

## Modell alapú hibainjektáló környezet fejlesztése

Oláh János V. Inf., jantsee@gmail.com

**Konzulens: Majzik István, majzik@mit.bme.hu**  
**Méréstechnika és Információs Rendszerek Tanszék**

Üzleti szempontból kritikus illetve biztonságkritikus informatikai alkalmazások esetén kulcsfontosságú annak elemzése, hogy az alkalmazott szoftver komponensek hogyan viselkednek hibák bekövetkezése illetve szélsőséges működési feltételek (pl. extrém bemeneti adatok, túlterhelés) esetén. Ezek az elemzések a tervezési fázisban segítik az architektúra optimalizálását illetve a megfelelő tulajdonságokkal rendelkező komponensek kiválasztását, így hozzájárulva az alkalmazás megbízhatóságának (hibatűrésének) és robusztusságának növeléséhez. Az általánosan elfogadott elemzési és mérési módszereket megbízhatósági benchmark formájában rögzítették egyes alkalmazásokhoz [1]. A benchmark lényege meghatározott hibák injektálása adott terhelés mellett működő rendszerbe, közben megfigyelve a hibahatásokat.

A hibák injektálására számos eszközt és keretrendszert fejlesztettek ki. Ezek jól paraméterezhetők és sokféle hiba injektálására alkalmasak (szoftver illetve hardver szinten is). Legfőbb hátrányuk azonban az, hogy nem illeszkednek a korszerű modell alapú fejlesztési folyamatokhoz, nem használják ki a tervezői modellekben lévő információkat és az automatizálási lehetőségeket. A kísérletek tervezése és konfigurálása alacsony (tipikusan forráskód) szintű, aprólékos kézi munkát igényel.

A TDK dolgozat egy olyan hibainjektáló környezetet kidolgozását ismerteti, amely lehetővé teszi Java programokhoz a *hibainjektálási kísérletek modell alapú tervezését, automatikus konfigurálását és végrehajtását*. A fejlesztő az alkalmazás modelljén, grafikus felület használva specifikálhatja az injektálandó hibákat (modell kiterjesztések segítségével megadva a hibatípust, paramétereket, injektálási időt illetve gyakoriságot) valamint a kísérletek során megfigyelendő adatokat. A modell alapú specifikáció alapján az eszközkészlet automatikusan generálja a hibainjektáláshoz valamint a hibahatások megfigyeléséhez és rögzítéséhez szükséges extra kódot. Az extra kód beillesztése a Javassist technológia [2] használatával modulárisan történik. Ugyanez a technológia biztosítja a komponensek automatikus felderítését és az alkalmazás modelljének visszafejtését. Az injektálható hibák típusa egy hibakészletből választható (a hibakészlet jelenlegi verziója elsősorban a szoftver komponensek közötti interakciós hibák injektálását támogatja).

A dolgozat ismerteti a hibainjektáló környezet architektúráját, a kidolgozott komponenseket, az Eclipse [3] környezetben történő megvalósítást és integrálást, valamint bemutatja a hibainjektálás módszerét egy példa alkalmazás vizsgálatán keresztül.

Hivatkozások:

1. DBench Dependability Benchmarks ([www.dbench.org](http://www.dbench.org), report date 27 February 2004)
2. Javassist: <http://www.csg.is.titech.ac.jp/~chiba/javassist/> (2008. 09. 22.)
3. Eclipse RCP: [http://wiki.eclipse.org/index.php/Rich\\_Client\\_Platform](http://wiki.eclipse.org/index.php/Rich_Client_Platform) (2008. 09. 22.)