

**Review of Cabodi G., Murciano M., and Violante M.
Boosting Software Fault Injection for Dependability
Analysis of Real-Time Embedded Applications.
ACM Transactions on Embedded Computing Systems
10, 2 (2010). 1-32.**

T.H. Tse,
Department of Computer Science,
The University of Hong Kong,
Pokfulam, Hong Kong

Embedded systems today are large and complex, composed of a variety of components from different sources. These may be commercial off-the-shelf components, which are often less fault tolerant than their traditional counterparts. It is essential to assure the dependability of such systems, including robustness and availability, especially if the systems are safety critical or if system failures would result in heavy financial losses.

This paper proposes an integrated approach to user-level runtime software fault injection, which covers transient faults in both virtual and physical hardware. It is grounded on the loadable kernel modules of the operating system, which is architecture independent, and is exploitable as soon as a system prototype is available. No instrumentation in the kernel code or application under investigation is necessary. With the authors' approach, rather than having to wait for hardware delivery, one can conduct early deployment of the dependency analysis on a virtual hardware platform, and then follow up on the physical hardware platform. Decoupling dependency analysis from hardware availability makes concurrent development of software and hardware feasible. Experiments indicate that the evaluation results on virtual hardware are comparable with those on physical hardware.

In order to truly confirm the contributions of the proposed approach, it would be useful if the authors could validate the significance of the experimental results using standard statistical techniques.

Furthermore, I am quite surprised about the number of typos present in an *ACM Transactions* paper, including text and figures. Apparently, the authors and the publisher should also examine the *dependability* of the proofreading process.