



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Szakterület-specifikus nyelvek használata alkalmazásmenedzsment feladatokra



Nyiri András I. évf., (MSc) mérnök-informatikus szakos hallgató
Konzulens: Szatmári Zoltán, MIT
Szolgáltatás biztos rendszertervezés ágazat
Önálló laboratórium 2. összefoglaló
2014/15. II. félév

A feladat célja az volt, hogy megvizsgáljam az iparban jelenleg elterjedt domain-specifikus nyelveket és közülük néhányat kiválasztva készítsem el egy egyszerű számítási felhőben telepíthető alkalmazás leíró nyelvét. A nyelv elkészítésénél fontos szempont volt az olvashatóság, szintaktikai elemek egyszerűsége és minél kevesebb kódismétlődés megjelenése, mivel az ezen a nyelven íródott alkalmazás leírók olvashatóak és könnyen megérthetőek kell, hogy legyenek emberi szem számára is, nem csak gépi értelmezésüknek kell egyszerűen kivitelezhetőnek lenni.

A feladat elvégzéséhez rendelkezésre álltak a külső, már megvalósult projectben elkészült XML formátumú alkalmazás leírók. Ezek már terjedelemben és tartalomában is kezdték kinőni azt a domain teret és felhasználási kört, amelyet XML formátumban még kényelmes manuálisan szerkeszteni, átlátni. Megjelent tehát az igény egy újfajta megközelítésre.

A megvalósítás során meg kellett ismerkednem többféle domain-specifikus nyelvvel is. Megállapítottam, hogy alapvetően kétféle fő megközelítés létezik a kivitelezés szempontjából. Az egyik egy alulról-fölfele architektúra szerint egy már létező, erős nyelvi elemeket tartalmazó (általában script) nyelv fölött valósítja meg a domain-specifikus nyelvek számára nélkülözhetetlen absztrakciós képességeket. Ez azt jelenti, hogy ezen erős nyelvi eszközök segítségével rejti el az elkészült domain-specifikus nyelv olvasójának szeme elől a háttérben megvalósuló műveleteket, ezzel elősegítve a könnyű érthetőséget és olvashatóságot. A másikatípus megközelítés egy felülről-lefele architektúra szerint működik. Itt egy magasabb absztrakciós szinten történik a domain specifikációja egy modell létrehozásával. Ebből a modelltől történhet aztán a nyelvi elemek generálása automatikus eszközökkel.

Az első megközelítési formát ismertem meg először a Groovy programozási nyelv tanulmányozása során. A Groovy egy, a Java virtuális gépre épülő különálló programozási nyelv, amely azonban a Java nyelvi elemein túl mutató, magasabb szintű és erősebb elemeket is tartalmaz. Legerősebb eszköze az anonim kód blokkok írásának lehetősége (closure) aminek a segítségével futás időben helyezhetőek más kontextusba a futtatni kívánt kód sorok. Részben ezen eszköz képességeit kiaknázva hoztam létre az egyik prototípus alkalmazás leírót és a hozzá tartozó programot. Ezzel az elkészült programmal szimulálható is a felhős alkalmazás létrehozása.

A második megközelítési formát az Xtext keretrendszer kipróbálásán keresztül ismerhettem meg. Habár az Xtextben is lehetőség van manuálisan specifikálni a nyelvi szabályokat, a feladat szempontjából célszerűbb volt EMF modelltől automatikusan generáltatni a nyelvet. Ehhez készítettem egy egyszerűsített alkalmazás Ecore modellt, majd ebből állítottam elő a prototípus értelmezőjét. Majd ehhez is írtam egy alkalmazás leírót.

A feladatot a két nyelv összehasonlításával és kiértékelésével zártam. Összességében az Xtextet szigorúbbnak és nagyobb odafigyelést igénylőnek találtam. Míg a Groovy esetében sokszor körültekintően kell eljárni, hogy mi az, amit érdemes megengedni az implementáció során, mivel igen nagy szabadságot ad a fejlesztő és az alkalmazás leíró készítőjének a kezébe is. Az Xtext előre elkészült integrált fejlesztői eszközöket ad Eclipsehez, ugyanakkor a Groovynak pont a fejlesztői eszköz függetlenség válik az előnyére.

A továbbfejlesztés során esettanulmányt célszerű végezni az alkalmazás leírók készítőjének körében az elkészült domain-specifikus nyelvek használhatóságát illetően. Majd nagyobb alkalmazások leíróját is el kell készíteni és az értelmezőt az alkalmazás menedzselő szoftverhez kell integrálni.