



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Multiplatform mobil alkalmazások generálása

Fénykép
helye
2x3 cm

Lóránd Bálint Tamás (H1B5Y9), I. évf, (MSc) mérnök inf. szakos hallgató
Konzulens: Ráth István tudományos munkatárs, MIT
Szolgáltatásbiztos rendszertervezés szakirány
Önálló laboratórium 1 összefoglaló
2011/12. II. félév

Napjainkban az okostelefonok robbanásszerű elterjedése megnövelte az igényt a mobilalkalmazásokra, viszont az ezzel járó gyorsabb fejlesztési ütem a felszínre hozta a hagyományos, natív eszközökkel történő fejlesztés hátrányait. Kezdetben egy új programnyelv megtanulása, majd a nagy komplexitás és sok hibaforrás mind mind időt követel a fejlesztőtől. Ha pedig meg akar felelni a mai trendeknek és multi-platform megoldást szeretne, akkor minden rendszeren meg kell küzdenie ezekkel a problémákkal. Továbbá egy modern szoftver felépítését tekintve a használt függvénykönyvtárakon és keretrendszereken túl a manuálisan írt kód egy része mindig egy sémára épül. Mivel ez magasabb szinten is megragadható, érdemes elemezni az előforduló elemeket, fogalmakat majd egy szakterület-specifikus nyelvből és kódgenerátorból álló céleszközt készíteni a feladatra. Ezzel az alkalmazásfejlesztőknek már csak a lényegre, saját ötletük megvalósítására kell koncentrálniuk.

Ezt az elvet tartották szem előtt az Eclipse alapú Applause¹ mobilalkalmazás generáló eszköz alkotói, mely segítségével többek közt iOS platformra készíthetünk alkalmazásokat mindössze egy modell felépítésével. Ehhez egy Xtext² szabályokból álló modellezési nyelvet kell használni, majd egy Xpand³ sablonokból álló kódgenerátor kész Xcode⁴ projektet generál nekünk, mellyel egy már azonnal fordítható és futtatható iOS alkalmazást kapunk.

Hogy magam is kipróbáljam, milyen az Applause-zal dolgozni, készítettem vele egy iOS tesztalkalmazást. Mivel elég korlátozottak a lehetőségek - lényegében csak webről letöltött XML fájlok adatait megjelenítő szoftver konstruálható - a napiarfolyamok.hu API-ját felhasználva egy árfolyam megtekintő alkalmazást modelleztem, és generáltattam.

A félév következő lépéseként, hogy megismerkedjem az Applause-t működtető technológiával, és megtaláljam a korlátait, hibáit, egy bővítést készítettem hozzá. Ehhez felvettem, mint probléma, hogy az ezzel az eszközzel generált alkalmazások nagyon hasonlítanak egymásra. Miután megvizsgáltam az Applause-t alkotó projektek felépítését, kiegészítettem azokat a testreszabható színezés lehetőségével. Új szabályokkal bővíttem az Applause modellezési nyelvet, majd kiegészítettem kódgenerátor sablonjait a szükséges kódrészletekkel.

Miután mind felhasználói, mind fejlesztői szempontból megvizsgáltam az Applause-t, pozitív tapasztalatként értékeltem, hogy a lehetőségek korlátain belül jó eredményt nyújt, működő alkalmazásokat generál, miközben használata kényelmes és egyszerű. Azonban ez hosszútávon eltörlődik amellyel, hogy a projekt stagnál, már nem fejlesztik aktívan. Az egyes platformokra használt natív UI függvénykönyvtárak használata pedig a hatékonyság rovására megy, mivel nagyon lekorlátozza a lehetőségeket, miközben megnehezíti az eszköz fejlesztését. Nem megoldott továbbá a generált alkalmazások kiegészítése manuálisan írt kóddal.

A félév során szerzett tapasztalatok alapján végül kialakult a terv egy új, saját eszköz fejlesztésére a következő félévekben, melyhez több lehetséges út áll előttem. Dönthetek úgy, hogy magam is Eclipse alapú rendszert készítek, de már modernebb technológiákat felhasználva, akár a template alapú kódgenerálás helyett absztrakt szintakszisfa alapú megoldást készítve. Azonban ebben az esetben érdemes lehet a JetBrains Meta Programming System⁵ szoftverét választanom egy szerkesztő létrehozására, majd ezt egészíthetném ki egy úgynevezett modellépítés módszerrel működő generátorral. Emellett megfontolandó a natív UI függvénykönyvtárak helyett OpenGL alapút használni a generált alkalmazásokhoz, hogy ezáltal a későbbiekben egyszerű legyen több célplatformra is kódgenerátort készíteni.

¹ Applause - <http://applause.github.com>

² Xtext - <http://www.eclipse.org/Xtext/>

³ Xpand - <http://www.eclipse.org/modeling/m2t/?project=xpand>

⁴ Xcode - <https://developer.apple.com/xcode/>

⁵ JetBrains MPS - <http://www.jetbrains.com/mps/>