An Exercise in Systematically Deriving Fault-Tolerance Specifications

Felix Gärtner

TU DARMSTADT, GERMANY
DEPARTMENT OF COMPUTER SCIENCE

Fault-Tolerance Specifications …

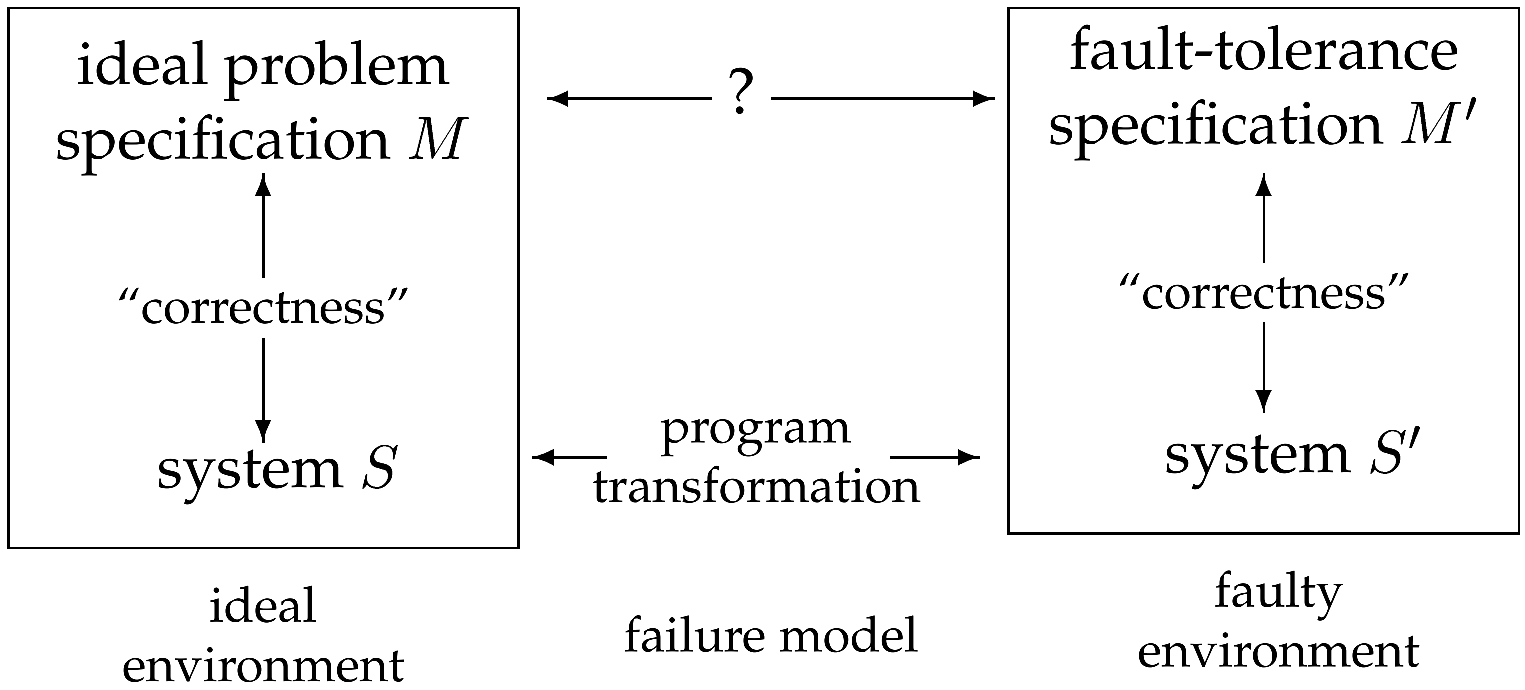
- are necessary in the formal verification of fault-tolerant systems
- are (weakened variants of) original correctness specifications
- reflect effects of faults

Low-Level Fault Modeling of *crash*

crash = processes stop executing steps

$$\begin{array}{rcl} \mathsf{true} & \longrightarrow & up := \mathsf{false} \\ up \wedge \langle \text{guard} \rangle & \longrightarrow & \langle \text{command} \rangle \end{array}$$

Ideal vs. Fault-Tolerance Specification



High-Level Fault Modeling of *crash*

Consider some process p_i as part of a larger system. Local impact of *crash*:

- add a crashed state c_i to state space of p_i
- (safety) p_i operates normally unless it has reached c_i
- (safety) once in c_i, always in c_i
- (liveness) eventually p_i will enter c_i (crash!)

Global assumption: no more than t processes crash.

Exercise: *consensus* under *crash*

How does ideal *consensus* specification change under crash?

- *crash* influences only liveness properties
- *crash* is a specification transformation defined by local and global parts
- conjoining global part yields *uniform consensus* [Hadzilacos and Toueg 1994]

Conclusions and Open Questions

- high-level fault modeling can help understand fault models
- transformation yields exactly one specification
- comparison with “lattice methods” [Herlihy and Wing 1991; Arora and Kulkarni 1998]
- can we prove that *uniform consensus* is the strongest form achievable under *crash*?
- study form and quality of transformations independent of fault model and problem specification

References

Arora, A. and Kulkarni, S. S. 1998. Component based design of multitolerant systems. *IEEE Transactions on Software Engineering 24*, 1 (Jan.), 63–78.

Hadzilacos, V. and Toueg, S. 1994. A modular approach to fault-tolerant broadcasts and related problems. Technical Report TR94-1425 (May), Cornell University, Computer Science Department.

Herlihy, M. P. and Wing, J. M. 1991. Specifying graceful degradation. *IEEE Transactions on Parallel and Distributed Systems 2*, 1 (Jan.), 93–104.