Dynamic Invariant Detection for Program Understanding and Reliability
MIT2001-01

Progress Report: July 1, 2001—December 31, 2001

Michael Ernst

Project Overview

The Program Analysis Group (PAG) of MIT is collaborating with Tadashi Araragi of NTT on problems raised by NTT's Erdös internet agent system. The goal is to formally verify mobile agent systems in which agents may be widely separated and may move about a network, yet must communicate, coordinate, and negotiate in order to perform useful work. Examples include internet marketplaces (in which directories of shops must be maintained), auctions (in which programs may be dynamically downloaded), and airline ticket reservation (in which multiple ticket agencies must have a consistent view of a database). In each of these cases, it is crucial that the system perform correctly, yet it is difficult to understand and verify it. PAG has provided several technologies that can increase confidence in such systems.

Progress Through December 2001

PAG's foremost contribution is a technique for automatically generating program specifications, and an implementation (named Daikon) of the technique. Program specifications are required for most program understanding, maintenance, and verification tasks, but are rarely present. One reason is that people find it tedious, and error-prone to write specifications; another is that it is as difficult to write a good specification as to write the program in the first place. Therefore, automating the task holds great promise. The Daikon system automatically generates specifications; it requires only a program and a test suite. Tadashi Araragi of NTT visited MIT in August and gained experience with the Daikon system; he then took it back to NTT with him in order to use it there. PAG made several enhancements (notably to the documentation, but also to some functionality) in response to his feedback, and five new releases of Daikon have been published during the last six months. While NTT is able to use Daikon in Japan, PAG has been hindered in replicating those experiments at MIT because of intellectual property concerns. However, in the meanwhile PAG has made progress on several closely related research areas that are also required to meet the research goals.

Because the final goal of this NTT-MIT project is program verification, PAG has been investigating proofs of program correctness in two domains: Java and the IOA programming language. The Java experiments have used the ESC/Java program checker from the Compaq Systems Research Center. Given a program annotated with specifications, it checks the specifications and indicates which specifications are inconsistent and whether

any run-time errors (such as null pointer dereferences and array bounds errors) are possible. Because programmers are reluctant to write such specifications, Daikon's specifications can be used as a starting point. Our first result is an automated system that integrates Daikon and ESC/Java. Our second result is an experimental analysis that shows that the system is usually very accurate: about 90% of the properties it reports are automatically verifiable, and it reports about 90% of properties required in order to perform verification. Our third result is a user study that shows that humans perform better on a program verification task when provided with Daikon output.

IOA is a language for modeling distributed systems that is being developed by Prof. Nancy Lynch's Theory of Distributed Systems (TDS) group. It is the subject of another NTT-MIT collaboration -- one which we have very fruitfully collaborated with and intend to continue working with closely. In fact, much of this work has been done in conjunction with TDS. The IOA experiments use the LP theorem-prover written by Dr. Stephen Garland (another NTT-MIT PI); there is an automatic mapping from IOA to LP formulas. However, the human user must still provide many small facts about the program; these are distracting and easy to forget. We have recently extended Daikon so that it produces many of the desired invariants and intermediate steps required for a proof in LP. Human direction is still required for the high-level direction of the proof and for some of the smaller steps. The results for both Java and IOA programs suggest that similar verification results may be possible in the Erdös system.

Another goal of this NTT-MIT project was detecting temporal invariants -- properties that describe behavior over time. This problem has proved less tractable than we initially hoped, but we have recently started to make some better progress on it, and it will be a focus for the upcoming six months.

Research Plan for the Next Six Months

Our future plans include continuing to improve and support the Daikon system, including scaling it to larger programs and continuing to support users with direct assistance and new distributions; investigating new techniques for generating specifications, with a particular emphasis on temporal invariants, which NTT is particularly interested in; extending our successful work on proofs of correctness in both the Java and IOA languages; and beginning to work directly on NTT codes as they become available to us.