Static Extraction and Conformance Analysis of Hierarchical Runtime Architectural Structure using Annotations

Marwan Abi-Antoun
Wayne State University

Tuesday, November 17, 2009
3:00pm Rm 110 Purdy-Kresge Library

Abstract:

An object diagram makes explicit the object structures that are only implicit in a class diagram. An object diagram may be missing and must extracted from the code. Alternatively, an existing diagram may be inconsistent with the code, and must be analyzed for conformance with the implementation. One can generalize the global object diagram of a system into a runtime architecture which abstracts objects into components, represents how those components interact, and can decompose a component into a nested sub-architecture. A static object diagram represents all objects and inter-object relations possibly created, and is recovered by static analysis of a program. Existing analyses extract static object diagrams that are non-hierarchical, do not scale, and do not provide meaningful architectural abstraction. Indeed, architectural hierarchy is not readily observable in arbitrary code. Previous approaches used breaking language extensions to specify hierarchy and instances in code, or used dynamic analyses to extract dynamic object diagrams that show objects and relations for a few program runs.

Typecheckable ownership domain annotations use existing language support for annotations and specify in code object encapsulation, logical containment and architectural tiers. These annotations enable a points-to static analysis to extract a sound global object graph that provides architectural abstraction by ownership hierarchy and by types, where architecturally significant objects appear near the top of the hierarchy and data structures are further down.

Another analysis can abstract an object graph into a built runtime architecture. Then, a third analysis can compare the built architecture to a target, analyze and measure their structural conformance, establish traceability between the two and identify interesting differences.

Bio:

Marwan Abi-Antoun, received his Ph.D. from Carnegie Mellon University in June 2009. Prior to CMU, Dr. Abi-Antoun received a M.S. in Computer Science from the University of Southern California. His research interests are in software architecture, type systems and program analysis to specify design intent in code, extract, check, enforce and analyze the architectures of real-world object-oriented systems.