



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Állapotgép-konstrukciós feladatok automatizált előállítás



Tóth Bálint V. évf, (BSc) mérnökinformatikus szakos hallgató
Konzulens: dr. Bergmann Gábor tudományos munkatárs, MIT
Informatikai technológiák szakirány / Rendszertervezés ágazat
Önálló laboratórium összefoglaló
2014/15. II. félév

A feladatom az új tanrendben induló második féléves Rendszermodellezés nevű tantárgy házi feladat generálási módszerének javítása, továbbfejlesztése volt. A házi feladat egy sakkóra állapotgép alapú működésének megvalósítása, melyet a hallgatók egy Eclipse beépülő modul, a Yakindu segítségével tudtak megoldani. A Yakindu lehetőséget nyújt arra, hogy egy eseményvezérelt alkalmazás állapotgép alapú működését grafikus felületen meg lehessen valósítani, illetve szimulációs, hibakereső funkciókat is képes ellátni.

A házi feladatok hallgatónként eltérőek bizonyos variációs pontokon. A különbségek megjelenhetnek egyszerű numerikus paraméterek, illetve strukturális különbségekként, mint a beállító menüben található menüpontok száma, maguk a menüpontok, esetleg, hogy milyen sorrendben tudunk a menü végigterelni. Mivel minden házi feladat különböző, törekedni kell arra, hogy nehézségük közel azonos legyen.

A jelenlegi házi feladat generálás programatikusan történik, egy Xtend nyelven megírt program segítségével. A program véletlenszerűen rakja össze feladatokat, használandó értékeket, figyelembe véve az egyes részfeladatok nehézségét.

A feladatom az, hogy a jelenlegi módszert lecseréljem egy Design Space Exploration (tervezési tér felderítés, továbbiakban „DSE”) alapú generálási módszerre. Ez egy keresés alapú tervezési folyamat, melynek feladata a tervezési kényszereknek megfelelő modellek felkutatása. Ez azt jelenti, hogy ahelyett, hogy explicit összerakjuk a házi feladat modelleket, egy szoftver segítségével megadunk transzformációs szabályokat, a DSE szoftver pedig a megfelelő modelleken végrehajtja azokat, az összes lehetséges sorrendben.

A félévem során a Viatra DSE modellvezérelt tervezési tér felderítésre alkalmas szoftverrel ismerkedtem meg. A szoftver működéséhez szükség van egy kiinduló házi feladat modellre, amely a felderítés alapjául szolgál. Definiálnunk kell szabályokat, amelyek két részből állnak, a bal oldali része egy IncQuery minta, a jobb oldali pedig Java kód, maga a transzformáció. Egy transzformáció akkor futhat le az adott modellen, amennyiben illeszkedik (vagy - beállítástól függően - nem illeszkedik) a szabály bal oldalán meghatározott mintára.

Azt, hogy melyik megoldások felelnek meg a kényszereinknek, erős és gyenge célkitűzések definiálásával adhatjuk meg, melyek együtt alkotják magát a célt. Az erős célkitűzések egy adott megoldás érvényességét, a gyenge célkitűzések pedig a minőségét határozzák meg. A cél több célkitűzést - akár erőset és gyengét vegyesen is - magába foglalhat.

Többféle keresési algoritmus közül választhatunk; ilyenek a mélységi és szélességi keresés. Természetesen mélységi keresés esetén van lehetőség limit megadására.

Ahhoz, hogy a felderítés sikeres legyen szükség van egy állapotkódolóra. Ez az állapotkódoló felel azért, hogy a megtalált állapotokat nyilvántartsa, enélkül a szoftver végtelen ciklusba kerülhetne, hiszen egy-egy már korábban felderített állapotot mindig újnak ismerne fel. A Viatra DSE tartalmaz egy beépített állapotkódolót, de ezt a próbálkozásaim alapján a mi problémánkra nem tudjuk felhasználni, ezért készíteni kell egy saját problémáspecifikus állapotkódolót.

További fejlesztési feladatok: A Viatra DSE szoftver használata a házi feladat generálásra, állapotkódoló írása a probléma megoldására. Kiinduló állapot létrehozása, DSE szabályok definiálása. A két generálási módszer különböző szempontok szerinti összehasonlítása. Ilyen szempontok lehetnek: sebesség, karbantarthatóság, módosíthatóság.