

Usably Expressing and Enforcing Design in Wyvern

Jonathan Aldrich
Carnegie Mellon University
aldrich@cs.cmu.edu

Alex Potanin
Victoria University of Wellington
alex@ecs.vuw.ac.nz

Abstract

The engineering properties of programs derive from their design. As programs grow in scale and complexity, ensuring fidelity to design becomes more critical yet also more difficult. The goal of the Wyvern programming language is to help programmers cleanly express and enforce design as an integral part of programming. Wyvern accomplishes this with a capability-safe object model that expresses design constraints constructively, a strong system of types and effects to enforce abstractions at both component and object scales, and an extensible syntax that can express designs in a variety of domains while preserving important security and modularity properties. Our NOOL talk will flesh out these design goals and demonstrate how the features of the language support them, while highlighting some research directions that have arisen from the language's goals.

CCS Concepts • Software and its engineering → General programming languages

Keywords keyword1, keyword2

References