Comparing and Validating Measurements of Dependability Attributes

Daniel Skarin    Raul Barbosa    Johan Karlsson

Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden

EDCC-8
April 28, 2010
Introduction

- Fault injection is becoming widely used in the development of safety-related systems.
  - The upcoming ISO 26262 lists fault injection as highly recommended for ASIL C and D.
- Numerous fault injection tools and techniques exist today.
- These techniques and tools all have their pros and cons.
- Many factors can affect the results of fault injection experiments.

Open issues

- Can results be reproduced using different fault injection techniques?
- What are the sources of uncertainties?
Contributions

We evaluate the metrological compatibility of measurement results obtained by three diverse fault injection techniques.

- Identify sources of uncertainty.
- Investigate if results obtained by the three fault injection techniques can be compared.
- Validate the correctness of the implementation of each technique.
Fault injection tools can be seen as measurement instruments.

The measurand is the quantity intended to be measured.

Measurement results in general consist of
  - A measured value
  - A measurement uncertainty

Uncertainty expresses doubt about the validity of a measurement result.
  - Characterizes the dispersion of values that could be attributed to a measurand.

Measurement results are metrological compatible if they refer to the same measurand.
Sources of uncertainty in fault injection

Measurement results are affected by

- Initialization uncertainty
  - Non-initialized memory values in a computer system
- Spatial and temporal intrusiveness
  - Changes made to the target system needed for the experimental evaluation.
- Assumptions in the measurement procedure
  - Unforeseen external events, or internal events affecting the target system, e.g., caused by design flaws in the hardware or software.
- Instrumental uncertainty
  - The tools used for the measurements are not always validated.
Investigation of uncertainties

Objectives

▶ Identify sources of uncertainty in our experimental setup
▶ Validate the implementations of three diverse techniques

The same set of faults was injected using:

▶ Test port-based injection
▶ Instrumentation-based injection
▶ Exception-based injection

Experiments were conducted with two target systems.

▶ A brake-by-wire controller running on a MPC565 microcontroller.
▶ Secern – A real-time kernel running on a MPC5554 microcontroller.
The GOOFI-2 fault injection tool

The previous version of GOOFI provides:

- **Test port-based injection**
  - Uses a Nexus-compliant debugger to inject faults.
  - No spatial intrusiveness, but high temporal intrusiveness.

The current version adds support for two well-known techniques:

- **Instrumentation-based injection**
  - Software-implemented technique that instruments the executable file with a fault injection routine.
  - Low temporal intrusiveness, but also low observability

- **Exception-based injection**
  - Injects faults using exception routines that are activated by hardware exceptions.
  - Low temporal intrusiveness, but high spatial intrusiveness.
Brake-by-wire controller

- Designed specifically for future brake-by-wire systems.
- Software-implemented mechanisms for detection and recovery of transient hardware faults.
- Injected the exact same faults using all three techniques.
  - Single bit-flip errors in general purpose registers and the data memory (1000 injections each).
Results – Brake-by-wire controller

- All three techniques produced identical results for errors injected in memory.
- Faults injected in general purpose registers had different outcomes in several cases.
  - Test port/instrumentation-based: 99% experiments matched
  - Test port/exception-based: 93% experiments matched
  - Exception/instrumentation-based: 92% experiments matched
Results – Brake-by-wire controller

Errors injected into registers (GPRs)

Proportion of incorrect outputs and the 95% confidence interval

![Graph showing the proportion of incorrect outputs for different methods. Test port-based has 76.4%, Instrumentation-based has 75.5%, and Exception-based has 77.5%.]
Extension to the \(\mu\)C/OS-II real-time kernel that provides robust partitioning.

- Ensures that a faulty task cannot affect the kernel or other tasks using, *e.g.*, memory protection.

A task’s accessible memory is initialized at startup.

\[\Rightarrow\] Initialization uncertainty is minimal.

Injections of single bit-flip errors using the test port-based and the instrumentation-based technique.

- The fault set included 1000 injections in general purpose registers and 1000 injections in the data memory.
The two techniques obtained identical results, except for a single fault. This fault was injected into an unused register. Instrumentation-based injection caused a task to be removed. Test port-based injection did not have any impact. We could not reproduce this behavior by repeated injections of the specific fault. Subsequent injections were all classified as having no impact.
Identified sources of uncertainty

- Brake-by-wire controller
  - *Initialization uncertainty* caused differences in the outcome of many individual experiments for all three techniques.
  - The high *spatial intrusiveness* of the exception-based technique caused additional differences in the obtained results.

- Secern
  - The different outcome for the single fault was attributed to *assumptions in the measurement procedure*. 
Tool validation

- The three techniques produced similar average results over a large number of experiments.
- The cases in which the outcomes differed were not caused by bugs in the tool, but rather by measurement uncertainties.
- We can therefore be confident that the instrumental uncertainty is minimal.
- If we also consider the temporal aspect, the techniques may produce different results due to variations in their temporal intrusiveness.
Conclusions

- We identified three sources of uncertainty affecting our three fault injection techniques.
- The outcome of many individual experiments was affected by these uncertainties.
- These uncertainties were related to injections in general purpose registers.
  - The proportion of individual experiments with different outcome varied from 1% to 8%.
- Despite these uncertainties, the results obtained from the different fault injections techniques were similar.
- Our results show that the three techniques obtain metrologically compatible results.
Future work

- Quantify the identified uncertainties.
- Investigate how results for a given system in a specific environment can be extrapolated.
- Study how the number of experiments affects fault forecasting estimations and fault removal effectiveness.
Fault injection results available on-line
http://www.amber-project.eu

COMPARING DEPENDABILITY MEASUREMENTS OBTAINED USING DIFFERENT INJECTION TECHNIQUES

Study Description

Raw Data & Documents

OLAP

Data Mining

SQL

Information Retrieval

This study examines the compatibility of measurements obtained using different fault injection techniques. Two software-implemented techniques and one test point-based technique are used to inject the same set of faults, which are defined in a shared database. Due to the uncertainties associated with each technique, the results of injecting a given fault may differ to some extent. This study analyzes the results of using the three techniques to inject faults into a brake-by-wire controller.

Author(s): Raul Barbosa, Daniel Skarin and Johan Karlsson