



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## Modell alapú hibainjektáló eszköz fejlesztése

Gerencsér György, FRE7DO, 3. évfolyam, Mérnök informatikus szak  
Konzulens: Oláh János, MIT  
Informatikai technológiák / Rendszertervezés  
Önálló laboratórium összefoglaló  
2010/11. II. félév

Egy szoftver fejlesztése során a tesztelési fázis fontos eszköze a hibainjektálás. Ennek során szándékosan és reprodukálható módon idézünk elő hibás működést egy adott cél rendszeren, így az instrumentált kód futtatásával kapott eredmények megtekintése után a hibakezelésért felelős kódrészleteket tudjuk ellenőrizni. Biztonságkritikus, illetve nagy rendelkezésre állású rendszereknél alkalmazzák ezt a technológiát.

Az eddig készült hibainjektáló eszközök nem alkalmazkodnak a korszerű modell alapú fejlesztési folyamatokhoz. A modellvezérelt programozási szemlélet (Model Driven Engineering – MDE) szerint egy fejlesztési folyamat különböző lépései során meghatározott típusú modelleket használunk fel, ügyelve a modellek közötti kapcsolatokra is.

Az Önálló laboratórium során megismertem a tanszéken készült hibainjektáló eszközt, ami már alkalmazkodik a fent említett modellvezérelt szemlélethez. Az alkalmazás segítségével különböző típusú hibákat injektálhatunk tetszőleges, Java-ban implementált programba, amennyiben annak class fájljai elérhetők. A keretrendszer ezek alapján elkészít egy modellt, melyen a felhasználó konfigurálhatja az injektálandó hibákat valamint a monitorozási pontokat, amelyek segítségével az alkalmazást figyelhetjük.

Mivel a modell felépítése, a hibainjektálás és a monitorozási pontok elhelyezése is lényegében a kód futásidőben történő módosítása, kiegészítése, ezért ezekhez a folyamatokhoz reflektív programozásra van szükségünk. Ennek során egy program kódját futásidőben megtekinthetjük, valamint változtathatjuk. A keretrendszerben ezekhez a feladatokhoz a Javassist programkönyvtárat használták fel.

A hibainjektáló alkalmazás maga egy Eclipse kiterjesztés (ezáltal a user interface beépül az Eclipse-be), és ezen kívül is használ különféle Eclipse alapú technológiákat. A fent említett modellvezérelt fejlesztés például az EMF keretrendszer segítségével valósul meg.

Egy ilyen hibainjektáló eszköz használatánál rengeteg adat jöhet létre (pl. milyen eredményekkel futnak az instrumentált alkalmazások). Ezeket érdemes kimenteni relációs adatbázisba, az eredmények további feldolgozása végett. A félév során a MySQL adatbáziskezelőt használva létrehoztam egy sémát ezen adatok tárolása végett, valamint kibővíttem az alkalmazást ezzel a funkcióval.

A létrehozott séma egy ún. csillag séma, ami egy központi ténytablából, és hozzá kapcsolódó dimenziótáblákból áll. A ténytablában külső kulcsok vannak az összes dimenziótáblához. Ez az elrendezés azért szükséges, hogy a későbbiek során analízis műveleteket hajtsunk végre az ún. OLAP (Online Analytical Processing) technológia segítségével. Az OLAP analízis megismeréséhez egy, a létrehozott sémához hasonló, tesztadatbázist kaptam, ami több ezer sornyi eredményt tartalmaz. Az OLAP használatához a JasperForge nyílt forráskódú szoftvereit használtam.

Források:

1. MDE: <http://www1.cse.wustl.edu/~schmidt/PDF/GEI.pdf>
2. Eclipse EMF: <http://www.eclipse.org/modeling/emf/docs/>
3. Javassist: <http://www.csg.is.titech.ac.jp/~chiba/javassist/>
4. MySQL: <http://www.mysql.com/>
5. OLAP - Mondrian: <http://jasperserver.sourceforge.net/docs/3-7-0/Mondrian-3.0-Technical-Guide.pdf>