

Modelltranszformációk statikus ellenőrzése

Ujhelyi Zoltán V. Inf., uz602@hszk.bme.hu

**Konzulensek: Horváth Ákos (BME-MIT, ahorvath@mit.bme.hu),
Dr. Varró Dániel (BME-MIT, varro@mit.bme.hu)**

Manapság a szoftverfejlesztés területén egyre többen sorakoznak fel az OMG Modellvezérelt Architektúra (MDA) kezdeményezése mellé, amelynek alapja, hogy egy magasszintű platform specifikus modelltől kiindulva transzformációs lépések sorozatán keresztül ér el futatható alkalmazásig. A modellvezérelt fejlesztés sikerének egyik létfontosságú eleme a modelltranszformációk fejlesztésének (MT) hatékony támogatása.

A modelltranszformációk támogatására készült a Méréstechnika és Információs Rendszerek tanszéken fejlesztett és több kutatási projektben használt VIATRA2 keretrendszer. A keretrendszer a gráftranszformáció és az absztrakt állapotgépek magasszintű nyelveit ötvözi egy egységes formális specifikációs nyelvbe. A nyelvben a gráf alapú modellek elemi transzformációját gráftranszformációs szabályok végzik, míg az elemi lépésekből egy komplex transzformációs programot az absztrakt állapotgépek segítségével építhetünk fel.

E TDK dolgozatban a VIATRA2 modelltranszformációs nyelvéhez általam készített statikus kódellenőrzőt mutatom be. A statikus forráskód elemzők célja, hogy a lehetséges hibákat a program futtatása nélkül, kizárólag a kód struktúrájából, illetve a nyelv szintaktikájából és szemantikájából felismerje, ezzel gyorsítva a fejlesztés folyamatát.

Az ellenőrzés elvégzéséhez a Sicstus Prolog CSP (Constraint Satisfaction Problem, kényszerkielégítési probléma) megoldó eszközét integráltuk a VIATRA2 keretrendszerbe. A kényszer megoldó rendszerek deklaratív megközelítést adnak kombinatorikus problémák megoldására. Változók egy halmazán működnek, ezek lehetséges értékkészletét a kényszerek egymás után történő alkalmazásával próbálják folyamatosan szűkíteni. Ha valamelyik változónak a lehetséges értékkészlete az üres halmazra csökken, akkor a problémának nincs megoldása. A CSP megoldó használatának legfontosabb előnye, hogy a kényszerek hatásait képes visszafelé is terjeszteni és így korábbi megállapításokat is befolyásolni

A dolgozatban a VIATRA2 teljes transzformációs nyelvét leképeztem kényszerekké, hogy egy részben típusos nyelvben többek között kiterjedt típusellenőrzést végezhesünk. Az ellenőrzés eredményét a VIATRA2 keretrendszer Eclipse alapú környezetébe integráltan jelenítem meg. A dolgozatban esettanulmánnyal igazolom a módszer gyakorlati alkalmazhatóságát és megállapítom a koncepcionális és gyakorlati korlátokat.