Static Analysis of Task Interactions in Bristlecone for Program Understanding

Brian Demsky and Sivaji Sundaramurthy
Institute for Software Research
University of California, Irvine
{bdemsky, srsundar}@uci.edu

ISR Technical Report # UCI-ISR-07-7

October 2007

Abstract

We have developed a static analysis to help developers understand the interactions between objects and tasks in Bristlecone applications. The Bristlecone language was designed to help developers construct robust applications out of potentially unreliable components. Bristlecone applications can adapt their behavior, potentially degrading their functionality, in response to software errors in order to avoid catastrophic failures.

Bristlecone applications are composed of a set of tasks. The description of the behavior of these tasks is split into two orthogonal specifications: a set of high-level task specifications that describe both when the task should be invoked and which objects the task operates on and low-level imperative specifications that describe the operational behavior of the task.

The Bristlecone compiler and runtime use the high-level specifications to detect software errors, to recover the application from an error to a consistent state, and to reason how to safely continue the application's execution after the error.

This paper presents a static analysis that automatically extracts information about the interaction of objects and tasks, a set of graphical representations that capture the relevant information about these interactions, and a web-based, interactive tool that uses these graphical representations to communicate this information to the developer. We have used this tool to explore the behavior of several benchmark applications including an online game, a web server, and a chat server. Our experience indicates that these analysis results are useful for understanding the interaction between tasks and objects in our benchmark applications and correcting several software bugs.