



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

ANSI C programok integritásának hivatkozási gráf alapú vizsgálata

Fejes Endre (G2MMW3), I. évf., (MSc) mérnök-informatika szak

Konzulens: Majzik István docens, MIT

Külső konzulens: Piroska László, Robert Bosch Kft.

Szolgáltatás-biztos rendszertervezés szakirány

Önálló laboratórium 1 összefoglaló

2009/10. II. félév

A Bosch Kft. által gyártott automata sebességváltók mikrokontrollerét beágyazott C nyelvű programok irányítják. Mivel biztonságkritikus rendszerről van szó, ezért szigorú kritériumoknak kell az alkalmazásnak megfelelnie, aminek ellenőrzése többek között teszteléssel történik. A tesztelés során felmerülő jellegzetes probléma, hogy az alkalmazás mérete és komplexitása miatt nehézkessé válik a tesztek hatókörének pontos meghatározása.

A probléma megoldása két részre bontható: egyrészt meg kell határozni a tesztek által lefedett program részeket, ehhez pedig meg kell határozni a lefedendő elemek teljes halmazát. Az önálló labor félév során ez utóbbi részfeladat megoldásával foglalkoztam, a modulok közötti interfészek tesztek általi lefedettségének vizsgálatához.

A konkrét probléma tehát egy C nyelvű, moduláris szerkezetű alkalmazás interfészének meghatározása. Mivel az ANSI C nyelvben nem létezik interfész, névtér vagy osztály, így definiálni kell az interfész fogalmát. Potenciális interfésznek tekintjük jelen esetben az olyan függvényeket illetve változókat, amik fejléc fájlban vannak definiálva. Azonban gyakorlatilag minden modulban használt függvény definíciója szerepel fejléc fájlban az átlátható programozás gyakorlatából kifolyólag, így csak azokat tekintjük tényleges interfész elemnek, amelyekre más modulból történik hivatkozás.

A megoldáshoz szükségünk van arra, hogy egy-egy állományról eldöntsük, melyik modulhoz tartozik. Mivel ennek egyértelmű jelölése nyelvi szinten nem megoldott és az állományok elnevezéséből illetve elhelyezkedéséből csak sok és sokféle potenciális kivétel figyelembe vételével lehetne meghatározni, ezért az állományok modulhoz tartozásához az ellenőrzési folyamat más lépései miatt rendelkezésre álló modulonkénti állománylista felhasználását választottam.

A készítendő eszközzel szemben támasztott követelmények egyike a könnyű használhatóság volt, ami miatt egy fejlesztőrendszerbe integrálható beépülő modul elkészítése mellett döntöttem. Mivel a cégnél nincs meghatározott egységes fejlesztő környezet, így az egyik legelterjedtebb IDE, az Eclipse mellett döntöttem.

Az elemzés statikus ellenőrzéssel történik, *Abstract Syntax Tree* (AST) technológiával, melyhez az Eclipse nyújtotta *C Development Tools* (CDT) eszközt használtam. A CDT-beli AST nem más, mint a C nyelv meta-modelljét leíró szerkezet, amelynek elemeinek használatával C forráskód elemzése lehetséges. Ebből egy konkrét programot leíró példány a *Concrete Syntax Tree* (CST), amely gyakorlatilag egy objektum orientált reprezentációja a kódnak. Ennek bejárására *Visitor* minta alapú lehetőség van, ami leegyszerűsíti a különféle nyelvi elemek (pl. függvény deklaráció, értékadás, stb.) különböző kezelését. Ezzel lehetséges a fejléc fájlok-beli deklarációk és kód fájlokban lévő hívások illetve hivatkozások kigyűjtése.

A függvény illetve változó hivatkozások alapján felépíthető egy hivatkozási gráf, illetve egy modulok közti hivatkozási mátrix is, további elemzésekhez. Ezek illetve a potenciális interfészelemek listája szöveges jelentésként és Eclipse nézetként is elérhető, így könnyen áttekinthető és felhasználható az elemzés eredménye.