

Modell alapú tesztelés mobil környezetben

Önálló laboratórium feladat összefoglalója (2. félév)

Hamvas Áron Gergely (HIFRD7)

Konzulens: Micskei Zoltán

BME Méréstechnika és Információs Rendszerek Tanszék
Informatikai infrastruktúra tervezése szakirány, 2007/2008. II. félév

Az önálló laboratórium keretei között a feladatom a modell alapú tesztelési módszerek megismerése és kipróbálása.

Az előző félévben sikerült megismerni az UML 2.0 Testing Profile-t [1], ami egy alkalmazástesztelés modellezésére alkalmas kiterjesztése az UML nyelvnek.

Ebben a félévben sikerült egy kicsit megismernem az IBM Rational Functional Tester [2], funkcionális, regressziós és grafikus felhasználói felület tesztelés elkészítéséhez és automatizálásához használható IBM terméket. Sajnos, a már az előző félévben is ismertetett, a HIDENETS projekthez kapcsolódó mintaalkalmazás tesztelésére ez a szoftver nem tűnt alkalmasnak, ezért ez a feladat inkább a tesztelési eszközökkel kapcsolatos általános ismeretek megszerzését segítette. A mintaalkalmazás esetében, ennek az eszköznek az alkalmasságát azért kérdőjeleztem meg, mert az elérhető dokumentációk, és értékelések alapján a Functional Testerrel tesztelt programok grafikus felülettel kell, hogy rendelkezzenek.

A félév során sikerült megismerkednem a JUnit [3] unit tesztelési keretrendszerrel, amelynek eszközkészlete lehetővé teszi alkalmazások modul szintű funkcionális tesztelését is.

Elkészítettem továbbá az UML 2.0 Testing Profile (U2TP) egy „implementációját”, amelyet modellezésnél már teljes mértékben felhasználhatok az IBM Rational Software Architect programban. Ennek a profilnak a felhasználásával elkészítettem a mintaalkalmazás tesztelési modelljét. Mivel a mintaalkalmazás szimulációs környezetben fut, és ráadásul C++ nyelven, de nem objektum-orientáltan implementálták, ezért az U2TP-nek csak viszonylag kis részét lehetett kihasználni, pl. osztály szintű tesztelés modellezésének nincs értelme.

A tesztelés részleges automatizálása úgy valósítható meg, hogy konfigurációs fájlokat generál a tesztkörnyezet, amelyek paramétereket tartalmaznak, amelyeket a szimulátor felhasznál a szimuláció végrehajtása során. A tesztelési környezet legközelebb a napló fájlok analízise során aktivizálódik, amely fájlokból a szimulációs eredményeket kiolvastva eldönti, hogy az egyes tesztek, vagy a tesztelés egésze sikeres volt, vagy sem. Mivel a szimulációs környezet részletes megismerésére nem maradt idő, ezért a tesztek még nem sikerült végrehajtani.

A félév végére konzulensem részéről felmerült egy igény, hogy készítsek egy szekvencia diagramokat beolvasni, analizálni képes Java alkalmazást. Ezt a későbbiekben modell transzformáció végrehajtására, kódgenerálásra is lehetne használni. Mivel ez igencsak a félév végén merült fel, ennek implementációjára a nyár folyamán kerül sor.

[1] UML Testing Profile, http://www.omg.org/technology/documents/formal/test_profile.htm

[2] Rational Functional Tester, <http://www.ibm.com/software/awdtools/tester/functional/>

[3] JUnit, <http://www.junit.org/>