

Korszerű modelltranszformációs rendszer tervezése és alkalmazásai

Terék Zsolt V. évf.

**Konzulens: Varró Dániel, doktorandusz,
Méréstechnika és Információs Rendszerek Tanszék**

UML modellek verifikációjának illetve validációjának elterjedt megközelítése szerint a felhasználó rendszermodelljét automatikusan transzformáljuk különféle matematikai leírásokká (Petri hálók, automaták, processz algebrák, stb.), majd a matematikai analízis eredményeit szintén automatikusan vetítjük vissza a rendszermodellbe, ezáltal a rendszermérnököknek nem szükséges ismerniük a különféle matematikai formalizmusokat.

A tanszéken kifejlesztett VIATRA (VIsual Automated model TRAnsformations) rendszer célja e transzformációs feladatok támogatása, melynek során egy UML Profile által megadott (gráf)transzformációs szabályhalmaz szemantikájának megfelelő Prolog programot automatikusan generáljuk és hajtjuk végre. A VIATRA rendszer gyakorlati alkalmazásai azonban rámutattak arra, hogy nagyméretű rendszerek transzformációi során hatékonysági problémák lépnek fel.

A modelltranszformációs rendszerek hatékonyságát döntően az alkalmazott gráfmintaillesztési algoritmus sebessége befolyásolja. Mivel a mintaillesztés problémája általános gráfok esetében NP-teljes, polinom idejű algoritmus kidolgozása (jelen ismereteink alapján) reménytelen. A vizsgálódások célja az volt, hogy az algoritmus *átlagos esetben* hatékony legyen.

A dolgozat egy új modelltranszformációs rendszert mutat be, amely az elmúlt másfél év kutatási eredményein alapuló mintaillesztő algoritmus köré épül. Ennek az implementációnak két fontos követelményt kellett teljesítenie:

- a VIATRA rendszer transzformációs magját funkcionálisan helyettesítenie kell úgy, hogy annál lényegesen nagyobb hatékonyságra legyen képes;
- alkalmas legyen a modelltranszformáció alapú kódgenerálás támogatására is.

A bemutatott modelltranszformációs rendszer a *Scheme* programozási nyelvet használja fel szkriptnyelvként és beágyazott interpreterként. A Scheme egy Lisp dialektus, amely többek közt rendkívül tiszta szemantikája miatt közkedvelt magasszintű programozási nyelv, mi több kimutatták, hogy hatékonyságát tekintve a C nyelvvel vetekszik.

Transzformációs rendszert kiterjesztettük kódgenerálási képességekkel is, melynek során a szabályok végrehajtásával egyidejűleg a megadott parametrikus kódvázak futtatásával a potenciális alkalmazási területek (pl. debug információk, logging, tesztesetek generálása stb.) egy szélesebb skáláját kapjuk.