

Robusztusság tesztek modell alapú generálása

Kárer Győző V. Inf., karer.gyozo@gmail.com

Konzulens: dr. Majzik István, MIT, majzik@mit.bme.hu

Napjaink szoftvereinek komplexitása miatt a fejlesztés során igen sok a hibaforrás. A helyes szoftver iránti igény kielégítésére a fejlesztési idő és költségek nagy hányadát (akár 40%-át is) tesztelésre fordítják. Ezt a nagy időigényt a tesztek kézi előállítása és végrehajtása indokolja, ennek csökkentése többek között a tesztek automatikus, már a szoftver fejlesztés kezdeti szakaszától induló generálásával történhet. Az automatikus tesztgenerálásra lehetőséget ad a modell alapú szoftverfejlesztés, hiszen ennek során megtörténik az alkalmazás funkcióinak és interfészeinek modellezése, pl. UML diagramok segítségével.

A tesztelés egyik specifikus területe az úgynevezett robusztusság tesztelés. A robusztusság teszteléssel az alkalmazásnak azt a képességét kívánják felmérni, hogy mennyire stabil működést tud garantálni extrém bemeneti paraméterek (pl. szélső értékek illetve a specifikáció által rögzített normál működési tartományból kilógó értékek) és környezeti feltételek (pl. túlterhelés, hibák bekövetkezése) esetén. Ennél a tesztelési módnál alapvetően azt vizsgáljuk, hogy a specifikációban rögzített funkciókat el tudja-e látni a szoftver (ennek része lehet megfelelő hibajelzések adása is) összeomlás illetve nem megengedett leállás nélkül. A robusztusság olyan tulajdonság, ami eseményvezérelt beágyazott rendszerek, felhasználói felületek, köztesrétegek és operációs rendszerek esetén különösen fontos.

A napjainkig kidolgozott, elérhető robusztusság teszt generátorok nem illeszkednek a modell alapú fejlesztés folyamatába: a robusztusság teszteket nem a specifikáció, hanem az elkészült alkalmazás forráskódja alapján generálják. Ezenkívül általában csak egy-egy interfészre koncentrálnak, és az összetett hívási szekvenciákat (használati forgatókönyveket) nem kezelik. Dolgozatomban egy általam fejlesztett módszert és eszközt mutatok be robusztusság tesztek modell alapú generálására. Az eszköz bemenetként UML osztály diagramokat és szekvencia diagramokat fogad (a tesztelendő interfészek illetve a hívási szekvenciák megadására), kimenete pedig a C++ szintaxist követő végrehajtható teszt hívások sorozata. Az elkészült eszköz mint Eclipse plug-in illeszkedik az Omondo EclipseUML modellező környezethez, főbb feladatai a következők:

- Teszt hívásokat generál az osztály diagramból kiválasztott metódusokhoz. A használt tipikus illetve extrém paraméter értékek egyszerű típusonként illetve metódusonként is testre szabhatók, az extrém és a normál értékek kombinálásáról az eszköz gondoskodik.
- A szekvencia diagramban megjelenő hívási sorrenden többféle mutációt hajt végre. A teszt hívások generálása során azok a metódushívások, amelyek nem vesznek részt a mutációban, tipikus bemeneti értékekkel szerepelnek. A paraméterek testre szabása itt három szinten történhet, ebből kettő az előző pont szerinti, a harmadik szinten pedig adott üzenetekhez közvetlenül rendelhetőek paraméterek.

Robusztusság tesztelés esetén az elvárt viselkedés ellenőrzése nem a teszt hívások eredményeinek precíz vizsgálatával, hanem csak a nem elfogadható állapotok illetve az időtúllépés vizsgálatával történik. Ezen elvárások (mint elérhetőségi követelmények) hatékonyan leírhatók lineáris temporális logika segítségével, amelyek alapján (egy már meglévő technológia felhasználásával) a teszt végrehajtást monitorozó alkalmazás kódja is automatikusan generálható.

A dolgozatban a módszerek és az eszközök bemutatása mellett a robusztusság tesztek generálását gyakorlati modelleken is demonstrálom.