

## **Complex software systems -- heal thyself**

May 3 2010



(PhysOrg.com) -- Software underlies modern life, keeping everything from mobile phone networks functioning to planes in the air, but ensuring increasingly complex systems stay free of faults has become an epic task. What if software could heal itself?

Researchers from Israel and six EU countries have carried out pioneering work on self-healing <u>software</u> capable of automatically and autonomously detecting, identifying and fixing errors in the copious lines of code that make up <u>complex systems</u>. The results of their research are already being used internally by several companies and could feed into commercial products in the near future.

"Software systems have grown increasingly large and complex as we come to rely on them to do more things. Just making a single mobile phone call may involve hundreds of systems operating behind the scenes



and all of them need to work properly," notes Onn Shehory, a researcher at IBM in Haifa, Israel.

We are talking about hundreds of systems containing hundreds of thousands or even millions of lines of code. And if just a tiny part of that code is wrong - due to design flaws or faults introduced while in use performance will be degraded or the system may not function at all. Fixing software faults has, until now, meant calling on <u>software</u> <u>engineers</u> to sift through the code to identify the cause, locate it and repair it, a process that could be compared to searching for a needle in a digital haystack.

Tools developed by a team of researchers, coordinated by Shehory and funded by the European Union in the SHADOWS project, do the sifting, identifying and fixing automatically. The approach relies on a set of detection-localisation-healing-assurance loops that function in the background of complex software systems, without the need for human intervention.

The detection stage reveals or predicts the presence of problems, such as functional deviations, performance bottlenecks or concurrency problems. The localisation stage identifies the fault that caused the issue. The healing stage provides automatic or semi-automatic problem remediation. And, finally, the assurance stage examines the healing that has been done to ensure it solved the problem and no new problems were introduced.

A unified framework, based on open standards, such as Eclipse, provides a single methodology and architecture.

"Say you have several hundred thousand lines of code. We don't analyse all of it but instead look at those areas - perhaps 10,000 lines - that have been identified as being at greater risk of faults. Monitoring it all would



be too costly as the load on the system from the healing software would be greater than from the software that is being monitored," Shehory explains.

When a fault is detected and its cause found, the tools can automatically apply a series of predefined solutions until it is resolved. In addition, the tools can be used to generate a model describing how a software system should function in a set of typical scenarios. These models can then be used to make comparisons with how it is functioning in reality.

"This is particularly useful when comparing different versions of the same software," Shehory says.

By using aspect-oriented development, the researchers designed their tools to function with legacy systems, ensuring that companies do not have to "reinvent the wheel" and redesign their existing software in order to incorporate self-healing features. This makes the SHADOWS tools cost-effective and relatively simple to implement.

## **Contained risks**

The team also worked on tools and drew up guidelines for developers creating new software, to encourage the development of software for complex systems with built-in self-healing capabilities.

"It will take time for this to be widely accepted by developers as they have to be able to trust tools that are going to act autonomously," Shehory notes. "Yet, results already achieved by project partners using the SHADOWS technologies demonstrate that risks due to autonomy are well contained."

Companies have already started applying the tools with success, with one telecommunications firm having used the SHADOWS approach to



identify and correct a long-running fault in its call servers.

"There is a very real need for self-healing solutions among users of software... although I think the biggest initial demand is from software developers who want to reduce software testing times," Shehory explains.

"They tell us that if our tools can reduce the time it takes to test for bugs and errors by just a few weeks it would be a major advantage." Several of the project partners are continuing to work together, and a follow-up project is planned with that goal in mind.

Provided by ICT Results

Citation: Complex software systems -- heal thyself (2010, May 3) retrieved 24 May 2024 from <u>https://phys.org/news/2010-05-complex-software-thyself.html</u>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.