Jordan University of Science and Technology

An Extensible Debugging Architecture Based on a Hybrid Debugging Framework

Authors: Ziad A. Al-Sharif

Abstract: The cost of writing debuggers is very high. Most debuggers are written employing low level operating system and hardware specific code, which is hard to port to new platforms or architectures and to extend with new debugging techniques. Moreover, current debuggers are usually limited in the amount of analysis that they perform and the level of detail that they provide in order to assist with debugging. Most debuggers are well suited for a specific class of bugs. Different bugs call for different debugging techniques, so experimentation is needed in order to develop the features that will someday be widely adopted in debuggers. This dissertation contributes three primary results. First, it introduces an event-driven debugging framework named AlamoDE (Alamo?Debug Enabled). The role of this framework is analogous to an abstraction layer upon which to build debuggers. AlamoDE 1) provides in-process debugging support with simple communication and no intrusion on the buggy program space, 2) enables debugging tools to be written at a high level of abstraction, and 3) facilitates developers of experimental automatic debugging features in a very high level language. AlamoDE supports construction of a variety of user-defined debugging tools that range from classical source-level debuggers to automated and dynamic analysis techniques. Second, this dissertation presents an extensible agent-based debugging architecture named IDEA (Idaho Debugging Extension Architecture). IDEA offers novel debugging techniques that break the rigidness, closeness, and inextensibility of most current debuggers. It provides programmers with the ability to easily implement, test, and combine user-defined debugging agents, and offers a simple dynamic and static extension mechanism. Finally, this dissertation provides a production source-level debugger for the Unicon language named UDB. UDB leverages the classical interactive debugging process with 1) built-in agents employing automatic detect