Robustness Patterns: coping with software bugs at run-time

P. Felber, U. of Neuchatel and C. Fetzer, Dresden U.

Only for the most safety-critical applications can we expect, or at least hope, that few or no faults remain in software deployed to the field. For most other domains, a variety of economical and technical factors lead to the deployment of software that contains faults, i.e., software bugs. For example, time-to-market requirements might demand the shipment of premature software and only during the lifetime of a system does the software dependability need to increase.

In this tutorial, we will present several widely used robustness patterns that were introduced over the last few decades. A first focus will be on mechanisms that have been successfully applied in practice. A second focus is on recently proposed patterns like Rx or Failure Oblivious computing.

The purpose of this tutorial is first to give practitioners a good overview of robustness patterns to improve the dependability of software, and second to give graduate students and researchers an up-to-date overview of the current research in this field.