Contributions to the Design for Testability of Distributed Systems

dipl. eng. Enyedi Szilárd

Doctoral thesis summary

THESIS ANALYSIS COMMITTEE

PRESIDENT
Prof.dr.eng. Marina Țopa – dean, Faculty of Electronics, Telecommunications and Information Technology, Technical University of Cluj-Napoca

MEMBERS
Prof.dr.eng. Gavril Toderean – scientific advisor, Technical University of Cluj-Napoca
Prof.dr.eng. Sergiu Iliescu – reviewer, “Politehnica” University Bucharest
Prof.dr.eng. Mircea Vlăduțiu – reviewer, “Politehnica” University of Timișoara
Prof.dr.eng. Șerban Agachi – reviewer, Babeș-Bolyai University, Cluj-Napoca
Prof.dr.eng. Liviu Miclea – reviewer, Technical University of Cluj-Napoca
Chapter 1 – Introduction

1.1 Theme
Faults in design and manufacturing can be very expensive. Bad publicity and repairing or replacing the product may cost the manufacturer more than testing it before shipping.

Today, complex products are not only extensively tested, but their testability is taken into account already in the design phase.

In distributed systems, where many independent parts of the system cooperate for a common goal, design for testability is more difficult.

This work presents some aspects of design for testability of distributed systems, with the problems and their solutions developed by the doctoral candidate.

1.2 Objectives
The main objective of this thesis was to contribute to the field of testing and testability of distributed systems. These contributions materialized in four major research directions: analysis and case studies on testing methods and design for testability; remote management methods for the testing process; theory, simulation and real-world use of software agents; design and implementation of hardware agents.

1.3 Structure
The thesis contains seven chapters. The first part, approximately 25% - chapter 2 and, partially, chapter 3 – presents the knowledge necessary for understanding the contributions. The second part, about 75%, encompassing the rest of the 3rd chapter and chapters 4, 5, 6 and 7, presents the author’s contributions and conclusions.
Chapter 2 – Design for Testability and Distributed Systems

With the increased complexity of today’s systems, their testability must be tackled in parallel with other design aspects. Thus, they need to be designed also for efficient testability.

Distributed systems comprise subsystems that can function also independently, but collaborate to achieve a common goal. Testing these systems rises various problems, like the heterogeneous nature of their components or the intense communication between the subsystems.

Chapter 3 – Study on the Analysis and Implementation of Testable Structures

Traditionally, circuits are tested with automated test equipments (ATE), which are external. However, these are very expensive and the test’s speed is slower than nominal.

An important milestone of design for test was the introduction of Boundary Scan (a.k.a. JTAG, IEEE 1149.1). This offered better internal and external test for the components and the systems, with minimal overhead.

An advanced concept is BIST (Built-In Self-Test), which introduces the testing circuitry right into the main circuit.

Chapter 4 – Contributions to the Design for Remote Test

4.1 The Experimental BIST Module

The experimental BIST module is a software module, written in Visual Basic. It accepts various test commands through TCP/IP sockets, on port 2811. It also sends back the results in ASCII form. The “end” character for the messages is ‘.’.

4.2 Developing the CORBA Application

CORBA (“Common Object Request Broker Architecture”) is an inter-platform RPC (“Remote Procedure Call”) platform.

The application developed by the author contains the “BistSlave”, “BistServant”, “BistMaster” classes, written in Java. It connects to the aforementioned BIST module.

![Figure 1 – The BistMaster GUI, after connecting to BistSlave.](image-url)
4.3 The web Application

The idea is that the user is able to access the BIST module from wherever on the Internet, start the desired test and receive the test results, using only a simple browser, e.g. Internet Explorer.

The following code connects to the BIST module, sends the name of the desired test, waits for the results and creates the web page containing them.

```
$fp = fsockopen ($addr, 2811);
if (!$fp)
    {
        echo "$errstr ($errno)<br>\n";
    }
else
    {
        echo "done."
        echo "<br>Sending test command "$test"...";
        fputs ($fp, strtoupper($test).".");
        echo "done."
        echo "<br>Reading response from BIST module...";
        $recvd = fgets ($fp,128);
        // read until "." (end of result) is met (using regular expression)
        while (!preg_match("/(.+)/",$recvd))
            {
                $recvd = $recvd . fgets ($fp,128); // "." is concatenation
            }
        echo "$recvd"
    }
```

4.4 The Mobile Version

The mobile version of the above application is based on the WAP ("Wireless Application Protocol") and WML (Wireless Markup Language) standards.

The main problems were the limited resources of the mobile devices and of the above mentioned standards.

Chapter 5 – Contributions to the Design for Distributed Testing with Agents

5.1 Tester/Repairer Agent Society Simulator

An agent is any entity that perceives the environment through sensors and alters the environment through some effectors.

BIST is good for testing, but for heterogeneous distributed systems, Distributed BIST (DBIST) is more appropriate. A good way to achieve efficient DBIST is with agents.

ADK (“Agent Development Kit”) was used to develop the software agents and the simulator, due to its widespread use.

5.2 Microagents in Testing Mobile Phones

The platform of choice here was JADE-LEAP, due to the ability of the microagents to communicate with their bigger brothers.
5.3 Study on Test and Periodic Repair with Embedded Processors, using Software Agents

The study analyzes and experiments the development of an agent system able to test, and, if possible, repair the embedded processors and connected peripherals in a distributed system, using software agents.

5.4 Study and Experiments Regarding Mobile Hardware Agents

The notion of Mobile Hardware Agents is based on two paradigms: Mobile Data Agents and Reconfigurable Systems, mostly FPGAs (“Field Programmable Gate Array”).

The equipment:

- ARM Integrator Board with ARM7TDMI and Xilinx Virtex2000E FPGA (peripheral node);
- uCLinux real-time operating system, on the peripheral node;
- Windows-XP, Java 1.5.0, on the Agent Host.

Chapter 6 – Application in the Remote Control and Monitoring System of a Hydroelectric Power Plant Chain

The main problems here were the huge amount of data and the heterogeneous nature of the system.
Chapter 7 – Contributions

1) The implementation of a demonstrative application for remote testing, based on CORBA technology. Results published in [21, 27, 28].

2) The analysis and development of a solution with web applications for test remote management. Results in [8, 9, 12, 19, 20, 25, 26, 28, 30].

3) The development of the management interface for the mobile phone. Results in [8, 9, 18, 25].

4) The implementation of a distributed test simulator, with software agents. Results in [1, 7, 10, 11, 15, 16, 17, 22, 31, 32].

5) The development and experiments with a mobile phone testing method, using software microagents. Results in [5, 6, 29].

6) The analysis and introduction of a periodic test and repair method, with embedded processors and intelligent agents. Results in [1, 14].

7) The introduction and experiments on a new test and repair solution, with mobile hardware agents. Results in [1, 4].

8) The implementation of an experimental monitoring and control system, with software agents, for a hydroelectric power plant chain. Results in [2, 3, 13, 23, 24].

Referred publications


International Test Conference 2005, Austin, TX, USA, November 8 - 10, 2005, accepted paper.


