Verification Tools Secure Online Shopping, Banking

Originating Technology/NASA Contribution

Much is made of the engineering that enables the complex operations of a rover examining the surface of Mars—and rightly so. But even the most advanced robotics are useless if, when the rover rolls out onto the Martian soil, a software glitch causes a communications breakdown and leaves the robot frozen. Whether it is a Mars rover, a deep space probe, or a space shuttle, space operations require robust, practically fail-proof programming to ensure the safe and effective execution of mission-critical control systems.

Java Pathfinder has been used to analyze ground control software for the space shuttle.

Just as rovers are rigorously tested in simulated Martian conditions on Earth before actual mission launch, the software components must also be thoroughly analyzed to ensure the absence of bugs that might cause complications or critical failures. NASA's Robust Software Engineering (RSE) group, part of the Intelligent Systems Division at Ames Research Center, works to develop automated verification and validation technologies for aerospace and aviation software, making certain these programs are correctly written. In 1999, the group began developing a powerful verification tool, called Java Pathfinder (JPF), for programs written in the popular Java programming language.

"Java Pathfinder started as a model checker, which exhaustively goes through all of the possible behaviors of a program," says Corina Pasareanu, a Carnegie Mellon University and Ames RSE researcher. Since then, she says, JPF has evolved into a toolset including several technologies, one being a symbolic execution tool called Symbolic Pathfinder, which is the focus of Pasareanu's work. In 2005, JPF became one of the first NASA programs to be open-sourced, and JPF and Symbolic Pathfinder have garnered multiple awards, including ones from NASA, the Federal Laboratory Consortium, and IBM among others.

Partnership

JPF was the original tool for verifying native Java code. Fully testing Java code typically requires massive quantities of input to explore every possible program path—hence the Pathfinder name, also inspired by the Mars Pathfinder mission—and to verify the absence of bugs. These inputs are often generated manually, which requires significant time and effort. Symbolic Pathfinder is NASA's answer to that problem. The tool uses symbolic execution, in which inputs are specified as symbolic variables, able to take any value within a numeric range. The tool analyzes the code structure of the target program and generates symbolic constraints for the program's variables. These constraints are solved to automatically create test cases that reach all parts of the program code. Essentially, Symbolic Pathfinder has the capability to automatically execute a program on all possible inputs and in all possible ways to find defects and what causes them. The tool offers extreme thoroughness at less time, effort, and cost.

"We have applied this technology to several projects at NASA, generating test cases automatically and uncovering bugs in NASA software," says Pasareanu. Flight control software, scripts for commanding robots, and air traffic management software are a few of the NASA components that have benefited.

A significant deficiency with Symbolic Pathfinder, however, came to the fore when Fujitsu Laboratories of America Inc., based in Sunnyvale, California, began exploring the use of the open-source tool for testing Java programs for Web applications. Fujitsu's Software Validation Program, part of its Trusted Systems Innovation Group, works to develop verification and validation techniques to enhance public "trust" in information technology systems that play a major role in daily life, like wired and wireless networks, e-commerce, banking, and government systems, says Sreeranga Rajan, senior researcher for Fujitsu.

"The problem was that the NASA tool only handled integers, Booleans, real numbers, and recursive data structures as inputs. As soon as we tried to apply it to Web and enterprise applications, we found that the inputs are text strings," says Fujitsu researcher Indradeep Ghosh. Building on the Symbolic Pathfinder tool, Fujitsu created a new capability to handle text input variables as symbolic

inputs, thus allowing significantly broader verification of Java programs.

Product Outcome

If you do your banking or shopping online, you are most likely using Web sites enabled by Java applications. These applications (shopping carts, for example) can now be tested using Symbolic Pathfinder.

> ***"This is the ideal case for government lab and industrial lab collaboration."***

"If you go to a banking application, you might have to input a check routing number, which is alphanumeric," says Rajan. Rajan explains that safety-critical classes of software, such as those involving financial transactions or for medical use, necessitate much deeper bug finding than typical testing techniques. Fujitsu's enhanced Symbolic Pathfinder tool, he says, "explores the program's behavior, analyzes the potential problem cases, and thereby helps cover a larger set of behaviors that typical testing programs cannot cover. That means we are able to unearth bugs that would not be found with existing technologies."

For commerce applications like shopping carts, Symbolic Pathfinder becomes a tool for ensuring security. With the new string input extension, Rajan says, Symbolic Pathfinder can help detect security holes that originate by string inputs.

Fujitsu plans to continue its work with JPF, scaling up its capabilities for verification of large-scale, complex Java programs. Rajan anticipates the company will use the toolset for in-house software development. The benefits of Fujitsu's work are also returning to NASA: The company has open-sourced its string input extension to allow the Ames JPF team and others to continue the tool's development.

"NASA was able to develop this complicated technology, something an industrial lab probably wouldn't have the time or resources to do," says Ghosh. "Once the technology matured, we could immediately see that, by improving it in a year or 2-year time frame, we could consider using it in an industrial setting. This is the ideal case for government lab and industrial lab collaboration."

This photo, taken during Mars Pathfinder testing, shows the Pathfinder lander opening its petals to expose the yellow Sojourner rover (on left petal). Just like robotics and other hardware, control software must be extensively tested to ensure proper execution of commands.

Rajan notes that a significant amount of software development goes into security systems, and that the sharing of experience and capabilities between government and private industry can improve security for industry and for the Nation alike.

"NASA had developed in JPF something nobody else had developed for Java, and because of that we were able to get to the stage we are at now," he says.

"Fujitsu Laboratories is glad to see this advancement in symbolic execution technology for strings transitioning to practicality," says Sanya Uehara, corporate senior vice president and director of Fujitsu Laboratories Ltd. "Such a milestone that has so far eluded verification researchers has been reached through collaboration between Fujitsu Laboratories in Japan and the United States, inspired by NASA's work on Symbolic Pathfinder." ❖