Felhasznált technológiák

A szakdolgozat elkészítése alatt a legtöbbet használt eszköz az Eclipse nevű platform volt. Az Eclipse egy széles körben használt integrált fejlesztőkörnyezet (integrated development environment, IDE), melyben alkalmazásokat fejleszthetünk Java vagy egyéb programozási nyelven. Az Eclipse fejlesztője, az Eclipse Foundation azonban az Eclipse IDE-n kívül, az Eclipse platformot is fejleszti.

Az Eclipse platfrom biztosítja az alapokat az Eclipse IDE működésének. Az Eclipse platform plug-inek formájába épül fel, vagyis személyre szabhatjuk, melyik plug-inokat szeretnénk használni. Létezik plug-in a különböző programozási nyelvekhez, pl. C/C++, Python, Ruby, de akár fejlesztést segítő plug-ineket is letölthetünk, pl. Git vagy LaTex. 2001 óta az Eclipse platform szabad, nyílt forrás kódú szoftver, jelenlegi licence az Eclipse Public License. Ennek következményeként az elérhető plug-inek száma növekszik. Az Eclipse Foundation 2006 óta minden év júniusában kiadják az Eclipse platform egy új verzióját, melyek mindig valamilyen tudományhoz kötődő elnevezést kaptak.

Modellezés szempontjából az Eclipse egyik fontos plug-inja az Eclipse Modeling Framework, röviden EMF. Az EMF projekt modellek készítését, leírását és kódgenerálást tesz lehetővé. Az EMF főbb eleme egy metamodell, amit Ecore modellnek is neveznek. Ezt az Ecore modellt más metamodellek létrehozására tudjuk használni, ebből kifolyólag, rengeteg más modellezési Eclipse plug-in az EMF-et használja alapjául.

Az egyik ilyen plug-innal foglalkozok szakdolgozatom keretein belül. Ennek a neve Xtext, és kifejezetten DSL-ek implementálására készítették. Az Xtext használható parser generátorként, de e funkción kívül sok egyébbel is rendelkezik. Legfontosabb közülük az EMF-fel való integrációja. Egy DSL nyelvtanának elkészítése után, ami legtöbb esetben egy szöveges objektum, az Xtext képes azt feldolgozni és működőképes EMF Ecore modellt készíteni belőle. Ez a művelet az Xtext legmeghatározóbb képessége, ezzel gyorsabbá és gyorsabban tanulhatóvá teszi a modellkészítés lépését, és a szoftver fejlesztésére szánt erőforrásokkal is gazdaságosabban bánhatunk. Ami még lényegesebb, hogy egy Ecore modell meglétével az Xtext felhasználhatja az EMF egyéb funkcióit is, például a kódgenerálást.

Az Xtext és az EMF a fejlesztők számára értékes Java osztályokat generálnak, melyek az általunk írt DSL nyelvtan reprezentációja, de egyéb osztályok is készülnek. Azonban fontos megjegyezni, hogy az Xtext fő célja nem osztályok létrehozása, hanem egy Eclipse plug-in generálása, amelyben az általunk létrehozott DSL nyelvben vagyunk képesek programozni. Az Xtext által generált egyéb osztályok ehhez a személyre szabott plug-inhez tartoznak és a plug-inben történő programozást segítik. Például egyik osztály a content assistot hozza létre a DSL nyelvünkben, míg másik a refaktorálással foglalkozik.

A DSL, mint modell leképezésével létrejövő Java osztályok a DSL-ben megírt kód feldolgozását segítik. A fejlesztőkre van bízva, milyen műveleteket végeznek ezekkel az osztályokkal/objektumokkal, de gyakori módszer a kódgenerálás (vagy bármilyen dokumentum generálása, a továbbiakban a kódgenerálás magába foglalja ezeket is). Ehhez is nyújt segítséget az Xtext, osztályok és a Java nyelv kódgenerálás központú változatával, melynek neve Xtend.

Egy kódgeneráló programrészlet nagyon átláthatatlan tud lenni. Ugyanis rengeteg kiírató függvényt tartalmaz, melyek argumentumaiban csakugyan programparancsok találhatóak. Ezt a problémát oldja meg egy sémanyelv használata, mely a megírt kódot valamilyen kimenetre irányítja. Egy ilyen sémanyelv az Xtextnél használatos Xtend is, melyben objektum orientált programokat lehet írni, többnyire kódgenerálás céllal. Az Xtend letisztult, „zajmentes” Javanak tekinti magát, ahol a Javaban lévő szintaxisok elhagyhatóak, például a parancs végi pontosvessző. Ezzel a kód valóban átláthatóbb lesz, de némi problémát okozhat egy adott programsor értelmezése. Érdekesség, hogy az Xtendet az Xtextben készítették el és Java nyelvet generálnak belőle.

A szakdolgozatom alatt egyéb technológiákat is felhasználtam, melyek a fejlesztéshez nem kapcsolódnak szorosan. A feladatot Windows 10 64-bites operációs rendszeren végeztem el. Verziókövetéshez Gitet és Githubot használtam. A generált C programokat Codeblocksban szerkesztettem és futtattam. Dokumentálást Microsoft Wordben készítettem.