

Elosztott objektum-orientált rendszerek verifikációja

Darvas Ádám, Műszaki informatika szak V. évf.

Konzulens: dr. Majzik István, Méréstechnika és Inf. Rendszerek Tanszék

Napjainkban egyre több az olyan alkalmazás, amely elosztott szoftver rendszerek használatát követeli meg. A rendszerek összetettsége ún. szoftver-krízishez vezetett, vagyis a programkódok átláthatatlanok lettek. Erre a problémára részben megoldást kínál az objektum-orientált szemlélet, amely mára uralkodóvá vált.

Az elosztott objektum-orientált rendszerek tervezése azonban rendkívül bonyolult nagyfokú komplexitásuk miatt. Gyakran az adott feladat mérete túlnövi a mérnökök látókörét, ami könnyen vezet hibás rendszertervekhez. Ez különösen nagy problémát jelent akkor, ha a hiba csak a már megvalósított rendszer tesztelése, esetleg működésbe lépése után derül ki. A rendszertervező számára jelentős támogatást adna tehát egy olyan eszköz, amely még a modellezés fázisában ellenőrizné a modell logikai helyességét.

Az objektum-orientált rendszerek tervezésére legelterjedtebb az UML szabványos grafikus modellező nyelv. Ez azonban csak egy leíró eszközt ad, a modell ellenőrzésére nem nyílik mód. A formális ellenőrzéshez matematikai modelleken alapuló eszközöket dolgoztak ki. Ilyen a SPIN, amely elsősorban elosztott rendszerek verifikációjára, vagyis a logikai helyesség ellenőrzésére alkalmas.

A fentiek alapján adódik az ötlet, hogy a két eszközt célszerű lenne összekapcsolni, ezzel lehetővé téve a megtervezett modell logikai ellenőrzését. A feladat egy transzformáció megvalósítását jelenti az UML modellből a SPIN bemeneti formátumára. Ennek kidolgozásához már rendelkezésre állt egy formális szemantika, amely egyszerű (nem elosztott) rendszerek esetén megadja a transzformáció matematikai háttérét.

Munkám során először kiterjesztettem a meglévő eredményeket úgy, hogy dinamikus elosztott OO rendszerek vizsgálata is lehetővé váljon. Meg kellett oldanom az aszinkron eseménykezelés (eseménysorok), objektum létrehozás és törlés, valamint az objektum mozgás (dinamikus átkonfigurálás) transzformációjának problémáit.

A transzformációt Java programként implementáltam. Ezt a Tanszéken lévő keretrendszerbe illesztve a tervező képes automatikusan ellenőrizni alapvető viselkedési jellemzőket (pl. holtponmentesség, veszélyes állapotok elkerülése). A transzformáció használatát mintapéldákon keresztül mutatom be.