GOOFI: Generic Object-Oriented Fault Injection Tool

Joakim Aidemark, Jonny Vinter, Peter Folkesson, and Johan Karlsson
Department of Computer Engineering
Chalmers University of Technology
S-412 96 Göteborg, Sweden
{aidemark, vinter, peterf, johan}@ce.chalmers.se

1. Introduction

Most fault injection tools have been developed with a specific fault injection technique in mind targeting a specific system, and using a custom designed user interface. The GOOFI (Generic Object-Oriented Fault Injection) tool [1] is designed to be adaptable to various target systems and different fault injection techniques in a user-friendly way.

The GOOFI tool has been used to evaluate several new EDMs and ERMs for control systems such as brake-by-wire systems and engine control systems, e.g. see [2].

2. Architectural design

GOOFI is highly portable to different host platforms since it is developed using the Java programming language and relies on an SQL compatible database. The design is divided into a three-layered architecture (see Figure 1). At the top layer is the graphical user interface (GUI). The middle layer contains the Fault Injection Engines package that handles the fault injection, while the bottom package includes the classes needed to communicate with the database.

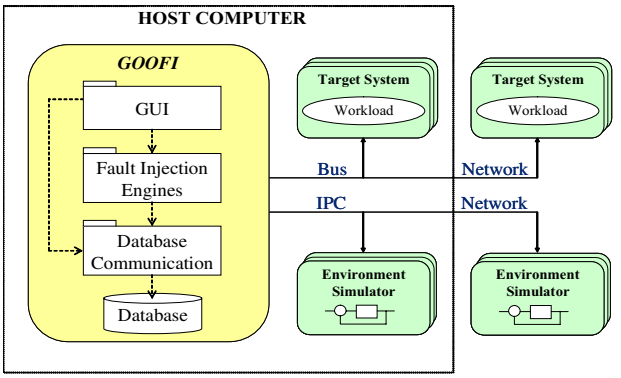

Figure 1. The GOOFI architecture.

The tool is developed to support validation of fault-tolerant embedded computer systems which can be connected to the GOOFI host computer either via local buses or to other computer hosts connected remotely via the Internet allowing the user to setup, perform and supervise experiments remotely from the target system.

GOOFI also supports the use of environment simulators, which can be executed either on the host computer or on another computer on the network.

3. Adapting and using GOOFI

To adapt the tool to a new fault injection technique, new GUI windows and a new Fault Injection Engine must be added. The engine is an algorithm defined by abstract methods (undefined procedures) that describes how fault injection is conducted. Different engines are required for different techniques, e.g. for SCIFI or SWIFI.

When GOOFI is adapted to a new target system, a new class is created that inherits the engine for a chosen fault injection technique. This enables the fault injection algorithm to be reused. Only the abstract methods used by the algorithm need to be implemented for the specific target. The user stores the possible fault injection locations, fault models, etc., into the database. The GUI is updated with this information.

To perform a fault injection campaign, the user selects a target system, fault locations, fault model, the points in time faults should be injected, environment simulator, termination conditions, etc. via the GUI. The defined campaign data is stored in the database and is used to perform the fault injection experiments. The outcome from the experiments is saved in the database and can later be analyzed to obtain various dependability measures such as error detection coverage etc.

4. References

[1] J. Aidemark, J. Vinter, P. Folkesson, and J. Karlsson, "GOOFI: Generic Object-Oriented Fault Injection Tool," *DSN 2001*, Gothenburg, Sweden, July 2001.

[2] J. Vinter, A. Johansson, P. Folkesson, and J. Karlsson, "On the Design of Robust Integrators for Fail-Bounded Control Systems," *DSN 2003*, San-Francisco, USA June 2003.