

Nissan Increases Software Reliability with PolySpace™ Products for C/C++

Nissan Motor’s overriding concern is quality. “Nissan has an important responsibility to its customers,” explains Mitsuhiko Kikuchi, leader of the Software Quality Group of Nissan Motor Company. “This is why we ensure that our vehicles meet stringent quality standards. These standards also apply to the software embedded aboard our vehicles. To ensure that quality processes are effectively used in all electronic control units (ECUs), the Software Quality Group examines the software review process with all suppliers right from the beginning of the project, and then again at each major milestone. The Software Quality Group also reports the status of software quality directly to Nissan executives.”



The Nissan Fairlady Z.

and other issues. Our coding rules and internal structure inspection tools could address some of these problems. However, these tools could not locate run-time errors, such as divisions by zero, overflows, and out-of-bound array accesses.”

THE SOLUTION

Mr. Kikuchi first learned about PolySpace™ Client for C/C++ and PolySpace Server for C/C++ from colleagues within the global Renault-Nissan alliance.

Nissan performed benchmark testing to evaluate the ability of PolySpace products for C/C++ to find run-time errors. “We were already using two static tools. We decided to try PolySpace products for C/C++ during a pilot project. Upon the successful completion of the pilot, we added PolySpace products for C/C++ to the two other tools and extended utilization to all software we would review. Since suppliers have their own specific software, including different development environments, micro-processors, and cross-compilers, we also designed an efficient process so we could analyze their code quickly.”

THE CHALLENGE

The Software Engineering Evolution Program (SWEEP) is in charge of assessing the software development processes of Nissan’s suppliers—including design, coding, and testing. Until 2001, SWEEP defined its quality goals mainly in terms of classical development techniques based on unit, integration, or system tests. These were crossed-checked against robustness objectives defined for each development activity.

“Nissan and the suppliers had to spend a lot of human resources to ensure flawless software robustness during the testing phase,” Mr. Kikuchi says.

At that time, the Software Quality Group was using two static tools that automated coding rules checking and the inspection of the source code’s internal structure. That left a real unmet need. As Mr. Kikuchi explains, “Nissan was experiencing problems with software bugs that were caused by increasingly complex architecture, coding errors,

THE CHALLENGE

Identify hard-to-find run-time errors to improve software quality

THE SOLUTION

Use MathWorks tools to exhaustively analyze Nissan and supplier code

THE RESULTS

- Supplier’s bugs detected and measured
- Software reliability improved
- PolySpace™ products for C/C++ adopted by Nissan suppliers

“PolySpace products for C/C++ can ensure a level of software reliability that is unmatched by any tools in the industry.”

Mitsuhiko Kikuchi, Nissan

Nissan defines three levels of gravity for every bug identified: major (must be fixed immediately), medium (should be fixed in a future release), and minor.

Mr. Kikuchi identifies the constraints and criteria that will be used to review previously validated code for the presence of run-time errors using PolySpace products for C/C++.

“PolySpace products for C/C++ not only find which operations can experience run-time errors, they also identify those that will never have one, no matter the operating conditions”, says Mr. Kikuchi. “Furthermore, they can do so during coding, thus before unit testing. This is of tremendous value to our suppliers.”

“While we do not force members of our value-added chain to use PolySpace products for C/C++ in their development process, we regard a systematic utilization by a supplier as a great plus. Knowing how and when PolySpace products for C/C++ were used during coding gives us great confidence in code reliability. It provides us with a guarantee that software robustness and reliability are ensured in the most efficient way possible,” concludes Mr. Kikuchi.

THE RESULTS

■ **Supplier’s bugs detected and measured.** “With PolySpace products for C/C++, results are quite easy to measure,” says Mr. Kikuchi. “We find that about 5 bugs per project—about 30K lines of code, or 100K of ROM—are major bugs. These must be fixed immediately by the supplier,” says Mr. Kikuchi. These projects were supposed to be validated already. “That means that these bugs could have been found earlier and for less by using PolySpace products for C/C++.”

■ **Software reliability improved.** PolySpace products for C/C++ ensure that applications perform reliably while costing a lot less to perform than conventional tests. Not to mention the fact that functional testing is no longer impeded by run-time errors!”

■ **PolySpace products for C/C++ adopted by Nissan suppliers.** Many Nissan suppliers are working to integrate PolySpace products for C/C++ into their internal development process. “Our suppliers did not adopt PolySpace products for C/C++ just because we highly recommended them. In fact, they know that ensuring software reliability earlier in the process is the best way for them to deliver quality applications at a fraction of the cost. They also know that the only tools that deliver the kind of exhaustive results that make this possible are PolySpace products for C/C++,” Mr. Kikuchi says.

To learn more about Nissan, visit

www.nissan-global.com

www.mathworks.com

APPLICATION AREAS

- Automotive
- Embedded code verification

PRODUCTS USED

- PolySpace™ Client for C/C++
- PolySpace™ Server for C/C++