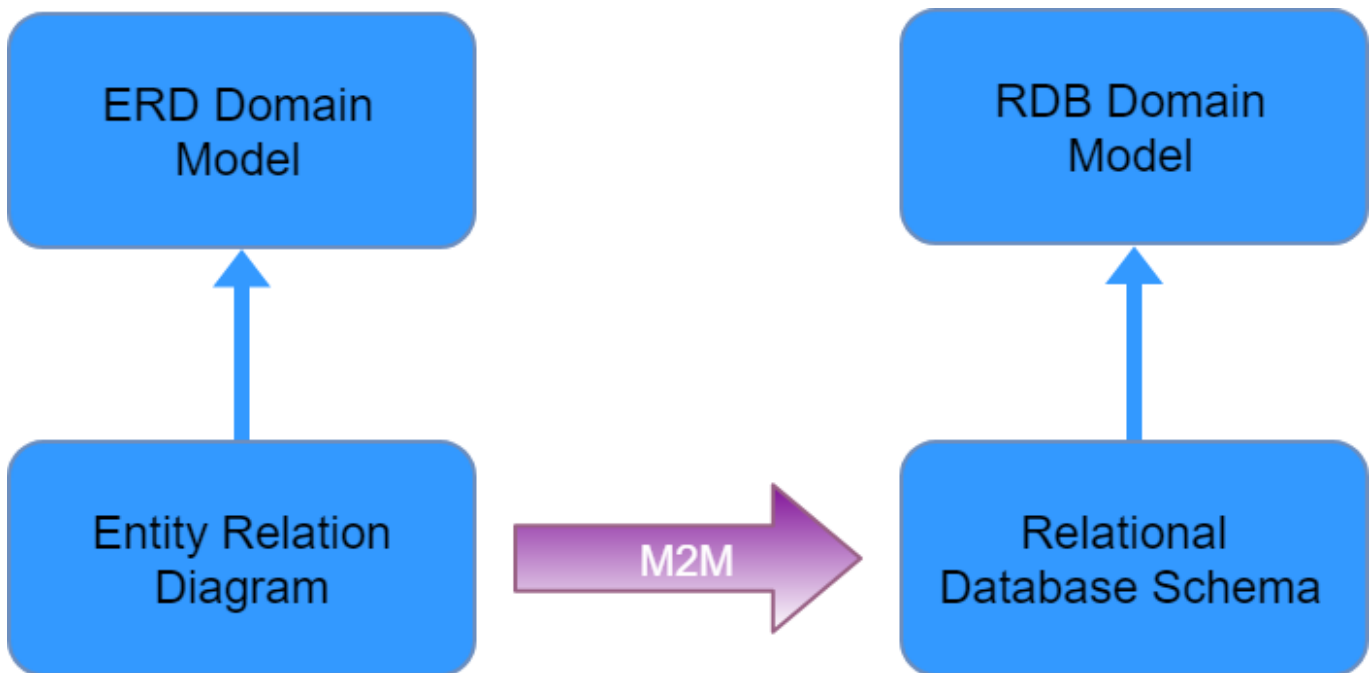


Modell-modell transzformáció

A feladat

Ezen a gyakorlaton példán keresztül mutatjuk meg, hogyan lehet eseményalapú modelltranszformációkat fejleszteni. Ez a transzformáció egy relációs adatbázis séma kialakítását fogja létrehozni (majd folyamatosan frissíteni, amíg aktív) bármely egyed-kapcsolat modellből.



Környezet beállítása

Szükséges Pluginek telepítése

Először is, néhány plugint kell telepítenünk, ha ezek még nem lennének telepítve. A példakód az alábbi verziókkal volt tesztelve:

- Eclipse Modeling 2019-12
- EMF - Eclipse Modeling Framework SDK 2.20.0
- Ecore Tools 3.3.2
- Xtend 2.20
- Sirius 6.3.0
- VIATRA 2.3.0
- EGit 5.6.0

Nyelv projektek beállítása

1. Kezdjünk egy üres Eclipse workspace-szel: zárjuk be vagy távolítsuk el (nem kell letörölni a merevlemezről) az összes projektet az aktuális munkaterületből.
2. Git clone <https://github.com/ftsrg/mdsd-examples>
3. Töltsük le az összes tag-et és álljunk át a `m2m-2020-phase1` tag-re

- Az Eclipse EGit GUI segítségével:
 1. Váltson a Git perspektívára, keresse meg a repository-t a git repositories nézetben (ha a parancssorból klónoztta, akkor hozzá kell adnia)
 2. Jobb klikk a repository-n, Remote > Fetch...
 3. Nyomjuk meg a next gombot, kattintsunk az Add All Tags Spec gombra, majd a Finish gombra.
 4. A repository alatt jobb klikkeljük az `m2m-2020-phase1` elemet, majd hozzunk létre egy új branch-et.
 5. A branch neve legyen `m2m`, és győződjünk meg róla, hogy be van jelölve a "Check out new branch" jelölőnégyzet.
 - Vagy parancssorból: `git fetch --all --tags && git checkout tags/m2m-2020-phase1 -b m2m`
4. Importáljuk a következő metamodelleket / nyelvfejlesztő projekteket (de az összes többi projektet NE):
- `hu.bme.mit.mdsd.erdiagram` - *erdiagram.ecore projekt (azaz a forrás metamodell)*
 - `hu.bme.mit.mdsd.erdiagram.design` - *erdiagram Sirius-alapú szerkesztő definíció*
 - `hu.bme.mit.mdsd.rdb` - *rdb.ecore projekt (azaz a cél metamodell)*
 - `hu.bme.mit.mdsd.rdb.design` - *rdb Sirius-alapú szerkesztő definíció*
 - `hu.bme.mit.mdsd.trace` - *trace.ecore projekt (azaz a megfelelési / követhetőségi metamodell)*
5. Néhány kezdeti fordítási hiba van; mindegyiket fel tudjuk oldani a `.genmodel` fájlok megtalálásával és a "Generate all" parancs kiadásával a következő sorrendben:
- `hu.bme.mit.mdsd.erdiagram/model/erdiagram.ecore`
 - `hu.bme.mit.mdsd.rdb/model/rdb.ecore`
 - `hu.bme.mit.mdsd.trace/model/trace.ecore`

► Nézze meg a videót ►

Transzformációs projektek beállítása

1. Indítsa el a runtime Eclipse-et, majd zárja be vagy távolítsa el (nincs szükség a lemezről való törlésre) az összes projektet az aktuális workspace-ből. (Megjegyzés: lehetséges a host Eclipse-ben folytatni, de ebben az esetben a VQL szerkesztő nagyon lassú lenne, mivel az azonos workspace-ben fejlesztett metamodelleink dolgoznának.)
2. Importálja a transzformációs projekteket:
 - `hu.bme.mit.mdsd.m2m2c` - *az alapvető modell-to-modell transzformációs kód*
 - `hu.bme.mit.mdsd.m2m2c.ui` - *felhasználói interfész kötések az említett transzformációk meghívásához*
3. Clean és build után nem kellene, hogy fordítási hiba legyen.
4. Hozzon létre egy új "Eclipse Application" launch konfigurációt, nevezze el például "RuntimeEclipse2"-nek, jelöljön meg egy munkaterületi mappát (pl. `${workspace_loc}/../runtime2-workspace`) és indítsa el. Győződjön meg róla, hogy a workspace üres.

► Nézze meg a videót ►

Felhasználói környezet beállítása

1. Importálja az [hu.bme.mit.mdsd.erdiagram.examplediagrams](https://github.com/bme-m2m/mdsd-erdiagram-examplediagrams) példaprojektet a Git munkapéldányból.
2. Váltson a Modeling perspektívára. A Model Explorerben nyissa le az forrásmodellfájlt ([My.erdiagram](#)), kattintson a modell gyökérelemére, majd válassza ki az "Prepare EDT" lehetőséget. Ez inicializálja a rdb sémát és az trace (követhetőségi) modellfájlokat.
3. Győződjön meg arról, hogy mind az ERD és az RDB példánymodell rendelkezik Sirius diagramokkal (ha hiányzik, jobb klikk > new representation). Nyissa meg mindkét diagramot egymás mellett.
4. Ismét a Modell Explorerben, jobb-kattintson a modell gyökérelemére, és válassza ki a "Start EDT" lehetőséget. Látható lesz a modelltranszformáció működés közben: a betöltés miatti rövid késleltetés után modellelemek jönnek létre a cél modellben.
5. A cél modell (RDB) diagram eszköztárából használja az első két gombot (Select All, majd Arrange) annak érdekében, hogy a Sirius megpróbálja elrendezni a diagramot. Látható lesz, hogy az minden Entitáshoz van egy hozzá tartozó Tábla, amelynek ugyanaz a neve, de az attribútumok még nem lettek leképezve (bár a Táblának lesz egy automatikusan generált alapértelmezett kulcsoszlopa, ha a megfelelő Entitásnak nincs attribútuma, amelynél isKey=true). Hasonlóan az "is-a" kapcsolatok és a Relációk még nem lettek leképezve.
6. Győződjön meg róla, hogy az implementált leképezések működnek és a transzformáció eseményvezérelt: kísérletezzen a forrásmodell módosításával, és az eseményvezérelt transzformáció leállításával és újraindításával ("Finish EDT" és "Start EDT" az ERD modell gyökérelemének kontextusmenüjéből a Model Explorerben). Lásd még a standard kimenetre naplózott eseményeket.

► [Nézze meg a videót](#) ►

Feladatok

1. Ismerkedjen meg a transzformációval kapcsolatos implementációval, különösen nézze meg a legfontosabb fájlokat:
 - mindkét .vql fájlt a Viatra lekérdezésekhez
 - ErdToRdbRuleSet.xtend a tényleges transzformációs szabályokhoz
 - TracingHelperUtility.java segédfüggvények a nyomkövetési modell manipulálásához
 - CommonHelperMethods.xtend egyéb hasznos statikus segédfüggvények a leképezésekhez
 - ErdToRdbLiveTransformation.xtend maga a transzformáció belépési pontja
2. Bővítse ki az eseményvezérelt transzformációt a hiányzó szempontokkal (kulcs- és nem kulcsattribútumok, "is-a" kapcsolatok, relációk végződésekkkel). Bár nem szükséges, megengedett a nyomkövetési metamodell módosítása (több nyomkövetési objektumtípus bevezetése, vagy többféle hivatkozás, vagy a nyomkövetési objektumok attribútumainak bevezetése), vagy prioritások bevezetése a transzformációs szabályokhoz (ld. kikommentelt sorok az ErdToRdbLiveTransformation.xtend fájlban). Megnézheti az [m2m-2020-phase2](#) és az [m2m-2020-phase3](#) tag-eket a megoldásokhoz.

Hivatkozások és további olvasnivalók

- Xtend nyelvi elemek (beleértve a sablonokat) - https://eclipse.org/xtend/documentation/203_xtend_expressions.html

- VIATRA Query API - <https://www.eclipse.org/viatra/documentation/query-api.html>
- VIATRA Transformations API - <https://www.eclipse.org/viatra/documentation/transformations.html>