



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## Szabványos viselkedésleíró-nyelv (ALF) szemantikai vizsgálata



**Horváth Benedek, I. évf, (MSc) mérnök inf. szakos hallgató**  
**Konzulensek: dr. Micskei Zoltán adjunktus, MIT és**  
**Vörös András tudományos segédmunkatárs, MIT**  
**Kritikus rendszerek főspecializáció**  
**Önálló laboratórium 1 összefoglaló**  
**2014/15. II. félév**

A modellvezérelt szoftvertervezés során a követelményekből és a specifikációból szisztematikusan származtatják a mérnöki modelleket, melyekhez modell-transzformációkon keresztül formális verifikációval támogatott helyességellenőrzés nyújtható.

A modellvezérelt tervezés másik előnye, hogy a tervezés korai fázisától kezdve a modellek végrehajtásával, szimulációjával ellenőrizhetőséget és tesztelési lehetőségek biztosíthatók, ezáltal a hibák késői felfedezéséből és kijavításából adódó többletköltségek megelőzhetők.

Viselkedésleíró modellek (pl. állapotgép, aktivitás diagram) végrehajtásához az ALF (*Action Language for Foundational UML*) célszerű választásnak tűnhet. Ahhoz, hogy a szimulációk helyesen elvégezhetők legyenek, szükséges az ALF szemantikájának ismerete.

Az ALF az UML aktivitás diagramokhoz és egyéb dinamikai elemekhez szemantikát definiáló részének, az fUML-nek egy konkrét (szöveges) szintaxisa. Mivel az ALF az fUML egy konkrét szintaxisa, ezért az ALF szemantikáját az egyes nyelvi elemeknek megfelelő fUML konstrukciókra való leképezésével (*mapping*) definiálták. Ennek ellenére önálló laboratórium témám során több ellentmondást és hiányosságot is találtam az egyes nyelvi elemekkel kapcsolatban.

Az ALF nyelvi elemei közül elsősorban az utasításokra (*Statements*) fókuszáltam, melyeket több csoportra lehet osztani: annotációk, in-line utasítások, blokkok, üres utasítások, helyi név deklarációk (*Local Name Declarations*), kifejezések, feltételes elágazások, ciklusok, függvény visszatérések, esemény és jelzésre várakozás, helyi név típusának megváltoztatása.

Az ALF *Model Driven Community* által készített referencia-implementációjában próbáltam ki az ellentmondásosnak talált konstrukciókat, mint például:

- a *helyi nevek* különböző deklarációs formái miatt könnyen összekeverhetők a változó értékadással, így fejlesztőeszköz készítése esetén nehéz feladatot jelent a programozónak visszajelezni az esetleges *helyi név* elgépeléseket
- a hagyományos programozási nyelvekben (Java, C, C++, C#, Python) megszokott változó láthatósági szabályoktól (*scoping*) teljesen eltérő ALF-ban az egyes *helyi nevek* („változók”) láthatósága, pl. blokkban létrehozott „változók” a blokkon kívül is láthatóak
- feltételes elágazó utasításoknál lehetőség van konkurensen kiértékelődő feltétel ágakat is felvenni, melyekben lehet a C-szerűen értéket adni egy változónak a feltétel kiértékelése során. Emiatt nemtriviális, hogy két konkurens feltételben használt változónak mi lesz az értéke az egyes ágakban.

A feltárt hiányosságokat és ellentmondásokat tervezzük visszajelezni az implementációt fejlesztő közösségnek, illetve a szemantikai ellentmondások feltárása után tervezem a BSc Önálló laboratórium során fejlesztett ALF – UPPAAL leképezés továbbfejlesztését és pontosítását.