SPECIFICATION AND DEVELOPMENT OF A MODULE FOR CORRELATION OF UMTS TRACE FROM DIFFERENT SOURCES: SIEMENS NETWORK ELEMENTS AND TEST MOBILES
"Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles" in Siemens, S. A. – COM RD1 NMP

Curricular Internship Report of LEIC 2004/2005

Luís Freire Duarte

Supervisor from FEUP: Prof. Manuel Ricardo
Supervisor from Siemens, S.A.: Eng. Inês Santos

August de 2005
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

To my family, girl-friend and friends
Resumo

Este relatório descreve o estágio realizado no contexto do Estágio Curricular da Licenciatura em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto. O estágio teve lugar na Siemens, S.A. COM RD1 NMP 2 e foi designado de “Especificação e desenvolvimento de módulo de correlação de dados de UMTS trace de diferentes fontes: elementos de rede Siemens e Test Mobiles”. O objectivo principal foi desenvolver o sistema Data Correlation para correlacionar informação de UMTS IMSI trace de diferentes fontes, numa análise única. Tal sistema seria desenvolvido para ser futuramente integrado no modulo UCAT da ferramenta RNA da Siemens, embora a integração não fosse uma parte integrante do estágio.

O relatório descreve o problema Data Correlation de sincronizar e correlacionar mensagens de diferentes fontes, que se encontram desordenadas e dessincronizadas. Descreve cada consideração que deve ser tomada enquanto se monta uma análise correlacionada. Sendo um conceito original não existe nenhum estado da arte porém, as tecnologias relacionadas foram estudadas e descritas com algum detalhe. Uma introdução à arquitectura do sistema WCDMA está presente, assim como uma descrição das tecnologias Test Mobile utilizadas no projecto.

A especificação do sistema Data Correlation é descrita, os seus requisitos são especificados em detalhe e a arquitectura do sistema é descrita através de estruturas e vistas arquitecturais. Depois da especificação do sistema, a sua implementação é descrita através da imagens e exemplos de como o mecanismo de Sincronização e Correlação funciona.

Os resultados do estágio são apresentados e o sistema é considerado pronto para ser integrado no UCAT. O primeiro contacto com os clientes foi positivo, visto que estes se encontravam muito interessados em obter do sistema Data Correlation, devido à sua utilidade e originalidade. O sistema realizou todos os objectivos iniciais e todos os requisitos foram respeitados. Do mesmo modo, uma lista de trabalhos futuros é apresentada, descrevendo funcionalidades extras e interessantes, que permitiriam melhorar o sistema de Data Correlation.

Para finalizar, a conclusão resume os resultados mais importantes e alguns dos problemas encontrados, além de apresentar algumas considerações pessoais sobre o estágio.
Abstract

This report describes the internship made in the context of the curricular internship of the “Licenciatura em Engenharia Informática e Computação” of the “Faculdade de Engenharia da Universidade do Porto”. The internship that took place in the Siemens, S.A. COM RD1 NMP 2 and was designated “Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles”. The internship’s primary objective was to develop the Data Correlation system for correlating UMTS IMSI trace information from different sources into a seamless analysis. Such system would be developed to be integrated in the UCAT feature of the Siemens RNA tool, although the integration was not part of the internship.

The report describes the Data Correlation problem of synchronizing and correlation messages from different sources, which are unsorted and desynchronised. It reports every consideration that must be taken while assembling the seamless correlated analysis. Being an original concept, there wasn’t any previous state of the art, but the related technologies were studied and described with some detail. An introduction to the WCDMA architecture is present, as well as a description of the Test Mobile technologies used in this project.

The Data Correlation system specification is described, its requirements specified in detail, and the system’s architecture described by using architectural structures and views. After the system’s specification been implemented, the system is described by the use of print screens and examples on how the Synchronization and Correlation mechanism works.

The internship’s results are reported and the system is considered ready to be integrated in the UCAT feature. Initial feedback from customers was positive, seeing that they were eager to obtain the Data Correlation system, due to its utility and originality. The system accomplished all the initial objectives and all the specified requirements were respected. Also, a list of things to do in the near future is presented, describing interesting extra functionalities that would improve the Data Correlation system.

Finally, the conclusion resumes the major achievements and some of the problems faced, as well as some personal considerations about the internship.
Acknowledgments

Now that I look back and witness the long path I made since I first came to Siemens, I cannot go without thanking some very special people for their help, support and friendship. Without you, I wouldn’t have come this far.

First I would like to express my thanks to a team that would make anyone proud of. The RNA team gathers some of the most qualified and experienced professionals that I have ever known. I can only say that I’m greatly thankful for everything that you taught me. My special thanks to Inês Santos; your advice and support guided me throughout the whole internship. My thanks to Verena Niepoort, our Project Leader; before meeting you I never dreamed there could be such a friendly, cool and capable boss. My thanks to Nuno Sousa and José Sá for every advice and help you gave me. To the rest of the team, Nuno Guedes, Hugo Magalhães, Dario Freire, Filipa Gomes, Eduardo Lourenço and Pedro Sobral, my thanks for making this an enjoyable experience. You are truly great people.

Second I would like to express my thanks to my teachers at FEUP. I learned a lot from you and you have my greatest respect. Special thanks to Professor Manuel Ricardo for being my coordinator and for your advice on my work and on how to write this report. To the rest I’m greatly thankful for your patience and counsel. I cannot thank you all for everything you taught me, but I would like to give special recognition to some teachers. I apologize to the ones not mentioned but it would be impossible to remember the names of every teacher I had. I would like to express thanks to Professor Raul Fernando de Almeida Moreira Vidal, our director and teacher; you always defended the interests of the students and now, more than ever, I value your advice. I thank Professor João António Correia Lopes for your humorous classes. I thank Professor Ademar Manuel Teixeira de Aguiar for your Software Architecture class; it showed me a new way to see Software. I also would like to thank every teacher from FEUP, who helped me during the five years I spent there. I learned almost everything I know of software engineering from you, and for that I’m very thankful.

Third I would like to express my thanks to all the people that make my life special. I thank Tânia Marisa Maia Campos for her love, friendship and company; you make every moment of my life worth living. My thanks to my best friend Filipa da Rocha Neves Couto Maia; you have been there for nearly 18 years, truly a lifetime of friendship. To my good friends from LEIC Ricardo Almeida, Ricardo Castro, Hugo Monteiro, Claudio Silva, Guilherme Silva, João Ferreira, Helder Coelho, Ivo Relvas, Helder Ferreira, José Ferreira, Tiago Lázaro, Helder Magalhães and Sérgio Rasmado my very very special thanks for your friendship. I thank my friends from everywhere I’ve been: Diana Pinho, Inês Domingues, Maria Inês Ferraz, Filipa Gonçalves, Marisa Laje, and Sara Monteiro.

Forth but most important, I thank my fathers Laurentino da Silva Duarte and Maria da Conceição Jesus Rocha Freire Duarte for being the greatest fathers that a son could ever hoped for. I thank my brother, Alexandre Freire Duarte for... well for not stressing me to much. I thank my grandfathers for all my memories.

Last I would like to thanks PRODEP for funding this internship.
# Table of Contents

Table of Figures ......................................................................................... viii

Table of Tables ......................................................................................... x

1 Introduction ............................................................................................... 1
   1.1 Siemens AG ....................................................................................... 1
      1.1.1 Siemens AG’s History ................................................................. 1
      1.1.2 Siemens, S.A. Portugal .............................................................. 2
   1.2 The Radio Network Analysis project .................................................. 2
      1.2.1 The UCAT feature ................................................................... 3
   1.3 The UMTS Data Correlation Prototype ............................................. 4
      1.3.1 The Data Correlation concept ................................................. 4
      1.3.2 Development of the Data Correlation prototype ...................... 5
      1.3.3 Most important Results ............................................................ 5
   1.4 Report’s structure and covered subjects .............................................. 5
      1.4.1 Introduction .............................................................................. 5
      1.4.2 Problem Analysis .................................................................... 6
      1.4.3 Technological Review .............................................................. 6
      1.4.4 System Specification ............................................................... 6
      1.4.5 Development .......................................................................... 6
      1.4.6 Results and future work ........................................................... 6
      1.4.7 Conclusions ............................................................................ 6
      1.4.8 References and Biography ....................................................... 6

2 Problem Analysis ....................................................................................... 7
   2.1 Problem Overview ............................................................................ 7
      2.1.1 Radio Network Optimisation ................................................... 7
      2.1.2 IMSI Call Trace ....................................................................... 7
   2.2 Objectives ......................................................................................... 8
      2.2.1 Project scheduling ................................................................... 9

3 Technological Review .............................................................................. 12
   3.1 UMTS General Concepts ................................................................ 12
      3.1.1 Historical Introduction - From 1G to UMTS Phase 1 ................ 12
      3.1.2 UMTS architecture ................................................................. 13
      3.1.3 Application protocols ............................................................. 15
   3.2 Test Mobile Technology .................................................................. 16
      3.2.1 TEMS Investigation ............................................................... 16
      3.2.2 Agilent .................................................................................. 17
   3.3 Tracing Messages ............................................................................ 18
   3.4 State of the Art Technology .............................................................. 20
   3.5 Project’s technological choices ......................................................... 20
      3.5.1 Java ..................................................................................... 20
      3.5.2 JUnit .................................................................................... 22
      3.5.3 ANT ..................................................................................... 23
      3.5.4 UML .................................................................................... 23
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

3.5.5 Eclipse .................................................................................................................23

4 System Specification..................................................................................................25
  4.1 System’s requirements .........................................................................................25
    4.1.1 Functional requirements .................................................................................25
    4.1.2 Non-functional requirements .........................................................................34
  4.2 Architectural documentation ...............................................................................35
    4.2.1 Architectural overview ...................................................................................35
    4.2.2 Modular decomposition ..................................................................................36

5 Development .............................................................................................................45
  5.1 Development environment ..................................................................................45
  5.2 Development report .............................................................................................45
  5.3 User Interfaces .....................................................................................................47
    5.3.1 Data Correlation Binary File & Analysis Area Window ..................................47
  5.4 The Trace Correlation Mechanism ......................................................................55
    5.4.1 Example based on real IMSI trace files .........................................................60

6 Results and future work ............................................................................................66
  6.1 Data Correlation system results .........................................................................66
  6.2 Future Development .............................................................................................67

7 Conclusion ................................................................................................................68

8 Biography and References .........................................................................................70

9 Acronyms ..................................................................................................................71
## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RNA Architecture</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>RNA Information flow</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Project’s Gantt Diagram</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>UMTS Evolutionary Diagram</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>UMTS Architecture</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Drive Test equipment in vehicle</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Walk Test equipment in backpack</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>GUI Interface prototype for the File Area</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Trace File Management Use Case Diagram</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>GUI Interface prototype for the Analysis Area</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>Analyses Management Use Case Diagram</td>
<td>31</td>
</tr>
<tr>
<td>12</td>
<td>GUI Interface Prototype for the Analysis Visualization Window</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Analysis Visualization Use Case Diagram</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>Data Correlation System High-Level diagram</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>File and Analysis Area Module’s MVC Diagram</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>The Model Component class diagram</td>
<td>39</td>
</tr>
<tr>
<td>17</td>
<td>The Controller Component Thread diagram</td>
<td>40</td>
</tr>
<tr>
<td>18</td>
<td>The Controller Component class diagram</td>
<td>40</td>
</tr>
<tr>
<td>19</td>
<td>The View Component class diagram</td>
<td>41</td>
</tr>
<tr>
<td>20</td>
<td>Analysis Visualization Window module class diagram</td>
<td>42</td>
</tr>
<tr>
<td>21</td>
<td>Synchronization and Correlation Engine class Diagram</td>
<td>42</td>
</tr>
<tr>
<td>22</td>
<td>Decoding Functionalities class diagram</td>
<td>43</td>
</tr>
<tr>
<td>23</td>
<td>File and Analysis Main Window GUI</td>
<td>47</td>
</tr>
<tr>
<td>24</td>
<td>File Menu GUI</td>
<td>48</td>
</tr>
<tr>
<td>25</td>
<td>Toolbar GUI</td>
<td>48</td>
</tr>
<tr>
<td>26</td>
<td>Action Bar GUI</td>
<td>48</td>
</tr>
<tr>
<td>27</td>
<td>File Format Selection GUI</td>
<td>49</td>
</tr>
<tr>
<td>28</td>
<td>File Table and respective Details Panel GUI</td>
<td>49</td>
</tr>
<tr>
<td>29</td>
<td>Analysis Table and respective Details Panel</td>
<td>51</td>
</tr>
<tr>
<td>30</td>
<td>Test Mobile Input Window</td>
<td>52</td>
</tr>
<tr>
<td>31</td>
<td>Invalid IMSI warning message</td>
<td>52</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>32</td>
<td>Calendar for choosing the Date</td>
<td>52</td>
</tr>
<tr>
<td>33</td>
<td>Correlation Window GUI</td>
<td>53</td>
</tr>
<tr>
<td>34</td>
<td>Analysis Visualization Window GUI</td>
<td>54</td>
</tr>
<tr>
<td>35</td>
<td>Message Details displaying a tree-like structure</td>
<td>55</td>
</tr>
<tr>
<td>36</td>
<td>Message Details displaying a table-like structure</td>
<td>55</td>
</tr>
<tr>
<td>37</td>
<td>Correlation’s Step Evolution Diagram</td>
<td>56</td>
</tr>
<tr>
<td>38</td>
<td>Sequential Messages</td>
<td>57</td>
</tr>
<tr>
<td>39</td>
<td>Non-sequential Messages diagram</td>
<td>57</td>
</tr>
<tr>
<td>40</td>
<td>Anchor Generation Activity Diagram</td>
<td>57</td>
</tr>
<tr>
<td>41</td>
<td>Sequential sets and sub-groups relations diagram</td>
<td>58</td>
</tr>
<tr>
<td>42</td>
<td>Sub-group 1 synchronization</td>
<td>63</td>
</tr>
<tr>
<td>43</td>
<td>Sub-group 2 synchronization</td>
<td>63</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1: Tasks Descriptions ................................................................. 9
Table 2: Radio Network's IMSI trace workflow .................................... 18
Table 3: Advantages and Disadvantages of Test Mobile Trace vs. Radio Network Trace ...... 19
Table 4: Necessary file details list .......................................................... 27
Table 5: Trace file management use cases ............................................. 28
Table 6: Necessary analysis details list .................................................. 30
Table 7: Analyses management use cases .............................................. 31
Table 8: Required Message List Column names .................................... 32
Table 9: Analysis Visualization use cases .............................................. 33
Table 10: TEMS Investigation required version support ......................... 35
Table 11: Decoding Functionalities Elements and Components .................. 43
Table 12: Times before synchronization (example) .................................. 59
Table 13: Times after synchronization (example) ..................................... 59
Table 14: Siemens IMSI trace analysis messages contents ....................... 60
Table 15: TEMS Investigation IMSI trace analysis messages contents ........... 61
Table 16: Sequential Sets from example ................................................. 62
Table 17: Sub-groups from example ...................................................... 63
Table 18: Resulting correlated analysis .................................................. 63
1 Introduction

1.1 Siemens AG

Siemens AG is a German company, one and a half centuries old, with headquarters in Munich. Nowadays it's a multinational company present in 190 countries with around 460,000 employees, from who 57,000 are dedicated to Research & Development.

1.1.1 Siemens AG’s History

"Formed in 1847, the company Telegraphenhaubanstalt von Siemens & Halske grew within the space of a few decades from a small precision-engineering workshop, producing mechanical warning bells for railways, wire insulation made of gutta-percha, and electrical telegraph systems, into one of the world's largest companies in electrical engineering and electronics. Landmark inventions, an immense readiness to innovate, and a strong international commitment have driven the company's success since its very beginnings. When in 1866 Werner Siemens (known as Werner von Siemens after 1888) discovered the dynamoelectric principle, the potential applications for electricity were limitless. Heavy-current engineering began to develop at a breathtaking pace, producing one triumphant innovation after another: In 1879, Siemens & Halske presented the first electric railway and installed the first electric streetlights in Berlin; in 1880 came the first electric elevator; and in 1881 the electric streetcar. Following the death of the company's founding father, Werner von Siemens, in 1892, his successors followed the course he had set, constantly advancing the company with trailblazing innovations. Lighting, medical engineering, wireless communication, and, in the 1920s, household appliances, were followed after World War II by components, data processing systems, automotive systems and semiconductors. The guiding principle that had applied since the company's beginnings - of concentrating solely on electrical engineering, "but on the whole of electrical engineering" - helped make Siemens the only company in its industry to operate in both light- and heavy-current electrical engineering, and by the mid-1920s it was again one of the world's five leading companies in its field. When the National Socialists seized power, Siemens, like the rest of German industry, was drawn into the system of the war economy. Through to 1944, higher state demand and military orders led to substantial increases both in sales and in the size of the workforce. In order to fill its manufacturing quotas, Siemens partly used forced labour. After World War II, Siemens began rebuilding in Germany first, but gradually moved into foreign countries from the 1950s on. Technological advances, expansion into new business segments, and the reestablishment of a presence in traditional export markets laid the foundations for the company's return to its old strength in the world marketplace in the 1960s. To give the company a stronger identity and consistent market presence, Siemens & Halske, Siemens-Schuckertwerke AG, and Siemens-Reiniger-Werke AG, the three main companies in the group, merged in 1966 to form Siemens AG. In 1969, the company's main business segments were assigned to six largely independent operating groups, creating an organizational structure that has been adapted on numerous occasions through to the present day. Today, Siemens is a transparent organization
comprising fast-acting business units that are making important and significant contributions
to the future of electrical engineering and electronics.\[1\]

1.1.2 Siemens, S.A. Portugal

Siemens has a long history in Portugal, having first invested here in 1905. Since then,
Siemens has grown to become the Portuguese leading company in electrical engineering and
electronics, employing about 5,100 people. These are divided among Siemens S.A., the
company's Portuguese headquarters; Indelma S.A., which manufactures electro-mechanical
automotive systems; Siemens-Nixdorf, the sales division for information technology; and
Osram, the sales office for lighting technology products.

The internship took place in the Siemens, S.A. Northern Region offices in Freixieiro (Porto –
Portugal), more specifically in the Siemens, S.A. – COM RD1 NMP 2. This is part of the OG
COM (Communications Operating Group). This operating group in the Portuguese national
market has its strategy oriented towards the leadership of the IP (Internet Protocol), Mobility
and e-business. With several initiatives, the OG COM achieved the leadership of several
domains of the IT and Telecommunications sectors, leading to the consolidation of the
Siemens Group in Portugal.

1.2 The Radio Network Analysis project

The Radio Network Analysis (RNA) is a troubleshooting application for managing
GSM/UMTS radio networks. Its targeted clients are the Telecom Operators, mainly Operation
& Maintenance Centres, Radio Optimisers and Network Planners. This project is exclusively
developed in Freixieiro (Porto – Portugal) since its first version was launched in 2000.

Developed entirely in Java and optimised for use in Windows and Solaris, RNA benefits from
the advantages inherent to the Java language, which allows is to be operating system
independent. It has a single implementation, yet it can be distributed in two distinct systems:
- A client-server system, in which the RNA server runs in a remote computer, while the
  user accesses the server using a conventional web browser.

![RNA Architecture](image)

**Figure 1: RNA Architecture**

[1] - from the Siemens Intranet
Security wasn’t forgotten in this RNA architecture, by being an application that runs through a browser, the RNA server resided in a Demilitarised Zone (DMZ), guaranteeing security, either from external access as from internal access from the operator’s radio network management network. With this architecture all of the RNA’s control is made remotely: from trace activation to the analysis of the processed results.

- Also there is a local version of RNA. The local version of the RNA is installed in a normal local computer, and then the user can load trace log files into the local installation for processing and analysis.

The information flow, when a trace is activated, follows the scheme defined in Figure 2: RNA Information flow. Remotely, trace requests are generated to a determined equipment or user. When the trace ends, the generated binary file is gathered by the RNA, saved on disk and then processed for latter analysis.

![RNA Information flow](image)

**Figure 2: RNA Information flow**

In technical terms, RNA architecture is Feature Oriented, providing a different way of organizing code. By allowing the programmer to compose layers of code, each layer representing a feature, the code complexity is reduced and existing components are reused. This result in an augmented capability to define, test, change and compose new features easily.

The RNA currently has a following Troubleshooting features:

- GSM IMSI and Cell Trace
- UMTS Call Trace
- GSM History of dropped calls
- GSM Smart Carrier Allocation
- UMTS Call Alarms
- GSM Traffic Localization

### 1.2.1 The UCAT feature

The UMTS Call Trace (UCAT) feature is part of the RNA tool, and it is used for analysing Siemens UMTS trace files. The trace files, that contain UMTS calls, are decoded into analyses. It is possible to analyse a single call, by performing a Call Analysis, or analyse several calls simultaneously by performing a Statistical analysis. When performing a Call Analysis, the user can make use of the UCAT’s analysis tools: Messages List, Messages Details, Message Flow, Coverage Analysis, Service Analysis and Capacity Analysis. The UCAT’s purpose is to complement Drive Test Evaluations and so help solving costumers’ complaints about trouble areas, and improve general indicators.

The Data Correlation system is planned to be integrated within UCAT in the near future, and so extend its base functionalities.
1.3 The UMTS Data Correlation Prototype

1.3.1 The Data Correlation concept

During a UMTS call, messages are continually being exchanged between the cell phone and the network, and also between the different network elements within the network. These messages can be recorded into trace files by using a procedure called IMSI call trace. When performing IMSI call trace, two possibilities are available:

- Trace the messages using the Network’s trace capability. When tracing messages using the Network, the recorded messages are the ones that the Network sent and received.
- Trace the messages using a mobile phone with trace capability. These mobile phones, called Test Mobiles, are equipped with special software that enables them to perform traces. When tracing messages using a Test Mobile, the recorded messages are the ones that the Test Mobile sent and received.

Both approaches have its advantages and disadvantages, and since the approaches aren’t mutually exclusive, it has become common the simultaneous use of both. This allows the network optimiser to use the best of what each approach has to offer, however this enhanced advantage doesn’t come without a price. The network optimiser has to use two different tools and try to detect errors and discrepancies in message lists that can thousands of messages long. The perfect solution would be a single tool capable of importing trace files from both approaches, and then automatically correlate the messages into a single seamless analysis. This not only would, not only, save considerable time but would also be much more reliable than performing the correlation by hand.

In order to achieve a seamless correlated analysis from two trace files, the following steps must be performed:

1. Import both traces and decode them into individual messages: each file is decoded into a list of messages ordered by time;
2. Identify matching messages: for each message in one of the message lists, check if it is contained in the other message list;
3. Synchronize the trace files using the found matches. Synchronization is necessary because there is a time difference between the Network’s and the Test Mobile’s internal clocks;
4. Create an analysis containing all messages from both message lists;
5. Transform the analysis into a seamless analysis by:
   a. Removing repeated messages;
   b. Identifying lost messages;
   c. Identifying missing messages.

The Siemens’ RNA is a tool that analyses Siemens Network’s trace files. It was requested to extend the tool should be extended in order to allow importing and correlating Test Mobile’s trace files with Siemens Network’s trace files. For that purpose this internship was created.
1.3.2 Development of the Data Correlation prototype

The internship, which lasted 6 months, was divided in four distinct phases. Each phase took the previously developed work into the following stage in a process of iterative development. New ideas were easily adopted and ineffective concepts were abandoned without risking the integrity of the entire project.

The first phase was the introductive stage of the internship. It was dedicated to the integration in the RNA team and Siemens’ internal processes. Also, it was a time for learning basic concepts on UMTS and Test Mobile technology. During this period the project’s requirements were discovered and refined, although they were continuously redefined as the project progressed.

The second phase was dedicated to the development of a decoder for Test Mobiles trace files. This was an extensive phase because no description was available on the Test Mobiles trace files formats. It was necessary to discover the internal structure of the files in order to develop a viable decoder. Since new formats are constantly being developed it was necessary that the developed decoder would support future version so it was design to be highly extendable.

The third phase was dedicated to the development of a correlation mechanism to generate seamless analysis with data from Siemens Network Elements and TEMS and Agilent trace files. A fully functional prototype was developed to test and validate all developed components.

The forth and final phase of the project was the writing of this report.

1.3.3 Most important Results

During the internship two major components were developed. The first component is the Test Mobile decoder module, capable of decoding TEMS Investigation and Agilent trace files. The second component is the Synchronization and Correlation module, capable of synchronizing and correlating Siemens traces with Test Mobile traces. Also, a prototype was developed that allows a user to test and make use of two developed modules. The prototype reuses several of UCAT’s components, modules, and its global architectural design. This will allow an easier integration of the Test Mobile decoder and the Synchronization and Correlation module in the UCAT feature.

1.4 Report’s structure and covered subjects

1.4.1 Introduction

The Introduction helps the reader understand the context in which this internship was made, as well as the internship’s purpose. It describes Siemens’ history from its early foundation in 1847 to present day, with special attention given to Siemens S.A. Portugal northern region offices in Freixieiro, where the internship took place. The RNA project, in which this internship was integrated, is succinctly presented, describing its normal use cases and basic architectural distribution. Furthermore the UMTS Data Correlation prototype is briefly presented, depicting the driving forces behind the project, its development and final conclusions. At the end of the chapter there is a brief description of the contents of each chapter of the report.
1.4.2 Problem Analysis

The second chapter presents the Problem behind the necessity to develop the Data Correlation prototype. It introduces the concept of network optimisation, followed by a brief description of IMSI Call Trace – a method for optimising radio networks. The internship’s objects, the high-level decisions taken, and the problems found are presented as the driving forces behind the decisions taken. At the end of the chapter there is the project’s calendar comprised by the tasks performed during the internship.

1.4.3 Technological Review

The third chapter analyses the technological context in which the Data Correlation prototype took place. It starts by introducing general concepts about UMTS: its history, main elements and protocols; as well as the Test Mobile technology employed in the project: TEMS Investigation and Agilent. This technological background is used for reasoning the necessity to develop the Data Correlation prototype. In addition, at the end of the chapter, the technologies used in the project are described: Java, JUnit, ANT, UML and Eclipse.

1.4.4 System Specification

The fourth chapter describes the proposed solution to the Data Correlation problem. The high-level requirements reported in chapter 2 are redefined in terms of functional requirements, non-functional requirements, system constraints, and use cases. In addition to the system’s requirements, the system is described in terms of its architectural design. The system’s architecture is depicted, using several views and representative structures that reveal its main architectural qualities (modifiability and usability). The architectural description also points out the components reused from UCAT.

1.4.5 Development

The fifth chapter narrates what happened during the development of the Data Correlation prototype. The development events – important decisions, problems faced, difficulties found and projects breakthroughs – are narrated by chronological order. The final developed prototype is introduced, pointing out its Graphical User Interfaces and the Synchronization and Correlation module that implements the Correlation Mechanism.

1.4.6 Results and future work

The sixth chapter evaluates the final results, taking into consideration the system requirements, the problems faced and all decisions taken during development. At the end of the chapter the perspectives for future evolution are reported.

1.4.7 Conclusions

The seventh chapter draws the final considerations of the work done. It describes what was achieved during the internship and what value was created.

1.4.8 References and Biography

The eighth and final chapter lists the resources used during this internship.
2 Problem Analysis

This chapter will introduce the problem behind the necessity to develop the Data Correlation prototype. It introduces the concept of network optimisation, followed by a brief description of IMSI Call Trace – a method for optimising radio networks. The internship objectives, as well as the high-level requirements and problem found, are presented as the driving forces behind the decisions taken during the development. At the end of the chapter there is the project’s calendar comprised by the tasks scheduled for the internship.

2.1 Problem Overview

2.1.1 Radio Network Optimisation

WCDMA system deployment, after successful field trials and service launches, entered a new critical stage: the phase of network optimisation and network troubleshooting. The objective of network optimisation is to evaluate and improve the quality of services, whereas network troubleshooting is concerned with detecting problems, and then finding and eliminating the problem’s causes. Problems are evaluated using special defined indicators based on measurement results, and the fewer problems one finds, the higher the quality of the services provided. The indicators that should be monitored and optimised are for example: Traffic, Traffic deviation, Traffic mixture, Soft handover percentage, Average TX power, Average RX power, Drop calls, Interference, Handovers per cell, Inter-system handover, Throughput, Bit error rate and frame error rate, etc.

The WCDMA system, like the GSM system, needs continuous optimisation and monitoring. This monitoring requirement is emphasised in WCDMA, as in all CDMA systems, because the traffic demands vary continuously, which in turn directly influences the radio network’s quality. WCDMA also operates in a much higher frequency band than those typically used in GMS and TDMA systems (2100 MHz in WCDMA in opposition to 900MHz and 1900MHz in GMS). Also, the higher data rates for UMTS require better signal strength, Eb/N0. These operating frequency differences, plus the higher data rates, mean that the radio propagation will not be equivalent. As a result, the GSM base station coverage areas are not necessarily valid in WCDMA and so must be reconsidered.

2.1.2 IMSI Call Trace

The IMSI Call Trace is a common troubleshooting, verification, optimisation, and maintenance procedure used to assure the quality of radio networks. The IMSI Call Trace can be used for ensuring adequate coverage and call quality, during a radio network bring up. It can also be used latter during operation/management, as user awareness rises, to achieve optimal quality. It helps improve voice quality, increase accessibility, increase number of successful call attempts, and achieve better service performance, while at the same time keeping new investments at a minimum level.

Throughout the duration of an UMTS call (or data session), messages are continuously exchanged between the user equipment and the network (via air-interface), and between different network elements. These messages can be measurements of call quality, signalling
events, requests, answers, etc. If necessary, the exchanged messages can be traced and have its messages registered into a log file. When tracing a call, two approaches are commonly available:

- Tracing the messages from the Radio Network’s perspective, by monitoring the messages exchanged between different Network Elements and also between the Radio Network’s Node B and the User Equipment. With this perspective, the messages are monitored by the Radio Network, once the Trace functionality is activated for a User Equipment’s IMSI. Usually the Radio Networks have trace support built-in as a basic functionality.

- Tracing the messages from the User Equipment’s perspective, by monitoring the messages exchanged between the User Equipment and the Radio Network’s Node B. With this perspective, the messages are registered by the User Equipment, either by using a special User Equipment or by configuring a mobile phone to actively monitor incoming/out coming messages. Usually a Test Mobile solution like TEMS Investigation or Agilent is adopted. A Test Mobile solution is normally composed by a User Equipment capable of monitoring messages and a software application for analysing information.

### 2.2 Objectives

Although both trace files contain valuable information, no single file contains all the available information. This is due to each file holding exclusive information to its own. A person doing network optimisation would use a Radio Network Analysis tool and a Test Mobile tool to better understand the whole picture. However using two tools for recognizing errors is difficult and time consuming, considering that the user will have to manually detect and identify errors in messages lists that can be thousands of messages long.

Since the Radio Network Analysis (RNA) tool only worked with the Radio Network Trace files, it was proposed that a module should be developed to allow RNA to support also Test Mobile Trace files. RNA would then generate seamless analysis with information from both perspectives.

Several Test Mobiles solution were considered when it came the time to chose which ones the Data Correlation mobile would support. In the end, TEMS Investigation and Agilent were chosen. This decision was taken because TEMS Investigation and Agilent have the biggest market share in Test Mobile Solutions.

Several objectives were considered at the beginning of the project. These included both functional and non-functional requirements. These requirements, by some degree shaped and guided the whole development process, and were the source of all the development problems.

As so, it was decided that the Data Correlation prototype should be able to:

- Decode the Test Mobile trace files from the TEMS Investigation solution into a manageable format;
  
  - Since the TEMS Investigation trace files contain both L3 messages and TEMS proprietary messages, it is necessary to support to both messages types;

- Decode The Test Mobile trace files from the Agilent solution into a manageable format;
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

- Correlate the Test Mobile trace files with the trace files from the Siemens Network Elements. By Correlation we understand:
  - Identify identical messages;
  - Synchronize messages;
  - Identify if messages are missing or if they are present in both traces;
- The Data Correlation will be in the future integrated with RNA’s UCAT feature, so extra attention must be taken and reuse components and design wherever possible;
  - Reuse the architectural design of the UCAT;
  - Reuse UCAT’s Components and Modules;
  - Reuse RNA’s global graphical appearance.

2.2.1 Project scheduling

The project was scheduled and planned for six months. The project’s started on the 1st of March and ended on the 31st August. The planned tasks were:

**Table 1: Tasks Descriptions**

<table>
<thead>
<tr>
<th>Task ID²</th>
<th>Task Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Internship Introduction</strong> - Understand how things worked in Siemens, what was the work process of RNA, and also studying UMTS and the UTRAN Architecture.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Data Correlation Requirements Specification</strong> - Specify what was supposed that the Data Correlation system would do. It was necessary to understand what the requirements (both functional and non-functional) were.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Data Correlation Architectural Design</strong> - Design a generic architecture capable of handling with the modifiability requirements inherent to the Data Correlation system.</td>
</tr>
<tr>
<td>7</td>
<td><strong>TEMS Investigation Decoder Implementation</strong> - Study, specify, design and implement a decoder for the TEMS Investigation files.</td>
</tr>
<tr>
<td>8</td>
<td><strong>TEMS Investigation Decoder Documentation</strong> - Document the TEMS Investigation Decoder module, the TEMS Investigation file’s structure and all the functional specifications.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Agilent Decoder Implementation</strong> - Study, specify, design and implement a decoder for the Agilent files.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Agilent Decoder Documentation</strong> - Document the Agilent Decoder, the Agilent file’s structure and all the functional specifications.</td>
</tr>
<tr>
<td>13</td>
<td><strong>File and Analysis Area Implementation</strong> - Develop an application for managing Trace files, their respective analyses. The application would use the decoding modules developed in the previous tasks and reuse some RNA components.</td>
</tr>
</tbody>
</table>

² Task ID according to ID in Figure 3: Project’s Gantt Diagram
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td><strong>File and Analysis Area Documentation</strong> - Document the management application developed in the previous task.</td>
</tr>
<tr>
<td>16</td>
<td><strong>Synchronization and Correlation Engine Algorithm Specification</strong> - Study and write down an algorithm capable of synchronizing and correlating IMSI trace data from different sources.</td>
</tr>
<tr>
<td>17</td>
<td><strong>Synchronization and Correlation Engine Implementation</strong> - Implement the synchronization and correlation algorithm studied in the previous task into a module and incorporate it in the management application developed in task 13.</td>
</tr>
<tr>
<td>18</td>
<td><strong>Synchronization and Correlation Engine Documentation</strong> - Document the implementation of the synchronization and correlation algorithm implemented in the previous task.</td>
</tr>
<tr>
<td>20</td>
<td><strong>System Testing and Optimisation</strong> - Test and optimise the Data Correlation system.</td>
</tr>
<tr>
<td>21</td>
<td><strong>Requirements Compliance Verification</strong> - Validate the Data Correlation system against the requirements written in the beginning of the project in task 2.</td>
</tr>
<tr>
<td>22</td>
<td><strong>Internship Report Elaboration</strong> - Write this report.</td>
</tr>
</tbody>
</table>
Figure 3: Project's Gantt Diagram
3 Technological Review

3.1 UMTS General Concepts

3.1.1 Historical introduction – From 1G to UMTS Phase 1

Electromagnetic waves were first discovered as a communications medium at the end of the 19th century. The first systems offering mobile telephone service (cell phone) were introduced in the late 1940s in the United States and in the early 1950s in Europe. Those early single cell systems were severely constrained by restricted mobility, low capacity, limited service, and poor speech quality. The introduction of cellular systems in the late 1970s and early 1980s represented a quantum leap in mobile communication. Semiconductors technology and microprocessors made smaller, lighter weight and more sophisticated mobile systems a practical reality for many more users.

The development of 2G cellular systems was driven by the need to improve transmission quality, system capacity, and coverage. Speech transmission still dominates the airways, but the demands for fax, short message, and data transmissions are growing rapidly. 2G cellular systems include GSM, Digital AMPS (D-AMPS), code division multiple access (CDMA), and Personal Digital Communication (PDC). Phase 1 of the standardization of GSM900 was completed by European Telecommunications Standards Institute (ETSI) in 1990. GSM standards were enhanced in Phase 2 (1995) to incorporate a large variety of supplementary services that were comparable to digital fixed network integrated services digital network (ISDN) standards.

![UMTS Evolutionary Diagram](image)

Figure 4: UMTS Evolutionary Diagram
The main characteristics of 3G systems, known collectively as IMT-2000, are a single family of compatible standards that have the following characteristics:
- Used worldwide
- Used for all mobile applications
- Support both packet-switched (PS) and circuit-switched (CS) data transmission.
- Offer high data rates up to 2Mbps (depending on mobility/velocity)
- Offer high spectrum efficiency

The most important IMT-2000 proposals are the UMTS (also known as wideband CMDA, W-CMDA) as the successor to GSM. UMTS allows many applications to be introduced to a worldwide base of users and provides a vital between today’s multiple GSM systems and IMT-2000. The new network also addresses the growing demand of mobile and Internet applications for new capacity in the overcrowded mobile communications sky. UMTS increases transmission speed to 2Mbps per mobile user and establishes a global roaming standard.

### 3.1.2 UMTS architecture

In UMTS release 1 (Rel.’99), a new radio access network UMTS terrestrial radio access network (UTRAN) is introduced. UTRAN, the UMTS radio access network (RAN), is connected via the Iu to the GSM Phase 2+ core network (CN). The Iu is the UTRAN interface between the radio network controller (RNC) and CN; the UTRAN interface between RNC and the packet-switched domain of the CN (Iu-PS) is used for PS data and the UTRAN interface between RNC and circuit-switched domain of the CN (Iu-CS) is used for CS data.

UMTS/GSM dual-mode user equipment (UE) will be connected to the network via UMTS air (radio) interface (Uu) at very high data rates (up to almost 2Mbps). Outside the UMTS service area, UMTS/GSM UE will be connected to the network at reduced data rates via the Um.

![Figure 5: UMTS Architecture](image)
The UMTS standard can be seen as an extension of existing networks. Two new network elements are introduced in UTRAN, RNC, and Node B. UTRAN is subdivided into individual radio network systems (RNSs), where each RNS is controlled by a RNC. The RNC is connected to a set of Node B elements, each of which can serve one or several cells.

Existing network elements, such as MSC, SGSN, and HLR, can be extended to adopt the UMTS requirements, but RNC, Node B and the handsets must be completely new designs. RNC will become the replacement for BSC, and Node B fulfils nearly the same functionality as BTS. GSM and GPRS networks will be extended, and new services will be integrated into an overall network that contains both existing interfaces such as A, Gb, and Abis, and new interfaces that include Iu, UTRAN interface between Node B and RNC (Iub), and UTRAN interface between two RNCs (Iur).

### 3.1.2.1 RNC

The RNC enables autonomous radio resource management (RRM) by UTRAN. It performs the same functions as the GSM BSC, providing central control for the RSN elements (RNC and Node Bs).

The RNC handles protocol exchanges between Iu, Iur, and Iub interfaces and is responsible for centralized operation and maintenance (O&M) of the entire RNS with access to the OSS.

The RNC uses the Iur interface, which has no equivalent in GSM BSS, to autonomously handle 100 percent of the RRM, eliminating that burden from the CN. Serving control functions such as admission, the RRC connection to the UE, congestion and handover/macro diversity are managed entirely by a single serving RNC (SRNC).

If another RNC is involved in the active connection through an inter-RNC soft handover, it is declared a drift RNC (DRNC). The DRNC is only responsible for the allocation of code resources. A reallocation of the SRNC functionality to the former DRNC is possible (serving radio network subsystem [SRNS] relocation). The term controlling RNC (CRNC) is used to define the RNC that controls the logical resources of its UTRAN access points.

**RNC Functions:**
- Autonomous RRM
- ATM Switching & Multiplexing
- Control Over RNS
- O&M Interface

### 3.1.2.2 Node B

Node B is the physical unit for radio transmission/reception with cells. Depending on sectoring (omni/sector cells), one or more cells may be served by a Node B. A single Node B can support both FSS and TDD modes, and it can be co-located with a GSM BTS to reduce implementation costs. Node B connects with the UE via the W-CDMA Uu radio interface with the RNC via the Iub asynchronous transfer mode (ATM) - based interface.

**Node B Functions:**
- Conversion of data to and from Uu radio interface
- Forward error correction (FEC)
- Rate adaptation
- Spreading/disпreading
- Quadrature phase shift keying (QPSK) modulation on the air interface
• Measures Quality and Strength
• Determines Frame Error Rate (FER)
• Responsible for the FDD softer handover

3.1.2.3 **UMTS UE**

The UMTS UE is based on the same principals as the GSM MS – the separation between mobile equipment (ME) and the UMTS subscriber identity module (SIM) card (USIM).”

UE Functions:
• UE as Node B Counterpart:
  o FEC (Encoding & Interleaving)
  o Power Control (Open & Inner Loop)
  o Radio Measurement (FER, SIR, Quality & Power)
  o Spreading/De-spreading
  o Modulation/De-modulation
• UE as RNC Counterpart
  o BEC (Acknowledge Mode: NRT)
  o RRC (Radio Resource Control)
  o Handover (CS) & Cell Selection (PS)
  o De-/Ciphering
• UE as CN Counterpart
  o Mobility Management (Location Registration, Authentication, IMEI Check, Attach/Detach)
  o Session Management (PDP), Context De-/Activation
  o Bearer Negotiation/Service Request

3.1.3 **Application protocols**

Application protocols are Layer-3 protocols that are defined to perform UTRAN-specific signalling and control. UTRAN-specific control protocols exist in each of the four interfaces.

3.1.3.1 **IU: Radio Access Network Application Part (RANAP)**

This protocol layer provided UTRAN-specific signalling and control over the lu. The following are a subset of the RANAP functions:
• Overall radio access bearer (RAB) management, which includes the RAB’s step, maintenance and release.
• Management of lu connections.
• Transport of nonaccess stratum (NAS) information between the UE and the CN; for example, NAS contains the mobility management signalling and broadcast information.
• Exchanging UE location information between the RNC and CN.
• Paging requests from the CN to the UE.
• Overload and general error situation handling.

3.1.3.2 **Iur: Radio Network Sublayer Application Part (RNSAP)**

UTRAN-specific signalling and control over this interface contains the following:
• Management of radio links, physical links, and common transport channel resources.
• Paging
• SRNC relocation
• Measurement of dedicated resources
3.1.3.3 *Iub: Node B Application Part (NBAP)*

UTRAN-specific signalling and control of the Iub includes the following:
- Management of common channels, common resources, and radio links
- Configuration management, such as cell configuration management
- Measurement handling and control
- Synchronization (TDD)
- Reporting of error situations

3.1.3.4 *Uu: Radio Resource Control (RRC)*

This layer handles the control plane signalling over the Uu between the UE and the UTRAN. Some of the functions offered by the RRC include the following:
- Broadcasting information
- Management of connections between the UE and the UTRAN, which include their establishment, maintenance, and release.
- Management of the radio bearers, which include their establishment, maintenance, release, and the corresponding connection mobility
- Ciphering control
- Outer loop power control
- Message integrity protection
- Timing advance in the TDD mode
- UE measurement report evaluation
- Paging and notifying

Two modes of operation are defined for the UE – the idle mode and the dedicated mode. In the idle mode the peer entity of the UE’s RRC is the Node B, while in the dedicated mode it is at the SRNC. Also, higher-layer protocols to perform signalling and control tasks are found on top of the RRC.

3.2 *Test Mobile Technology*

This section introduces the Test Mobile technologies supported by the Data Correlation prototype: TEMS Investigation and Agilent E7476A. Each technology is introduced and their main features described. Also, the Post processing functionality of each technology is presented.

3.2.1 *TEMS Investigation*

TEMS Investigation is the industry-leading tool for troubleshooting, verification, optimisation, and maintenance of wireless networks. Offering data collection, real-time analysis, and post-processing all in one, TEMS Investigation is a complete solution for all of a network operator’s daily network optimisation tasks. This complete solution eliminates the need for multiple tools, reducing costs and saving time and effort for operations staff.

With equal support for GSM, GPRS, EDGE, and WCDMA technologies, TEMS Investigation is used for rolling out new networks and to ensure seamless integration with existing networks. By using TEMS Investigation, operators can achieve improved voice quality, increased accessibility, more successful call attempts, and better service performance. A wide range of features makes TEMS Investigation practical throughout the network’s life cycle.
TEMs Investigation also offers multi-vendor support as well as multi-technology infrastructure support. It supports handsets from all major vendors across multiple technologies. TEMs Investigation can be used to:

- Tune and optimise WCDMA and GSM/GPRS networks
- Perform fault-tracing and troubleshooting
- Verify true terminal behaviour with phone based measurements
- Verify cell coverage, capacity, accessibility, etc.
- Troubleshoot the network
- Perform indoor, pedestrian, and outdoor measurements
- Post-processing of multiple log files.

3.2.1.1 Multi-mode functionality

TEMs Investigation supports multi-mode functionality for system verification, troubleshooting, and optimisation of WCDMA and GSM/GPRS/EDGE networks. The multi-mode functionality makes it possible to:

- Verify and optimise WCDMA and GSM intersystem handover and cell reselection.
- Verify compressed mode behaviour.
- Verify and compare GSM/WCDMA coverage and performance.
- Verify WCDMA and GSM system accessibility.

3.2.1.2 Post processing functionality

TEMs Investigation includes the possibility of performing Post processing of collected data. This feature is exploited to allow the Data Correlation system to import TEMs Investigation files and then correlate the information with Siemens Networks’ IMSI trace files. TEMs Investigation can export its files into .FMT files (tab separated ASCII files).

3.2.2 Agilent

The Agilent E7476A (UTMS) drive test system is used to obtain RF coverage and service performance measurements for wireless communications networks that use the advanced 3GPP W-CDMA (UTMS) technology. The full system planned for the future will include a 3GPP phone, when available, to work in conjunction with the key measurements of the receiver. The system software runs on a PC that interfaces with an Agilent digital RF receiver. Future plans include adding a 3GPP phone to the system. The system can control up to four receivers and four phones simultaneously. The drive test system is a platform product: features such as carry around testing, indoor testing and real-time map information can be added, as well as measurement capabilities in other technologies such as IS-95 CDMA, cdma2000, GSM, and TDMA. Further points that should be noted are that the SW platform can support testing of any combination of different technologies simultaneously with the addition of the appropriate licenses. All measurements can be made with reference to position, using both the receivers’ internal GPS or with the use of pen tablet option offered primarily for the monitoring of network performance indoors.
3.2.2.1 Post-processing measurement data

Data logged with the Agilent E7476A can easily be further analysed after data collection by third party applications. This feature is exploited to allow the Data Correlation system to import Agilent E7476A IMSI Trace files and then correlate the information with Siemens Networks’ IMSI trace files. The export functionality allows filtering of information so it is possible to select only the information required for the Data Correlation system, and so reducing considerably the size of the exported files. Agilent E7476A IMSI Trace files can be exported to .csv, .mif, .qle, and .txt formats.

3.3 Tracing Messages

The workflow for generating trace data from the Radio Network’s perspective has two steps: Trace Activation and Trace Data Generation, which are briefly explained below in Table 2: Radio Network’s IMSI trace workflow:

Table 2: Radio Network’s IMSI trace workflow

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1. Activate Trace     | The activation can be done in the RNC or in the RC using CLI commands. The activation specifies the specific IMSI and the Trace Type that can be one of the following:  
  - Layer 3  
  - Layer 3 + Dedicated Measurements  
  - Layer 3 + RNC Resource  
  - Layer 3 + Dedicated Measurements + RNC Resource |
| 2. Trace Data Generation | The specified IMSI is traced in the RNC so that each call will generate a binary trace file that is uploaded from the RNC to the RC.  
  
The RC notifies RNA automatically about the new trace files. The corresponding files are automatically copied to RNA (Binary Trace file area) and the corresponding Analysis is created (Analysis Area).  
  
  **NOTE:** At start-up RNA synchronises with RC. Afterwards RC notifies RNA every time there are new trace files. Therefore the Binary Trace file area is always synchronised with the RC. |

For tracing messages with a Test Mobile solution, usually a drive test campaign is performed. During one of these campaigns, the Test Mobile equipment is placed within a vehicle. Then, people drive around a pre-determined area with the Test Mobile, continually making calls and measurements.
Test Mobiles are very useful for thoroughly testing specific situations, as per example: consider two cells distancing one from the other by 1Km. Somewhere, between the two cells, a handover should occur. Someone can drive repeatedly back and forward between the cells, while checking in the Test Mobile equipment if the handover took place correctly. In addition to drive tests, walk tests can also be performed. Walk tests usually take place inside buildings or within areas where normally vehicles could not be sent. When performing a walk test, the Test Mobile equipment is placed within a backpack for easier transportation.

**Table 3: Advantages and Disadvantages of Test Mobile Trace vs. Radio Network Trace**

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Mobile Trace</strong></td>
<td><strong>Radio Network Trace</strong></td>
</tr>
<tr>
<td>- Test Mobile Trace files contain very detailed information on the Air Interface.</td>
<td>- Information is available on messages exchanged between different network elements.</td>
</tr>
<tr>
<td>- It is easy to simulate critical events like handovers, call dropped, etc.</td>
<td>- Data can be collected without the necessity of moving around.</td>
</tr>
<tr>
<td>- Test Mobile equipment can make measurements even on idle status, i.e. when no call is being made.</td>
<td>- Radio Network’s normally have built-in Trace capacity.</td>
</tr>
<tr>
<td></td>
<td>- Information on Air Interface not as detailed as with Test Mobile technology</td>
</tr>
<tr>
<td></td>
<td>- Trace data only available when call is made; no information available on idle status.</td>
</tr>
<tr>
<td></td>
<td>- Difficult to produce critical events.</td>
</tr>
</tbody>
</table>

For the Data Correlation project two Test Mobile solutions were chosen to be supported. The chosen solution
3.4 State of the Art Technology

The concept of correlating IMSI trace data, generated by different sources, into a seamless analysis is unprecedented, and no other solution on the market supports this functionality. There are solutions capable of importing data from different sources (e.g. Actix), but none of them is capable of correlating the imported into a seamless analysis. Being unique in a competitive market creates this concept an important commercial advantage to RNA, but being unique also makes development difficult.

With no previous work in which to base the development of the Data Correlation system it was necessary to study what could be reused (mainly RNA components), what had to be reverse-engineered (components that belong to competitors), and had to be created from scratch (components completely original).

The study revealed that the Data Correlation system required:

- A component capable of decoding Siemens IMSI trace files. RNA already possessed such a component so it was decided that it would be reused by the Data Correlation system.
- A component capable of decoding TEMS Investigation IMSI trace files. There wasn’t any component available for doing this, and furthermore, being a competitor’s technology, not even the files’ formats where available. It was necessary to reverse-engineer the files’ formats, and then build a decoder.
- A component capable of decoding Agilent IMSI trace files. There was a library capable of doing this, but due to legal requirements imposed by Agilent, it was prohibited to use it. There was no other solution other than reverse-engineer the files’ formats and then build a decoder.
- A component (algorithm) capable of synchronizing and correlating IMSI trace data from different sources. Being a very specific problem, the component had to be developed from scratch.

Concisely it can be said that the Data Correlation system is pioneering, and that there aren’t any existing solutions to which it can be compared.

3.5 Project’s technological choices

The technological choices were somewhat limited and dictated by the technologies in which RNA was developed. The main technology was Java, helped by the usage of JUnit, Ant, and UML. Installation

3.5.1 Java

Java technology is both a programming language and a platform.

3.5.1.1 The Java Programming Language

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java bytecodes – the platform-independent codes interpreted by the interpreter on the Java platform.
You can think of Java bytecodes as the machine code instructions for the Java Virtual Machine (Java VM). The bytecodes can be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows, a Solaris workstation, or on a Mac.

The Java programming language is a high-level language that can be characterized by the following characteristics:

- **Simple** – Primary characteristics of the Java programming language include a simple language that can be programmed without extensive programmer training while being attuned to current software practices.

- **Familiar** – Even though C++ was rejected as an implementation language, keeping the Java programming language looking like C++ as far as possible results in it being a familiar language, that allows programmers can migrate easily to the Java platform and be productive quickly.

- **Object Oriented** – The Java programming language is designed to be object oriented from the ground up. The needs of distributed, client-server based systems coincide with the encapsulated, message-passing paradigms of object-based software.

- **Robust** – The Java programming language is designed for creating highly reliable software. It provides extensive compile-time checking, followed by a second level of run-time checking. Language features guide programmers towards reliable programming habits. The memory management model is extremely simple: objects are created with a new operator. There are no explicit programmer-defined pointer data types, no pointer arithmetic, and automatic garbage collection.

- **Secure** – Java technology is designed to operate in distributed environments, which means that security is of paramount importance. With security features designed in the language and run-time system, Java technology lets you construct application that can’t be invaded from outside.

- **Architectural Neutral** – Applications must execute atop a variety of hardware platforms and operating systems, and interoperate with multiple programming language interfaces. To accommodate the diversity of operating environments, the Java CompilerTM product generates bytecodes – and architecture neutral intermediate format designed to transport code efficiently to multiple hardware and software platforms.

- **Portable** – Java technology takes portability a stage further by being strict in its definition of the basic language. The programs are the same on every platform – there are no data type incompatibilities across hardware and software architectures. The Java virtual machine is the specification of an abstract machine for which Java programming language compilers can generate code.

- **High Performance** – Performance is always a consideration. The Java platform achieves superior performance by adopting a scheme by which the interpreter can run at full speed without needing to check the run-time environment. The automatic garbage collector runs as a low-priority background thread, ensuring a high probability that memory is available when required, leading to better performance.

- **Interpreted** – The Java interpreter can execute Java bytecodes directly on any machine to which the interpreter and run-time system have been ported. In an interpreted platform such as Java technology-based systems, the link phase of a program is simple, incremental, and lightweight. You benefit from a much faster development cycles, prototyping, experimentation, and rapid development are the normal case, versus the traditional heavyweight compile, link, and test cycles.

- **Threaded** – The Java platform supports multithreading at the language level with the addition of sophisticated synchronization primitives: the language library provides the
Thread class, and the run-time system provides monitor and condition lock primitives. At the library level, moreover, Java technology’s high-level system libraries have been written to be multithread safe: the functionality provided by the libraries is available without conflict to multiple concurrent threads of execution.

- **Dynamic** – While the Java Compiler is strict in its compile-time static checking, the language and run-time system are dynamic in their linking stages. Classes are linked only as needed. New code modules can be linked in on demand from a variety of sources, even from sources across a network.

### 3.5.1.2 The Java Platform

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces, these libraries are known as packages.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code.

### 3.5.2 JUnit

JUnit is a wildly popular unit testing framework, used primarily by developers, originally written by Erich Gamma and Kent Beck. One of its main advantages is that the unit tests area developed in the same language as the product code, so developers can work in an environment that they’re comfortable with.

JUnit features include:

- Assertions for testing expected results.
- Test fixtures for sharing common test data.
- Test suites for easily organizing and running tests.
- Graphical and textual test runners.

JUnit is a good choice for a unit test tool, especially if the developers are doing “test-drive development” (TDD), where the unit testing is an integral part of the development process. The JUnit’s functionalities are accesses from the Java API, but there is also a command line and a graphical interface for running tests.

A selling point for JUnit is that fact that the unit tests are developed in the same language as the product code, so developers can work in an environment that they’re comfortable in. This selling point might not be as strong if you want an independent test team to develop black-box tests. If the testers are already familiar with a different automation language, and they don't need white-box access to the product implementation, then they would likely be more productive with a tool designed for black-box testing.
3.5.3 ANT

Apache Ant is a Java-based build tool, created for surpassing others build tools like make, gnumake, nmake. These tools have several limitations, especially when developing software across multiple platforms. Make-like tools are inherently shell-based; they evaluate a set of dependencies, and then execute commands not unlike what someone would issue in a shell. This means that someone can easily extend these tools by using or writing any program for the OS that you are working on. However, this also means that someone is limited to the OS, or at least the OS type such as UNIX, that someone is working on.

Ant is different. Instead of a model where it is extended with shell-based commands, Ant is extended using Java classes. Instead of writing shell commands, the configuration files are XML-based, calling out a target tree where various tasks get executed. Each task is run by an object that implements a particular Task interface.

3.5.4 UML

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of object oriented software development and the global software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML were:
- Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modelling Language (UML) was designed to respond to these needs.

3.5.5 Eclipse

Eclipse is an open source, Java-based, extensible development platform. By itself, it is simply a framework and a set of services for building a development environment from plug-in
components. Opportunely, Eclipse comes with a standard set of plug-ins, including the Java Development Tools (JDT). While most users use Eclipse as a JAVA IDE, it has additional functionalities. Eclipse also includes the Plug-in Development Environment (PDE), which is mainly of interest to software developers who want to extend Eclipse. It allows user to build tools that integrate seamlessly with the Eclipse Environment.

This parity and consistency aren’t limited to Java development tools. Although Eclipse is written in the Java language, its use isn’t limited to the Java language; it supports languages such as C/C++, COBOL, and Eiffel. The Eclipse framework can also be used as the basis for others types of applications unrelated to software development, such as content management systems.

Eclipse is open source. Open source software is software that is released with a license intended to ensure that certain rights are granted to users. The most obvious right, of course, is that the source code must be made available so that users are free to modify and redistribute the software. This protection of users’ rights is accomplished with a device called a copyleft: the software license claims copyright protection and prohibits distribution unless the user is granted these rights. The copyleft also requires that any re-distributed software be covered by the same license. Since this, in effect, stands the purpose of copyright on its head – using the copyright to grant rights to the user rather than reserve them for the developer of the software – copyleft is often described as "all rights reversed."
4 System Specification

This chapter describes the proposed solution to the Data Correlation problem. The high-level requirements reported in chapter 2 are redefined in terms of functional requirements, non-functional requirements, system constraints, and use cases. In addition to the system’s requirements, the system is described in terms of its architectural design. The system’s architecture is depicted, using several views and representative structures that reveal its main architectural qualities. The architectural description also points out the components reused from UCAT.

4.1 System’s requirements

In order to understand some of the design decisions taken, it is necessary to describe the Data Correlation requirements with further detail. Throughout the project, several modifications were made to the initial requirements. This was due to the fact that in the beginning of the project, since the Test Mobile technology belonged to competitors, it was unknown what could or couldn’t be done. The requirements written in the following pages reflect the most mature version of the requirements, i.e. those against which the prototype was validated.

4.1.1 Functional requirements

The functional requirements here reported are requirements that describe what the Data Correlation’s functionalities are and how they are expect to perform. There are three major user requirements the Data Correlation prototype:

1. The system must allow the user to manage Siemens’ trace files and Test Mobile’s trace files.
2. The system must allow the user to manage Siemens’ analysis and Test Mobile’s analysis.
3. The system must allow the user to visualize the contents of an analysis.

Besides the major user requirements, there are also two major system requirements:

4. The system must support and decode TEMS Investigation and Agilent trace files.
5. The system must allow correlation between Siemens analysis and Test Mobile analysis.

The following section will address each of these user and system requirements individually. They will be decomposed into further user and system requirements and their use cases will be described.

4.1.1.1 Siemens trace file and Test Mobile trace file management functionalities

The first requirement, a user requirement, declared that the Data Correlation prototype needed to have the functionality of managing Siemens and Test Mobile’s trace files. In order to better understand what was required, the requirement was decomposed into secondary user requirements and system requirements:

1. The user must have the possibility of performing trace file management operations.
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

a. The user must have the possibility of adding trace files into the system.
   i. The system must have an Open file button.

b. The user must have the possibility of deleting trace files from the system.
   i. The system must have a Delete file button.

c. The user must have the possibility of importing trace files.
   i. The system must have an Import file button.
   ii. If the file imported is a TEMS Investigation file, the system must request the file’s IMSI and date.
   iii. If the file imported is an Agilent file, the system must request the file’s IMSI.

2. The user must have the possibility of viewing which trace files are in the system and see their details.

   a. The system must have a graphical area for displaying the Siemens’ trace files in the system.
      i. The system must have a table where the Siemens’ trace files are displayed.
         1. There is a Reuse Requirement of reusing the GenericTable class for the implementation of the table where are displayed the Siemens’ trace files.
         ii. Each row of the table corresponds to a single Siemens’ trace file.
         iii. Whenever a new Siemens’ trace file is opened, a new row is added to the Siemens trace files table representing the new Siemens’ trace file.
         iv. The table must have several columns displaying information about the Siemens’ trace files. The necessary file details are listed in Table 4: Necessary file details list.

   b. The system must have a graphical area for displaying the Test Mobile’s trace files in the system.
      i. The system must have a table where the Test Mobile’s trace files are displayed.
         1. There is a Reuse Requirement of reusing the GenericTable class for the implementation of the table where are displayed the Test Mobile’s trace files.
         ii. Each row of the table corresponds to a single Test Mobile’s trace file.
         iii. Whenever a new Test Mobile’s trace file is opened, a new row is added to the Test Mobile files table representing the new Test Mobile’s trace file.
         iv. The table must have several columns displaying information about the Test Mobile’s trace files. The necessary file details are listed in Table 4: Necessary file details list.
c. The system must have a graphical area for displaying the details of a selected trace file.

i. The system must have a panel displaying several fields reporting some of the file’s information. The necessary file details are listed in Table 4: Necessary file details list.

<table>
<thead>
<tr>
<th>Detail Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>The name of the file.</td>
</tr>
<tr>
<td>File Format</td>
<td>The Format of the File; it can be either: “Trace” for Siemens trace, “TEMS” for TEMS Investigation trace and “Agilent” for Agilent trace.</td>
</tr>
<tr>
<td>File Size</td>
<td>The size, in bytes, of the file.</td>
</tr>
<tr>
<td>Server Date</td>
<td>The date of when this file was added to the system.</td>
</tr>
<tr>
<td>State</td>
<td>The State of the File; it can be “New” if the file as been newly added to the system, and “Processed” if the file has been already decoded.</td>
</tr>
<tr>
<td>Analysis Available</td>
<td>An indication if there is an analysis of the file available.</td>
</tr>
<tr>
<td>In Server</td>
<td>An indication if the file is still in the system.</td>
</tr>
</tbody>
</table>

Based on these requirements, it was possible to design the following GUI diagram to give a rough idea of what was desired in graphical terms.

![GUI Interface prototype for the File Area](image)

**Figure 8: GUI Interface prototype for the File Area**

Also, based on the requirements, it was also possible to extract the use cases present in the Figure 9: Trace File Management Use Case Diagram.
Table 5: Trace file management use cases

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open file</td>
<td>The user can open a Siemens of Test Mobile trace file into the application</td>
<td>None</td>
</tr>
<tr>
<td>Delete file</td>
<td>The user can delete files from the File Area. After selecting a file, if the user presses the Delete button, the selected file will be removed for the application</td>
<td>There is one file selected</td>
</tr>
<tr>
<td>Import file</td>
<td>The user can generate analysis from a file. After selecting a file, if the user presses the Import button, the selected file will be imported, and an analysis will be created with the file’s contents</td>
<td>There is one file selected</td>
</tr>
<tr>
<td>Input IMSI</td>
<td>When the user imports a TEMS Investigation file or an Agilent file, he must input the file’s IMSI</td>
<td>The imported file is either a TEMS Investigation or an Agilent file</td>
</tr>
<tr>
<td>Input Date</td>
<td>When the user imports a TEMS Investigation file, he must input the file’s Date</td>
<td>The imported file is a TEMS Investigation file</td>
</tr>
</tbody>
</table>
4.1.1.2 Siemens analysis and Test Mobile analysis management functionality

The second requirement, a user requirement, declared that the Data Correlation prototype needed to have the functionality of managing Siemens and Test Mobile’s analyses. In order to better understand what was required, the requirement was decomposed into secondary user requirements and system requirements:

1. The user must have the possibility of performing analyses management operations.
   a. The user must have the possibility of deleting analyses from the system.
      i. The system must have a Delete analysis button.
   b. The user must have the possibility of viewing the contents of an analysis.
      i. The system must have a View analysis button.
   c. The system must have the possibility of correlation Siemens analyses with Test Mobile analyses.
      i. The system must have a Correlate button.
      ii. The system must have a panel where the user can choose which files the user wishes to correlate with.

2. The user must have the possibility of viewing which analyses are in the system and see their details.
   a. The system must have a graphical area for displaying the Siemens’ analyses in the system.
      i. The system must have a table where the Siemens’ analyses are displayed.
         i. There is a Reuse Requirement of reusing the GenericTable class for the implementation of the table where are displayed the Siemens’ analyses.
         ii. Each row of the table corresponds to a single Siemens’ analysis.
         iii. Whenever a new Siemens’ analysis is created, a new row is added to the Siemens analyses table representing the new Siemens’ analysis.
         iv. The table must have several columns displaying information about the Siemens’ analyses. The necessary analysis details are listed in Table 6: Necessary analysis details list.
   b. The system must have a graphical area for displaying the Test Mobile’s analyses in the system.
      i. The system must have a table where the Test Mobile’s analyses are displayed.
There is a Reuse Requirement of reusing the GenericTable class for the implementation of the table where are displayed the Test Mobile’s analyses.

Each row of the table corresponds to a single Test Mobile’s analyses.

Whenever a new Test Mobile’s analysis is created, a new row is added to the Test Mobile files table representing the new Test Mobile’s trace file.

The table must have several columns displaying information about the Test Mobile’s analyses. The necessary analysis details are listed in Table 6: Necessary analysis details list.

The system must have a graphical area for displaying the details of a selected trace file.

The system must have a panel displaying several fields reporting some of the file’s information. The necessary analysis details are listed in Table 6: Necessary analysis details list.

### Table 6: Necessary analysis details list

<table>
<thead>
<tr>
<th>Detail Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSI</td>
<td>The IMSI of the analysis.</td>
</tr>
<tr>
<td>Analysis Format</td>
<td>The Analysis Format; can be “Trace” for analyses holding Siemens trace data, “TEMS” for analyses holding TEMS Investigation trace data, “Agilent” for analyses holding Agilent trace data, “Trace + TEMS” for analyses holding both Siemens and TEMS Investigation trace data, and “Trace + Agilent” for analysis holding both Siemens and TEMS Investigation trace data.</td>
</tr>
<tr>
<td>Number of Files</td>
<td>The number of files used for generating this analysis.</td>
</tr>
<tr>
<td>Start Date</td>
<td>The analysis’ start time.</td>
</tr>
<tr>
<td>Stop Date</td>
<td>The analysis’ stop time.</td>
</tr>
</tbody>
</table>

Based on these requirements, it was possible to design the following GUI diagram to give a rough idea of what was desired in graphical terms.
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

Figure 10: GUI Interface prototype for the Analysis Area

Also, based on the requirements, it was also possible to extract the following use cases:

Figure 11: Analyses Management Use Case Diagram

Table 7: Analyses management use cases

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete Analysis</td>
<td>Deletes the currently selected analysis</td>
<td>There is one analysis selected</td>
</tr>
<tr>
<td>View Analysis</td>
<td>Displays the contents of an analysis</td>
<td>There is one analysis selected.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Prerequisites</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Correlate Analysis</td>
<td>Correlates a Siemens analysis with a Test Mobile analysis</td>
<td>There is one Siemens analysis selected and there is at least one Test Mobile analysis.</td>
</tr>
<tr>
<td>View analysis' details</td>
<td>Displays the contents of an analysis.</td>
<td>There is one analysis selected.</td>
</tr>
</tbody>
</table>

**4.1.1.3 Analysis visualization functionality**

The third requirement, a user requirement, declared that the Data Correlation prototype needed to have the functionality to visualising the contents of an analysis, which could be either Siemens analysis or Test Mobile analysis. In order to better understand what was required, the requirement was decomposed into secondary user requirements and system requirements:

1. The user must have the possibility of viewing the contents of an analysis.
   a. The system must have an area for analysis visualization.
   b. The analysis visualization area must have a table with the messages of the analysis.
      i. Each line in the table corresponds to a message.
      ii. The table must have several columns, each column displaying a specific message detail. The necessary message details are represented in Table 8: Required Message List Column names.
   c. The user must be able to select individual messages from the table.
   d. When a message is selected, that message’s details are displayed in a proper area.

Table 8: Required Message List Column names

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Number</td>
<td>The number of the message in the analysis.</td>
</tr>
<tr>
<td>Date</td>
<td>The message’s Date and Time.</td>
</tr>
<tr>
<td>Data Type</td>
<td>The type of data of the message; can be “L3 Message”, “HW Resource”, or “TEMS Proprietary Message”</td>
</tr>
<tr>
<td>Message Type</td>
<td>The type of the message.</td>
</tr>
<tr>
<td>Message Kind</td>
<td>A String indicating the message’s kind.</td>
</tr>
<tr>
<td>Direction</td>
<td>The message’s direction; can be “Send” or “Received”.</td>
</tr>
</tbody>
</table>

32
Protocol | The message’s protocol.
---|---

Based on these requirements, it was possible to design the following GUI diagram to give a rough idea of what was desired in graphical terms.

![GUI Interface Prototype for the Analysis Visualization Window](image)

**Figure 12: GUI Interface Prototype for the Analysis Visualization Window**

Also, based on the requirements, it was also possible to extract the following use cases:

![Data Correlation Analyses Visualization System](image)

**Figure 13: Analysis Visualization Use Case Diagram**

**Table 9: Analysis Visualization use cases**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Message details</td>
<td>Visualizes the details of an message</td>
<td>There is one message selected.</td>
</tr>
</tbody>
</table>
4.1.1.4 TEMS Investigation and Agilent decoding functionality

The fourth requirement, a system requirement, declared that the Data Correlation prototype needed to support the decoding of TEMS Investigation trace files and Agilent trace files. In order to understand what was required, the requirement was decomposed into further requirements:

1. The system must decode Test Mobile trace files into manageable formats.
   a. The system must support and decode TEMS Investigation trace files in ASCII export format.
      i. The system must extract and decode L3 Messages from TEMS Investigation exported ASCII files.
         1. The system must reuse UCAT's L3 decoding capabilities.
      ii. The system must extract and decode TEMS Proprietary messages from TEMS Investigation exported ASCII files.
      iii. The TEMS Investigation files must be decoded into a tree-like structure, where the message details are represented in nodes and leafs.
   b. The system must support and decode Agilent trace files in ASCII export formats.
      i. The system must extract and decode L3 Messages from TEMS Investigation exported ASCII files.

For more information of the supported TEMS Investigation formats consult topic 4.1.2.5.

4.1.1.5 Correlation functionality

The fifth requirement, a system requirement, declared that the Data Correlation prototype needed to support the correlation of Siemens and Test Mobile analysis.

4.1.2 Non-functional requirements

The Data Correlation system has several requirements that aren't system specific functions, but rather emerging system properties that must be respected. Although these requirements don't specify functional features of the system, their non-compliance will make the system ineffective, or worse, useless.

4.1.2.1 Portability requirements

The RNA tool runs on several operating systems, consequently the Data Correlation system must also runs on the same operating systems. For that purpose, the system must work in Windows and Solaris.

4.1.2.2 Availability requirements

The Data Correlation system must be able to handle with system failure and its associated consequences. A system failure occurs when a system no longer delivers a service consistent with its specification and the user becomes aware of it. The Data Correlation system must be able to detect faults, mask them from the user, and provide proper notification back to the user whenever possible. The system must display the current status of every user action, informing
if the action is running, finished, or failed (notifying the fault reasons). Furthermore, for increased availability, the system must support simultaneous execution of multiple user actions.

4.1.2.3 Usability requirements

The Data Correlation usability requirements are related with how easy it is for the user to accomplish a desired task, and the kind of support the system provides. To make the Data Correlation system more usable, it is required that the system adopts a GUI similar to that of RNA. A similar GUI will allow a user, with previous experience with RNA, to use the Data Correlation system without requiring specific training. In addition, in order to minimize the impacts of errors, it is required that any input made by the user to be validated as far as possible.

4.1.2.4 Modifiability requirements

The system must be highly modifiable because many of its components will be latter reused in the final, commercial product. As so, the developed components must be generic enough so that they can be reused without too much work. Additionally, since the system is expected to have an extended lifetime, the decoding components must be highly modular and easily replaced. This will allow further support of new Test Mobile versions and formats. Also, since the Data Correlation system’s modules will be integrated in UCAT, it is required that the system adopts a similar design to that of UCAT.

4.1.2.5 Standard compatibility requirements

The decoder must be compliant with the following TEMS Investigation versions:

Table 10: TEMS Investigation required version support

<table>
<thead>
<tr>
<th>Versions 2.X</th>
<th>2.1</th>
<th>2.2</th>
<th>2.3.1</th>
<th>2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Versions 3.X</td>
<td>3.0.1</td>
<td>3.0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.2.6 Legal requirements

Due to legal constraints, imposed by the owners of TEMS Investigation and Agilent, it is required that the Data Correlation system works with ASCII exports of the binary trace files.

4.1.2.7 Time constraints

The system must be finished until the 31st August, 2005.

4.2 Architectural documentation

This section reports the architectural design of the Data Correlation system. The architectural elements and connections are described using architectural structures and views.

4.2.1 Architectural overview

The Data Correlation system is a prototype developed for testing the Test Mobile Decoder module and the Correlation Engine module. As so, the system was developed in a highly modular and multi-layered architectural structure.
4.2.2 Modular decomposition

The Data Correlation system is made by modules that can be decomposed into smaller ones recursively. Modular decomposition is useful for ensuring the system’s modifiability, by ensuring that changes are made considering all the system’s elements and their connections. Figure 14: Data Correlation System High-Level diagram is an informal high-level modular decomposition diagram that shows the system’s four basic modules.

![Diagram](image)

**Figure 14: Data Correlation System High-Level diagram**

The system can be decomposed into four basic modules:

- File and Analysis Management module – this is the module responsible for managing files and analysis. It is responsible for managing data, and making requests to other modules for answering to user requests.

- Analysis Visualization Window module – this is a module for visualizing the contents of an analysis in a graphical component.

- Synchronization and Correlation Engine module – this module is responsible for the synchronization and correlation of Siemens and Test Mobile analysis.

- Data Correlation System Decoding Functionalities module – this is the module responsible for the processing and decoding of Siemens and Test Mobile trace files. It decodes L3 Messages, Hardware Resources and TEMS proprietary messages.

4.2.2.1 File Area and Analysis Management Module

The Data Correlation Binary File & Analysis Areas is a module, of the Data Correlation feature. These areas allow the user to perform a group of use cases related with loading files into the Data Correlation prototype and generating the correspondent analysis.
The main function of the Binary File & Analysis Areas is to allow the user to manage the trace files (both Siemens NE trace files and external trace files) and the related analysis.

4.2.2.1.1 Binary File Area

Each Binary File Area has a table that lists all the files that have been loaded. A details panel is present to display detailed information for a selected file. There are two Binary File Areas. One for Siemens trace files and another for external trace files. The following options are available for the Data Correlation Binary File Area:

- The user can load files into the application by using the **Open File** option and selecting a valid file to load.
- The user can save the current status of the application by using the **Save Data** option. This option saves, into the current working directory, all files and analyses currently in the application.
- The user can remove a file from the application by using the **Remove** option.
- The user can import a file and generate an analysis using the **Import** option:
  - If the analysis is from a Trace binary file, the application automatically generates the analysis.
  - If the analysis is from an Agilent .csv file, the application asks the user to type the IMSI of the analysis.
  - If the analysis is from a TEMS .FMT file, the application asks the user to type both the IMSI and the Date of the analysis.

4.2.2.1.2 Analysis Area

Each Analysis Area has a table that lists all the generated analysis. A simple details panel is present to display detailed information for a selected file. There are two Analysis Areas. One for Siemens trace analyses and another for external trace analyses. The following options are available for the Data Correlation Analysis Area:

- The user can remove an analysis from the application by using the **Remove** option.
- The user can open the Analysis Window for the current selected analysis by using the **View Analysis** option.
- If the Analysis Area is the Trace Analysis Area the user can also perform a correlation between a Trace analysis and an External analysis by using the **Correlate Analysis** option.

The Data Correlation Binary File & Analysis Area was developed around a Model-View-Controller (MVC) design pattern. The MVC is a classic design pattern often used by applications that need the ability to maintain multiple views of the same data. The MVC pattern hinges on a clean separation of objects into one of three categories — models for maintaining data, views for displaying all or a portion of the data, and controllers for handling events that affect the model or view(s).

Because of this separation, multiple views and controllers can interface with the same model. Even new types of views and controllers that never existed before can interface with a model without forcing a change in the model design. This was taken in consideration to try to easy the integration in the UCAT feature. Some changes were made to the default architectural concept to simplify its implementation.
The Model is a component that controls the data model of the module. It is responsible for storing data and answering to change requests made by the Controller. Whenever a change is made in the data model, the Model sends an event to the View indicating that changes where made in the data model and that the View should refresh the data being displayed.

The Controller is a component that creates actions to answer to the user input. Each action is processed in a separated thread to make the application more user-friendly. The actions make requests to the Model to change its data. The Controller sends events to the View indicating the status of each action.

The View is a component that corresponds to the graphical user interface. Its purpose is to receive user input that will be translated to actions by making requests to the Controller. It reads data from the Model and displays it to the user. If changes are made in the data model, the View receives events to indicate that it should refresh the data being displayed.

Each of these three components is composed by several classes as the following sections demonstrate.

4.2.2.1.3 The Model Component

The Model Component is controlled by the Data Correlation Model Controller class. This class serves as gateway into the model and all requests are handled by it. It bases its functionalities in a group of related classes that working together creates a data model for storing files and analysis.

The Data Correlation Model Controller provides functionalities for:
- Accessing the Trace binary file collection
- Accessing the Trace analysis collection
- Accessing the external data binary file collection
- Accessing the external data analysis collection
- Add a file to the respective binary file collection
- Add an analysis to the respective analysis collection
- Remove a file from the respective binary file collection
- Remove a file from the respective analysis collection
- Save and Load data from disk
Various functionalities are executed directly by the Data Correlation Model Controller, as others are executed by classes associated to the Data Correlation Model Controller. In the following diagram, the Model Component is represented, showing the Data Correlation Model Controller class and all the other classes related with it. It also shows other previous developed classes like Analysis Data from the UCAT feature and File from Java.util.

![Diagram showing the Model Component class structure]

**Figure 16: The Model Component class diagram**

### 4.2.2.1.4 The Controller Component

The Controller Component is controlled by the Data Correlation Action Controller class. This class serves as gateway into the controller component and all requests are handled by it. It bases its functionalities in group of related classes that working together creates a multithread central processing component for handling user events received by the View Component and making the necessary change requests to the Model Component.

The Data Correlation Action Controller provides functionalities for:
- Add a file to the Model Component
- Correlate two analysis
- Generate an analysis from a Data Correlation File and add it to the Model Component
- Load Data from disk into the Model Component
- Remove a Data Correlation File from the Model Component
- Remove a Data Correlation Analysis from the Model Component
Each request received is processed in a separate thread to allow the View Component to be more responsive as the following diagram shows:

![Diagram of the Controller Component Thread diagram]

**Figure 17: The Controller Component Thread diagram**

Some functionality is executed directly by the *Data Correlation Action Controller*, as others are executed by classes associated to the *Data Correlation Action Controller*.

In the following diagram, the Controller Component is represented, showing the *Data Correlation Action Controller* class and all the other classes related with it.

![Diagram of the Controller Component class diagram]

**Figure 18: The Controller Component class diagram**
4.2.2.1.5 The View Component

The View Component is the Graphical User Interface. Being the GUI it has as main purpose is to display the data contained in the Model Component and receive user requests and transmitting such requests to the Controller Component.

The View Component receives events from the Model Component whenever changes where performed in the data, and also receives events from the Controller Component whenever the status of a thread changes.

In the following diagram, the View Component is represented, showing the Data Correlation Window Frame class and all the other classes related with it.

![Class Diagram]

Figure 19: The View Component class diagram

4.2.2.2 Analysis Visualization Window module

The Analysis Visualization Window is a module of the Data Correlation system. It is responsible for displaying the contents of the analyses in a graphical interface. The analysis messages are presented in a Messages List Frame. There is also a Message Details Frame for displaying the details of a selected message.
4.2.2.3 Synchronization and Correlation Engine module

The Synchronization and Correlation engine is a module of the Data Correlation system. It is responsible for synchronization and correlation of Siemens analysis and Test Mobile analysis. The module is composed of three elements:

The Synchronization and Correlation Engine component is responsible for performing the synchronization and the correlation. The Synchronization Message is a wrapper required for extending the basic functionalities of the Analysis’ messages and the Data Wrapper is a component for wrapping the analysis’ message in Message Wrappers.
4.2.2.4 Decoding Functionalities module

The Decoding Functionalities is a module of the Data Correlation system. It is responsible for decoding files into analysis. The module is composed by 6 elements and reuses 2 components from UCAT:

![Diagram of Decoding Functionalities class diagram](image)

Figure 22: Decoding Functionalities class diagram

This architecture was designed similar to the UCAT Decoder’s architecture. This similarity was adopted in order to simplify the future integration of the Test Mobile File Decoder into the UCAT feature. Each element is responsible for a specific task in the process of decoding files into analyses. The elements and components’ responsibilities are listed in the Table 11: Decoding Functionalities Elements and Components.

<table>
<thead>
<tr>
<th>Element</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing</td>
<td>Acts as an interface to the Decoding Functionalities module. Receives outside requests and returns decoded analysis in the form of an analysis. For Test Mobile trace files it calls the Test Mobile File Decoder element, while for Siemens trace files it calls the UCAT’s File Decoder component.</td>
</tr>
<tr>
<td>Test Mobile File Decoder</td>
<td>This element is responsible for handling Test Mobile IMSI trace files. It opens the files, breaks them into individual messages and then call for each message the necessary decoder. For L3 Messages and Hardware Resource Messages it calls the UCAT’s Message Decoder component, while for TEMS Proprietary Messages it calls the TEMS Proprietary Message Decoder.</td>
</tr>
<tr>
<td>Element</td>
<td>Responsibility</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TEMS Proprietary Message Decoder</td>
<td>This element is responsible for decoding individual TEMS Proprietary Messages into TEMS Proprietary Message objects. If the message is from the version 2 family it calls the TEMS Version 2 Decoder element, while if the message is from the version 3 family it calls the TEMS Version 3 Decoder.</td>
</tr>
<tr>
<td>TEMS Version 2 Decoder</td>
<td>Responsible for decoding TEMS Proprietary Messages from the version 2 family.</td>
</tr>
<tr>
<td>TEMS Version 3 Decoder</td>
<td>Responsible for decoding TEMS Proprietary Messages from the version 3 family.</td>
</tr>
<tr>
<td>Component</td>
<td>Responsibility</td>
</tr>
<tr>
<td>UCAT’s File Decoder</td>
<td>The UCAT’s File Decoder is a component of the UCAT feature. It is capable of decoding Siemens IMSI trace files into analysis.</td>
</tr>
<tr>
<td>UCAT’s Message Decoder</td>
<td>The UCAT’s Message Decoder is a component of the UCAT feature. Its purpose is to decode individual messages in Asn.1 format. It is capable of decoding L3 Messages and Hardware Resources Messages.</td>
</tr>
</tbody>
</table>
5 Development

This chapter narrates what happened during the development of the Data Correlation prototype. The development events – important decisions, problems faced, difficulties found and projects breakthroughs – are narrated by chronological order. The final developed prototype is introduced, pointing out its Graphical User Interfaces and the Synchronization and Correlation module that implements the Correlation Mechanism.

5.1 Development environment

Besides the technologies and tools described in section 3.5 (Project’s technological choices) the following technologies, tools, and equipment were used during the internship:

- Microsoft Visio 2003 and Dia for creating UML diagrams.
- Microsoft Project for project planning.
- IBM Clearcase for source backup.
- Editplus 2 for ASCII files visualization and editing.
- Computer was a PC PIII 700 MHz with 384 Mb RAM.

5.2 Development report

This section accounts the most important events that happened during the Data Correlation internship. It briefly describes the major problems faced, and how they were resolved or bypassed.

The first problem emerged almost immediately at the beginning of the project. The Test Mobile technology belonged to direct competitors and little information/support was available. This caused a great difficulty in assessing and measuring the necessary effort to each task. Also, it was unknown what could or couldn’t be done. It was necessary to continually explore the Test Mobile technology and adjust the project whenever found necessary. In order to deal with the continuous changes to the project, it was chosen to develop the project as a working prototype, in which approaches and solutions could be studied and classified as feasible or impracticable. Since the module was to be latter integrated in the RNA, as an optional module for the UCAT feature, special care should be taken, during developments, to create a working prototype that, not only should have a similar design to UCAT but that should, whenever possible, reuse UCAT’s components and classes.

Following an initial analysis of the requirements, it was clear that a mechanism to import information for external Test Mobile files (also called logs) should be developed. Given that TEMS Investigation and Agilent are the key players in Test Mobile technology, it was decided that the working prototype would support both of them. TEMS and Agilent files are available in two formats:

- A binary format: .log files for TEMS Investigation and .SD5 files for Agilent.
• An ASCII export format: .FMT files for TEMS Investigation and .csv files for Agilent.

The ASCII files are exports of the binary files, so both the binary and the ASCII formats contain the same information, however the binary files were more user-friendly and take considerably less space in storage than the ASCII files. On the other hand, since the binary formats were unknown, it would be significantly more difficult to decode the binary files than their ASCII exports. The RNA should be able to analyse all the information contained in the Test Mobile files. This included L3 messages and any Test Mobile proprietary messages. Test Mobile proprietary messages are internal messages registered by the Test Mobile that contain measurements.

Initial attempts to decode the TEMS binary formats proved to be in vain, because as it was found latter, after spending an entire week of work, although the binary file’s structure is decipherable, the individual messages are encoded and no public decoder was available. Erickson (TEMS Investigation) was contacted, but it refused permission to use the binary formats, but allowed the use of the ASCII exports. For this reason the TEMS Investigation decoder works with ASCII exports of TEMS Investigation binary files, and not with the binary files.

Likewise the Agilent binary proved to also undecipherable, but a decoder was available. Initial contacts with Agilent revealed that they refused the use of the binary formats, but fortunately they also allowed the use of ASCII exports. For this reason the Agilent decoder works with ASCII exports of Agilent binary files, and not with the binary files.

The RNA should allow importing Trace files and Test Mobile files from a same IMSI into an associated analysis. The analysis would then provide correlated information from both sources. When correlating messages it must take into consideration the following points:

• There is a time inconsistency between the clock times of the messages of the Test Mobile and those of the Radio Network. This is caused mainly because of two issues:
  o Internal clocks of the User Equipment and of the Radio Network aren’t synchronized;
  o Propagation times must be taken into consideration.

• Messages can be missing, so extra attention must be taken when correlating messages.

• Some of the messages are transmitted asynchronously, so messages can appear in the logs in the wrong order.

• Also, if a message is present in both logs, it should only appear once in the correlated analysis.

Such precise requirements for the Synchronization and Correlation mechanism created the necessity of building a specific algorithm for solving this very particular problem. The algorithm was built from scratch and works by steps; each steps building upon the previous until the final result is achieved (see 5.4 The Trace Correlation Mechanism for further details).

The Data Correlation system was developed by using a Test Driven Development (TDD) approach whenever possible. Using TFF helped to improve the overall system quality, and the written unit tests are useful when system modifications were made.
5.3 User Interfaces

5.3.1 Data Correlation Binary File & Analysis Area Window

The Data Correlation Binary File & Analysis Areas are a sub-module, of the Data Correlation feature. These areas allow the user to perform a group of use cases related with loading files into the Data Correlation prototype and generating the correspondent analysis.

The main function of the Binary File & Analysis Areas is to allow the user to manage the trace files (both Siemens NE trace files and external trace files) and the related analysis.

![Figure 23: File and Analysis Main Window GUI](image)

5.3.1.1 Menu Bar

The Menu Bar is a simple JMenuBar that holds basic user operations. The possible operations are:

- Load File – The user can load files into the application by using the Open File option and selecting a valid file to load.
- Save Data – The user can save the current status of the application by using the Save Data option. This option saves, into the current working directory, all files and analyses currently in the application.
- Close – The user can close the application by using the Close option.
5.3.1.2 **ToolBar**

The Tool Bar is a simple *JToolBar* that displays the principal user options. These options are divided in two distinct groups:

- The Fixed Toolbar Options – this group of options is fixed and is always available.
- The Variable Toolbar Options – this group of options is variable according to the primary selected panel (for example: the trace analysis panel will display different options of the trace binary file panel). For further information on the available options for each panel consult the Binary File & Analysis Panel description.

5.3.1.3 **ActionBar**

The Data Correlation *ActionBar Panel* is a simple *JPanel* that holds a *GenericTable* to show Actions.

<table>
<thead>
<tr>
<th>Status</th>
<th>Action</th>
<th>Start Time</th>
<th>Executed Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executed</td>
<td>Adding file</td>
<td>2005/07/04 14:17:48</td>
<td>2005/07/04 14:17:49</td>
<td>008</td>
</tr>
<tr>
<td>Executed</td>
<td>Adding file</td>
<td>2005/07/04 14:18:03</td>
<td>2005/07/04 14:18:03</td>
<td>549</td>
</tr>
</tbody>
</table>

In this panel the user can see the following information separated by columns:

- A Status column where the current status of an Action being performed is displayed.
- An Action column where the general description of an Action being performed is displayed.
- A Start Time column where the Action’s start time is displayed.
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

- An Executed Time column where the Action's executed time is displayed. The Executed time is the time when the Action finished, i.e. when the Action completed.
- A Message column where can be displayed messages related with an Action to the user. Usually this column is used to show error messages.

5.3.1.4 File Format Selection Panel

The File Format Selection Panel is a simple Panel that contains two JButtons. Each JButton selects one of the possible format groups. One JButton selects Trace Data and other selects External Data. A coloured border indicates the currently selected Data Format.

![Trace Data Button](image)

![External Data Button](image)

Figure 27: File Format Selection GUI

5.3.1.5 Binary File Panel and respective Details Panel

The Binary File Panel and respective Details Panel is one of the two main visualization panels. It displays a list of Files and their respective details. The Details are displayed in a separated Details Panel for better reading.

<table>
<thead>
<tr>
<th>Trace Analysis</th>
<th>Trace Binary File Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>File Format</td>
</tr>
<tr>
<td>IMSI_trace01000000_8608.log</td>
<td>Trace 3904</td>
</tr>
<tr>
<td>IMSI_trace01000000_8610.log</td>
<td>Trace 3536</td>
</tr>
<tr>
<td>IMSI_trace01000000_8612.log</td>
<td>Trace 5916</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>File Format</td>
</tr>
<tr>
<td>State</td>
</tr>
</tbody>
</table>

Figure 28: File Table and respective Details Panel GUI
When the user adds a file to the Binary File Area, it is displayed in a row of the table. For each added file the following information is displayed:

- **File Name** – the name of the imported file
- **File Format** – the format of the file. Currently the format can be:
  - Trace – for Siemens NE trace files
  - TEMS – for TEMS Investigation .FMT trace files
  - Agilent – for Agilent .csv trace files
- **File Size** – the size of the file in bytes
- **State** – the state of the file, that currently can be:
  - New – if it wasn’t been processed
  - Processed – if it as already processed at least one time
- **Analysis Available** – a check box indicating if there is an analysis available for the file.
- **In Server** – an indication if the file is in the server.
- **Server Date** – the date of when the file was stored in the server.

The Binary File Panel, and respective Details Panel, when selected add two options to the Toolbar:

- A Remove option that allows the user to remove a selected file.
- An Import option that allows the user to import a selected file.

If not file is selected neither option will be action. As soon as the user selects a file, both options will be available. This applies to both the Trace Data Binary File Area and the External Data Binary File Area. The Details Panel of the Binary File Area displays the same information as the information contained in the table but more readable.

5.3.1.6 **Analysis Panel and respective Details Panel**

The Analysis Panel and respective Details Panel is one of the two main visualization panels. It displays a list of Analysis and their respective details. The Details are displayed in a separated Details Panel for better reading. When the user adds an analysis into the Analysis Area, it is displayed in a row of the table. For each added analysis the following information is displayed:

- **IMSI** – the IMSI of the Analysis
- **Analysis Format** – the format of the Analysis that can be:
  - Trace
  - Trace + TEMS
  - Trace + Agilent
  - TEMS
  - Agilent
- Number of Files - the number of files that from which information was extracted into the analysis
- Start Date – the start date and time of the beginning of the analysis
- Stop Date – the stop date and time of the end of the analysis

![Table and Details Panel](image)

**Figure 29: Analysis Table and respective Details Panel**

The Analysis Panel, and respective Details Panel, when selected add two options to the Toolbar:
- A Remove option that allows the user to remove a selected analysis.
- An Import option that allows the user to view a selected analysis.

If not file is selected neither option will be action. As soon as the user selects a file, both options will be available.

This applies to both the Trace Data Analysis Area and the External Data Analysis Area.

In the Trace Data Analysis Area a third option is available. This is the Correlation analysis option. Also this option is only available if there is one selected analysis.

The Details Panel of the Analysis Area displays the same information as the information contained in the table but more readable.

### 5.3.1.7 Test Mobile Input Window

The Test Mobile Input Window is a simple Frame that allows users to input missing information on the Test Mobile files. This frame opens when the user imports a file. Since Test Mobiles trace files are missing some necessary information, it is up to the user to input that information. For TEMS Investigation trace files that user must input the file’s IMSI and
Data (see Figure 34: Analysis Visualization Window GUI). For Agilent files the user just has to input the file’s IMSI.

![Figure 30: Test Mobile Input Window](image.png)

The user must input an IMSI with exactly 15 numbers long or the application will show a warning message indicating that the user inserted an invalid IMSI (see Figure 31: Invalid IMSI warning message). For setting the date the user just can use the calendar application (see Figure 32: Calendar for choosing the Date) by pressing the button on the right side of the date.

![Figure 31: Invalid IMSI warning message](image.png)

The Calendar is a RNA component reused that allows the user to choose a specific data. In the Data Correlation it is used for allowing the user to choose the date of an analysis.

![Figure 32: Calendar for choosing the Date](image.png)

5.3.1.8 Correlation Window

The Correlation Window is a Frame that allows the user to correlate analysis. When the user selects a Siemens IMSI trace analysis and chooses to correlate it, the system opens this Correlation Window (see Figure 33: Correlation Window GUI). The Correlation Window is divided into three separated parts:
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

- The Siemens Analysis Details – This panel displays the information on the Siemens IMSI trace analysis the user selected, and to which he wishes to correlate a Test Mobile Analysis.

- The Test Mobile Analysis Details – This panel displays this information on the currently selected Test Mobile analysis of the Test Mobile Analysis List.

- The Test Mobile Analysis List – This is a list of all the Test Mobile Analyses available to be correlated with the Siemens IMSI trace analysis.

![Figure 33: Correlation Window GUI](image)

When the user selects a row in the Test Mobile Analysis List, that row’s details are displayed in the Test Mobile Analysis Details. Having the Test Mobile Analysis Details just bellow the Siemens Analysis Details allows the user to compare the IMSI and Date values and decide if he should correlate the analyses.

5.3.1.9 **Analysis Visualization Window**

The Analysis Visualization Window is a JFrame that allows the user to visualize the contents of a given analysis (see Figure 34: Analysis Visualization Window GUI). The analysis is displayed in term of a Messages List where are represented all messages from the analysis. The user can also see the details of any L3 Message in a tree like structure, and the details of any HW Resource Message in a table format.

The Analysis Visualization Window is divided in four areas:

53
The Menu Bar – the menu bar is where the user can access the high-level options. The user can:
  - Close the Analysis Visualization Window – by pressing the Close button in the File menu.
  - Tile the Internal Frames – by pressing the Tile button in the Window menu.
  - Cascade the Internal Frames – by pressing the Cascade button in the Window menu.

The Toolbar – the Toolbar allows the user to select any of the two internal frames.

Messages List – one of the two internal frames of the Analysis Visualization Window. It displays a list of all the messages from the analysis. The table is a reuse of the GenericTable which has built in functionalities for sorting rows (ascendant and descendent), filtering rows by values, hide and display columns, and change the columns order.

Message Details – one of the two internal frames of the Analysis Visualization Window. It displays the details of any message selected in the Messages List. The details can be either in a tree-like structure for L3 Messages (see Figure 35: Message Details displaying a tree-like structure) or in a table for Hardware Resource Messages (see Figure 36: Message Details displaying a table-like structure).

Figure 34: Analysis Visualization Window GUI
5.4 The Trace Correlation Mechanism

The Synchronization and Correlation algorithm is composed by several steps that allow to correlate two analysis from different sources (Siemens Network Elements trace files and External trace files). The algorithm is used upon two trace analyses populated with messages loaded from two different data sources – a Siemens Network Elements trace file, and a Test Mobile trace file. This algorithm synchronizes and correlates the messages into a single analysis that will hold the messages from both analyses. The algorithm is constituted by a set of constraints that must be respected, and a list of steps that must be followed. The mechanism’s constraints are:

1. No change can be performed upon the analysis holding the Siemens NE trace data (henceforward called Siemens NE Analysis).
2. All changes must be performed upon the analysis holding the Test Mobile trace data (henceforward called Test Mobile Analysis).
3. It must be taken into consideration that some messages can be missing. A missing message is a message that is only present in one trace file, but in normal conditions should be present in both trace files.
4. The correlation must generate a unique analysis holding all messages contained in both Siemens NE Analysis and Test Mobile Analysis, but:
   a. If a message is found in the Siemens NE Analysis and in the Test Mobile Analysis (i.e. a repeated message), the Test Mobile message’s should be removed to maintain message uniqueness.
5. In case there is a difference in the ordering of the messages between the Siemens NE Analysis and the Test Mobile Analysis, we will consider the Siemens NE Analysis to hold the correct ordering of the messages.

After listing the constraints for the synchronization mechanism we will consider the primary mechanism’s steps:

![Image of step evolution diagram]

**Figure 37: Correlation’s Step Evolution Diagram**

1. The first step is to synchronize and correlate the RRC messages:
   a. In order to synchronize and correlate the RRC messages it is necessary first to perform a message classification, where is determined for each message, the number of occurrences and the number of possible matches:
      i. The message’s binary representation is used for identifying identical messages
      ii. For each RRC message in the Siemens NE analysis, the number of times it is present in the Siemens NE analysis is counted. This step determines the number of occurrences of each RRC message in the Siemens NE analysis.
      iii. For each RRC message in the Test Mobile analysis, the number of time it is present in the Test Mobile analysis is counted. This step determines the number of occurrences of each RRC message in the Test Mobile analysis.
      iv. For each RRC message in Siemens NE analysis, the number of times it appears in the Test Mobile analysis is counted. This step determines the number of possible matches for every message in Siemens NE analysis.
      v. For each RRC message in Test Mobile analysis, the number of times it appears in the Siemens NE analysis is counted. This step determines the number of possible matches for Test Mobile analysis messages
      vi. Any message that hasn’t got any possible match is considered to be missing.
   b. Once the messages have been classified, the next step is to generate sequential sets:
      i. A sequential set is a group of messages that follow a certain group of constraints:
         1. Any message in a sequential set cannot be missing.
         2. The initial message of any sequential set must have only on occurrence and only one possible match.
3. For any two sequential Siemens NE messages in the sequential set, their respective matches must also be sequential; e.g. take for example the next two diagrams. The diagram on the left depicts a group of sequential Siemens messages and their Test Mobile counterparts; alternatively the diagram on the right depicts a group of non-sequential messages.

Figure 38: Sequential Messages  
Figure 39: Non-sequential Messages diagram

4. A sequential set must be at least five messages long, on each side.

ii. The sequential sets are generated by a state machine that runs over all messages on both analyses. The activities sequence is illustrated in the following diagram:

Figure 40: Anchor Generation Activity Diagram
c. Once the sequential sets have been generated and their messages synchronized and correlated, it is necessary to synchronized and correlate the remaining messages.

i. The remaining messages (not contained in the sequential sets) are gathered in sub-groups. A sub-group is delimited by either the start of the analysis, a sequential set, or the end of the analysis, e.g.:

![Sequential sets and sub-groups relations diagram.](image)

Figure 41: Sequential sets and sub-groups relations diagram.

ii. The synchronization and correlation at this point is very straightforward and is performed by running the following steps:

1. First the messages within the sub-group must be reclassified, counting the number of occurrences and possible matches, but this time only considering the messages within the sub-group.

2. For every message in the Siemens side that is not missing, find the first possible correspondent that has not been already connected, and match them.

3. Repeat the previous step until all Siemens messages that are not missing have been matched with their counterparts.

4. If a Siemens or Test Mobile message still has not a match by the end of processing the sub-group, it is considered to be missing. This could happen if there are two Siemens messages with the same counterpart – since both message have the same counterpart, only one of them will be matched and the other will be considered missing. The same applies to the Test Mobile messages.

iii. The first sub-group is synchronized and correlated backwards while all other sub-groups are synchronized and correlated forwards.

2. Once the time of each RRC Message is synchronized, we must correct the time of the remaining messages (missing RRC messages that were only found on the external Analysis and proprietary message like TEMS Proprietary message).

a. Since the time disparity is inconstant we must do a rough calculation. The proposal is to do an approximation proportional has the following tables show:
### Table 12: Times before synchronization (example)

<table>
<thead>
<tr>
<th>Trace Message Number</th>
<th>Trace Time</th>
<th>Trace Message</th>
<th>TEMS Message Number</th>
<th>TEMS Time</th>
<th>TEMS Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:15</td>
<td>RRC</td>
<td>1</td>
<td>00:05</td>
<td>RRC (missing)</td>
</tr>
<tr>
<td>2</td>
<td>00:20</td>
<td>NBAP</td>
<td>2</td>
<td>00:25</td>
<td>RRC</td>
</tr>
<tr>
<td>3</td>
<td>00:30</td>
<td>RRC</td>
<td>3</td>
<td>00:35</td>
<td>RRC (missing)</td>
</tr>
<tr>
<td>4</td>
<td>00:40</td>
<td>RRC</td>
<td>4</td>
<td>00:45</td>
<td>RRC (missing)</td>
</tr>
<tr>
<td>5</td>
<td>01:00</td>
<td>NBAP</td>
<td>5</td>
<td>00:55</td>
<td>RRC</td>
</tr>
<tr>
<td>6</td>
<td>01:15</td>
<td>NBAP</td>
<td>6</td>
<td>01:00</td>
<td>RRC (missing)</td>
</tr>
<tr>
<td>7</td>
<td>01:45</td>
<td>RRC</td>
<td>7</td>
<td>01:10</td>
<td>RRC</td>
</tr>
<tr>
<td>8</td>
<td>01:55</td>
<td>ALCAP</td>
<td>8</td>
<td>01:40</td>
<td>RRC</td>
</tr>
</tbody>
</table>

### Table 13: Times after synchronization (example)

<table>
<thead>
<tr>
<th>TEMS Message Number</th>
<th>TEMS Original Time</th>
<th>TEMS Corrected Time</th>
<th>Offset</th>
<th>TEMS Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:05</td>
<td>-00:05</td>
<td>-00:10</td>
<td>TEMS</td>
</tr>
<tr>
<td>2</td>
<td>00:25</td>
<td>00:15</td>
<td>-00:10</td>
<td>RRC</td>
</tr>
<tr>
<td>3</td>
<td>00:35</td>
<td>00:20</td>
<td>-00:15</td>
<td>TEMS</td>
</tr>
<tr>
<td>4</td>
<td>00:45</td>
<td>00:25</td>
<td>-00:20</td>
<td>TEMS</td>
</tr>
<tr>
<td>5</td>
<td>00:55</td>
<td>00:30</td>
<td>-00:25</td>
<td>RRC</td>
</tr>
<tr>
<td>6</td>
<td>01:00</td>
<td>00:375</td>
<td>-00:225</td>
<td>TEMS</td>
</tr>
<tr>
<td>7</td>
<td>01:10</td>
<td>00:40</td>
<td>-00:30</td>
<td>RRC</td>
</tr>
<tr>
<td>8</td>
<td>01:40</td>
<td>01:45</td>
<td>-00:05</td>
<td>RRC</td>
</tr>
</tbody>
</table>

As an example of how we will do this, let's take as example the Trace message numbered 1 and 3 in Table 12: Times before synchronization (example). The difference is 15 ms. On the TEMS side the same messages have numbers 2 and 5 (see Table 12: Times before synchronization (example)). Their difference is 30 ms. The proportion between TEMS and Siemens Trace is of 50% (30 ms -> 15 ms). Therefore we take that relation of 50% to correct the messages that are contained between messages 2 and 5. Between messages 2 and 3 in TEMS (see Table 12: Times before synchronization (example)), there is a disparity of 10 ms. Applying the proportion then the difference becomes 5 ms (10 * 50%). Between message 3 and 4 there is also a disparity of 10 ms, that results in a corrected disparity of 5 ms. The TEMS messages previous to the first RRC message cannot be corrected by this mechanism, because it is only capable of correcting message between two RRC messages. One possible solution is to apply the same offset applied to the first RRC message to all preceding messages. The same will be done to all TEMS messages that follow the last RRC message, but in this case the offset of the last RRC message will be applied. For each RRC message from the Siemens trace data it must be found the corresponding equivalent message in the
External trace data. For doing that we being by the idea that the same message on both sides will have the same binary representation.

5.4.1 Example based on real IMSI trace files

In order to provide a better understanding on how the Correlation Mechanism works, the following example has been prepared. The example is about a dropped call, from which are available the IMSI trace files from both the Siemens Network and the TEMS Investigation.

The Siemens Network IMSI trace file contained the RRC messages listed in Table 14: Siemens IMSI trace analysis messages contents and the TEMS Investigation IMSI trace file contained the RRC messages listed in Table 15: TEMS Investigation IMSI trace analysis messages contents.

Table 14: Siemens IMSI trace analysis messages contents

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Checksum Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2005/02/02 12:26:12.628</td>
<td>Downlink Direct Transfer</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2005/02/02 12:26:12.686</td>
<td>Uplink Direct Transfer</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>2005/02/02 12:26:13.006</td>
<td>Uplink Direct Transfer</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>2005/02/02 12:26:13.504</td>
<td>Radio Bearer Setup</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>2005/02/02 12:26:14.126</td>
<td>Measurement Report</td>
<td>E</td>
</tr>
<tr>
<td>6</td>
<td>2005/02/02 12:26:14.326</td>
<td>Radio Bearer Setup Complete</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>2005/02/02 12:26:14.346</td>
<td>Measurement Control</td>
<td>G</td>
</tr>
<tr>
<td>8</td>
<td>2005/02/02 12:26:14.624</td>
<td>Downlink Direct Transfer</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>2005/02/02 12:26:15.080</td>
<td>Measurement Control</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>2005/02/02 12:26:15.512</td>
<td>Active Set Update</td>
<td>J</td>
</tr>
<tr>
<td>11</td>
<td>2005/02/02 12:26:15.846</td>
<td>Active Set Update Complete</td>
<td>K</td>
</tr>
<tr>
<td>12</td>
<td>2005/02/02 12:26:15.846</td>
<td>Measurement Control</td>
<td>L</td>
</tr>
<tr>
<td>13</td>
<td>2005/02/02 12:26:16.166</td>
<td>Measurement Report</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>2005/02/02 12:26:16.200</td>
<td>Active Set Update</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>2005/02/02 12:26:16.246</td>
<td>Measurement Report</td>
<td>O</td>
</tr>
<tr>
<td>16</td>
<td>2005/02/02 12:26:16.526</td>
<td>Active Set Update Complete</td>
<td>P</td>
</tr>
<tr>
<td>17</td>
<td>2005/02/02 12:26:16.582</td>
<td>Measurement Control</td>
<td>Q</td>
</tr>
<tr>
<td>18</td>
<td>2005/02/02 12:26:19.794</td>
<td>Transport Channel Reconfiguration</td>
<td>R</td>
</tr>
<tr>
<td>19</td>
<td>2005/02/02 12:26:20.592</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>S</td>
</tr>
</tbody>
</table>

The Checksum Code is a unique code that is used in this example for make easier to find identical messages; i.e. two identical messages will share the same Checksum Code.
<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Checksum Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2005/02/02 12:26:23.074</td>
<td>Transport Channel Reconfiguration</td>
<td>T</td>
</tr>
<tr>
<td>21</td>
<td>2005/02/02 12:26:23.872</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>U</td>
</tr>
<tr>
<td>22</td>
<td>2005/02/02 12:26:23.896</td>
<td>Measurement Control</td>
<td>V</td>
</tr>
<tr>
<td>23</td>
<td>2005/02/02 12:26:28.864</td>
<td>Transport Channel Reconfiguration</td>
<td>W</td>
</tr>
</tbody>
</table>

Table 15: TEMS Investigation IMSI trace analysis messages contents

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Checksum Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2005/02/02 12:26:08.090</td>
<td>RRC Connection Request</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>2005/02/02 12:26:08.650</td>
<td>RRC Connection Setup</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>2005/02/02 12:26:08.760</td>
<td>RRC Connection Setup Complete</td>
<td>Z</td>
</tr>
<tr>
<td>4</td>
<td>2005/02/02 12:26:09.080</td>
<td>Initial Direct Transfer</td>
<td>AA</td>
</tr>
<tr>
<td>5</td>
<td>2005/02/02 12:26:09.250</td>
<td>Measurement Control</td>
<td>AB</td>
</tr>
<tr>
<td>6</td>
<td>2005/02/02 12:26:09.620</td>
<td>Measurement Control</td>
<td>AC</td>
</tr>
<tr>
<td>7</td>
<td>2005/02/02 12:26:09.920</td>
<td>Security Mode Command</td>
<td>AD</td>
</tr>
<tr>
<td>8</td>
<td>2005/02/02 12:26:09.950</td>
<td>Security Mode Complete</td>
<td>AE</td>
</tr>
<tr>
<td>9</td>
<td>2005/02/02 12:26:10.260</td>
<td>Uplink Direct Transfer</td>
<td>B</td>
</tr>
<tr>
<td>10</td>
<td>2005/02/02 12:26:10.700</td>
<td>Downlink Direct Transfer</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>2005/02/02 12:26:10.710</td>
<td>Uplink Direct Transfer</td>
<td>C</td>
</tr>
<tr>
<td>12</td>
<td>2005/02/02 12:26:11.740</td>
<td>Measurement Report</td>
<td>E</td>
</tr>
<tr>
<td>13</td>
<td>2005/02/02 12:26:11.970</td>
<td>Radio Bearer Setup</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>2005/02/02 12:26:12.090</td>
<td>Radio Bearer Setup Complete</td>
<td>F</td>
</tr>
<tr>
<td>15</td>
<td>2005/02/02 12:26:12.500</td>
<td>Measurement Control</td>
<td>G</td>
</tr>
<tr>
<td>16</td>
<td>2005/02/02 12:26:12.740</td>
<td>Downlink Direct Transfer</td>
<td>H</td>
</tr>
<tr>
<td>17</td>
<td>2005/02/02 12:26:13.180</td>
<td>Measurement Control</td>
<td>I</td>
</tr>
<tr>
<td>18</td>
<td>2005/02/02 12:26:13.580</td>
<td>Active Set Update</td>
<td>J</td>
</tr>
<tr>
<td>19</td>
<td>2005/02/02 12:26:13.590</td>
<td>Active Set Update Complete</td>
<td>K</td>
</tr>
<tr>
<td>20</td>
<td>2005/02/02 12:26:13.810</td>
<td>Measurement Report</td>
<td>N</td>
</tr>
<tr>
<td>21</td>
<td>2005/02/02 12:26:13.840</td>
<td>Measurement Report</td>
<td>L</td>
</tr>
<tr>
<td>22</td>
<td>2005/02/02 12:26:13.960</td>
<td>Measurement Control</td>
<td>O</td>
</tr>
</tbody>
</table>

The Checksum Code is a unique code that is used in this example for make easier to find identical messages; i.e. two identical messages will share the same Checksum Code.
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS network elements and Test Mobiles

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Checksum Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>2005/02/02 12:26:14.190</td>
<td>Active Set Update</td>
<td>M</td>
</tr>
<tr>
<td>24</td>
<td>2005/02/02 12:26:14.200</td>
<td>Active Set Update Complete</td>
<td>P</td>
</tr>
<tr>
<td>25</td>
<td>2005/02/02 12:26:14.700</td>
<td>Measurement Control</td>
<td>Q</td>
</tr>
<tr>
<td>26</td>
<td>2005/02/02 12:26:18.260</td>
<td>Transport Channel Reconfiguration</td>
<td>R</td>
</tr>
<tr>
<td>27</td>
<td>2005/02/02 12:26:18.330</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>S</td>
</tr>
<tr>
<td>28</td>
<td>2005/02/02 12:26:21.250</td>
<td>Transport Channel Reconfiguration</td>
<td>T</td>
</tr>
<tr>
<td>29</td>
<td>2005/02/02 12:26:21.560</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>U</td>
</tr>
<tr>
<td>30</td>
<td>2005/02/02 12:26:21.950</td>
<td>Measurement Control</td>
<td>V</td>
</tr>
</tbody>
</table>

A quick verification of the contents of the files (specially the Checksum Code) will clearly reveal that the files are quite different, but that share many common messages, that can be or not in the same order.

5.4.1.1 Correlating the analysis

The first step is to classify the messages in term of the number of possible matches and in term of the number of occurrences of each message in the same analysis. After classifying the analysis the following results were obtained:

- Siemens’ messages 1 to 22 have only one occurrence and only one possible match.
- Siemens’ message 23 is missing because it doesn’t have any possible match in the TEMS Investigation analysis.
- TEMS Investigation’s messages 1 to 8 are missing because they don’t have any possible match in the Siemens analysis.
- TEMS Investigation’s messages 9 to 30 have only one occurrence and only one possible match.

The second step is to generate sequential sets. Two sequential sets are present in the example analysis and are declared in Table 16: Sequential Sets from example.

<table>
<thead>
<tr>
<th>Table 16: Sequential Sets from example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Set Number</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

The third step is to generate sub-groups with the remaining messages. Messages 1 to 8 from the TEMS Investigation analysis are missing messages, and so they will not be included in the sub-groups. Two sub-groups are present and are declared in Table 17: Sub-groups from example.
Table 17: Sub-groups from example

<table>
<thead>
<tr>
<th>Sub-group Number</th>
<th>Siemens Messages Numbers</th>
<th>TEMS Investigation Numbers</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 to 5</td>
<td>9 to 13</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12 to 15</td>
<td>20 to 23</td>
<td></td>
</tr>
</tbody>
</table>

Once the sub-groups have been generated they must be synchronized. For synchronizing sub-groups one must start by performing a new classification of the messages within each sub-group. In this specific example all messages have only one occurrence and only one possible match. The mechanism synchronizes the first subgroup backwards and all the others forwards. The synchronization of the first sub-group is displayed in the Figure 42: Sub-group 1 synchronization and the synchronization of the second sub-group is displayed in the Figure 43: Sub-group 2 synchronization.

![Figure 42: Sub-group 1 synchronization](image)

![Figure 43: Sub-group 2 synchronization](image)

After synchronizing the sub-groups the final step is to adjust the timestamps of the TEMS Investigation messages that haven’t been already synchronized in the precious steps. The only TEMS Messages that haven’t been synchronized are messages 1 to 8. Since message 9 changed is timestamp in + 2426 milliseconds, the remaining messages will undergo the same correction. After finishing correlating the analysis, both analyses are merged together, and the repeated messages are removed. The final result is present in

Table 18: Resulting correlated analysis

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Present in</th>
<th>Checksum Code⁵</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2005/02/02 12:26:10.516</td>
<td>RRC Connection Request</td>
<td>TEMS only</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>2005/02/02 12:26:11.076</td>
<td>RRC Connection Setup</td>
<td>TEMS only</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>2005/02/02 12:26:11.186</td>
<td>RRC Connection Complete</td>
<td>TEMS only</td>
<td>Z</td>
</tr>
</tbody>
</table>

⁵ The Checksum Code is a unique code that is used in this example for make easier to find identical messages; i.e. two identical messages will share the same Checksum Code.
<table>
<thead>
<tr>
<th>Message Number</th>
<th>Date</th>
<th>Message Type</th>
<th>Present in</th>
<th>Checksum Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2005/02/02 12:26:11.506</td>
<td>Initial Direct Transfer</td>
<td>TEMS only</td>
<td>AA</td>
</tr>
<tr>
<td>5</td>
<td>2005/02/02 12:26:11.676</td>
<td>Measurement Control</td>
<td>TEMS only</td>
<td>AB</td>
</tr>
<tr>
<td>6</td>
<td>2005/02/02 12:26:13.046</td>
<td>Measurement Control</td>
<td>TEMS only</td>
<td>AC</td>
</tr>
<tr>
<td>7</td>
<td>2005/02/02 12:26:12.346</td>
<td>Security Mode Command</td>
<td>TEMS only</td>
<td>AD</td>
</tr>
<tr>
<td>8</td>
<td>2005/02/02 12:26:12.376</td>
<td>Security Mode Complete</td>
<td>TEMS only</td>
<td>AE</td>
</tr>
<tr>
<td>9</td>
<td>2005/02/02 12:26:12.628</td>
<td>Downlink Direct Transfer</td>
<td>Both</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>2005/02/02 12:26:12.686</td>
<td>Uplink Direct Transfer</td>
<td>Both</td>
<td>B</td>
</tr>
<tr>
<td>11</td>
<td>2005/02/02 12:26:13.006</td>
<td>Uplink Direct Transfer</td>
<td>Both</td>
<td>C</td>
</tr>
<tr>
<td>12</td>
<td>2005/02/02 12:26:13.504</td>
<td>Radio Bearer Setup</td>
<td>Both</td>
<td>D</td>
</tr>
<tr>
<td>13</td>
<td>2005/02/02 12:26:14.126</td>
<td>Measurement Report</td>
<td>Both</td>
<td>E</td>
</tr>
<tr>
<td>14</td>
<td>2005/02/02 12:26:14.326</td>
<td>Radio Bearer Setup Complete</td>
<td>Both</td>
<td>F</td>
</tr>
<tr>
<td>15</td>
<td>2005/02/02 12:26:14.346</td>
<td>Measurement Control</td>
<td>Both</td>
<td>G</td>
</tr>
<tr>
<td>16</td>
<td>2005/02/02 12:26:14.624</td>
<td>Downlink Direct Transfer</td>
<td>Both</td>
<td>H</td>
</tr>
<tr>
<td>17</td>
<td>2005/02/02 12:26:15.080</td>
<td>Measurement Control</td>
<td>Both</td>
<td>I</td>
</tr>
<tr>
<td>18</td>
<td>2005/02/02 12:26:15.512</td>
<td>Active Set Update</td>
<td>Both</td>
<td>J</td>
</tr>
<tr>
<td>19</td>
<td>2005/02/02 12:26:15.846</td>
<td>Active Set Update Complete</td>
<td>Both</td>
<td>K</td>
</tr>
<tr>
<td>20</td>
<td>2005/02/02 12:26:15.846</td>
<td>Measurement Control</td>
<td>Both</td>
<td>L</td>
</tr>
<tr>
<td>21</td>
<td>2005/02/02 12:26:16.166</td>
<td>Measurement Report</td>
<td>Both</td>
<td>M</td>
</tr>
<tr>
<td>22</td>
<td>2005/02/02 12:26:16.200</td>
<td>Active Set Update</td>
<td>Both</td>
<td>N</td>
</tr>
<tr>
<td>23</td>
<td>2005/02/02 12:26:16.246</td>
<td>Measurement Report</td>
<td>Both</td>
<td>O</td>
</tr>
<tr>
<td>24</td>
<td>2005/02/02 12:26:16.526</td>
<td>Active Set Update Complete</td>
<td>Both</td>
<td>P</td>
</tr>
<tr>
<td>25</td>
<td>2005/02/02 12:26:16.582</td>
<td>Measurement Control</td>
<td>Both</td>
<td>Q</td>
</tr>
<tr>
<td>26</td>
<td>2005/02/02 12:26:19.794</td>
<td>Transport Channel Reconfiguration</td>
<td>Both</td>
<td>R</td>
</tr>
<tr>
<td>27</td>
<td>2005/02/02 12:26:20.592</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>Both</td>
<td>S</td>
</tr>
<tr>
<td>28</td>
<td>2005/02/02 12:26:23.074</td>
<td>Transport Channel Reconfiguration</td>
<td>Both</td>
<td>T</td>
</tr>
<tr>
<td>29</td>
<td>2005/02/02 12:26:23.872</td>
<td>Transport Channel Reconfiguration Complete</td>
<td>Both</td>
<td>U</td>
</tr>
<tr>
<td>30</td>
<td>2005/02/02 12:26:23.896</td>
<td>Measurement Control</td>
<td>Both</td>
<td>V</td>
</tr>
<tr>
<td>31</td>
<td>2005/02/02 12:26:28.864</td>
<td>Transport Channel Reconfiguration</td>
<td>Siemens only</td>
<td>W</td>
</tr>
</tbody>
</table>
Looking at the correlated analysis there is all the information that was contained in the two separated analysis. The Siemens analysis has been complemented with the missing messages present in the TEMS Investigation analysis; and the TEMS Investigation messages’ timestamps have been corrected to match the Siemens messages’ timestamps.

The user looking at the correlated analysis can notice something very important: the last message (number 31), was only present in the Siemens analysis. Since the Transport Channel Reconfiguration is a message sent from the network to the UE, it can be stated that the message after being sent from the network never reached the UE and the call dropped. The location of the problem has been identified, and now is up to the user to find the causes and correct them.
6 Results and future work

6.1 Data Correlation system results

The developed Data Correlation system accomplished what was specified at the beginning of the project. The system was validated against the functional requirements, and respected the non-functional requirements. It proved that correlating IMSI trace information into a seamless analysis is not only possible, but is also a valuable tool in the domain of radio network optimisation and troubleshooting.

The system as reached a mature design, where the modules are loosely coupled to improve modifiability and reusability. The elements, which can be used to process a considerable amount of data, have been redesigned several times to improve performance, testability and usability. The system has identified, and reused, many of UCAT’s components, including the decoding components and the data model components. The system can be used as base for developing and testing new correlation functionalities and new decoders, reducing this way the risks and costs of integrating new components.

Currently the TEMS Proprietary Message Decoder decodes the following messages:

- From version 2:
  - GSM Dedicated Mode Report
  - GSM Idle Mode Report
  - DL Inner Loop Power Control Report
  - Random Access Report
  - RLC Statistics Report
  - Uplink Signal Strength and Power Control Report
  - RAT State Report
  - RRC State Report
  - UE Identify Report
  - Downlink Transport Channel Report
  - UE Intra-Freq Report
  - Cell Reselection Measurement Report

- From version 3:
  - DRX Mode, dual UE
  - Power Control Compressed Mode
  - AGC
  - Transport Channel Uplink
  - Common Physical Channels Downlink 4.x
  - Physical Channel Uplink
  - PRACH
  - Active Set
  - Neighbour Set
  - BLER
  - Uplink Mac Logical Channel
  - Downlink MAR Logical Channel
  - RRC State
  - Cell Id
  - UL RLC AM Statistics Extended
  - DL RLC AM Statistics Extended
Specification and development of a module for correlation of UMTS trace from different sources: SIEMENS
network elements and Test Mobiles

- List Search 5.x
- RACH
- Transport Channel Downlink 4.x
- Common Physical Channels Downlink 4.x

The decoder was developed having in mind the necessity of reverse engineering new messages and values. A logging mechanism reports all messages that aren’t according to the specification, so it helps developers to improve the decoders.

6.2 Future Development

Initial feedback from clients was very positive and indicated that the Data Correlation system has good perspectives of becoming an important and useful feature of radio network optimisation and troubleshooting. According to a costumer, the Data Correlation system can save up to 90% of the time it takes to identify some types of errors. This opens the Data Correlation system to future development and improvement. The first step is to integrate the Data Correlation system into the UCAT feature and make it commercial. The correlation mechanism must be further studied and optimised so that extra functionalities can be added.

The Data Correlation system decoders will require constant update as new version of TEMS Investigation and Agilent are released. Also the Test Mobile File Decoder will be extended to support further Test Mobile solutions like Nemo, Nec, or any other solution with significant commercial weight.

It as also been thought to develop a similar system for GSM, but since most of the Test Mobile technology is used for bringing up new radio networks, and since most of the new investments are focused in WCDMA systems, it was considered, for the time being, preferable to concentrate resources in further developing the Data Correlation system for UMTS and ignore GMS.
7 Conclusion

The Data Correlation internship was a project with a few major objectives and many system requirements. The internship objectives were accomplished, and the system’s requirements were respected throughout development until being fully validated in the finishing internship’s tasks.

In the system requirements, the Data Correlation system was required to have a module for managing Siemens IMSI trace files and Test Mobile IMSI trace files, and respective analyses. For that purpose, the File management module was developed. This module was made graphically similar to UCAT and shared, by reuse, some of the graphical components for a similar look and feel. The module answered to all the specified requirements and proved to be user-friendly, mainly because it had a similar look to that of UCAT.

The Data Correlation system was also required to have a module for visualizing the contents an analysis, in which the messages would appear in a list, and the user could select a message for visualizing its contents. For this purpose, the Analysis Visualization Window module was developed. This module was design to be similar to the UCAT’s Analysis Window, and so facilitate the reutilization of some of its components and code. The module answered to all the specified requirements, being very simple to use, but at the same time, offering many analysis options such as message filtering, message details visualization and message sorting (by any direction and column of the messages list table).

The Data Correlation internship had two major objectives: to develop a decoder for Test Mobile IMSI trace files and a synchronization and correlation mechanism for creating seamless analyses from Siemens and Test Mobile IMSI trace information. The two objectives were mapped into system requirements, but the success, or failure, of the Data Correlation internship would be dictated by the accomplishment of these objectives.

To accomplish the decoding objective, much time was invested in developing a Test Mobile decoder module, capable of decoding TEMS Investigation IMSI trace files and Agilent IMSI trace files. TEMS Investigation and Agilent were chosen because they are the most used solutions in the Test Mobile market. By supporting the two primary solutions, the Data Correlation guarantees the support of a substantial portion of clients. For these purpose, the Test Mobile Decoder module was developed. This module was capable of decoding TEMS Investigation’s .FMT files and Agilent’s .csv files. The two supported file types (.FMT and .csv) are ASCII exports of binary files, but due to legal requirements it was the only possibility. The Test Mobile Decoder module is capable of decoding TEMS Investigation files from version 2.X (any file from the 2\textsuperscript{nd} generation family) and version 3.X (any file from the 3\textsuperscript{rd} generation family), and Agilent files from version 6.0.

To accomplish the correlation objective, which was the primary objective of the internship, the Synchronization and Correlation Engine module was developed. This module was developed around the Synchronization and Correlation mechanism. The mechanism is capable of correctly classifying, synchronizing and correlating IMSI trace information from different sources, into a seamless analysis. This module successfully answered to the system requirements, and was made robust so to manage many different possibilities, and keep on working.
Many problem faced during development were due to the fact that the Test Mobile technology, involved in the project, was unknown at start, and much time was spent in obtaining basic comprehension on the technology. It was necessary to reverse engineer the Test Mobile files, because no file format was available. Initially it was attempted to decode the files in binary format, but after much effort it was discovered that the files were encrypted, so an attempt to get the files formats from the owners was made. Due to legal requirements that wasn’t possible so there was no other solution as to use the ASCII exports.

Initial feedback from costumers on the Data Correlation system was excellent, and indicated that the system is on the right direction. They stated that the Data Correlation system could save up to 90% of the time they take to find discrepancies and errors. By being so helpful and at the same time being an original concept, the Data Correlation system is expected to have a significant commercial success in an extremely competitive market. To become a commercial product, the Data Correlation system will be integrated in the UCAT feature, in the next version of RNA.

On a personal level, the internship proved to be a unique experience. Coming from a University environment, were I was used to develop some small to medium projects, and then facing alone such a task, proved to be an excellent introduction to the business world. The RNA team is a group of excellent professionals that helped me integrate into Siemens quickly, and become productive very fast. Although it was a great experience, it wasn’t without difficulties. One of the major difficulties I faced, beside the fact I was working with Test Mobile technology from which I didn’t have any previous knowledge, was the development machine: a PC Pentium III 700 MHz with 384 MB of RAM, which reduced my productively and seriously putting my patience to test.

In the end, I considered the internship to have been excellent in terms of personal and professional development, and I feel fully integrated in the team and ready to start integrating the Data Correlation system in the UCAT feature.
8 Biography and References


Sommerville, I., “Ingeniería de software”, 6va Edición, Addison Wesley, 2002


Parys, J., “Memorandum: Introduction to UMTS”, Alcatel Telecom, 2000


9 Acronyms

ALCAP – access link control application part
AMPS – Advanced Mobile Phone System
API – Application Programming Interface
ATM – Asynchronous Transfer Mode
BTS – Base Transmitting Station
CDMA – Code Division Multiple Access
CLI – Command Line Interface
CN – Core Network
CRNC – Controlling Radio Network Controller
CS – Circuit Switched
DMZ – Demilitarized zone
DRNC – Drift Radio Network Controller
ETSI – European Telecommunications Standards Institute
FEC – Forward Error Correction
FER – Frame Error Rate
FEUP – Faculdade de Engenharia da Universidade do Porto
GPS – Global Positioning System
GSM – Global System for Mobile Communication
GUI – Graphical User Interface
HW – Hardware
IMSI – International Mobile Station Identity
ISDN – Integrated Services Digital Network
IT – Information technology
JDT – Java Development Tools
MVC – Model View Controller
NBAP – Node B Application Part
NE – Network Element
NMP – Network Management Products
O&M – Operation and Maintenance
OG – Operating Group
OO – Object Oriented
PDE – Plug-in Development Environment
PS – Packet Switched
QPSK – Quadrature Phase Shift Keying
RAN – Radio Access Network
RANAP – Radio Access Network Application Part
RC – Radio Commander
RNA – Radio Network Analysis
RNC – Radio Network Controller
RNS – Radio Network System
RNSAP – Radio Network Sublayer Application Part
RRC – Radio Resource Control
RRM – Radio Resource Management
SRNS – Serving Radio Network System
TDD – Test Driven Development
TDMA – Time Division Multiple Access
UCAT – UMTS Call Trace
UE – User Equipment
UML – Unified Modelling Language
UMTS – Universal Mobile Telecommunications Service
VM – Virtual Machine
WCDMA – Wideband Code Division Multiple Access