

# Szoftver megbízhatósági modellezés

## Önálló labor feladat összefoglaló

Készítette: **Krasznai Péter**

Konzulens: **Majzik István**

A megbízhatósági modellezés általában már a rendszertervezés korai szakaszában nagy hangsúlyt kap, főként biztonságkritikus rendszerek tervezése során. Segítségével már tervezési időben, tehát még az implementációs szakasz előtt megbecsülhető a készülő rendszer megbízhatósága, valamint azonosítani lehet a rendszer gyenge pontjait, szűk keresztmetszetét.

A rendszer együttműködő modulok kompozíciója, a nyújtott szolgáltatás minősége nyilván az egyes komponensek megbízhatósági jellemzőitől függ. A megbízhatósági modellezés célja a modulok paramétereiből és a modulok közti kapcsolatrendszerből kiindulva megbecsülni a rendszerre (mint egészre) jellemző megbízhatóságot.

Az elmúlt félévek során megismerkedtem egy automatizálható módszerrel, ami egy rendszer UML diagramjából kiindulva annak időzített Petri háló alapú modelljét állítja elő. A módszer két lépésben – egy közbülső ún. *Intermediate Model* –en keresztül – végzi el a transzformációt. A transzformáció eredményeként előálló általánosan leírt Petri hálós modellt egy konkrét modellező eszköz bemeneti formátumára konvertálva elvégezhető a rendszer megbízhatóságának elemzése.

Ezt követően a transzformáció megvalósításával kezdtem foglalkozni. A cél tehát egy rendszer osztálydiagramjának – ahol az egyes elemek megbízhatósági szempontból fontos jellemzői is adóttak – automatikusan történő transzformálása időzített Petri hálóvá. Ehhez a tanszéken fejlesztett VIATRA nevű modelltranszformációs rendszert kezdtem használni. Ez az eszköz a vizuálisan megtervezett gráftranszformációs szabályokra, valamint a forrás- és célmodell MOF alapú modelljére támaszkodva transzformációs kódot generál a forrás- és a célmodell között.

Az előző félév végére a transzformáció közbülső (IM) modelljéig – egy újabb közbülső (preIM) modell bevonásával – tartó transzformációs szabályok elkészültek, tesztelésre azonban nem maradt idő. Jelen félév a teszteléssel kezdtem – volna – a dolgot, de némi problémák adódtak az UML modellező eszközök körül, ugyanis szükséges modellelemek (pl. tagged value) hiányoztak belőlük (Rational Rose), illetve nem jó formátumban exportáltak XMI-ba (argoUML). Végül is a Rational Rose-nál maradtam, a hiányzó modellelemek helyett azonban másokat kellett használni, és ezért egy előtranszformációt is be kellett vezetni, ami a kívánt formátumra hozza a modell XMI leírását.

Jelen állapotban az előtranszformáció működik. Az osztálydiagram → preIM transzformációban sok szabály módosult a tesztelése eredményeként, de a tesztelés még nem fejeződött be. A preIM → IM transzformáció pedig az előző félévi állapotban maradt.