



M Ű E G Y E T E M 1 7 8 2
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

JavaScript forráskódok gráfalapú statikus analízise

Stein Dániel II. évf., MSc mérnökinformatikus szakos hallgató
Kritikus rendszerek főspezializáció

Konzulens: Szárnyas Gábor, doktorandusz (MIT)
Lippai Ádám (Tresorit)

Diplomatervezés 1. összefoglaló

2015/16. II. félév

Nagyméretű, komplex szoftverek fejlesztésekor a közreműködők számának növekedésével a kódban is gyakrabban fordulhatnak elő hibák. Ezek kiküszöbölésére jelenleg is rendelkezésre állnak megoldások, mint például a statikus analízis, amely többek között automatikusan ellenőrzi, hogy a kód megfelel-e a kódolási szabályoknak.

A hagyományos általános célú programozási nyelvek (pl. C++, Java) esetén több eszköz is létezik a statikus analízis támogatására (pl. Clang Static Analyzer, Facebook Infer, PMD, FindBugs), ill. bevett gyakorlat, hogy az analízis a folytonos integráció (continuous integration) része. Ezzel szemben a weben jelenleg legnépszerűbb programozási nyelv, a JavaScript esetén a kód statikus analízise nem elterjedt gyakorlat.

A diplomamunka megvizsgálja és bemutatja a JavaScript forráskódok elemzésére jelenleg rendelkezésre álló statikus analízis technikákat. Emellett megvizsgálja ezen technikák megvalósításának lehetőségét gráfalapú adatstruktúrán történő mintaillesztés alkalmazásával.

A tervezett rendszer prototípusa megfelelő megkötések mellett képes JavaScript kódbázisok feldolgozására és abban hibagyanús kódrészletek, illetve a rendszert használó fejlesztő számára jelző konstrukciók megkeresésére. (Ilyen például azon deklarációk megkeresése, melyre kódbázis szinten nem található hivatkozás, így az adott kódrészlet feltehetőleg felesleges.)

Az elkészült elemző modul prototípus fejlesztőkörnyezetbe integrálása szintén a diplomamunka része. Ennek használatával a fejlesztőkörnyezet elhagyása nélkül értesülhet az azt használó fejlesztő a modul által kiadott figyelmeztetésekről (a fejlesztőkörnyezet jelzésein felül).

Központi verziókezelő rendszer támogatása mellett a prototípus egyszerre több felhasználó egyidejű munkásságát képes feldolgozni és a megfelelő fejlesztőnek eljuttatni a jelzéseket, így egyszerre igazodva a folytonos integrációs munkafolyamathoz és az egyedi fejlesztési ciklushoz.

Emellett a diplomamunka által érintett analízis területek sorát bővíti a típuskövetkeztetési rendszer gráf adatstruktúrán történő megvalósítás lehetőségének vizsgálata és alapjainak prototípuson keresztüli bemutatása.