



Budapesti Műszaki és Gazdaságtudományi Egyetem
Mérésztchnika és Információs Rendszerek Tanszék

Keresés alapú szoftver- és rendszertervezés



László Dániel András, BSc mérnök inf. szakos hallgató
Konzulens: Nagy András Szabolcs doktorandusz, MIT
Rendszertervezés (MIT) specializáció
Témalaboratórium összefoglaló
2016/17. I. félév

A Témalaboratórium (VIMIAL00) tárgy során azt a feladatot kaptam, hogy olyan programot írjak, amely egy (aknakeresőhöz hasonló) játékot old meg ismerve annak egy általam megfogalmazott szakterület-specifikus nyelvű leírását.

A játék. Az előbb említett játék neve [Fill-A-Pix](#). Bemenetként adott egy $n \times n$ darab mezőből álló tábla, ahol egy mező értéke lehet üres, vagy $\{0-9\}$ szám értékű. Ez a szám azt mutatja meg, hogy a szám környezetében hány darab mező satírozott. Ezután a feladat az, hogy eldöntsük minden mezőről, vajon az satírozott-e vagy sem.

A probléma. Természetesen a leírt játék bonyolultsága nem haladja meg egy általános célú programnyelv képességeit, könnyedén lehetne egy <200 soros programot írni, amely a beolvasás után az adatokat elrendezi, a játékot megoldja, majd a kapott eredményt értelmezhető formátumban ki is írja. A cél tehát nem ez, hanem inkább ismerkedés olyan technológiákkal, amelyek egyrésről a megkötés alapú programozás révén sokszor megkönnyíthetik használójuk életét (Choco Solver), másrésről olyanokkal, melyek specifikus problémák leírását teszik sokkal egyszerűbbé, csupán olyan eszközöket adva a felhasználó kezébe, amelyek az adott problématerületen hasznosnak bizonyulnak. Egy ilyen felhasználói környezet kialakítására létrejött technológia az Xtext.

A szakterület-specifikus nyelvek és az [Xtext](#). Az Xtext többek között a következő lehetőségeket biztosította számomra egy szakterület-specifikus nyelv létrehozásánál:

1. A nyelv szintaxisának leírása: Pontosan megadhatjuk a nyelvünkben használatos kulcsszavakat, változótípusokat és, hogy azok milyen értéket fogadhatnak el. Mindezt szabályokba (nyelvtanokba) elrendezve. A Fill-A-Pix játék szintaxisát két lépésben írtam le. Először a pálya méretét kell megadni, majd ezután egy vagy több mező értékét.
2. Osztálystruktúra létrehozása: Miközben leírtuk a szintaxist, az Xtext egy osztálystruktúrát hozott létre a szabályaink alapján. Minden egyes szabályban leírt változó rendelkezik egy `get()` metódussal.
3. Validáció: A validáció segítségével figyelmeztethetjük a felhasználót, hogyha olyan problémát fogalmaz meg, amely az általunk létrehozott nyelvben nem értelmezhető. Esetemben például nem szeretném azt megengedni, hogy egy mező értéke 9-nél nagyobb legyen. Ezt ERROR-ral jelzem a felhasználónak.
4. Quickfix-ek: Bizonyos problémákat könnyű észrevenni és javítani. A felhasználó értesül a problémáról a validáció révén és ezen túl egy lehetséges megoldási javaslatot is kap. Ilyen kis segítség nagyban tudja növelni a felhasználói élményt. Esetemben ha egy mező értéke nagyobb, mint 9, a program felajánlja, hogy azt átírja a maximálisan megengedett 9 értékre.
5. Szemantikai értelmezés: Miután a felhasználó leírta a problémát, lehetőségünk van azt szemantikusan feldolgozni, azaz ismerve minden adategység pontos jelentését. Többek között erre is jó a 2. pontban leírt osztálystruktúra. A Fill-A-Pix esetén ekkor jön a megoldás.

A Fill-A-Pix játék megoldása és a Choco Solver. A probléma leírása a felhasználó által megtörtént. A bemenet az Xtext szemantikai értelmezőjének és egy „parser” segítségével előállt. A játék

megoldásához egy kényszerkielégítési problémák (CSP, Constraint Satisfaction Problems) megoldására alkalmas Java könyvtárat használtam, amelyet Choco Solvernek hívnak. Ez azért hasznos program, mivel elegendő csupán bizonyos mezőkön értelmezett kényszereket felvenni, ezután a megoldások kiszámítását önállóan elvégzi a könyvtár, amelyet elkérhetünk és kiírathatunk a felhasználónak.