

Beágyazott rendszerek modell alapú fejlesztése

Önálló laboratórium feladat összefoglalója (2. félév)

Kadlecsik Ferenc Péter (FQN5OS)

Konzulens: Balogh András

**BME Méréstechnika és Információs Rendszerek Tanszék
Informatikai infrastruktúra tervezése szakirány, 2007/2008. II. félév**

A világ egyre inkább megköveteli az autonóm és mindenhol jelen lévő informatikai rendszerek létét. Ezek a rendszerek mind a beágyazott informatika és elektronika termékei.

Az ilyen rendszerek tervezése és üzemeltetése a növekvő bonyolultság mellett nem kis feladat. Az előbbihez jön még a követelmények és általános felhasználói elvárások szigorodása is. Külön kiemelt terület a biztonság kritikus alkalmazások, ahol nagy rendelkezésre állás, hibatűrés és megbízhatóság kap fokozottan kitüntetett szerepet.

A mérnöki gyakorlatban számos módszertan segíti a beágyazott informatikai rendszerek fejlesztését. Ilyen módszertan a modell alapú tervezés (Model Driven Architecture, MDA). Megfigyelhető az előbb említett módszertanon, hogy a fejlesztői szerepkör egyre inkább a specifikálás és konfigurálás irányába tolódik el.

A nagyfokú integráltságnak köszönhetően egyre gyakrabban beszélhetünk „System-On-a-Chip”-ekről (SoC), melyek az adott követelményeket megvalósító eszközök egy chipen történő integrálásával jöttek létre. Ilyen SoC rendszerek egy kommunikációs csatornára kapcsolva alkothatnak egy beágyazott informatikai rendszert.

Az önálló laboratórium témájaként a Matlab Simulink-kel megalkotott SoC-k modelljeinek feldolgozásával foglalkoztam, mely alapján konfigurációk lesznek generálhatóak, valamint az átjárhatóság biztosítását tűztem ki céloknak az Eclipse-es világ és a Matlab Simulink között.

Az első feladat a Matlab Simulink megismerése volt. A fájlformátum megvizsgálása után a következő lépés egy megfelelő parser megalkotása volt, amihez két eszközt is kipróbáltam. Az első a JavaCC volt, ami nem vált be a célunk annyira, mint a második kipróbált parser generátor, az LPG, amely végül alkalmazásra került.

A parser egy bemeneti fájlban lefutott eredményként jön létre egy AST (Abstract Syntax Tree). A következő fázis ezen AST értelmezése, a fogalmaknak megfelelő modell építése.

Az munkám elején egy absztrakt modellt terveztem meg, melyet felhasználhatok a későbbi transzformációk során és a Simulink fájlba történő visszafelé íráskor is. Ez nem bizonyult előnyös megoldásnak. Jobbnak találtam ha, nem egy, hanem két modellt épül: egy a Simulink fájl AST-je alapján, a második a kapott első modellt felhasználva.

Az első felépülő modell magának a fájlban a modellje. A második absztraktabb modell tükrözi az igazi felhasználási célt, a beágyazott rendszerek tervezését.

A modellezéshez az EMF (Eclipse Modelling Framework) Eclipse plugin-t használtam, tekintve, hogy a cél alkalmazási környezet is Eclipse alapokon működik. A két modellt egy-egy „modelbuilder” osztály építi fel: az egyik az első modellt az AST alapján, a másodikat a másik az első modell alapján.

A második önálló laboratórium keretében megismertem a Matlab Simulink-et, annak fájlformátumát, a JavaCC és az LPG parser generátorokat, az EMF-et. Elkészítettem a szükséges nyelvtanokat, megvalósítottam alkalmas parser-eket, megterveztem a szükséges metamodelleket illetve implementáltam az egyes modelleket felépítő Java osztályokat, bizonyos kódgenerálási képességekkel. Mindezek mellett folyamatosan figyelembe vettem a jövőbeli felhasználási lehetőségeket.