Harmonizing Classes, Functions, Tuples, and Type Parameters in the Virgil Programming Language

Ben L. Titzer
Google Munich

Where: Jakob-Haringer-Str. 2, Room T02

When: Friday, January 24, 2014, 11:00 s.t.

Languages are becoming increasingly multi-paradigm. Subtype polymorphism in statically-typed object-oriented languages is being supplemented with parametric polymorphism in the form of generics. Features like first-class functions and lambdas are appearing everywhere. Yet existing languages like Java, C#, C++, D, and Scala seem to accrete ever more complexity when they reach beyond their original paradigm into another; inevitably older features have some rough edges that lead to nonuniformity and pitfalls. Given a fresh start, a new language designer is faced with a daunting array of potential features. Where to start? What is important to get right first, and what can be added later? What features must work together, and what features are orthogonal? We report on our experience with Virgil III, a practical language with a careful balance of classes, functions, tuples and type parameters. Virgil intentionally lacks many advanced features, yet we find its core feature set enables new species of design patterns that bridge multiple paradigms and emulate features not directly supported such as interfaces, abstract data types, ad hoc polymorphism, and variant types. Surprisingly, we find variance for function types and tuple types often replaces the need for other kinds of type variance when libraries are designed in a more functional style.

Ben L. Titzer is a Software Engineer at Google in Munich working on the V8 JavaScript virtual machine, primarily on the optimizing compiler. Prior to V8, he worked on internal tools to process Java server exception logs at Google-scale and profiling tools for Java servers. From 2007 to 2010, he was a Researcher at Sun Labs in Menlo Park where he worked on the Maxine VM, a meta-circular Java-in-Java virtual machine. He initially developed the C1X optimizing JIT compiler for Java which became the precursor the Oracle Labs Graal and Truffle projects. He received a PhD in Computer Science at UCLA in 2007 under Professor Jens Palsberg. In 2005-2007 he designed and implemented the first version of Virgil for embedded devices. In 2004 he wrote the Avrora sensor network simulator, which is still widely used and is one of the DaCapo benchmarks. At Purdue from 2001-2003, he was part of the team that developed OVM, another Java-in-Java virtual machine targeted at real-time applications, under the direction of Jan Vitek.

UNIVERSITÄT SALZBURG

embedded Software & Systems Center
Colloqium Series

host: Andreas Haas