Erik Meijer & Guy Steele

-- latest news from the programming language research experts

Date: Thursday, October 5, 2006
Time: (**11:15 – 12:15** & **12:30 – 13:30**)
Place: [meeting room & big aud., IT Huset]

Talk 1: ***"A Language Geek Perspective of LINQ, XLINQ, DLINQ"*** **12:30-13:30**
-- by **Erik Meijer** (Microsoft Research, Redmond) meeting room

Modern applications operate on data in several different forms: Relational tables, XML documents, and in-memory objects. Each of these domains can have profound differences in semantics, data types, and capabilities, and much of the complexity in today's applications is the result of these mismatches. The future "Orcas" release of Visual Studio aims to unify the programming models through integrated query capabilities in C# and Visual Basic, a strongly typed data access framework, and an innovative API for manipulating and querying XML.

Talk Rescheduled ~~10:15-11:15~~ → **12:30-13:30**

This talk explains the programming language theory roots (monads and monad comprehensions, lazy/co-inductive functional programming, meta-programming) behind language integrated queries (LINQ) and briefly discusses the language enhancements to support them. We will give an in-depth treatment of the three domain specific APIs that constitute the LINQ framework namely the standard query operators for objects, the new XLinq API for manipulating XML, and the new DLinq and ADO.Net object-persistence infrastructures.

Talk 2: ***"Parallel Programming and Code Selection in Fortress"*** ***11:15-12:15***
-- by **Guy L. Steele Jr.** (Sun Fellow, Sun Microsystems Laboratories) big auditorium

As part of the DARPA program for High Productivity Computing Systems, the Programming Language Research Group at Sun Microsystems Laboratories is developing Fortress, a language intended to support large-scale scientific computation with the same level of portability that the Java programming language provided for multithreaded commercial applications. One of the design principles of Fortress is that parallelism be encouraged everywhere; for example, it is intentionally just a little bit harder to write a sequential loop than a parallel loop. Another is to have rich mechanisms for encapsulation and abstraction; the idea is to have a fairly complicated language for library writers that enables them to write libraries that present a relatively simple set of interfaces to the application programmer. Thus Fortress is as much a framework for language developers as it is a language for coding scientific applications. We will discuss ideas for using a rich parameterized polymorphic type system to organize multithreading and data distribution on large parallel machines. The net result is similar in some ways to data distribution facilities in other languages such as HPF and Chapel, but more open-ended, because in Fortress the facilities are defined by user-replaceable and -extendable libraries rather than wired into the compiler. A sufficiently rich type system can take the place of certain kinds of flow analysis to guide certain kinds of code selection and optimization, again moving policymaking out of the compiler and into libraries coded in the Fortress source language.

Special BRICS [π-λ] Seminar (open to all interested) / Claus Brabrand