

Eclipse alapú fejlesztés és integráció

Házi feladat telepítés

Hellner Gábor (XPAW4D)

Eclipse Android-os fejlesztések támogatásához:

Telepítés

A telepítés és a használat alapvetően két féle képen történhet. A *plug-in* fejlesztés közbeni használatához érdemes beimportálni a fejlesztőeszköz forrásprojektjeit az *Eclipse*-be (9 darab projekt). Ezek közül a **.design* projekten kívüli *componentmodel* projekteket kell a *Host Eclipse*-be telepíteni (Export -> Plug-in development -> Deployable Plug-ins and fragments -> Install into Host). Ezek után pedig egy *Runtime Eclipse*-et indítva a fejlesztőeszköz használható a *Runtime IDE*-ben. Viszont, végfelhasználóként a legegyszerűbb úgy a használata, ha az összes fejlesztőeszközbe importált projekteket telepítjük a *Host Eclipse*-be.

Használat

A fejlesztőeszköz alapvetően három részből áll. Egy adatmodellező fejlesztőeszközből, mely a szakdolgozatom alatt készített *Eclipse plug-in*, ami némi átalakításon ment át a házi készítése során. Használatához egy sima *Eclipse*-es vagy *Java*-s projektet kell indítani és az *src* mappában **.dm* fájlokban lehet megadni az adatmodellt szöveges szintaxissal és egy **.conf* fájlba kell megadni a célprojekt konfigurációit. A kódgenerálás és az exportálás a kontextus menü *Modeling* füléről indítható el.

A komponensmodellező eszköz ugyanezt a sémát használja, csak a modellfájlok kiterjesztése **.cm* és a *Java* típusok használatához a nyelvben érdemes *Java*-s projektbe tenni a forrásfájlokat. A kódgenerálás és az exportálás ugyanabból a menüből érhető el.

A komponensmodellező eszköz ki van egészítve grafikus szintaxissal is, mely a szöveges, illetve exportált modellek egy grafikus reprezentációját képes mutatni és azt szerkeszteni. Ennek eléréséhez a *Sirius* perspektívára kell váltani és ott a projekt kontextus menüében a *Configure -> Convert to Modeling* projekt-re kell kattintani, majd pedig újra a kontextus menüből ki kell választani a *Viewpoints Selection* opciót és ott ki kell választani a *Componentmodel Viewpoint*-ot.

Ezek után a projektben található összes modell, akár **.cm*, akár exportált **.xmi* formában, elérhető a grafikus szerkesztővel. A megadott modellt a projektben ki kell addig bontani, amíg egy *Content* elemhez nem érünk és erre kattintva a kontextus menüből ki kell választani a *new Representation -> new Component Diagram* elemet. Ha ez megtörtént, akkor a modell elemekhez létrejön a grafikus megjelenítése a modellnek, amiben egy szerkesztőn keresztül módosíthatjuk is a modellt, aminek eredménye rögtön látható a forrásfájlokban.

A modellből generált kód felhasználása a *Generation Gap* mintán alapul, tehát a generált osztályokat a belőlük történő leszármaztatással lehet testre szabni. A komponensek indítását a megfelelő függvények meghívásával lehet elérni (melyeknél néhol szükséges paraméterül átadni a leszármazott osztály típusát), illetve a működés testre szabását a megfelelő függvények felüldefiniálásával lehet elérni. A *Fragment*-ek esetén a leszármaztatott osztályokat be kell regisztrálni az *Application* osztályon keresztül az *onCreate* metódusban a megfelelő generált függvénnyel.

Szükséges eszközök

- *Eclipse Luna* szükséges hozzá a *Java 8* és a *Sirius plug-in* miatt.
- A fejlesztőeszközök az *Xtext* és *EMF* keretrendszerre építenek így ezen *plug-in*ek megléte szükséges az *Eclipse*-ben (ezek mind részét képezik az *Eclipse IDE for Java and DSL Developers* disztribúciónak, az *Eclipse Luna*-hoz kiadottat érdemes használni).
- Ezeken felül a grafikus szerkesztőhöz szükséges a *Sirius plug-in* is.
- És mivel *Android*-os fejlesztéseket támogatnak az eszközök szükséges az *Android Developer Tools plug-in* (nincs függőség, de generált kód hasznosításához szükséges)

Megjegyzések

Az összességében megalkotott *Android*-os fejlesztőeszköz tehát három eszközrészletet tartalmaz (adatmodellező eszköz, komponensmodellező eszköz és ehhez a grafikus szerkesztő). Ezek közül az adatmodellező eszköz teljesen függetlenül használható a komponensmodellező eszköztől. A komponensmodellező eszköz is ugyanígy fordítva, de az képes a modellből a generált adatmodellre hivatkozni, így együtt összehangolva is használhatók. De a grafikus megjelenítő csak a komponensmodellezővel együtt tud működni.