Software Engineering Seminar

# Automated Repair of Deployed Software

## Description

Under certain circumstances, fixing errors in already deployed – or maybe even running – software systems can be necessary. Reasons could be, for example, that the source code of the system is not available or that the system is indefinitely running and must not be stopped. Obviously, the applied techniques in this area vary greatly from approaches that deal with the debugging on source code level. For example, the tool ASSURE [3] introduces so-called *rescue points* that provide points of safe return to a software system in case of an occurring error. That means that ASSURE does not actually try to repair existing errors, but that it lets the software system recover from occurring faults that would otherwise harm the integrity of the system or lead to a crash. Another tool, ClearView [1], detects and corrects errors in running software systems by learning invariants from correct executions and enforcing those invariants in case of an occurring error.

The goal of this topic is to examine and discuss techniques to automatically debug and repair deployed or running software systems.

## References

[1] Jeff H. Perkins, Sunghun Kim, Sam Larsen, Saman Amarasinghe, Jonathan Bachrach, Michael Carbin, Carlos Pacheco, Frank Sherwood, Stelios Sidiroglou, Greg Sullivan, Others, Weng-Fai Wong, Yoav Zibin, Michael D Ernst, and Martin Rinard. Automatically patching errors in deployed software. *Symposium on Operating Systems Principles*, pages 87–102, 2009.

[2] Eric Schulte, Jonathan DiLorenzo, Westley Weimer, and Stephanie Forrest. Automated repair of binary and assembly programs for cooperating embedded devices. *SIGARCH Comput. Archit. News*, 41(1):317–328, March 2013.

[3] Stelios Sidiroglou, Oren Laadan, Carlos R Perez, Nicolas Viennot, Jason Nieh, and Angelos D Keromytis. ASSURE : Automatic Software Self-healing Using REscue points. *Architectural Support for Programming Languages and Operating Systems*, pages 37–48, 2009.

## Contacts

Simon Heiden (heiden@informatik.hu-berlin.de)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin