

Biztonságkritikus rendszerek felhasználói kezelőfelületeinek tesztelése

Önálló laboratórium feladat összefoglalója (2. félév)

Jahn Gergely (E0GESB)

Konzulens: Majzik István

**BME Méréstechnika és Információs Rendszerek Tanszék
Informatikai infrastruktúra tervezése szakirány, 2006/2007. II. félév**

A félév során grafikus felhasználói felületek tesztelésével ismerkedtem meg, valamint olyan technikákkal, melyek segítségével tesztszekvenciák generálására alkalmas szoftvert készítettem.

Az előző félévben megismert PetShop nevű alkalmazásban elkészíthető interfész modellt alapul véve egy olyan szoftvert készítettem el, mely bemenetként egy XML fájlban kódolt Petri Hálót kap, majd ebből véges állapotú automatát készít, végül ezt megfelelő formátumba egy `_sal` kiterjesztésű fájlba írja ki, melyet a SAL (Symbolic Analysis Laboratory) modellellenőrző beolvas és szekvenciákat generál.

A munkát szakmai cikkek, tanulmányok olvasásával kezdtem, és kiválasztottam egy technikát, melyet alaposabban megismertem. Ennek lényege, hogy a rendszer állapotváltozásait a felhasználói felületen végbemenő interakciók triggerelik. Ennek megfelelően egy véges állapotú automata segítségével leírhatjuk a működését. Ezt követően a kezdő állapotból kiindulva különböző utakat járunk be, melyek tesztszekvenciákra képezhetők le.

A PetShop alkalmazásban megtervezett felhasználói felület és prototípus XML fájlba exportálható, melyet első lépésben be kell olvasni. Ehhez egy SAXParsert használok, mellyel végigpásztázom az XML fájlt és egyes részeket feldolgozok, másokat nem. Csak a helyek, tranzíciók, élek, valamint a kezdő tokeneloszlás fontos, a többi résszel nem kell foglalkozni. A következő konvenció bevezetése szükséges: azokat a tranzíciókat, melyeket közvetlenül felhasználói interakció triggerel a `'_ui'` postfixel látunk el az elnevezésben.

Ha kezünkben van az eredeti modell, akkor szisztematikusan mélységi bejárást alkalmazva, minden lehetséges tokeneloszlásnak egy állapotot feleltetünk meg, vagyis Petri háló \rightarrow véges állapotú automata transzformációt végzünk.

Az elkészült automatában egyszerűsítési lehetőségeink vannak, melyek ekvivalencia-relációkon alapulnak. Ezen a téren még finomításokra lesz szükség, egyelőre egy triviális eset beépítésére került sor.

Ezt követően nincs más dolgunk, mint a SAL által feldolgozható formátumba exportálni ezt az automatát. A SAL egy modellellenőrző rendszer, melynek van véges állapotú automatákon adott mélységű szekvenciákat generáló funkciója.

A SAL-nak átadott automata élei felelnek meg az egyes felhasználói interakciónak, és az egyes szekvenciák ezek egy meghatározott sorrendjét fogják kiadni.

A program folyamatosan naplózza a műveleteket, így nyomon tudjuk követni a beolvasás, a transzformáció és az exportálás lépéseit, illetve azok sikerességéről győződhetünk meg.

A robosztusság teszteléséhez az elkészített szekvenciák végállapotaiban végrehajtjuk egyenként az összes nem megengedett műveletet is. Ehhez a program exportál egy listát, mely tartalmazza minden állapotban a nem megengedhető műveleteket.

További feladatként tesztvégrehajtókhoz illesztést, a teszt fedettség mérését, további absztrakciós lépések beépítését, illetve más tesztgeneráló algoritmusok használatát foglalmaztuk meg.