Improve Fault Handling in Radio Base Stations by applying Self-test and Visualization

Master’s Degree Project
Stockholm, Sweden January 2013

XR-EE-LCN 2013:008
Master of Science Thesis

Improve Fault Handling in Radio Base Stations
by applying
Self-test and Visualization

A Case Study at Ericsson Radio Software Department Kista-Stockholm

Atiyeh Goudarzi

Company Supervisor: Johnny Lindqvist, James Centerstam
Examiner: Professor Panos Papadimitratos

KTH School of Information and Communication Technology
The Royal Institute of Technology, Stockholm, Sweden
Preface

This thesis work was conducted in the premises of Ericsson in Kista. The greatest part of the report contains sensitive information that is Ericsson’s intellectual property. As such and to avoid any infringement, the large part of the report is kept confidential.

The full report was presented to the KTH examiner (Prof. P. Papadimitratos) and it was evaluated jointly with the responsible supervisors (J. Lindqvist and J. Centerstam) from the side of Ericsson.

The publically accessible part of the thesis herein includes the abstract, the introduction and the conclusion.

Johnny Lindqvist
johnny.lindqvist@ericsson.com

Panagiotis Papadimitratos
papadim@kth.se

James Centerstam
James.Centerstam@ericsson.com
Abstract

Trouble Reports (TRs) encompass the details of a fault-ended test scenario during Radio Base Stations (RBS) verification phase. TR handling is a daily routine for designers in Ericsson Radio Software Department which has been reversed into a bottleneck. TR handling is a log-centric procedure and getting right information from log files is not easily applicable due to various reasons.

In this research, we went first to pinpoint those barriers which put difficulty in TR handling procedure. This phase accomplished with comments from TR handlers combined with standard principles for fault handling. A list of hindrances is recognized by the end of this part of project which introduced the main area we should work for performing improvements in TR handling.

In second phase of project, we concentrated on cryptic data as a major barrier in log analysis and consequently in TR handling. Cryptic data refers to hardware state during various intervals which pass useful information about hardware in different settings. Troubleshooters can not easily receive the message from this kind of data due to its low level essence.

A solution is proposed to handle this difficulty by helping users with the final message which is passed through cryptic data. We get test settings from log file and try to make an expectation on hardware state during test routine, then we make a comparison between what is expected and what is observed in log file. Any inconvenience between observed value and expected one might indicate a deficiency in relevant blocks.

Finding out the expected value in each step is done through two different solutions. At first solution, we dig into calculation details in each block and try to perform all steps in parallel with application but in a safe-mode to make sure that nothing might affect the results. For second solution, we do not engage with calculation complexity. Some test cases are designed to target selected blocks in application and watch out the behavior in any possible configuration. We have a database from all possible states which have impact on a block’s result based on that block’s characteristics. Functional test is run on each Radio separately and presented information in a TR should be compared with recorded data for that specific Radio.

These solutions are implemented as a web application which receives Trace and Error log as the source for catching test settings. Then correctness of presented data might be confirmed by recalculation or reading from results of functional test.
Acknowledgment

This thesis work is conducted in Ericsson Radio Software design department, Stockholm, Sweden.

My sincerest gratitude goes to my manager and supervisor, Johnny Lindqvist who provided me this opportunity, without whose help, this research would never have come to be.

I also offer my deepest gratitude to my technical supervisor, James Centerstam who patiently guided me in whole procedure with his invaluable assistance and abundant help.

Special thank also goes to the members of the Radio Software Trx-Ctrl team who were always supportive and helpful.

During this project I have collaborated with many colleagues for whom I have great regard, and wish to warmly thank the whole Radio SW department who guided me through this project and provided me with feedback on each step.

Moreover, I am indebted beyond measure to Peter Eriksson for all his kindness and invaluable support.

I’d also like to convey my thanks to professor Panos Papadimitratos for accepting my thesis proposal and supervising this research.

Finally, I express the most wholehearted gratitude to my parents for their love, support and encouragement during my entire academic carrier.

Atiyeh Goudarzi
Stockholm, January 2013
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCI</td>
<td>Board Control Interface</td>
</tr>
<tr>
<td>CPP</td>
<td>Connectivity Packet Platform</td>
</tr>
<tr>
<td>DL</td>
<td>Down Link</td>
</tr>
<tr>
<td>DS</td>
<td>Design Specification</td>
</tr>
<tr>
<td>HALI</td>
<td>Hardware Abstraction Layer Interface</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>Operation and Maintenance</td>
</tr>
<tr>
<td>RBS</td>
<td>Radio Base Station</td>
</tr>
<tr>
<td>RCI</td>
<td>Read Controller Interface</td>
</tr>
<tr>
<td>RU</td>
<td>Radio Unit</td>
</tr>
<tr>
<td>T&amp;E</td>
<td>Trace and Error</td>
</tr>
<tr>
<td>TR</td>
<td>Trouble Report</td>
</tr>
<tr>
<td>UL</td>
<td>Up Link</td>
</tr>
<tr>
<td>WARP</td>
<td>WCDMA ASIC Radio Platform</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Application testing is a process of technical exploration with the intention to reveal system's executive efficiency with respect to the context in which it is intended to operate. The ultimate goal in software testing is to identify errors, evaluate the environment's capacity and assure that system meets user expectations. According to Gartner, “Testing is an integral part of aligning IT and business, and it is essential in delivering a quality product to end users.” [6]

Experience indicates that no system/product can be deemed flawless, even best written applications are potentially vulnerable. This calls for a strong structure to handle application testing in any production.

Ericsson supplies various testing methods in application's lifecycle, these tests continue even after application releases. Several verification departments hold the responsibility to test various aspects of radio application releases on different hardware by applying pre-defined test cases which are designed based on the required expectations from the application. Trouble Report (TR) is an explanation on what happened during a test scenario which ended in a fault. A TR is supported with different kinds of log files, it usually entails the software version, hardware type and Data Collection Guideline (DCG).

TRs might be reported by customers as well, customer reported TRs gain the highest priority in TR handling queue and the handling procedure is almost similar to internal TRs, however it is different in some steps which are negligible in our case. TR handling is mainly done by SW designers in Radio, some of them are 100% dedicated to TR handling and some of them have this as one task in their weekly backlogs.

Due to the large number of TRs in Ericsson Radio SW department which is caused by various application releases and also high expectations from the product, TR handling is a permanent task and a notable number of specialists in system are involved in this procedure. TRs in Radio SW department are assigned to different teams based on the fault area and designers are responsible for troubleshooting. A number of barriers in this system, has reversed TR handling procedure into a bottleneck in production cycle, the procedure is just wasting of time in some cases without any improvement in fault pinpointing due to system's complexity, deficiencies in handling procedure
Radio application is a synthesis of various functional blocks which are broadly deployed. In addition to complexity that is inherent in the application due to various resources which ends in a misunderstanding situation, troubleshooters are often hamstrung by lack of applicable tools that can assist in understanding the sources’ behavior in practice.

1.1 Problem Definition

This project is aimed to study current TR handling procedure in Ericsson Radio SW department to investigate deficiencies the system suffer from and suggest solutions to improve fault handling procedure in terms of efficiency and time.

The major source of information in TRs is recorded log files during test routines. In many cases, figuring out what to search for in log files is difficult, since there is no transparent mapping between log messages and observed symptoms. Besides, SW designers in Radio department are divided into various development teams based on software architecture[4] and each team is more concentrated on a specific part of application, but while handling a TR, troubleshooters also have to understand the application’s behavior in general.

According to a survey [Appendix A] done as a part of this project in the company, 45% of troubleshooters believe that they can not get sufficient information from log files to handle TRs at first stage, this might indicate a deficiency in data collection, data presentation1 or even analysts’ competence. Log files are not provided with sufficient data in advance and this calls for redoing test in many occasions. Another problem is cryptic data represented in log files which are not easy to be interpreted by troubleshooters and they need to transfer it into a human readable message.

Apart from difficulties in analysis, time is a vital issue in today’s market. This announces the need for more efficient and faster methods in log analysis and consequently in TR handling.

With this background, this project is aimed to reach some solutions to enhance TR handling procedure in Radio SW department. First step is depicting a clear picture of current state, its strong and weak points and then decide if any practical movement might be applied to enrich the procedure from troubleshooting viewpoint.

Regarding the log-centric behavior of TR handling procedure, improvements in this area is tightly coupled with log analysis enhancements. In other words, we need to explore solutions to fortify log analysis in order to meet this approach.

1 These concepts are explained in detail in chapter 2
1.2 Method

This research is started with getting familiar with the Radio application, its characteristics and behavior. This is continued by conducting an observation on TR handling procedure from the time a TR is registered in system until the last step which is provided by a technical answer for the reported problem. Since we were responsible to evaluate whole the procedure, we started by joining Trace Classification Group (TCG) daily meetings, who conduct TR pre-analysis and then decide about the TRs’ priority in handling queue and also assign TRs to different SW teams based on reported fault. Then we followed the final handling procedure which is directed by SW designers. The major troublesome area is witnessed at this step, troubleshooters face a wide range of difficulties from lack of sufficient data in log files to huge plain text files with bunch of cryptic data.

Application’s complexity causes various kinds of barriers in TR handling procedure, some of them are pervasive in TRs in general and some are more significant in TRs written on specific parts of application which are categorized in lower priority for our project.

In order to make any improvement in this system we needed to narrow down the problem into separated hurdles and take steps on enhancements. This decision is difficult since we face large number of alternatives and need to prioritize them.

We designed a questionnaire to evaluate the system from SW designers’ viewpoint, the major audience for this project, and decide about the specific field we can chose as the first step in TR handling improvement. Analyzing survey results ended up with selecting cryptic data, as one of the system’s major barriers, to conduct more investigation into. Meeting this approach involves studying various documents on lower level functionalities in Radio application and understanding the characteristics of HW interface since we are working in closest layer to HW and designers need to map the code to its peer HW device. However, documents are not easy to understand in some cases specially for people who are not much familiar with the system and system recognition in this research is mainly done through some workshops with specialists in company both in Radio SW and Radio HW departments. Our ultimate solution for this barrier uses self-testing methods which is gained through two major ways in this project; re-calculation and automated functional test.

In re-calculation, we try to implement whole calculation procedure considering all relevant rules and exceptional areas in parallel with application to make sure that no other process might affect the expected result from a specific function.

Second solution is free from re-calculation difficulties and relies on some functional test cases which are designed with focus on those variables that affect a function’s behavior. This method is implemented by means of an internal Ericsson’s test framework named Mira and Ruby language. As a proof of concept, both solutions are tested for a functional block inside the application which represents cryptic data.

Another challenge in this system refers to the implementation of the proposed solution, in such a big company, it is not easy to decide about extra tools or applications, users might work on
different operating systems and editors, we should be careful not to add more difficulties for
them in some way in order to improve the procedure in another way.

After careful studies and observing current tools in company, we decided to show the
applicability of the solution in term of a web application. The application is written in Java,
using Spring framework.

1.3 Contribution

TR handling procedure in Ericsson is done through a web integrated system which is designed
based on support and maintenance purposes, but log file analysis is not assisted with any extra
tool and it is done all by human intelligence.

We know that a good analyst can not be replaced with any tool but customized tools might be
helpful in terms of time and resource saving in analysis procedure. The need for a tool helping
users in Trace and Error (T&E) log analysis seems vital for Radio SW department. However, the
company can not make benefit from commercial log analyzer tools for Radio application and
analysis might be done just by knowing application’s characteristics in each and every single
step which is unique by itself.

We propose a method which is based on self-test principles to validate data received in each
block during signal processing. The idea is new since reverse engineering in measurement
functions is not easy and troubleshooters normally use a kind of pattern from non faultly tests to
validate data presented in faulty test scenarios.

In order to tackle the barriers in handling procedure, we concentrated on functional blocks and
followed calculation details which serve to configure these functional blocks. We supported the
idea of self-testing even in existence of complex hardware dependencies. Designing an
automated functional test just by means of internal resources, shall provide a decent approach
towards enhanced methods for log analysis.

Both solutions are implemented as a web application and as we aimed from beginning, user does
not need to know the details behind signal processing measurement details. Moreover, this
application is a process-aware information system which can provide process filtration in log
files which is complicated by itself since each process might be called inside several other
processes and following a specific event means you have bunch of data from different processes.
1.4 Thesis Outline

Chapter 1 conducts an introduction on TR handling procedure in Ericsson and the importance of this research for the company, a summary on method and contribution is also provided in this section

Chapter 2 concludes a background on the research area and also relevant literature studies

Chapter 3 dives into the specific area of research, defines project scope in detail and provides a quick introduction on proposed solution

Chapter 4 Explains all details in applying re-calculation to perform self-test on T&E log files

Chapter 5 Supplies self-testing by means of results from automated functional tests and also implementation details on designed test cases

Chapter 6 provides the implementation details on web application that performs data validation in TE log files

Chapter 7 introduces the conclusion and also suggestions for future works

Appendix A presents a survey done in Radio SW department to evaluate current TR handling procedure.
This part is intentionally removed due to confidentiality
Chapter 7

Conclusion and Future Works

The purpose of this thesis was to evaluate current state of TR handling procedure in Ericsson Radio SW department and pinpoint vulnerabilities and suggest solutions for improvement. Application’s massive volume and complicated troubleshooting procedure make a hindrance in depicting a clear picture from system, its strong and weak points.

We concentrated first in providing a technical definition for system’s known deficiencies and categorized current characteristics based on standard requirements in fault handling procedures. This evaluation might help company to decide which areas they should emphasize for future enhancements in TR handling procedure.

Later, one of the identified troublesome areas from previous stage, selected for practical improvement. This phase shall be named as a proof of concept for what we proposed in first phase. During experimental stage, we concentrated on providing an interpretation for cryptic data represented in log files. Two major solutions proposed to handle self-test;

- **Re-calculation**: Which considers the idea of comparison between what happened during test procedure with what is expected in a safe-mode with the same configurations. This solution is supported by handling all exceptions in PA0 calculation to cover all possible states.

- **Automated Test Results**: This solution seems to be easier to implement in large scales, Mira framework provides many options in carrier configuration which are vital in PA0 behavior.

Both methods have advantages and disadvantages, the problem with automated test lies in its dependence to hardware and the need to run test cases on each radio separately which takes long time (approximately 8 hours just running the test cases on each radio and gathering data, in case we do not make any interpolation and test every possible state). On the other hand, re-calculation is free from hardware in principle but working with various measurement blocks show that providing baseline data for re-calculation is not always possible. Re-calculation might be trustworthy in case that input data is intact, we can not rely on data processed somewhere else, but in some blocks we need to get data from other functions without having the permission to check its resource and this might cause in unexpected Sid effects.
In our case, both methods work well but extension is necessary if we want to make it useful in whole procedure. Deep studies on each block will show which method is more efficient based on system requirements.

**Future Works**

Proposed solutions in re-calculation and automated test are both proved in as possible methods for performing self-test in DL, however it would be useful in case we implement them for all functional blocks inside DL. Primary studies show that PA0 has the most complicated calculation in compare with other blocks inside DL, so extending the solution would be easier for other blocks.

1. Deep studies in all measurement blocks inside DL to find out which solution (re-calculation and automated test) is more efficient for self-test improvements.

2. Decide on one solution and cover all blocks for self-test with that method.

About implementation, we need deeper studies, web application is selected for this project to meet to approaches; first of all we needed a context to visualize the idea more like a prototype and then we suggested this format to support integrity with another tool for log analysis, however this tool is not that much relevant to Radio SW but applicable in some cases. But some designers believe it would be better to have this solution implemented in command line interface since most of users work in that environment and this might be easier for them not to change context while handling a TR.

Apart from cryptic data which is our main approach in second phase of project, aforementioned deficiencies in data collection should be analyzed in detail, company is paying huge cost for re-doing tests just because of insufficient trace activation, and this might be improved by recognizing test case context and map available traces prior to running tests.
Bibliography

[1] SAL, 4/102 62-CN 160 6510 Uen (Ericsson Internal)

[2] TX Module, 13/0363-FCP 105 361 Uen (Ericsson Internal)


[5] DS TX Carrier Power , 58/102 62-CN 160 6510 Uen (Ericsson Internal)


[7] DS for RU SW(for WARP), 1/102 62-CAH 161 1490/1 Uen (Ericsson Internal)


[12] Trace Systemization in Radio DS, 10/159 41-FCP 105 361 Uen (Ericsson Internal)


[14] Event Trace DS 84/102 62-CN 160 6510 Uen (Ericsson Internal)


[18] 25/102 60-HRB 105 102/1 Uen (Ericsson Internal)


[20] TX Carrier Power, 58/102 62-CNH 160 6510 Uen (Ericsson Internal)