

Baráth Áron

Új nyelvi elemek multiparadigmás programozási nyelvekben

Kutatási terv

ELTE Informatikai Kar Doktori Iskola
Az informatika alapjai és módszertana

Témavezető: Dr. Porkoláb Zoltán

1. Témaválasztás

A modern programozási nyelvek felépítése egyre inkább multiparadigma jellegű. A legújabb C++ szabvány is tartalmaz procedurális, objektum orientált és funkcionális (lambda kifejezések) elemeket is [Str94, Str13]. A C++ nyelvben a lambda kifejezések azonban csak szintaktikus cukorkák, nem hozott be új mechanizmusokat a magnyelvbe.

A könnyen hozzáférhető többmagos rendszerek egyre inkább megkövetelik a többszálú, konkurrens programok fejlesztését, hogy a lehető legjobban ki lehessen aknázni a rendelkezésre álló számítási kapacitást [P.08]. Azonban a konkurrens programok új problémát vet fel: szinkronizáció. Ezért a programozási nyelveknek fel kell készülniük az ilyen rendszerek fejlesztésére is [D.99]. A C++ nyelvhez hasonlóan, a Java nyelv is több programozási paradigmát valósít meg: konkurrencia, objektum-orientált, generikus.

2. Kapcsolódó kutatások

Már a számítógépes grafikával foglalkozók körében is felmerült a kevert programozási paradigmák alkalmazásának a kérdése [Wis11]. Ez annak is betudható, hogy az egyes feladatokat jól megválasztott nyelvben tömörebben és hatékonyabban lehet leírni. A számítógépes grafika

eredendően több platformot céloz meg (pl.: Windows, Linux, Mac), így ha a program több, különálló nyelv segítségével készült, akkor felmerül annak a veszélye, hogy az egyes nyelvek máshogy viselkednek különböző platformokon, vagy esetleg nem is létezik hozzá fordítóprogram.

A konkurrens programozási feladatoknál is felmerül az az igény, hogy különböző feladatokat más és más nyelven írjunk le. Például a funkcionális nyelven írt programot nagyon könnyű párhuzamosítani a nyelv felépítéséből kifolyólag [WS08, HPJ]. Azonban vannak olyan feladatok, melyeket funkcionális nyelven leírva természetellenesnek tűnik – ilyenek például az I/O műveletek – ezért ezeket érdemesebb más nyelven elkészíteni a könnyebb átláthatóság érdekében. Ebből a szempontból az Erlang érdekes nyelv, mert nem klasszikus funkcionális nyelv (mohó kiértékelésű, szekvenciák írhatók, mellékhatásos kifejezések), de a nyelv alapvetően konkurrens rendszerek készítésére való [AJR96].

3. Tervezett kutatási irány

Kutatásaim során először a meglévő multiparadigma nyelvekkel szeretnék mélyebben foglalkozni, az egyes paradigmák és nyelvi eszközök felhasználási módjait megvizsgálni.

A kutatás részét képezi a jelenleg használt eszközök felmérésén kívül a javítási, és továbbfejlesztési lehetőségek vizsgálata is. Ilyen a generikusság kapcsán megfogalmazható kérdések, mint például a megszorított generikusság; az objektum-orientáltság kapcsán a statikus típusrendszer és a altípusosság kapcsolata (strukturált altípusosság, dependent types, family polimorfizmus); a többmagos rendszerek kihasználását támogató nyelvi eszközök (szálak, taszkok); a napjainkban egyre inkább divatos funkcionális nyelvi elemek.

A korábbi tanulmányaim során már foglalkoztam saját nyelv fejlesztésével, melyről egy TDK dolgozat is készült (a kari TDK konferencián első helyezést értem el vele). Ez a nyelv képezné a kutatásom alapját. Ebbe a nyelvbe építeném be kísérleti jelleggel az új nyelvi elemeket,

hogy meg tudjam vizsgálni azok használhatóságát.

Hivatkozások

- [AJR96] Wikstrom C. Armstrong J., Williams M. and Viriding R. *Concurrent Programming in Erlang*. Prentice-Hall, 1996.
- [D.99] Lea D. *Concurrent Programming in Java: Design Principles and Patterns*. Prentice-Hall, 1999.
- [HPJ] Peterson J. Hudak P. and Fasel J. *A Gentle Introduction to Haskell, Version 98*. See www.haskell.org/tutorial.
- [P.08] Van Roy P. *The Challenges and Opportunities of Multiple Processors: Why Multi-Core Processors are Easy and Internet is Hard*. Position statement, in ICMC 2008, Belfast, Northern Ireland, 2008.
- [Sar93] V. A. Saraswat. *Concurrent Constraint Programming*. MIT Press, Cambridge, MA, 1993.
- [Str94] Bjarne Stroustrup. *The Design and Evolution of C++*. Addison-Wesley, 1st edition, 1994. ISBN 0201543303.
- [Str13] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley Professional, 4th edition, 2013. ISBN-10: 0-321-56384-0.
- [W.76] Dijkstra E. W. *A Discipline of Programming*. Prentice-Hall, 1976.
- [Wis11] Peter Wisskirchen. *Object-Oriented and Mixed Programming Paradigms*. Springer, 2011.
- [WS08] Judy Rees Wendy Sullivan. *Clean Language: Revealing Metaphors and Opening Minds*. Crown House Publishing, 1st edition, 2008. ISBN-10: 1845901258.