

**Design and Application of an Electronic Logbook
for Space System Integration
and Test Operations**

Alicia T. Kavelaars

Stanford Linear Accelerator Center
Stanford University
Stanford, CA 94309

SLAC-Report-816

Prepared for the Department of Energy
under contract number DE-AC02-76SF00515

Printed in the United States of America. Available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

This document, and the material and data contained therein, was developed under sponsorship of the United States Government. Neither the United States nor the Department of Energy, nor the Leland Stanford Junior University, nor their employees, nor their respective contractors, subcontractors, or their employees, makes an warranty, express or implied, or assumes any liability of responsibility for accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use will not infringe privately owned rights. Mention of any product, its manufacturer, or suppliers shall not, nor is it intended to, imply approval, disapproval, or fitness of any particular use. A royalty-free, nonexclusive right to use and disseminate same of whatsoever, is expressly reserved to the United States and the University.

DESIGN AND APPLICATION
OF AN
ELECTRONIC LOGBOOK
FOR SPACE SYSTEM
INTEGRATION AND TEST OPERATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Alicia T. Kavelaars

April 2006

© Copyright by Alicia T. Kavelaars 2006
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor Elliott Bloom, Principal Advisor

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor J. David Powell

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Professor Robert Twiggs

Approved for the University Committee on Graduate Studies.

Abstract

In the highly technological aerospace world paper is still widely used to document space system integration and test (I&T) operations. E-Logbook is a new technology designed to substitute the most commonly used paper logbooks in space system I&T, such as the connector mate/demate logbook, the flight hardware and flight software component installation logbook, the material mix record logbook and the electronic ground support equipment validation logbook. It also includes new logbook concepts, such as the shift logbook, which optimizes management oversight and the shift hand-over process, and the configuration logbook, which instantly reports on the global I&T state of the space system before major test events or project reviews. The design of E-Logbook focuses not only on a reliable and efficient relational database, but also on an ergonomic human-computer interactive (HCI) system that can help reduce human error and improve I&T management and oversight overall. E-Logbook has been used for the I&T operation of the Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) at the Stanford Linear Accelerator Center (SLAC). More than 41,000 records have been created for the different I&T logbooks, with no data having been corrupted or critically lost. 94% of the operators and 100% of the management exposed to E-Logbook prefer it to paper logbooks and recommend its use in the aerospace industry.

For Christian.

For my parents, Paloma and Erik.

For my grandfather, Manuel.

In memory of Willem Wamsteker.

Acknowledgements

When I started to write this dissertation, I was told that the most difficult writing would take place in the acknowledgements section. I thought this not to be true, since writing this section would mean that I would be done with my thesis. But that person was right. The truth is that when you think about how to thank the people who helped you finish a Ph.D. project, you don't know where to start, or where to end.

So many people helped me in succeeding at this endeavor, both professionally and personally. First and foremost I have to thank Prof. Elliott Bloom, my advisor, who gave me the incredible opportunity of working for GLAST and SLAC, provided immense support, and always believed in me.

I would also like to specially thank Prof. Dave Powell, an inspiring figure who helped me maintain my priorities, and was always patient, wise and cheerful even during the most difficult times.

Thank you also to Robert Twiggs, for providing students with great opportunities to learn about how to build space systems, and to the rest of the Professors at Stanford from whom I learned so much. Thank you also to the Aeronautics and Astronautics Department, especially Lynn Kaiser, and to Stanford University, for the opportunity to come to the United States with a Physics degree to become an aerospace engineer.

This research would not have been possible without the tremendous support of the GLAST Integration and Test team. I would like to thank Kenneth Fouts for his support, and all the staff of the LAT I&T facility at SLAC, who were patient with my questions and enthusiastic about my research, and in the end, made it succeed thanks to their invaluable input: Brian Horwitz, Brian Grist, Eliazar Ortiz, Leo Manger, Tom Nieland, Dave Kiehl, Mark Mollini, Doug Bartholomiew, John Marshall, Phouc Hoang, Albert Nguyen, Roger Williams and Larry Wai. Kelly Burlingham, the quality assurance

engineer who sat with me in every verification and validation test of E-Logbook, was incredibly valuable in making E-Logbook meet high standards.

I would also like to thank the LAT I&T Online team: Lester Miller, Jim Panetta, Ric Claus and specially Selim Tuvi, an incredible programmer whose ideas made E-Logbook better, and was a calmed voice and friend in the most stressful times. And finally, Eduardo do Couto e Silva, a pleasure to work with, always ready to help and whose input and criticism improve the clarity, structure and strength of ideas.

I would not have been able to get this far without the support and love of my family, especially my parents, Paloma and Erik, who sacrificed everything to give us an education and the values to live a fruitful and happy life, and are and will always be an example to follow. And my grandfather, your genes are probably what keep us all afloat. And my siblings, Patricia, Sandra, Mark and Alain.

Since I came to Stanford, I have made wonderful friends like those I left in Spain, and I have met my future husband. Christian and I share the happiness of getting this degree, because, without him, this and many more good things in my life simply would not exist.

To all of you who became part of me in Spain and in the United States, either in the volleyball courts, in the classroom, out or at home; we made it!

Table of contents

Abstract.....	v
Acknowledgements.....	Error! Bookmark not defined.
Table of contents.....	xi
List of tables.....	xv
List of illustrations	xix
List of acronyms	xxiii
1 Introduction.....	1
1.1 Related Topics	5
1.1.1 The Gamma-ray Large Area Telescope (GLAST)	5
1.1.2 Space System Integration and Test (I&T)	17
1.1.3 Design and Optimization of Electronic Databases	26
1.1.4 Human-Computer Interactive (HCI) Systems	31
1.1.5 Software Development Considerations.....	38
1.2 Current Approach: Space System I&T Logs	40
1.3 Prior Art	46
1.4 Contributions.....	49
2 E-Logbook Components	51
2.1 E-Logbook Component Requirements	52
2.2 I&T Log Components	52
2.3 Design Components	60
2.4 Development Platform Components.....	62
2.4.1 Hardware Components.....	62
2.4.2 Software Components.....	62
2.5 E-Logbook Component Implementation	67
3 E-Logbook Electronic Database System	69
3.1 Database Requirements.....	69

3.2 Database Design.....	71
3.2.1 Requirements Analysis and Conceptual Design	71
3.2.2 Logical and Physical Design.....	80
3.3 Database Requirements implementation.....	88
4 Human-Computer Interactive (HCI) - Graphical User Interface (GUI) System	89
4.1 HCI-GUI System Requirements	89
4.2 HCI-GUI System Design	91
4.2.1 HCI Design	91
4.2.2 GUI Design	95
4.2.3 Problems and Solutions.....	106
4.2.4 Final HCI-GUI System Structure.....	111
4.3 HCI-GUI System Requirements Implementation.....	113
5 E-Logbook’s Experimental Setup for the LAT I&T Project.	115
5.1 Experimental Requirements.....	116
5.2 Software Architecture	117
5.2.1 Database mirroring.....	118
5.2.2 Database backup	119
5.2.3 Mobile Computing Rack (MCR) architecture	119
5.3 Hardware Architecture.....	120
5.4 Requirements Implementation Process.....	122
5.4.1 E-Logbook’s Code Structure	125
5.4.2 E-Logbook’s Website	127
5.5 Experimental Requirements Implementation	129
6 E-Logbook Analysis and Results.....	131
6.1 Database Performance Analysis	132
6.2 Human-Computer Interface (HCI) – Graphical User Interface (GUI) System Performance Analysis	139
6.2.1 Main Results	141
6.2.2 Usability Criteria Results	145

7	Conclusions and Future Work	163
	APPENDIX A: E-Logbook's I&T Logs Requirements.....	167
	APPENDIX B: E-Logbook's Database Schema.....	181
	APPENDIX C: JIRA Issues.....	199
	APPENDIX D: Usability Questionnaire.....	207
	APPENDIX E: Questionnaire Results	218
	Bibliography	237

List of tables

<i>Number</i>	<i>Page</i>
Table 1: Relational Database Model terms and definitions	27
Table 2: HCI Features	36
Table 3: Original field requirements for the FHW Log	77
Table 4: Original field requirements for the FHW Log	78
Table 5: Number of computers and location used for E-Logbook in the LAT I&T facility	120
Table 6: E-Logbook database statistics for the LAT I&T project as of 02/24/06	132
Table 7: E-Logbook database schema statistics for the LAT I&T project as of 02/24/06, on a table by table basis	133
Table 8: Data loss statistics for E-Logbook on the LAT I&T project	137
Table 9: Subject statistics.....	140
Table 10: E-Logbook main usability results.....	143
Table 11: Visual Clarity average statistics comparison between QNov05 and QFeb06.	148
Table 12: Consistency and Compatibility average statistics comparison between QNov05 and QFeb06	149
Table 13: Explicitness average statistics comparison between QNov05 and QFeb06	150
Table 14: Appropriate Functionality average statistics comparison between QNov05 and QFeb06.....	151
Table 15: Informative Feedback and User Guidance statistics comparison between QNov05 and QFeb06	152
Table 16: Error Prevention and Correction average statistics comparison between QNov05 and QFeb06	153
Table 17: Flexibility and Control average statistics comparison between QNov05 and QFeb06.....	154

Table 18: System Usability average statistics comparison between QNov05 and QFeb06.....	156
Table 19: E-Logbook Feedback average statistics.....	158
Table 20: Overall Criteria Rating Results on QNov05.....	160
Table 21: Overall Criteria Rating Results on QFeb06.....	161
Table 22: User Log Database Fields.....	167
Table 23: Shift Log Database Fields.....	169
Table 24: M/D Log Database Fields.....	170
Table 25: FHW Log Database Fields.....	171
Table 26: FHW Matrix Log fields retrieved from FHW Log.....	172
Table 27: MMR Log Database Fields.....	175
Table 28: FSW Log Database Fields	177
Table 29: EGSE Log Database Fields	178
Table 30: Configuration Request Log Database Fields	179
Table 31: Configuration Report Format	179
Table 32: E-Logbook JIRA issues	206
Table 33: Visual Clarity Results on QNov05	218
Table 34: Visual Clarity Results on QFeb06	219
Table 35: Consistency and Compatibility Results on QNov05	220
Table 36: Consistency and Compatibility Results on QFeb06	221
Table 37: Explicitness Results on QNov05	222
Table 38: Explicitness Results on QFeb06	223
Table 39: Appropriate Functionality Results on QNov05	224
Table 40: Appropriate Functionality Results on QFeb06.....	225
Table 41: Informative Feedback and User Guidance Results on QNov05.....	226
Table 42: Informative Feedback and User Guidance Results on QFeb06.....	227
Table 43: Error Prevention and Correction Results on QNov05	228
Table 44: Error Prevention and Correction Results on QFeb06.....	229
Table 45: Flexibility and Control Results on QNov05.....	230

Table 46: Flexibility and Control Results on QFeb06	231
Table 47: System Usability Results on QNov05	232
Table 48: System Usability Results on QFeb06	233
Table 49: E-Logbook Feedback Results on QNov05	234
Table 50: E-Logbook Feedback Results on QFeb06	235

List of illustrations

<i>Number</i>	<i>Page</i>
Figure 1: Pair-conversion and detection mechanism [7]	6
Figure 2: Third EGRET Catalog versus GLAST expected Catalog after one year of operation, for gamma ray sources of energy greater than 100 MeV	8
Figure 3: Simulation of the gamma-ray sky ($E > 100$ MeV) as observed by the GLAST Large Area Telescope (LAT) [5]	9
Figure 4: Cutaway view of the GLAST LAT instrument [7]	10
Figure 5: LAT flight 16 tower grid under integration at SLAC	11
Figure 6: TKR detector module about to be I&T in the LAT tower grid	12
Figure 7: CAL detector module about to be installed into the LAT flight grid.....	13
Figure 8: ACD detector subsystem.....	15
Figure 9: The LAT bottom up, showing locations of the various ELX modules with respect to the TKR modules [7]	15
Figure 10: Quality Engineering (I&T, QA and CM) Relationships [13].....	18
Figure 11: GLAST LAT I&T Group Organization Chart [15].....	22
Figure 12: LAT I&T Assembly and Integration Documentation Flow [15]	24
Figure 13: LAT I&T facility and I&T flow diagram [21]	25
Figure 14: Four phases of database design [28].....	30
Figure 15: Tree structure as proper GUI navigation pattern [44]	35
Figure 16: Block diagram of HCI system architecture	36
Figure 17: Separation of an interactive system into a computational and a communication element [55]	39
Figure 18: Mate/Demate Report used at NASA JPL	43
Figure 19: Sample paper M/D Report used initially for the LAT I&T Project	44
Figure 20: Sample paper EGSE Validation Report used initially for the LAT I&T Project	45

Figure 21: Partial snapshot of a Babar Shift, html based.....	48
Figure 22: Connector mate for one of the LAT TKR modules.....	54
Figure 23: First CAL Module Installation into the LAT	55
Figure 24: Coating of one of the LAT FHW Connectors using a mix recorded in the MMR Log	57
Figure 25: EGSE setup in the LAT I&T Facility.....	58
Figure 26: E-Logbook's I&T Log Configuration	59
Figure 27: E-Logbook's interactive component system.....	61
Figure 28: Programming tools architecture schema used in the development of E-Logbook	63
Figure 29: E-Logbook's Development Software Architecture.....	67
Figure 30: GLAST LAT web based Shift Log	72
Figure 31: Relationships between E-Logbook's database schema global tables and I&T Logs.....	86
Figure 32: E-Logbook's Database Schema.....	87
Figure 33: HCI-GUI system design path	92
Figure 34: E-Logbook's Main window.....	93
Figure 35: E-Logbook's Edit M/D Record GUI.....	96
Figure 36: E-Logbook's EGSE Validation Record GUI	97
Figure 37: E-Logbook's M/D Report GUI with record information	99
Figure 38: Paper based M/D Report generated by E-Logbook.....	100
Figure 39: Paper based EGSE Report generated by E-Logbook	101
Figure 40: E-Logbook's Shift Report GUI	102
Figure 41: E-Logbook's Run Report GUI	103
Figure 42: E-Logbook's FHW Matrix Report GUI	105
Figure 43: E-Logbook's GUI-HCI System Structure	112
Figure 44: E-Logbook's software architecture for the LAT I&T Project.....	117
Figure 45: E-Logbook's hardware architecture in the LAT I&T facility	121
Figure 46: The LAT I&T facility control room (Room 102, Bldg. 33, SLAC).....	122

Figure 47: E-Logbook’s requirements implementation loop.....	123
Figure 48: E-Logbook’s release history correlated to LAT I&T progress	126
Figure 49: E-Logbook’s website’s main page	128
Figure 50: Correlation between I&T Log records created and number of record entries in the database schema.....	135
Figure 51: Second sample of paper M/D Report used initially for the LAT I&T Project	157
Figure 52: Correlation of overall question ratings after for QNov05, QFeb06 and the initial set of subjects in QFeb06.....	162
Figure 53: User Record GUI.....	167
Figure 54: User Report GUI	168
Figure 55: FHW Record GUI	173
Figure 56: FHW Report GUI	174
Figure 57: MMR Record GUI.....	176
Figure 58: MMR Report GUI	176
Figure 59: FSW Record GUI	177
Figure 60: EGSE Report GUI.....	178
Figure 61: Configuration Request Report GUI.....	180
Figure 62: FHW Configuration Report GUI.....	180

List of acronyms

AIDS	Assembly Instruction Data Sheet
CM	Configuration Management
DBMS	Database Management System
DR	Discrepancy Report
EGSE	Electrical Ground Support Equipment
ESD	Electro-Static Discharge
eV	Electron Volt
keV	Kilo Electron Volt
MeV	Mega Electron Volt
GeV	Giga Electron Volt
FHW	Flight Hardware
FSW	Flight Software
GLAST	Gamma-ray Large Area Space Telescope
GUI	Graphical User Interface
HCI	Human-Computer Interface
IFCT	Integration, Facilities, Configuration, and Test
I/O	Input and/or Output
ISOC	Instrument Science Operations Center
I&T	Integration and Test
LAT	Large Area Telescope
MGSE	Mechanical Ground Support Equipment
MMR	Material Mix Record
MMR#	Material Mix Record Number
NCR	Non-Conformance Report
NRL	Naval Research Laboratory
QA	Quality Assurance
R/D	Reference Designator

SVAC	Science Verification, Analysis, and Calibration
TRR	Test Readiness Review
V&V	Verification and Validation

Chapter 1

1 Introduction

In the manufacturing industry, there are few products that need more qualifications and regulations than space systems. Space systems use expensive state-of-the-art technology, many times undisclosed to the general public. Unlike nuclear power plants or submarines, fighter planes or medical devices, space systems can not be reached once deployed. They cannot be maintained or recalled. Due to the scope of their mission, failure of integration implies the loss of millions of dollars. In many cases, space system technology is irreplaceable.

Due to the unique characteristics of each space system, their integration and test (I&T) continues to be done by hand today. It lays in the hands of the specialized I&T personnel to ensure that each and every component in the system will perform once its mission starts. Nowadays, space systems I&T has become a culture of discipline and transparency, regulation, special training of highly qualified personnel and rigorous control of operations. Provisions for every contingency in the I&T process are constantly investigated.

Documentation of operations is critical since it will be the only source of information once the system is shipped. Nowadays, documentation of space systems I&T operations is done on paper. This practice is justified by many factors: the need to enter information multiple times for each I&T operation, reliability, easiness of use around the I&T facility, tradition, the fear of an insufficient or inefficient system.

Nevertheless, the use of paper causes maintenance problems and is inefficient for problem-solving. By the time I&T operations are completed, thousands of records have been hand-written, organized in folders, and stored for future reference. In fact, the volume of paper is so great that entire archiving facilities are created. This means that once information is needed, someone needs to go to the archiving facility and find the right information. I&T personnel indicate that this process can sometimes take on the order of a week. Moreover, finding the right information is not the end of the problem. Since the records are hand-written, even if the right information is found it can be difficult or impossible to read. Loss of time and loss of information can cause project delay, or even failure.

The use of electronic relational databases to maintain large amounts of information is a proven technology in today's computer world. An electronic database can help optimize the organization and display of information in innumerable ways not possible on paper. It enables the instant retrieval of critical information in any format devised by the user. It performs incomparably well at its intended task – finding information instantly.

The difficulty of designing an electronic database for space systems I&T operations derives primarily from the fact that it must be completely reliable. Furthermore, the creation of a reliable electronic database does not ensure its acceptance by the disciplined I&T community. Ultimately, the electronic database must be usable, and better than paper.

When I&T personnel were asked about the feasibility of creating an electronic logbook to document space systems I&T operations, the initial reaction was generally negative. Most of the operators believe that its implementation would only cause problems and the slowdown of a process that currently is carried out efficiently through paper. On the other hand, the quality assurance and managing I&T team believe that an electronic system to document operations can greatly benefit the process of space system I&T over the long run.

The Gamma-ray Large Area Space Telescope (GLAST) project is a scientific satellite that aims to explore the universe in the high energy band, from 20 MeV to 300 GeV. Its goal is to help the fields of Cosmology and Astroparticle Physics answer fundamental questions such as the origin of gamma ray bursts, acceleration mechanisms in relativistic outflows or the origin of dark matter.

GLAST's main space instrument, the Large Area Telescope (LAT), is currently in the final stages of I&T at the Stanford University Linear Accelerator Center (SLAC). In order to take over I&T operations, SLAC has built a state-of-the-art facility from the ground up. It has become one of the few U.S. government laboratories in a university setting that hosts the I&T operation of a major scientific space instrument. Due to the scope of the project, the LAT I&T subsystem has experienced intense scrutiny, throughout its 19 months of operation at this writing.

Nevertheless, it is precisely because of the academic environment at SLAC, that research on the improvement of space systems I&T operations can advance. But there are special circumstances too; since there is no other mechanism put in place to document I&T operations, the research must be completed and deployed within the major constraints of the GLAST LAT I&T mission, and its implementation must not and can not fail.

The research contained in this thesis aims not only to create a reliable electronic logbook, or E-Logbook, for space systems I&T operations within the scope of the GLAST project, but to design a system that improves the overall process of I&T, both for the I&T personnel that use the system, and the overall scientific and engineering community that ultimately will be exposed to the results of the I&T process.

This research will investigate every possible source of optimization for the system, as based on the understanding of the practices of space system I&T operations, the scientific mission of GLAST, the design of electronic databases and the process of software implementation. As regards software implementation, considerable effort will be devoted

to creating an ergonomic system that satisfies the needs of the users, which in the view of this researcher will ultimately define the success of the resulting technology.

The following section (Section 1.1) in this Chapter is an introduction to the different results obtained by research performed in these areas. The first subsection is dedicated to the GLAST project and includes a brief description of its science and instrumentation. The second subsection provides an introduction to space systems I&T and its application to the GLAST LAT I&T effort. The third subsection introduces the relational model for the design and optimization of electronic databases. The fourth subsection concentrates on results obtained regarding the design and optimization of human-computer interfaces. Finally, the fifth subsection describes several approaches that can help optimize the process of software development.

The remainder of Chapter 1 proceeds with the exposition of the current approach for the documentation of space system I&T operations and includes prior work done in the use of electronic capabilities for this effort. It concludes with the specific contributions of this research. Chapter 2 formalizes the specific I&T logging components and their requirements defined for this research, as based on the results in Chapter 1, as well as the overall design approach and development platform devised to implement them. Chapter 3 describes the design of the electronic relational database created for E-Logbook, while Chapter 4 focuses on the design of its human-computer interface system. Chapter 5 describes the experimental setup that made possible the use of E-Logbook for the GLAST LAT I&T effort at SLAC. Chapter 6 analyzes the overall performance of E-Logbook throughout its 19 months of operation for this project, as based both on the reliability of the database and the usability of the human-computer interface system. Chapter 7 shows the conclusions of this research and proposes new directions for future work.

1.1 Related Topics

1.1.1 The Gamma-ray Large Area Telescope (GLAST)

1.1.1.1 Detecting the gamma ray sky

Exploring the Universe requires the use of highly technological instruments. Telescopes scan the sky in search of new phenomena and to explain outstanding questions in all fields of knowledge. Physicists and astronomers rely on spacecraft to place new generations of telescopes that can investigate light sources in all frequency ranges beyond the Earth's atmospheric constraints. At present, there are on the order of 100 scientific satellites exploring the Universe orbiting around the Earth.

Gamma rays belong to the most energetic frequency of the electromagnetic spectrum. While the energy of visible light is roughly 2 eV^1 , the GLAST gamma ray energy range starts at 20 MeV. In this high energy range, problems such as the origin of the galactic and extragalactic diffuse gamma ray background, the search for the nature of dark matter and extra dimensions, or new classes of astrophysical objects can be investigated, and data on gamma ray bursts, solar flares or active galactic nuclei jets, which generate gamma rays, can be obtained.

The gamma ray's high frequencies yield very short wavelengths, and therefore the gamma rays cannot reflect or refract in telescope optics as does visible light. Nevertheless, these high-energy photons can be made observable by a very dense medium in which they interact to convert into a charged pair of particle and anti-particle.

¹ eV: A unit of energy, the electron volt (eV) is equal to the work required to move one electron through a potential difference of 1 volt. The eV is a small unit of energy, so mega-electron volts (MeV) and giga-electron volts (GeV) are commonly encountered. The masses of elementary particles are frequently expressed in term of electron volts by making use of Einstein's equation, $E = mc^2$, where m is the mass of the particle and c is the speed of light [1].

These charged particles can then be tracked by modern particle physics detectors. The technological challenge is to design an instrument that can measure the energy, initial direction and time of arrival of the photon converted particles.

This challenge was met by GLAST's precursor, EGRET [2]. EGRET used a spark chamber full of gas as the tracker detector and Ta as the thin dense material to convert the incoming photons, where a set of wires across the chamber would detect the resulting particle direction. A NaI(Tl) calorimeter following the chamber detected the energy of the particles. A monolithic charged particle counter covering the rest of the detector rejected background charged particles.

A modernization of the 25-year-old design of the EGRET telescope gave birth to GLAST [3] [4] [5] [6] (see section 1.1.1.3). Layers of silicon-strip-position-sensitive detectors interleave with layers of dense Tungsten material where the photon pair conversion takes place, in what is called the tracker detector (TKR). A hodoscopic CsI(Tl) calorimeter (CAL) follows the layers to detect the position and energy of the conversion byproducts. These detectors are surrounded by a segmented charge particle detector, called Anti-coincidence detector (ACD), which helps rule out or veto incoming charged particles, such that only neutral particles like gamma ray photons are taken into consideration. Figure 1 shows a cartoon not to scale of this concept.

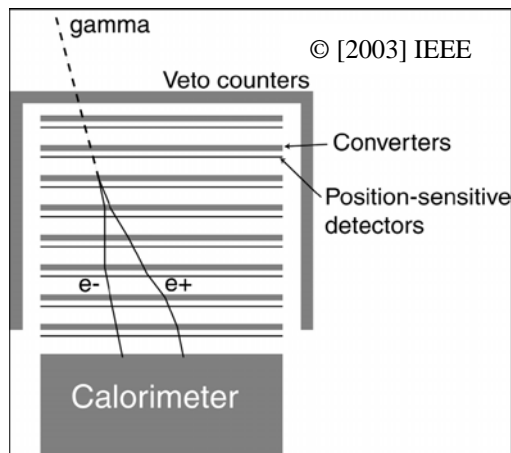


Figure 1: Pair-conversion and detection mechanism [7]

1.1.1.2 GLAST mission

The scientific mission of GLAST is to browse the gamma ray sky operating from 10 keV to 300 GeV, including a never-before-explored energy range from 30 to 100 GeV. To accomplish this it will use a pair conversion based large area telescope (LAT) and a secondary instrument called the GLAST Burst Monitor (GBM). While the LAT energy range is from 20 MeV to 300 GeV, the GBM will operate at a lower energy range, 10 keV to 30 MeV.

Scientists expect that GLAST will have as much of an impact as the Hubble telescope, even though it is a much smaller mission (roughly 0.5B\$ compared to >2B\$). At the time of this writing, GLAST is expected to be launched in the summer of 2007 from Cape Canaveral Space port on a Delta II 2920H-10 rocket. GLAST's orbit is 565 km circular and 28.5 degrees of inclination. GLAST's technical budget is 3000 kg of mass and 650 W of energy.

The LAT will provide a very large field of view of more than 2.4 steradians. The LAT has two observation modes, all sky survey and pointed observations, for which it is equipped with an autonomous re-pointing mechanism and a rapid slew speed of 75 degrees in less than 10 minutes.

The GBM is designed to observe gamma ray bursts, which are sudden, brief but very intense flashes of gamma rays that occur about once a day at random positions in the sky. Once the GBM detects a gamma ray burst it alerts both the LAT and ground telescopes within seconds. The GBM can also re-point for exceptionally bright bursts that occur outside the LAT field of view. Together with the LAT, the GBM will provide the widest range of energy detection in the x-ray and gamma ray regime for any satellite ever built.

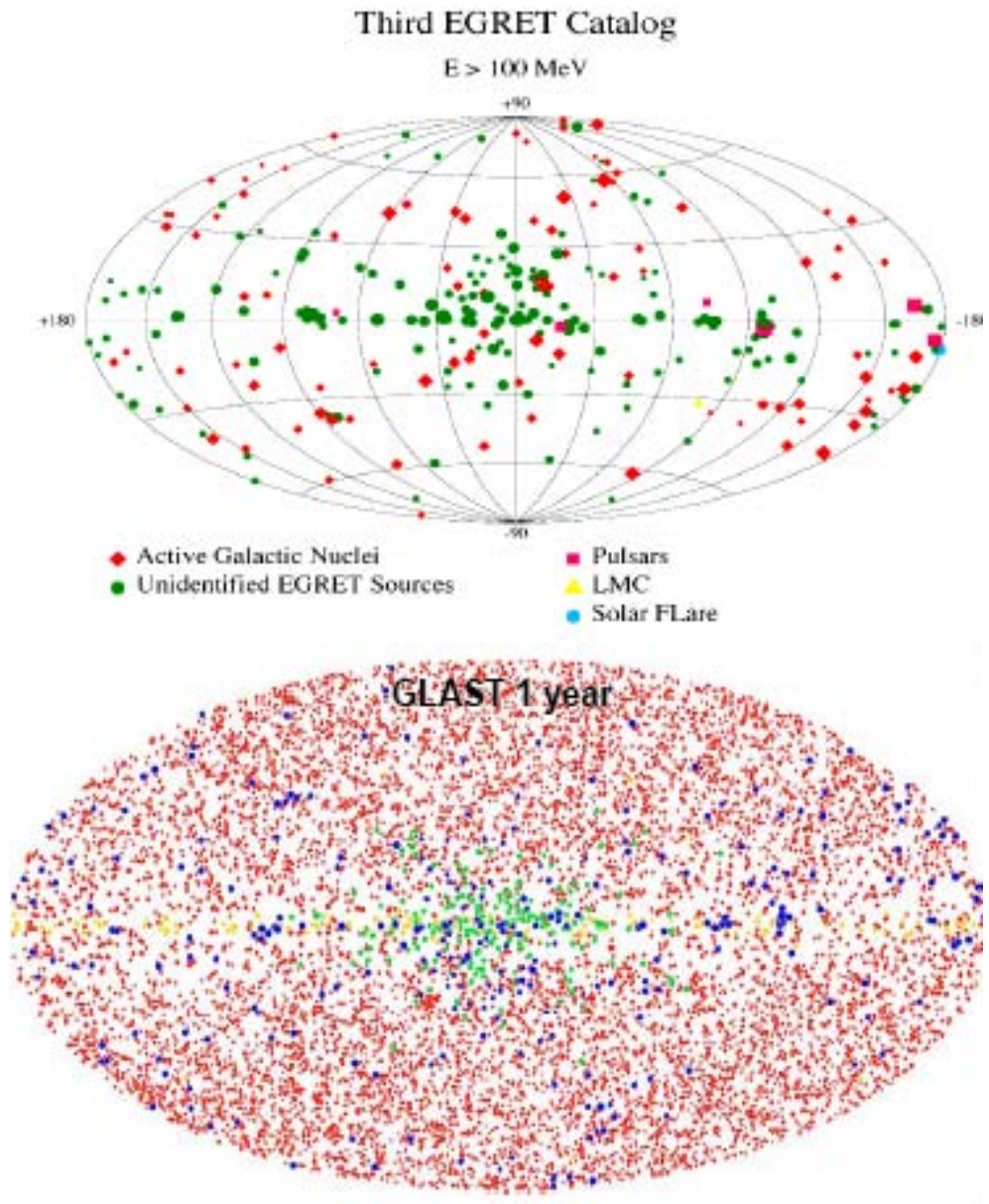


Figure 2: Third EGRET Catalog versus GLAST expected Catalog after one year of operation, for gamma ray sources of energy greater than 100 MeV

GLAST's new technologies will yield improved detector sensitivity. The top of Figure 2 [5] shows EGRET's third and final catalog (in galactic coordinates with the center of the Milky Way galaxy at the center of the figure) of gamma ray sources with energies above 100 MeV, both known and unidentified (not associated with any known astrophysical object). The bottom of the figure shows the expected GLAST catalog for the same energy range, after one year of all-sky-survey operation.

Figure 3 shows another simulation of the gamma ray sky for energies greater than 100 MeV as detected by the LAT.

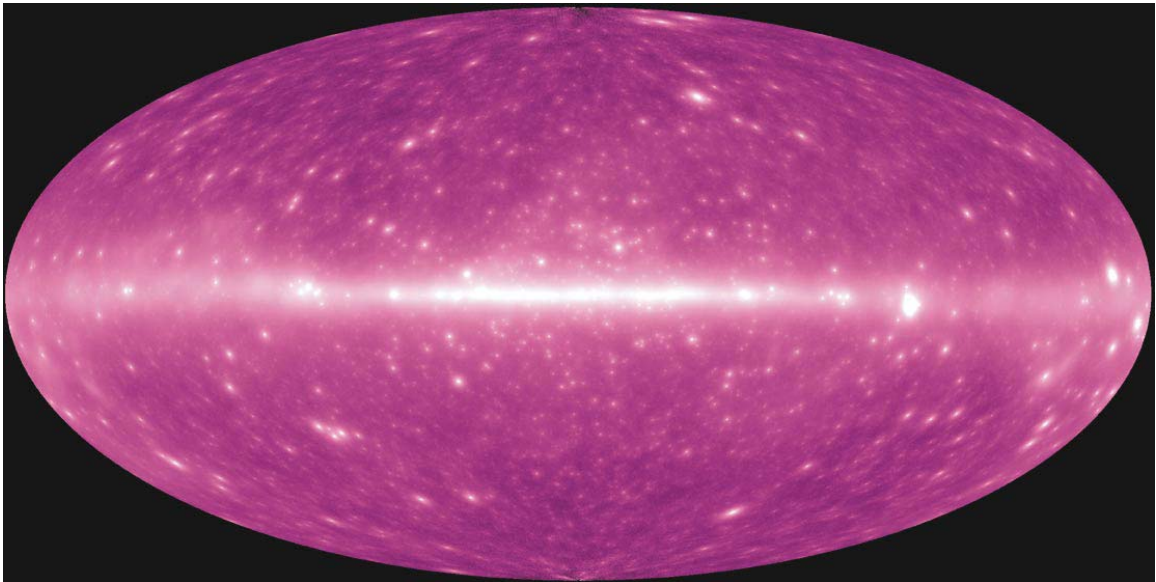
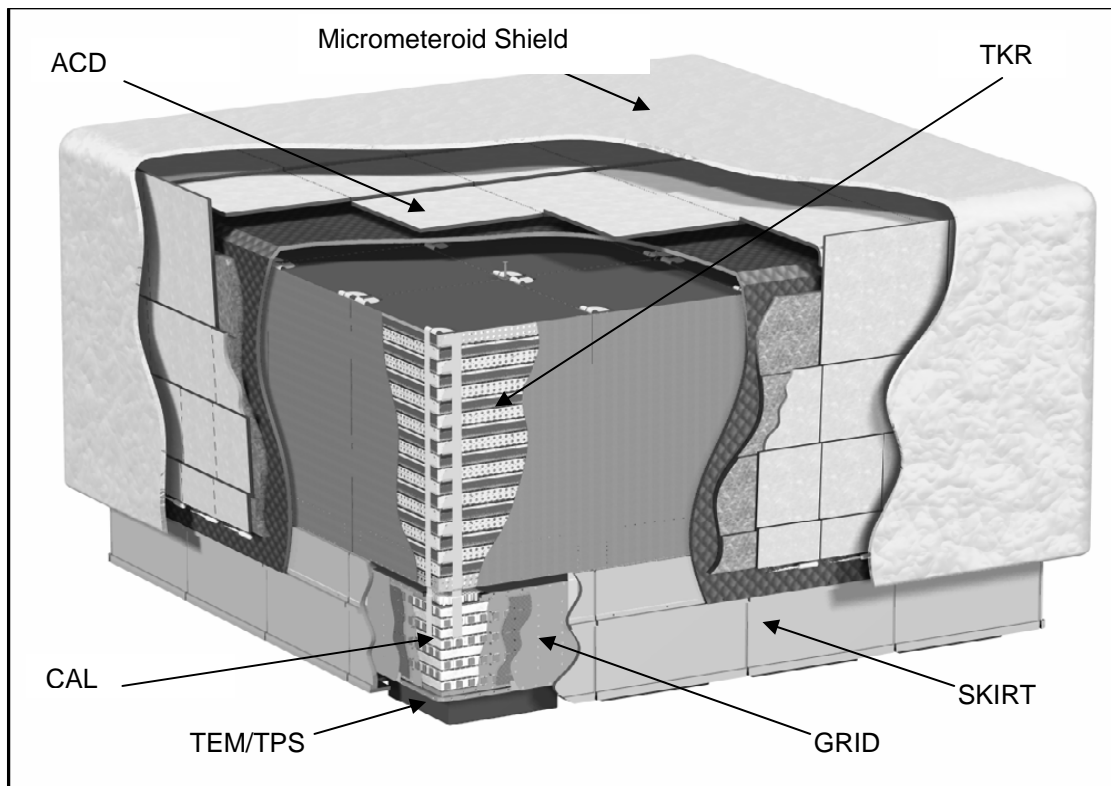


Figure 3: Simulation of the gamma-ray sky ($E > 100$ MeV) as observed by the GLAST Large Area Telescope (LAT) [5]

1.1.1.3 The LAT Detector System

The LAT detector system is composed of multiple subsystems. The core of the detector consists of 16 towers, each of which contains a pair conversion and detection system called tracker (TKR), a calorimeter (CAL) and the tower electronics and power supply (TEM/TPS) module. The towers are inserted in a support grid and surrounded by an anticoincidence detector (ACD), used to veto incoming charged particles, and a micrometeoroid shield. Underneath the towers are located the LAT data acquisition and trigger electronics (ELX) modules and the heat-pipe thermal system [7].

Figure 4 shows a cutaway view of the LAT. The ELX modules and heat-pipe thermal system are not shown. Figure 5 shows the LAT flight 16 tower grid under I&T at SLAC.



© [2003] IEEE

Figure 4: Cutaway view of the GLAST LAT instrument [7]

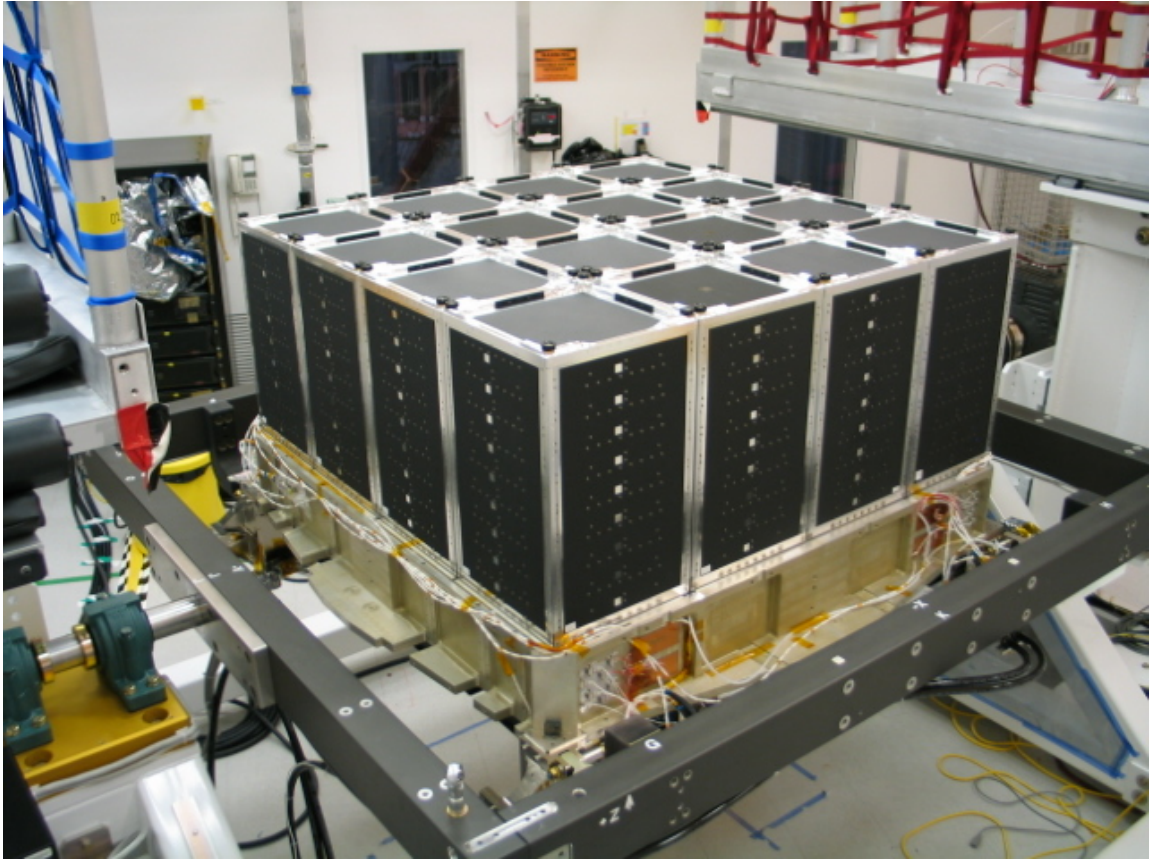


Figure 5: LAT flight 16 tower grid under integration at SLAC

1.1.1.3.1 The Tracker Detector (TKR)

Each of the 16 TKR detector modules consists of 36 layers of Silicon strip detectors mounted on both sides of a tray containing a Tungsten foil. The foil's thickness ensures interaction with the incoming gamma ray. The trays are mounted in the TKR alternating plane orientation, such that 18 planes of x,y layers separated by a few millimeters are formed. The layers are made of Silicon strip wafers with a 228 micron pitch separation between strips. The 228 micron pitch ensures good two-track separation (when multiple tracks hit the detector simultaneously) for event reconstruction and excellent angular resolution of high energy conversions. The main technological challenge in the TKR was

to read out the 36-cm long strips with only 175 microwatts of power per channel, including digital activity and data [8] [9].

Another major challenge for the TKR design was to build a low-mass, stiff carbon-composite structure to support the converter/tungsten foils and detectors/layers during launch, while minimizing interactions with gamma rays and the electron-positron pair.

The 16 TKRs were designed by a team led by UC Santa Cruz that included SLAC and several Universities in Japan. The TKRs were produced primarily by an Italian group of Universities led by the University of Pisa. Figure 6 shows a TKR flight unit under integration and test at SLAC.

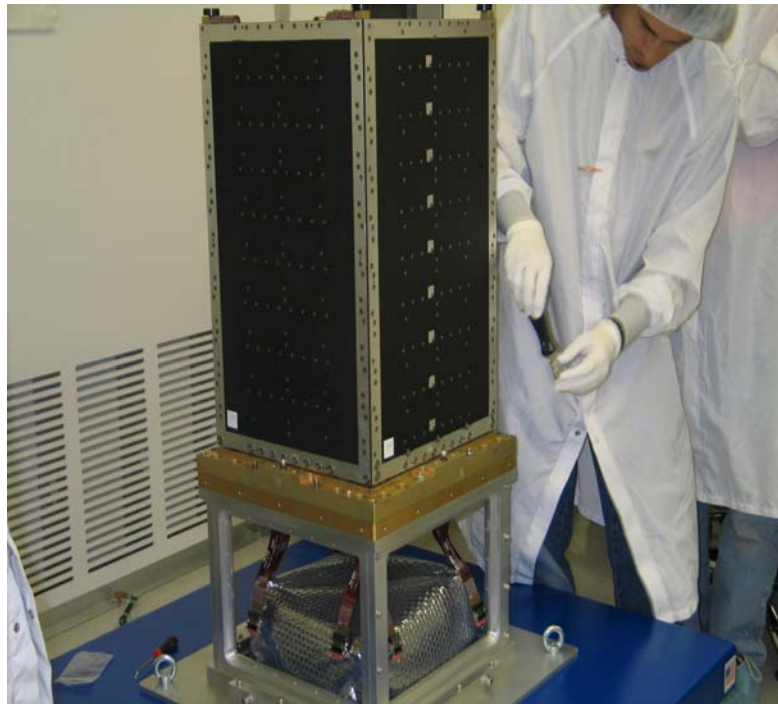


Figure 6: TKR detector module about to be I&T in the LAT tower grid

1.1.1.3.2 The Calorimeter Detector (CAL)

The CAL is the detector responsible for measuring the position and energy deposited by the cascade of particles produced by pair conversion in the TKR. The energy information is used to estimate the original energy of the gamma ray source.

The Calorimeter also consists of 16 modules, each with 8 layers of Cesium Iodide crystals. The crystals are horizontal bars with two diodes on each end, one small for response to high energy and one large for low energy. The alternating layers are oriented in perpendicular planes to ensure a three-dimensional reconstruction of each event shower. Each crystal is read out by an amplifier with two gain ranges, giving a total of 4 ranges to cover the overall required dynamic range to read out the energy deposition and location of each event. The 16 CALs were designed and produced by a team led by the Naval Research Laboratory (NRL), involving groups in Sweden, and France. Figure 7 shows a flight CAL module about to be installed into the flight grid at SLAC.

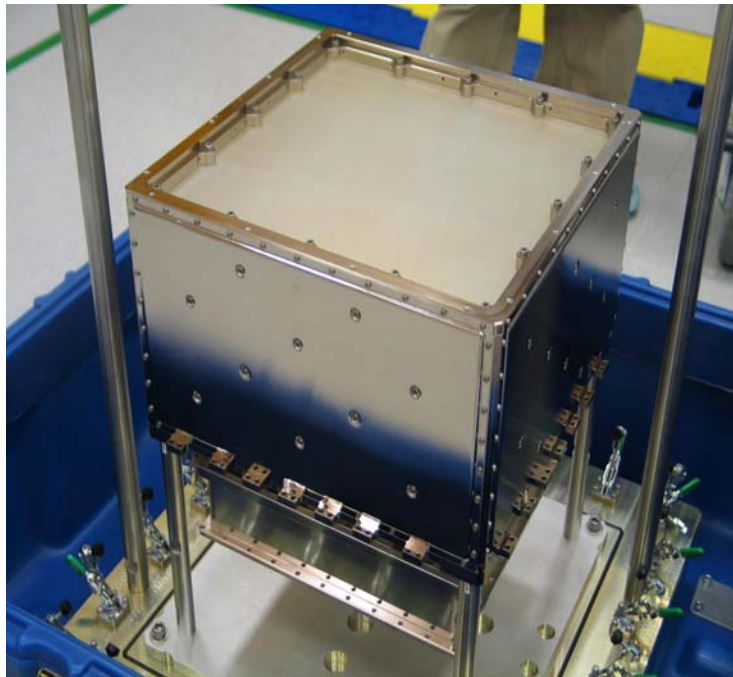


Figure 7: CAL detector module about to be installed into the LAT flight grid

1.1.1.3.3 The Anti-Coincidence Detector (ACD)

The Anti-Coincidence Detector (ACD) system is an array of 89 plastic scintillation tiles read out by a dual-redundant phototube system that hermetically covers the LAT 16 tower grid. Every time a charged particle (typically protons, electrons, helium nuclei or heavier cosmic rays) hits any of its tiles the ACD sends a trigger signal that can be used to rule out the event as originating from a gamma ray.

The ACD is required to be hermetic in order to achieve 99.97% efficiency when rejecting even the minimum ionizing particles [7]. But it also must be highly segmented (thus the tile design), to minimize particle rejection from back-splashing x -ray photons produced in gamma-ray induced electromagnetic showers in the LAT TKR and CAL.

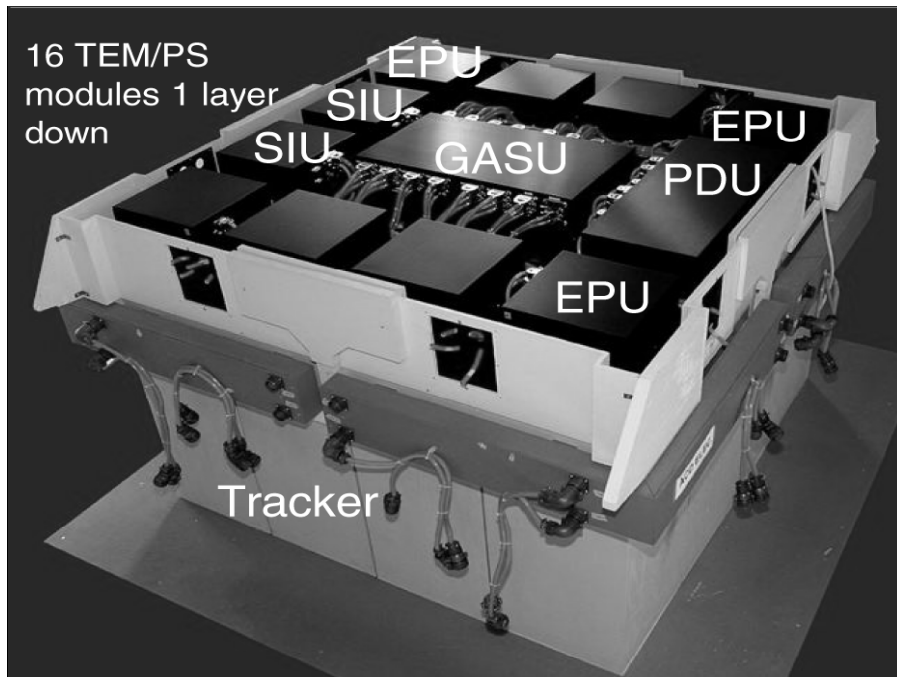
Another challenge for the ACD design is to support its large structure against launch loads with minimal extraneous material and mass. The support shell is fabricated from aluminum honeycomb panels with carbon-fiber face sheets. The plastic tiles are mounted on the panels with aluminum flexures. The structure also supports the outermost layer of the LAT, the thermal blanket and micro meteor shield. Figure 8 shows the flight ACD in production at GSFC.

1.1.1.4 Other subsystems

Figure 9 shows a schematic of the location of the electronics (ELX) subsystems with respect to the TKR/CAL modules. Each of the 16 TKR/CAL tower modules has a tower electronics module (TEM), including data acquisition and trigger electronics plus tower power supplies (TPS). The TEM/TPS modules are mounted just below the CAL. Below the TEM/TPS modules the rest of the ELX modules are located: a redundant global trigger and signal distribution module (GASU), which includes the ACD digital electronics, 3 event-processing modules (EPU) that make up a “computing farm” for event preprocessing before transmission to the ground, two redundant spacecraft interface units (SIU), and a redundant power distribution unit (PDU).



Figure 8: ACD detector subsystem



© [2003] IEEE

Figure 9: The LAT bottom up, showing locations of the various ELX modules with respect to the TKR modules [7]

The TEM and additional modules are enclosed on the sides by a “skirt” that helps to block electromagnetic interference (EMI). The data acquisition (DAQ) ELX modules and much of the front end TEM electronics were designed and built at SLAC.

The detector towers and electronic modules mount into the 4x4 aluminum grid. The 4x4 aluminum grid is the principal structural element in the LAT design. It also serves to conduct heat produced by the TKR, CAL and ELX modules out from the center of the instrument to its edges where variable heat pipe radiators mounted on the outside of the grid radiate the heat to space. The grid was designed and built at SLAC.

1.1.1.5 LAT Engineering Models

The GLAST technological concept to detect high-energy gamma rays could not be tested in the Earth’s environment. Instead the engineering models (EM) were tested in the beams of high energy particle accelerators and in high-altitude balloons.

An early prototype of a LAT tower, including a tracker, calorimeter, and anti-coincidence detector, was built and operated in test beams in 1999 and 2000 and flown on a high altitude balloon in 2001. Those prototypes served to validate the LAT instrument design concepts [10] [11] [12].

In 2004 a set of EM prototypes was built to the final flight design and subjected to both functional and environmental testing. The Calorimeter module was completely instrumented, while the Tracker module was instrumented with live silicon and electronics only in the lowest 3 x,y planes. Those two instruments successfully operated together while taking cosmic ray and low energy gamma ray (17 MeV) data, using a TEM for data acquisition.

Several calibration units (CU) have also been built in order to perform tests on the overall system.

1.1.2 Space System Integration and Test (I&T)

Experience has shown that a stringent set of rules, procedures, standards and management groups facilitate the successful integration and testing of products as complex and technologically advanced as space systems. Entities such as NASA, Lockheed Martin or Boeing are examples of pioneering efforts in the formalization of this process [13] [14].

The so-called space system I&T group or subsystem is defined as the entity responsible for verifying the correct integration of all of the space system elements through methodical testing and/or analysis. The main benefit of the existence of an I&T group is autonomy, or the capacity to act as an independent agent to validate and verify the system elements. On the other hand, the technical input and support of the Engineers who designed and built the hardware to be integrated also key to the success of the I&T group [14].

1.1.2.1 I&T Philosophy

A philosophy for the I&T group has been defined in the field to ensure the successful completion of its mission. It is based on three standards [13]:

- *Safety*: The primary responsibility of any I&T group is the safety of the space system components with which it is entrusted. Therefore, safety awareness is built into all phases of the I&T mission: test methods and procedures are developed and executed to ensure that the system and its test components are in no danger, due to improper handling, erroneous wiring, electrostatic discharge, contamination and power or operational problems, for example.
- *Quality Engineering*: Quality engineering must be accomplished by the I&T group, comprised of three essential subgroups:

- *Quality Assurance (QA)*: The QA group ensures the credible development of the test article by enforcing project standards throughout the system lifecycle.
 - *Configuration Management (CM)*: The CM group maintains the integrity and traceability of the test articles by controlling and tracking these throughout the system lifecycle.
 - *Integration and Test (I&T)*: The overall I&T group systematically proves that the test article satisfies requirements and serves as a forum for final test article or system acceptance.
- *Independent Testing*: The I&T group must be autonomous and free to act as an objective independent agent to validate and verify the space system elements.

Figure 10 [13] shows the established relationships among the quality engineering groups. The intersecting areas show the activities in which the three groups interface.

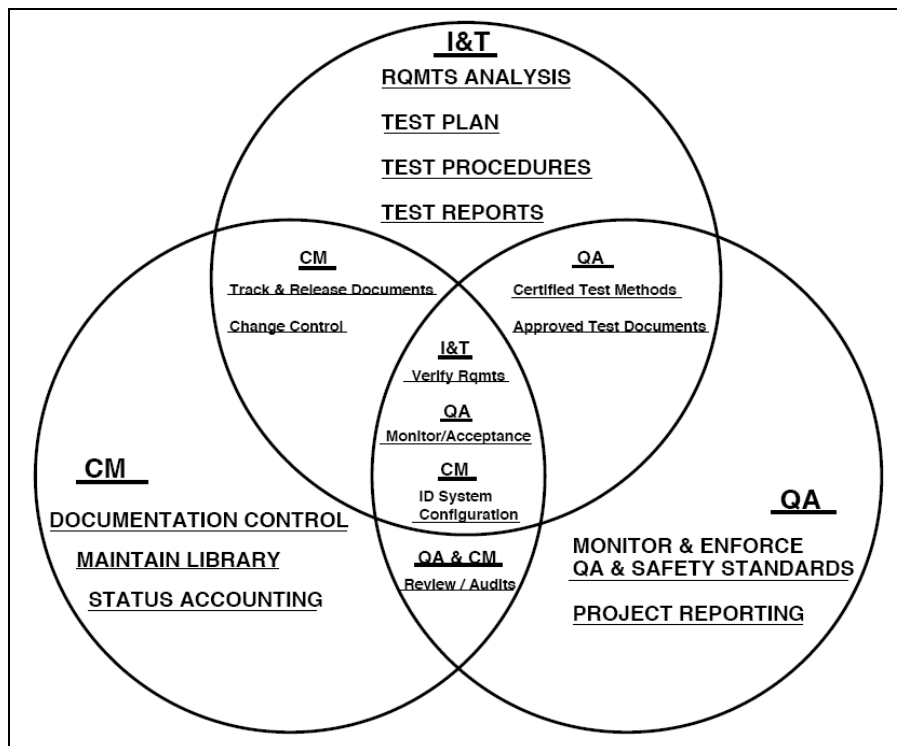


Figure 10: Quality Engineering (I&T, QA and CM) Relationships [13]

1.1.2.2 Building Block Test Approach

An example of the application of the I&T philosophy is the test approach. A standard test approach used by the I&T group is the so called *Building Block* approach. This approach is based on the progressive testing of only one item or process at a time in a stand-alone condition, and then integrating and testing it with other previously tested items or processes in the system.

The main benefit of this approach is that it simplifies the testing and analysis by building each test on the knowledge and results gained in the previous test and controlling the number of new integration parameters, thereby limiting the number of variables that have an effect on the test results [13] [15].

1.1.2.3 I&T Test Plan

Another example of the application of the I&T philosophy is the creation of a comprehensive Test Plan. Considered one of the most important tools at the disposal of the I&T group, the Test Plan is the complete set of requirements and ordinances defined for the successful completion of the space system I&T [13].

The Test Plan consists of several mechanisms:

- *Test Articles*: The global set of components that are going to be tested, such as space system hardware, Ground Support Equipment (GSE), software simulators, etc.

- *Test Support*: The necessary infrastructure to carry out the I&T effort, such as the I&T facility (which kind of clean room standard must be adopted based on mission requirements, safety mechanical and electrical infrastructure, physical and networking security, ...), the facility personnel (Test Manager, Director, Engineer, Conductor, Technician, QA, ...), their responsibilities, tools and communications.

- *Test Management*: The different groups of personnel on every level that are dedicated to the I&T subsystem, their organization and responsibilities.
- *Test Flow*: The different sets of tests that are defined to successfully integrate a piece of software and hardware into the system, defined chronologically. Each test is satisfied according to defined criteria.
- *Test Program*: The detailed description of the activities necessary for each test and milestone defined in the test flow of the I&T test plan.
- *Test Methodology*: The factual set of activities that comprise the different tests, and the overall methods used in the I&T effort.
- *Test Operational Controls and Procedures*: The set of tools defined to control the progress of the I&T group, such as daily and milestone meetings, scheduling or documentation.

Reference [13] describes the standard test control documentation² currently used for space systems I&T. Examples are the test plan itself, requirements documents, test procedures, standard operational procedures and orders, test reports, discrepancy reports (DR) that show problems when following procedures, and finally, the I&T Logs, which are the focus of this research.

1.1.2.4 GLAST LAT I&T

The GLAST LAT I&T group was the subsystem responsible for integrating the LAT from the ground up at SLAC. In the pursuit of this goal, and since space systems I&T did not exist at SLAC prior to the GLAST era, a clean room facility was also built from the ground up at SLAC.

² Documents such as user manuals, equipment instruction manuals, and any test article instruction manuals are not normally regarded as test documentation.

It was therefore of special relevance that both the I&T facility and the I&T team complied with the standards established in the industry, as described in the sections above. The personnel that operated the GLAST LAT I&T subsystem had extensive space systems I&T experience (see Chapter 6), which provided the project the necessary background for performing a professional operation.

The GLAST LAT I&T Test Plan [15] is based on the I&T philosophy of safety, quality engineering and independent and progressive testing utilizing the Building Block approach.

The LAT I&T group is responsible for the following elements of the GLAST project [15]:

- Developing and executing LAT I&T plans and procedures.
- Developing, prototyping, fabricating, assembling, and testing LAT I&T Mechanical Ground Support Equipment (MGSE) and elements of Electrical Ground Support Equipment (EGSE). This also includes environmental test planning and execution.
- I&T of the Engineering Model (EM) units, the Calibration Unit (CU) and the flight LAT.
- Verifying the GLAST LAT instrument by comparing data collected in beam tests from EM units, CU, and LAT to Monte Carlo (MC) predictions.
- Executing full operational testing and baseline testing to assure LAT performance before and after all configuration tests.

The I&T group uses these tests to demonstrate that the LAT instrument is working properly before the transfer of operations to the environmental testing group at NRL, and eventually the Instrument Science Operations Center (ISOC) at SLAC [16] [17].

Figure 11 [15] shows an organizational chart of the GLAST LAT I&T group. The I&T Group is supported by the subsystems and is coordinated in part by the SVAC and Mission groups. The Online group is responsible for developing, testing, monitoring and supporting the software used to collect data from the EM units, CU and LAT test stands

[18]. It is therefore responsible for the generation of the mechanism to archive the data generated by this and every I&T subsystem testing, which represents the focus of this research.

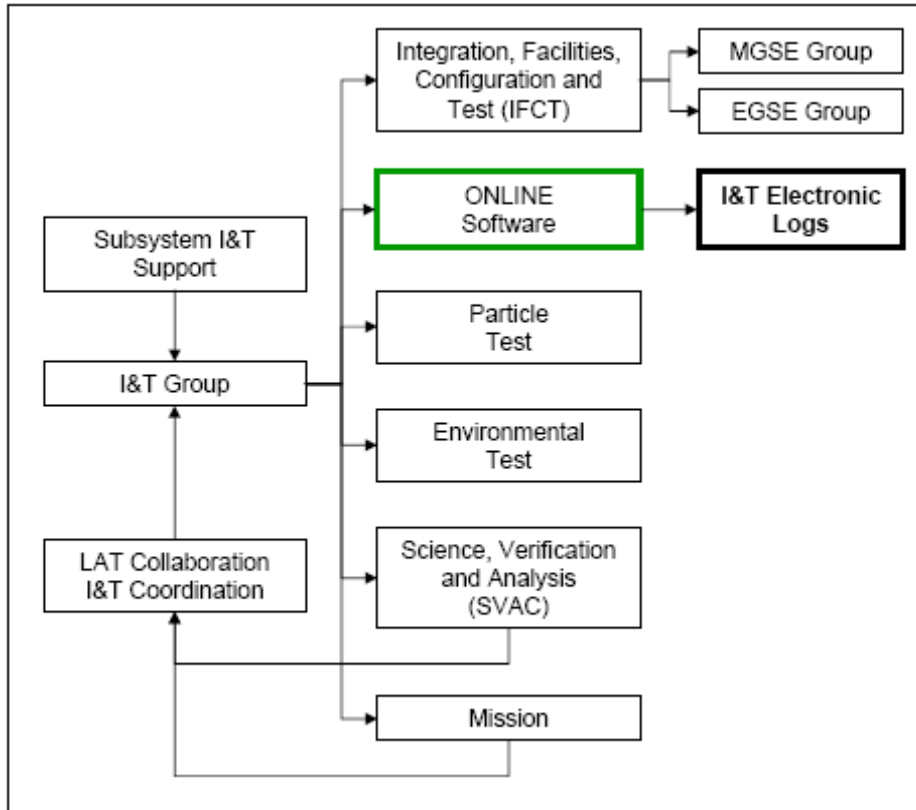


Figure 11: GLAST LAT I&T Group Organization Chart [15]

The LAT I&T test support infrastructure at SLAC consists of the LAT I&T Facility, located in Building 33, a Class 100,000 High Bay Clean Room environment per FED-STD-209D [19] designed to minimize the threat of all sources of contamination to the hardware, as shown in Reference [20].

The operation of the facility, following the GLAST LAT Facilities Plan [21], is based on safety and security. To this end the LAT I&T Facility benefits from a firewalled Local Area Network (LAN), isolated from the rest of the SLAC network and the world [15] [22].

The GLAST LAT I&T test support team consists of the following members [21]:

- *Test Manager*: I&T Manager.
- *Operations Manager*: Integration, Facilities, Configuration and Test (IFCT) Manager.
- *Systems Engineer*:
 - Mechanical Floor Engineer.
 - Electrical Floor Engineer.
- *Test Director (TD)*.
- *Test Conductor (TC)*.
- *Test Technician*:
 - Electrical: Specially trained to use the EGSE providing electronic troubleshooting skills to fault isolate problems that arise during testing [18].
 - Mechanical: Specially trained to use the MGSE, in particular flight hardware installation techniques, including connector mate/demate³ techniques. Responsible for ensuring all test equipment required to perform duties was available at the test location and is in calibration [20].
- *Quality Assurance Engineer (QAE)*: Required to be present and observe all flight hardware testing. Responsible for overseeing the following issues:
 - Ensuring that all hardware and test logs were maintained.
 - Ensuring that the procedures were performed as written and in the current released version.
 - Ensuring that all test requirements were accurately met.
 - Ensuring that any procedural deviation was approved and documented prior to being performed.

³ Demate: Term used to describe the opposite of mating or connecting two pieces of hardware. Although not included in the dictionary for the English language [23], the use of this word is widely accepted in the satellite industry.

- Ensuring that any unsatisfactory result is documented and understood prior to continuing the test.

The operation of the LAT I&T facility was based on the following controls and procedures [15] [21]:

- *Control Groups:*
 - Daily I&T Test Plan Meetings
 - Test Readiness Reviews (LAT TRRs)
 - Integration Readiness Review
 - System Test Readiness Review
 - Environmental Test Readiness Review
 - Pre-ship Readiness Review
 - LAT Configuration Control Board (LAT CCB) [24]
- *Schedule:* The near term and two week schedule was maintained by the IFCT manager with an integrated project management system. The schedule was reviewed daily in the I&T meeting.

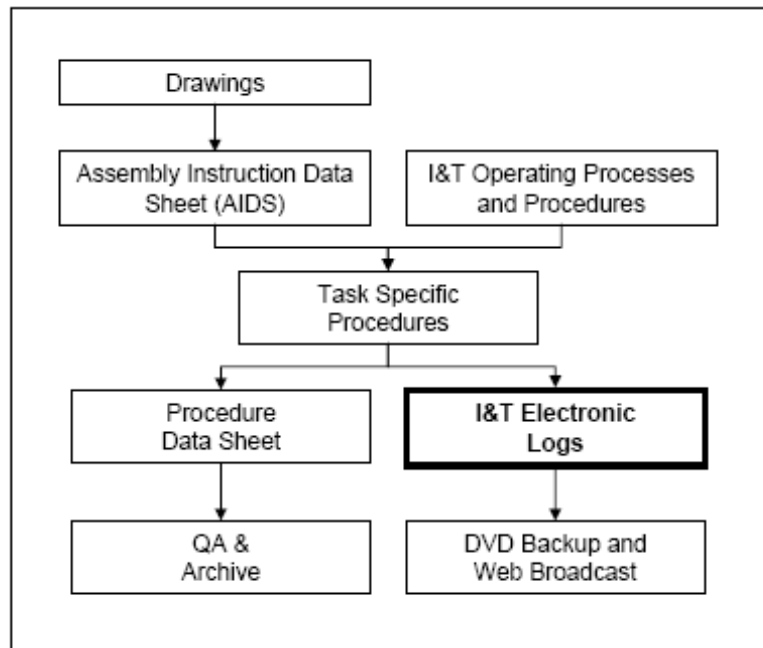


Figure 12: LAT I&T Assembly and Integration Documentation Flow [15]

- *Documentation:* Figure 12 [15] shows the LAT I&T assembly and integration documentation flow. Examples of the documents defined for LAT I&T [15] are requirements documents, the test procedures, the assembly instruction data sheets (AIDS), the assembly drawings, the non-conformance reports (NCRs) and finally, the I&T Electronic Test Logs which are the focus of this research.⁴

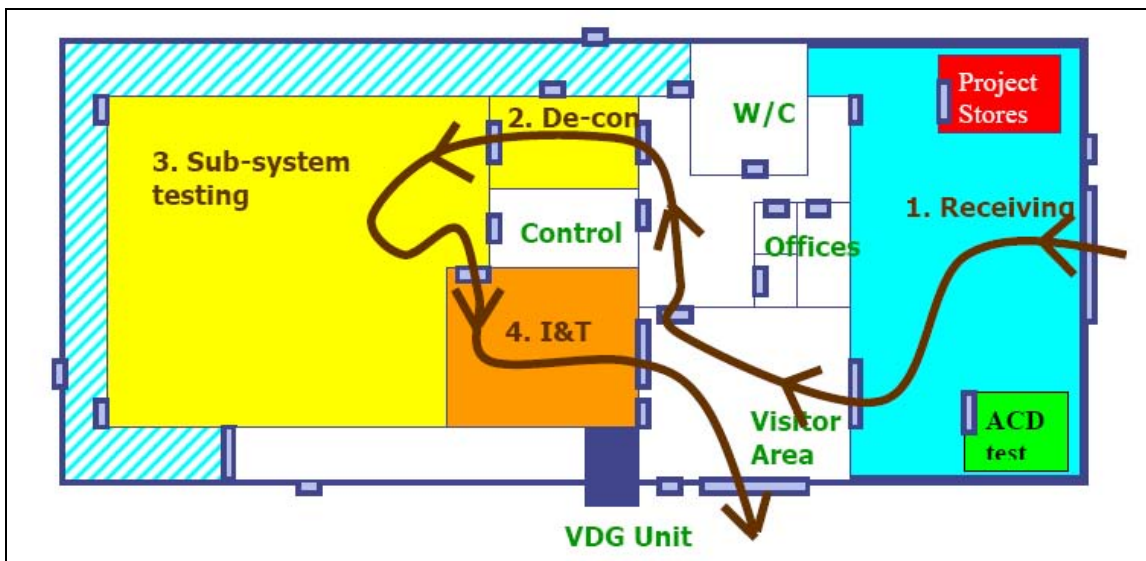


Figure 13: LAT I&T facility and I&T flow diagram [21]

Figure 13 [21] shows a layout of the LAT I&T facility at SLAC and the LAT I&T test flow indicated by the brown arrow in the figure. The starting point for I&T is receiving of flight hardware from subsystems (number 1 in Figure 13). Small parts will be stored in project stores, and larger parts (e.g. CAL or TKR modules) will be moved directly into the clean room decontamination area (number 2 in Figure 13). The exception to this rule is the ACD, which has its own special storage/test area (painted in green in Figure 13). Processing of flight hardware begins with post-ship functional tests executed in room 103

⁴ In the Current Approach Section (Section 1.2) of this thesis this aspect of I&T will be discussed in detail.

(number 3 in Figure 13) (TKR, CAL, and ELX) or the ACD test area. The flight hardware is then transferred into room 104 (number 4 in Figure 13) for integration into the LAT assembly. The ACD moves into the ACD test area for post-ship testing, and then directly into room 104 when ready for integration to LAT and does not pass through the standard clean room decontamination area. Upon completion of the LAT environmental test readiness review, the LAT instrument is shipped to the NRL for environmental testing.

1.1.3 Design and Optimization of Electronic Databases

The major part of the practical work done for this thesis focused on the design and development of the electronic database and system of human-computer interface (HCI) displays that would be used as the GLAST LAT I&T Logs.

The metaphor usually used to introduce the concept of electronic databases is the difference between piles of paper on a desk and those same papers organized in a filing cabinet. A lot of luck is needed to find the right paper in the piles on the desk, while in the filing cabinet you know exactly where to look in order to find it. Electronic databases represent the computerized version of this concept, electronic filing cabinets where piles of data can be organized and stored for easy and fast retrieval.

During the first decades in which electronic database systems were used, several data representation and storage systems were created. The hierarchical database system represents data in one or more tree structures. The system provides tools for locating a particular tree and then traversing the tree to find the desired piece of data. Hierarchical database systems are now widely used in the directory services realm, such as Microsoft's Active Directory and Netscape's Directory Server, as well as with Extensible Markup Language (XML). Another system is the network database. This system exposes sets of records and sets of links that then define relationships between the different records.

1.1.3.1 Relational databases

In 1970 Dr. E. F. Codd of IBM's research laboratory published the paper "A Relational Model of Data for Large Shared Data Banks" [25]. In this paper he proposed that data be represented as sets of tables. Instead of pointing from record to record in a defined structure, redundant data is used to link records in different tables.

Term	Definition
Column or Field	An individual piece of data stored in a table, such as Weight.
Row	A set of columns that together completely describe an entity or some action on an entity. Also called a record.
Index	Additional table parameter defined to identify a column for faster data retrieval.
Primary key	One or more columns that can be used as a unique identifier for each row in a table.
Foreign key	One or more columns that can be used together to identify a single row in another table.
Table	A set of rows, held either in memory during queries to the database (non persistent) or on permanent storage defining the database structure or schema (persistent).
Schema	Database structure composed of the tables, its elements and the underlying relationships defined by the foreign keys, populated with data.
Query	A command sent to the database to retrieve a subset of data (it will output a result set or non persistent table).
Result set	Another name for a non persistent table, generally the result of a query.

Table 1: Relational Database Model terms and definitions

Table 1 [26] shows the basic terms and definitions used in the relational database model. Each table in a relational database includes information that uniquely identifies a row in that table (known as the primary key), along with additional information needed to describe the entity completely. No other row (or record) is ever assigned that identifier,

and no other information is needed to locate that record in the table. The column size of the tables is typically not an issue other than the database server characteristics. The number of rows that a table may contain is a matter of physical limits; i.e. how much disk drive space is available.

Another element in the relational database model is the use of the so called foreign keys. Foreign keys are additional columns added to a table that include information used to navigate to other tables. They serve the same purpose as the lines that connect the entities in the hierarchical and network systems. However, unlike the rigid structure of the hierarchical/network models, relational tables can be used in various new ways (including some not envisioned by the people who originally designed the database) thanks to the use of foreign keys.

It might seem wasteful to store the same data many times, but the relational model is quite clear on what redundant data may be stored. The process of refining a database design to ensure that each independent piece of information is in only one place (except for foreign keys) is known as normalization.

After decades of usage the main features of relational databases that make them so attractive to the computer world are:

- Maturity: Decades of development.
- Reliability: Proven to be robust under operation.
- Fault tolerance: Robust against data corruption.
- Performance: Provides access to data in a timely fashion.
- Scalability: Even if the data grows the access time remains stable thanks to the use of indexes.

1.1.3.2 The SQL language.

Along with Codd's definition of the relational model, he proposed a language called DSL/Alpha for manipulating the data in relational tables. Shortly after Codd's paper was released, IBM commissioned a group to build a prototype based on Codd's ideas. This group created a simplified version of DSL/Alpha that they called SQUARE. Refinements to SQUARE led to a language called SEQUEL, which was finally renamed SQL⁵ [26].

SQL is now entering its fourth decade, and it has undergone a great deal of change along the way. SQL goes hand-in-hand with the relational model because the result of an SQL query is a table (the result set).

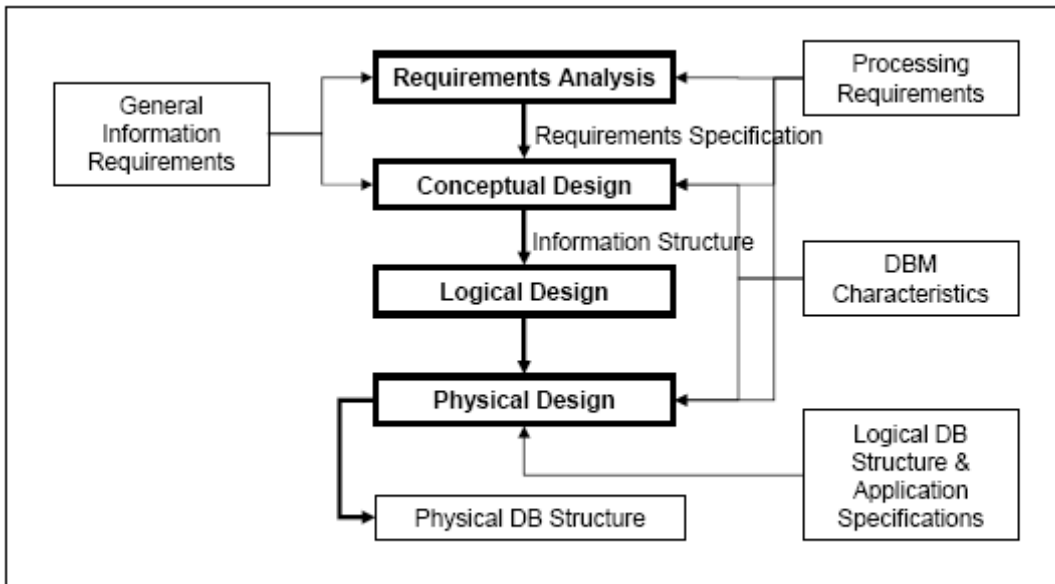
1.1.3.3 Design and optimization tools.

It became evident quite early on that there was a great deal of market share in the provision of software that would enable the user to create, maintain and manage electronic databases, the so called database management systems (DBMS).

MySQL, like mSQL and PostgreSQL, emerged out of the need for robust relational database management systems (RDBMSs) for small budgets, a gap that giants such as Oracle, Microsoft or Sybase had left in the market [27]. The main feature that makes MySQL stand out from other budget solutions is performance.

Even with the help of DBMS the design of databases is complex and usually requires revisions as new requirements or components are implemented. The purpose of database design is to transform the user-oriented description of the system requirements into a DBMS-oriented description.

⁵ SQL is not an acronym for anything (although many people will insist it stands for "Structured Query Language"). When referring to the language, it is equally acceptable to say the letters individually (i.e., S. Q. L.) or to use the word "sequel" [26].



© [1989] IEEE

Figure 14: Four phases of database design [28]

A generic database design methodology is presented in Figure 14⁶ [28]. The first and second phases are interrelated and consist of the analysis of the requirements and the conceptual design based on this analysis. Once the concept of the database is established, a third phase based on logical design follows, in which the different logical connections between the data are established and optimized. A final phase which attends to the physical or structural detailing of the database based on the selected DBMS language completes the design of the database.

Due to the complexity of this task, there are multiple studies on improving the process of database schema design [29] [30]. Some studies advocate an evolutionary database design, where the application development starts with a nucleus of an application which is extended, refined and delivered to the customer in small increments [31]. Evolutionary delivery is based on early and frequent iterations, much like the traditional software

⁶ Adapted from [28] with permission from publisher.

development process. But it also proposes ways of minimizing the changes in the database schema in order to simplify its maintenance.

A very *a propos* database architecture is based on the server/client concept, in which the application processing is divided into two or more logically distinct pieces. The database makes up half of the client/server architecture. The database is the server; any application that uses that data is a client. In many cases, the client and server reside on separate machines; in most cases, the client application is a user-friendly interface to the database.

1.1.4 Human-Computer Interactive (HCI) Systems

Satellite I&T activities are performed by a set of skilled operators trained to work in a clean room environment and to follow the practices established by a detailed test plan to ensure every level of success. Nevertheless, they, like all humans, occasionally make mistakes.⁷

In the computer world the human factor is becoming more important as the product developers increasingly realize that ergonomic designs eventually reduce cost and time [34]. The focus on the user and his/her interaction with the computer is a major factor in this research. It aims to provide a human-computer interface (HCI) that will help the user improve his/her performance and find information in a more efficient way.

1.1.4.1 Human Error

An error is a discrepancy between the operator's actual performance and the performance desired of him. Reference [35] defines error as the most common characteristic of human performance in a negative sense. It explains how a common division has been made

⁷ In September 2003 the satellite NOAA N Prime, slipped off a cart as it was being rotated from a vertical to horizontal position in Lockheed's satellite factory in California. The accident happened because bolts used to secure the satellite to the cart were missing, since in a previous shift the operators had "borrowed" them for another satellite in production without documenting this action [32] [33].

between systematic error (error that is repeated systematically; that has a pattern) and variable error (unforeseen random error). Another common distinction has been made between mistakes and slips, where a mistake reflects inappropriate attention and slips unintentional errors, like for example switching numbers when entering a sequence of numbers (enter 0545 instead of 0554).

While there is nothing we can do to avoid human variable error, there is a great deal of work that can be done to prevent systematic error or mistakes. While it is unlikely that error (and especially variable error) can be predicted, error data can be collected using several methods, such as manual collection in the real world, self-reporting, automatic data collection, experimental studies or expert judgment. This data will yield trends that can identify mainly systematic error. Once errors have been detected, they can be used to reveal a problem that exists in the equipment, personnel, job design or training, and therefore provide a starting point for fixing it.

As far as this research is concerned, operator error was collected in two ways. First, typical errors committed on paper-based documentation tools for satellite I&T activities were collected. Once the electronic approach taking these errors into account had been developed, new errors were collected and inspected via electronic data and questionnaires. It was expected that these new errors would be variable errors that could not be fixed by the computer approach, overseen errors that were not addressed, or completely new errors derived from the use of computers.

1.1.4.2 Use of Graphical User Interfaces (GUI) in HCI systems

One of the first decisions that need to be made in order to create human-computer interfaces (HCIs) is the nature of the interface. Historically the first HCIs were always text-based, where the human would interact with the computer via a set of commands fed into the command line. As computers improved, however, new operating systems appeared that made us shift to the world of graphical user interfaces (GUIs), displays, or windows, as they are called. Purists may say that text-based HCIs are more powerful,

since you are not limited by the display capabilities. Nevertheless those who favor text-based interfaces are consistently experts in the use of computers. Studies dating as far back as 1994 show that, for ordinary tasks, displays are slightly better than text-based HCIs [36]. This advantage has only increased with the extraordinary development of the computer industry in the past decade [37].

1.1.4.3 Milestones in GUI design for HCI systems

The use of displays as HCIs has become a science of its own. Its formalization has gone to some lengths and ranges from the focus on the semantic consistency of dialogs in GUIs [38] to the formulation of mathematical models to create single probabilistic finite state model representations of the HCIs to help with their evaluation [39].

Progress made on the development of systems that help design GUI systems has been especially useful [40] [41]. The goal is for the developer not to be distracted by the effort of creating the multiple elements of each display, since this tedious and redundant work is provided automatically, with the result that he/she can concentrate on the development of the functionality and cosmetic aspects of the design. In words of Trolltech's Qt Designer: "*Code less, create more*" [42] [43].

Once the creative gate of display design is open, special care needs to be made not to tread in the waters of creating a bad interface. Failing to communicate is a costly business, and not just in terms of lost productivity and frustrated users [44].

This aspect of display design has consequently also been formalized, starting with questions such as:

- Is it better that the display is static (or 2-D) or dynamic (3-D), in order to maximize information transfer while minimizing error in representation? [45]
- Which colors [46] or sounds should be used?
- Which kind of language should be developed to communicate concepts to the user?

The aspects that follow focus on the myriad of details involved in the design of a complex display HCI system. In the design effort of this research special attention has been paid to the ideas stated in References [44] [47] and [48].

Reference [44], in particular, is an invaluable summary of the philosophy that should be behind the design of any HCI. It offers clear guidelines:

- *Constraints*: Avoid useless complexity by limiting the user options without limiting what they need and can do.
- *Intelligence*: Create a system that takes care of the simple tasks automatically. This is not about AI, but about simple, practical touches that help the user get his/her job done. Herein resides the issue of error, and how to avoid it and draw attention to it with the proper alerting message system.
- *Elegance*: The 80/20 rule must be applied, which states “80 percent of the market will use 20 percent of the product features”. In terms of HCI design this means narrowing down the design to the essential needs and not creating complex mechanisms to provide useless detail that generally won’t be used and will obscure the overall product.
- *Transparency*: HCIs should serve the user as unobtrusively as possible, like a good waiter. They need to act subtly and attentively and should speak to the users in their own language. The users need to work, not work on their interface.
- *Attention to detail*: Once useless complexities have been avoided, all efforts should be used to attend to the detail of what is offered by the design. Every single element of the display needs to have a function and must have been thoroughly tested for its use and capabilities. If it’s not all right, then it’s all wrong; otherwise it should not have been there.

Another concept in [44] adopted in this research is the design of the display flow. While interconnectivity between windows would seem like a good idea in principle, practice shows that limiting the connectivity between windows to a so called tree structure creates a GUI system that is simpler to navigate and easier to understand, as shown in Figure 15.

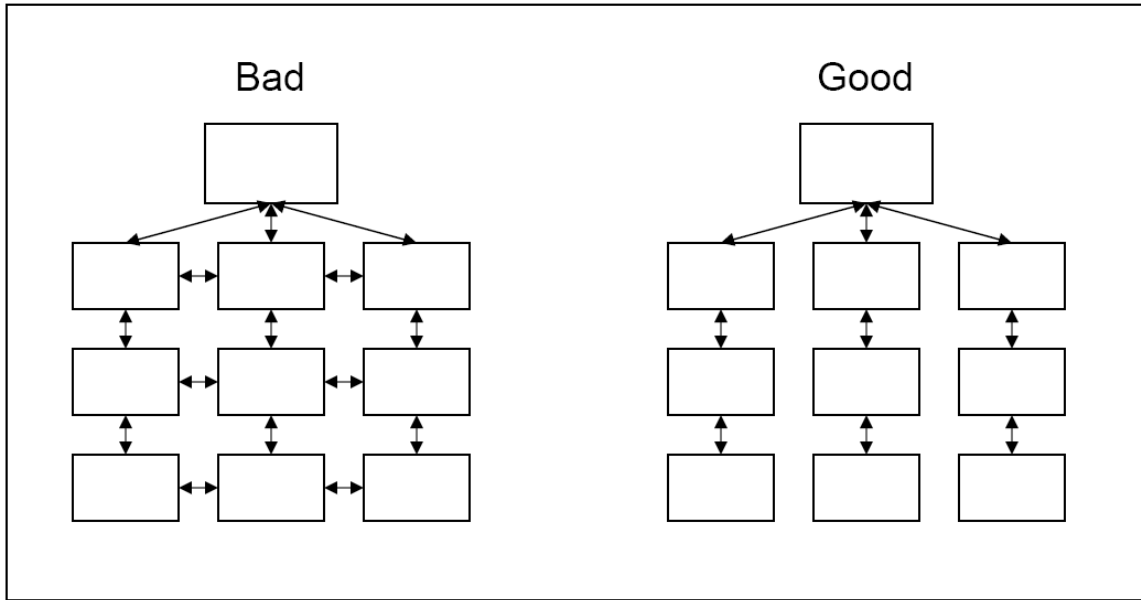
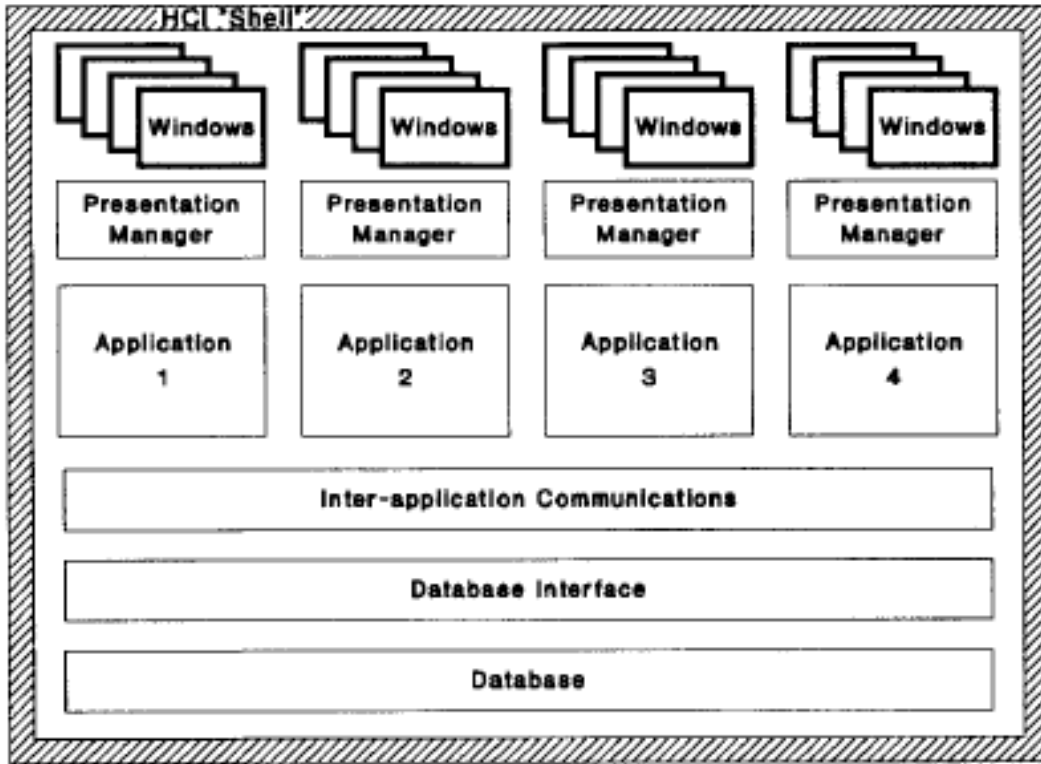


Figure 15: Tree structure as proper GUI navigation pattern [44]

Reference [47] advocates the creation of a simple, friendly and intuitive HCI with which non-specialist users can interact. It applies a similar tree structure concept into a global HCI frame, in which the task is split a number of sub-tasks. Each sub-task is performed by the operator using a software application running within the HCI, thus maintaining a consistent look and feel across the applications. The set of applications is tied by underlying connections created by an inter-application communication system and the database and database interface, as shown in Figure 16. Reference [47] also lists a comprehensive set of features that can produce a successful HCI, shown in Table 2.



© [1992] IEEE

Figure 16: Block diagram of HCI system architecture

HCI Features
• Interactive graphical user interface and software environment.
• Multiple concurrent windows and applications.
• Data-driven menu structure using buttons and pointing device.
• User configurable environment.
• Common user interface across applications yielding consistent “look and feel”.
• Ability to mix text and graphical display formats.
• Dynamic communication between applications.
• Underlying database which automatically flags changes.
• Ability to distribute applications over a network
• Historical traceability of data.

© [1992] IEEE

Table 2: HCI Features

Reference [48] explains the practical rules that developers have found, through experience, help avoid typical misuse and misconceptions of GUIs. Worth mentioning is the color selection, the location of the different widgets⁸ and their order.

By adopting the rules stated in Reference [48], this research also benefits from the fact that these rules are widely used, and therefore can endow it with the color of familiarity that can help its success so much. This is to say that a very early and conscious decision in the design of E-Logbook was to use tools already familiar to the user and to which he felt “friendly.” This would help minimize his/her learning curve and facilitate the acceptance of a new tool in the very well-tuned machine that is the satellite I&T domain.

Beyond the proper implementation of an HCI system, Reference [49] is a recommended practice on the design of HCI systems for space system operations. It formalizes the HCI requirements for HCI systems used for example in Mission Operation Centers, as for example the minimum information needed, the use of icons and default language, as well as proper GUI layouts.

1.1.4.4 User feedback as key element in the design of HCI systems

As mentioned above in dealing with the design of a Human-Computer Interface, it is increasingly understood that a lot of attention needs to be devoted to the human element. Especially complex is the treatment and inclusion of user feedback in the design and development process [50] [51].

⁸ Widgets: Input/Output (I/O) elements of the GUI, such as tables, line edits (single line windows), combo boxes (line edits with a predetermined list of values) or list views (tables that can group information in subfolders).

In the traditional approach, human factors and user feedback were relegated to the end of the development and implementation process, where attention to their recommendations had little effect. In the past years, an effort has been made to pay attention to the user feedback at earlier stages of a more inclusive design process. Reference [34] describes a new idea for a more profitable relationship with the user, called iterative design [52].

In iterative design the concept of incremental software development is explored, where the code is implemented in a step-wise refinement effort, philosophically similar to a top-down design. It results in a successively more robust version of the software, but also allows the introduction of user feedback much earlier in the development process.

Iterative design is an extremely effective approach to HCI design. It is summarized in four basic principles:

- Early and continual focus on the users.
- Integrated design.
- Early and continual user testing.
- Iterative design.

Reference [34] assimilates the idea of iterative design and expands it by introducing an 8-step process to include user feedback in the design effort, called the HCI Requirements Specification.

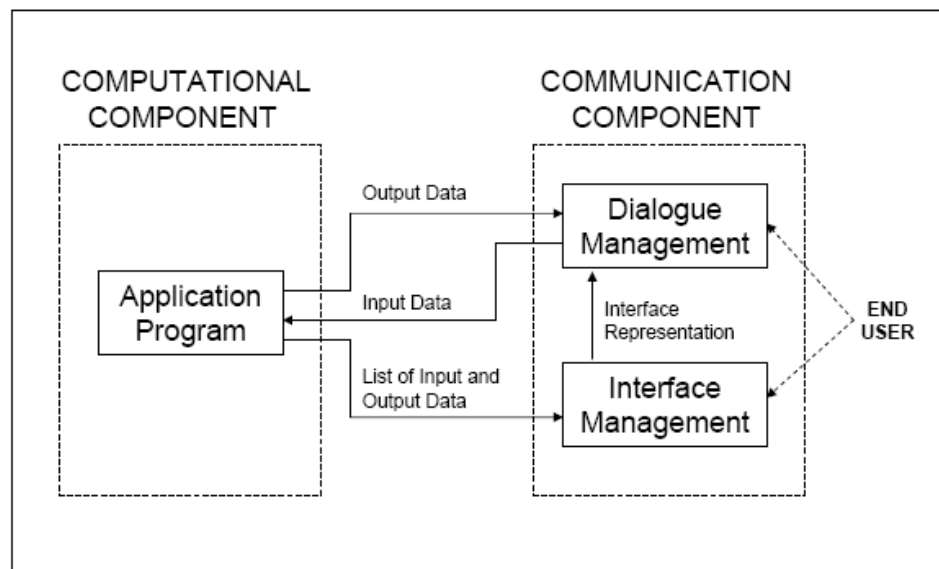
1.1.5 Software Development Considerations

In the previous sections I have alluded to the fact that the work of this thesis is two-fold. First it focuses on the design and optimization of an electronic relational database based on the requirements defined for the GLAST LAT I&T effort. Second it refines the development and optimization of the system of HCIs that will interface the database with the user.

In the traditional approach, software is implemented linearly from development to unit testing and debugging and then to function and system testing for more debugging, in a linear step-by-step cycle iterated over time as needed [34].

Some work has been made to formalize and improve the software implementation cycle. For example, the so-called cleanroom software engineering approach bases software development on statistics [53]. Nevertheless, such practices have not fully been assimilated by the software development community thus far, even though its benefits are quite evident [54].

In terms of software development of HCI systems, Reference [55] describes the idea of the separation of the implementation of such interactive systems into a computational and communicational element, such that the developer tends to each of these specialties one at a time. This practice proves quite useful in the coding process, since the task of development can be split in two categories, interface management and dialogue management (Figure 17⁹). This concept can be directly applied to the dual development of electronic databases and their HCI interfaces.



© [1991] IEEE

Figure 17: Separation of an interactive system into a computational and a communication element [55]

⁹ Adapted from [55] with permission from Publisher.

1.2 Current Approach: Space System I&T Logs

As seen in the introductory section of space systems I&T, documentation is one of the configuration control processes that ensures a successful operation of space systems I&T. In order to do this, several reporting formats called “logbooks” or “logs” are currently used [13] in the industry:

- *I&T Daily Log*: A day-by-day record of all events that occur or have a direct effect on the system (powered up or not) kept by the operations manager. It is the place to record daily events on the product structure (e.g., opening and closing of bay doors, mating and demating of flight hardware, installation or removal of any test article, system power-up and power-down, pre-test setup and post-test tear down) and the scheduled tests/re-tests, troubleshooting, repairs, etc. that are conducted on the system. The I&T Daily Log shows the date, time and the event and who accomplished it.
- *I&T Test Log*: Used to document any significant (i.e., normal or abnormal) chronological events that occur during I&T testing. It is kept by the I&T Test Director/Conductor. The Test Log tracks test events chronologically and is also used as an aid to identify test scenario problems and hardware and software discrepancy events. Items to be recorded in the Test Log include:
 - Date(s) and times of test to be conducted.
 - Test Director/Conductor name and names of all test personnel associated with a given test.
 - Identifies the specific test(s) to be run and the test configuration and test article(s) to be tested.
 - Any test limitations or constraints; the daily record of test progress to include the test events (i.e., type of test, time test started, time test halted, time test continue, time test completed).

- Observations and all abnormal events, as well as the test procedure, test step and the time that they occurred during testing.
 - All test problems including: Discrepancy reports (DRs); troubleshooting performed; and the actions taken by the test team to correct problems. Record the standard and troubleshooting operational orders use in the I&T test log along with any information or remarks pertinent to the testing.
- *Subsystem Log*: Used for individual product subsystems. They record the individual maintenance or events pertinent to that subsystem on a daily basis. The Subsystem Log shows the date, time and the event and who accomplished it. These logs are maintained by the appropriate subsystem engineers.
 - *Mate/Demate (M/D) Log*: Used for reporting connector mates and demates (connections and disconnections) performed on test and flight parts. This I&T Log is especially important since it contains the full connection history of the space system, and is heavily used throughout the I&T mission.
 - *Material Mix Record (MMR) Log*: Used to track the material mixes preparations used for adhesive applications on flight hardware.
 - *Discrepancy Report (DR) Log*: The Log is used to document and track the status of all DRs generated on the product system by either testing or daily operations. The DR Log contains the date it was generated, code to identify the type of DR, test procedure or test where DR was encountered, a short title to describe the DR, status code for each DR (i.e., in review, being investigated, being corrected and verified, awaiting parts, etc.), and a date when the DR was cleared.

For every I&T operation, a record must be created in the dedicated I&T Log. Each step documented in the record needs to be signed off on by both the I&T operator and the QA engineer. In paper-based I&T Logs, once the records are created, it is the responsibility of the QA engineer to archive them for future reference.

Figure 18 shows a sample paper-based M/D Report used at NASA JPL. Before the idea of E-Logbook took shape, paper-based reports needed for initial tests were created for the LAT I&T project, as seen in Figure 19, Figure 51 and Figure 20. These reports were used as reference in the process of defining the initial set of requirements for the I&T Logs, as will be described in the following Chapter of this dissertation.

Jet Propulsion Laboratory

T No 6311

WIRE RECORD & CONNECTOR

SERVICE RECORD TAG

Assembly _____

Ref Desig _____ Serial # _____

Wired On Fixture # _____

THIS TAG SHALL BE MAINTAINED UP TO DATE & MAY BE REMOVED ONLY AS AUTHORIZED BY THE COGNIZANT ENGINEER.

WIRE

SPEC	TYPE	MFGR	LOT NO.	SPEC	TYPE	MFGR	LOT NO.

CONNECTOR

Ref. Desig. & Serial No.	M	U	M	M	U	M	U	M	U

Figure 18: Mate/Demate Report used at NASA JPL

UNIT DESCRIPTION: TEM / TPS CONNECTOR MATE / DEMATE

Connector(s)	Authorized by	Date	Mate or Demate	Flight or Test	Verify Power Off	Pre-mate Inspect	ESD Bleed and Connector Mate	Final Inspect
Connector Reference Designator	Procedure & para of NCR	M/D/Y	M or D	F or T	Emp. ID#	Emp. ID# QA	Emp ID# QA	Emp. ID# QA
J34	D3-0497	8-23-04	M	T	TL	TL	TL	TL
J10	D3-0497	8-23-04	M	T	TL	TL	TL	TL
J10	"	8-23-04	D	T	JB	N/A		TL
J11	"	8-23-04	M	T	TL	TL	TL	TL
J12	"	"	D	T	TL	TL	TL	TL
J13	"	"	M	T	TL	TL	TL	TL
J14	"	"	M	T	TL	TL	TL	TL
J15	"	"	M	T	TL	TL	TL	TL
J16	"	"	M	T	TL	TL	TL	TL
J17	"	"	M	T	TL	TL	TL	TL
J18	"	"	M	T	TL	TL	TL	TL
J19	"	"	M	T	TL	TL	TL	TL
J20	"	"	M	T	TL	TL	TL	TL
J21	"	"	M	T	TL	TL	TL	TL
J22	"	"	M	T	TL	TL	TL	TL
J23	"	"	M	T	TL	TL	TL	TL
J24	"	"	M	T	TL	TL	TL	TL
J25	"	"	M	T	TL	TL	TL	TL
J26	"	"	M	T	TL	TL	TL	TL
J27	"	"	M	T	TL	TL	TL	TL
J28	"	"	M	T	TL	TL	TL	TL
J29	"	"	M	T	TL	TL	TL	TL
J30	"	"	M	T	TL	TL	TL	TL
J31	"	"	M	T	TL	TL	TL	TL
J32	"	"	M	T	TL	TL	TL	TL
J33	"	"	M	T	TL	TL	TL	TL
J34	"	"	M	T	TL	TL	TL	TL
J35	"	"	M	T	TL	TL	TL	TL
J36	"	"	M	T	TL	TL	TL	TL
J37	"	"	M	T	TL	TL	TL	TL
J38	"	"	M	T	TL	TL	TL	TL
J39	"	"	M	T	TL	TL	TL	TL
J40	"	"	M	T	TL	TL	TL	TL
J41	"	"	M	T	TL	TL	TL	TL
J42	"	"	M	T	TL	TL	TL	TL
J43	"	"	M	T	TL	TL	TL	TL
J44	"	"	M	T	TL	TL	TL	TL
J45	"	"	M	T	TL	TL	TL	TL
J46	"	"	M	T	TL	TL	TL	TL
J47	"	"	M	T	TL	TL	TL	TL
J48	"	"	M	T	TL	TL	TL	TL
J49	"	"	M	T	TL	TL	TL	TL
J50	"	"	M	T	TL	TL	TL	TL

Personnel that is Mate/Demate certified:

Connector / Bracket R/D:

Figure 19: Sample paper M/D Report used initially for the LAT I&T Project

TEST DATA SHEET		Unit S/N: CART # 1	Date/Temperature: 12/03/04	
Title: 5.1.3 Test Equipment		Operator: TL	QA:	
Para./ Step	Test Equipment Description, Manufacturer	Model/LAT Number	Serial/Rev. Number	*Cal./Val. Date
5.1.3.1 - 1	Record Model/LAT number, Serial/Revision number, Calibration due dates and Validation date for all equipment used in this procedure:			
	VME Crate, Dawn VME Products, 11-1011777-2119	VME64x (series 767)	GLAT0888	N/A
	VME, TST-STP Trans card	LAT-DS-00999	GLAT0220	N/A
	VME SBC MVME2304 card, Motorola PN	MVME2304-0123	GLAT0330	N/A
	VME LCB Mezzanine card	LAT-TD-00860	GLAT1308	N/A
	LCB Transition board cable	LAT-DS-03247	GLAT1223	N/A
	DC Power supply, BK Precision	1697 1786A	GLAT0944	06/05
	Uninterruptible Power Supply	Tripp-Lite SU1000 or equivalent	GLAT1594	N/A
	28 Volt supply cable	LAT-DS-03246	GLAT1586	12/05
	PS Control cable	LAT-DS-04831	GLAT1582	12/05
	TEM to GASU cable	LAT-DS-02106	GLAT1573	11/05
	CAT5 Ethernet cable	TRD855PL-50	N/A	N/A
	CAT5 Ethernet cable.RS-232	TDC003-7 (RECO98M connectors)	N/A	N/A
	Digital Multimeter, Fluke/Meterman	* 87-III/38XR	N/A	N/A

This column is for recording the calibration due date for a given piece of equipment or the date that EGSE was validated.

** Do not substitute other DMM's

CAUTION: Fluke 87-III and Fluke 87-V are not the same, Fluke 87-V is not allowed.

Figure 20: Sample paper EGSE Validation Report used initially for the LAT I&T Project

1.3 Prior Art

One of the initial steps taken in the development of this research was the investigation of whether the application of electronic database technology to document space system I&T operations had taken place in the field.

This investigation was to some extent impeded by the fact that most of the major companies that carry out space system I&T operations are required to maintain security clearance. Nevertheless this researcher was able to learn about the current practices in these companies in general terms through interviews with members (past and present) of their personnel.

In an interview with personnel of Lockheed Martin's space systems I&T facility at Sunnyvale in April 2004, this researcher learned that the completion of I&T logging tasks in this facility was performed in paper format. This was the case even though there existed a clear shift to the use of electronic database technologies for I&T operations, in particular through SAP.

In interviews with personnel in Space Systems Loral's I&T facility at Mountain View, this researcher learned that several electronic databases had been developed and were already in use for I&T operations. For example, a system called *Flexys* maintained the work order database and also provided links to other databases, like the Discrepancy Report database, for example. This was especially useful for the QA personnel.

This facility also maintained the only reference this researcher found at this writing of an electronic version of one of the I&T Logs, the MMR Log. In Space Systems Loral at Mountain View, this I&T Log was maintained electronically, using html format to display material mix record information. Nevertheless, the rest of the I&T Logs, including the M/D Log, were still maintained on paper.

With regard to the investigation of NASA practices, reference [13] shows how NASA maintains high configuration control standards for each I&T activity, but still performs these activities mainly on paper. When contacted early in 2004, NASA JPL demonstrated great interest in learning about the results of this research.

Investigation of Boeing showed that this company mounted a concerted effort to use innovative technologies to improve the process of both airplanes and space systems. In a reference found in 2005 [59], Microsoft described how Boeing used computer networking power as a solution to the storage of paper documentation, or its reduction:

“Boeing manufacturing facilities build planes, satellites, and other aerospace products. In those facilities, quality control and inventory control clerks perform critical quality and inventory control evaluations—a process that requires completing numerous paper-based forms. Later, after the quality control personnel walk from the manufacturing floor to their offices—a long distance that is sometimes referred to as the “4-mile trek”—they transfer this paper-based data into a computer database.”

Boeing also has spread the use of tablet PCs on the I&T plants to provide information to the I&T personnel on the spot about assembly drawings, work orders and test procedures. Nevertheless, this reference shows that computing power was used to maintain documentation created in paper.

No more references were found on the use of electronic technology to document space systems I&T operations.

The research shown in this thesis work also benefited from technology developed in the field of Particle Physics, thanks to the development framework at SLAC. In order to implement the I&T Daily Log, the GLAST project learned about the use of an electronic Shift Log at SLAC for the 24 hour operation of the BaBar experiment [56] [57].

Day Shift for Wed, 13-JUL-2005

	Previous	BaBar Shift Logbook						Next
	Day Shift for Wed, 13-JUL-2005							
BaBar Home	Help	Search	Detector Operations	Shift Index	Shift Accounting	List Runs	Shift Schedule	

You are logged in as chee

Help on page usage: [Field Usage](#)

Shift Takers:

LEADER: Julie Malcles **DQM:** Robert S. Kroeger

General Note:

```

Shift plan: take data

Shift summary: we took data with double trickle and lumi>6e33
most of the time.

- morning: lots of backgrounds + dead time. One beam abort due
to two
SVT diodes trips (trip summaries are shown in the log). MCC had
a
problem with the update of babar StripTool variable, but nothing
was
wrong on our side.

- afternoon: data taking smoother. There were few beams losses,
but

```

Figure 21: Partial snapshot of a Babar Shift, html based

Figure 21 shows a snapshot of the information displayed in html format for one of the BaBar Shifts. The Shift Log developed in this research was used by more subsystems beyond LAT I&T operations (see Figure 30, Chapter 3) through the use of an Oracle database similar to the one created by Paul Raines for the BaBar experiment at SLAC [57].

The results shown in this section indicate that, even though there is a clear shift to the use of electronic database technology in the industry, factors such tradition, reliability and difficulty of implementation have delayed its application to the documentation of space system I&T operations.

1.4 Contributions

The ultimate goal of this research is the creation of a successful electronic system that can replace the use of paper for space systems I&T operations. To this end the specific contributions made by this research are the following:

- *Design of a comprehensive system of logbooks to document space system I&T operations:* Researched the current practices in the process of space systems I&T and identified the documentation procedures and controls required to ensure the success of the I&T mission. Developed a complete list of requirements for each I&T Logs needed to fulfill this task.
- *Created and optimized an electronic database system based on the relational model in order to maintain the documentation of I&T operations:* Designed and optimized an electronic database structure based on the requirements defined for the documentation of space systems I&T operations. Focused on a database design that is independent of the experimental application for the LAT I&T effort and can easily expand for use, not only in any other space systems I&T project but also in any technological integration and test endeavor requiring an electronic logging system.
- *Designed a complete human-computer interactive system that improves the process of documentation of space systems I&T operations:* Created a human – computer interactive system of graphical user interfaces to enter and retrieve data from the electronic logging database schema. Focused the implementation of requirements on improving the logging procedures and alleviate the user’s workload. Ensured that the user’s interaction with the system represents an improvement upon the current process and use of paper.

- *Proved technology concept through the operation of a major space system I&T project:* Created the necessary infrastructure and applied the resulting technology to the operation of the Gamma Ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Integration and Test (I&T) facility at the Stanford Linear Accelerator Center (SLAC). Avoided permanent loss or corruption of the data stored in the system throughout its 19 months of operation.

The following Chapters will address how this research was accomplished. Chapter 2 is dedicated to formalizing the definition and requirements of the I&T Logs that were created for E-Logbook, as well as its design concept and tools used throughout its development. Based on these concepts, Chapter 3 describes E-Logbook's database schema, and Chapter 4 its human-computer interface. Finally, the experimental setup and results of its application for the GLAST LAT I&T project will be addressed in Chapter 5 and Chapter 6, respectively. Conclusions and future directions will be shown in the final Chapter of this thesis.

Chapter 2

“Requirements before analysis; requirements before design.”

Stanley I. Weiss and Michael S. Williams [60]

2 E-Logbook Components

Chapter 1 describes how in order to design an efficient electronic tool that replaces paper-based logging activities in space systems I&T operations, research must proceed on multiple fronts.

First, the process of space system I&T needs to be investigated in order to identify which critical I&T Logs are typically used. This effort also involves understanding the needs of the project that is going to serve as experimental platform for their electronic implementation: the GLAST LAT I&T project.

Second, understanding how electronic databases work based on the relational model is critical in order to ensure that all the capabilities of this technology can be made part of the design.

Third, research on the most optimum design process to implement interactive software needs to be performed in order to ensure that the software maximizes the successful transition by the front end user from paper to computer.

Finally, configuring an optimum developing platform from the initial steps of design, as well as the proper experimental setup for its use in the LAT I&T project, ensures the successful completion of this task, especially considering the significant time constraint and scope of the work at hand.

This Chapter describes the fundamental results obtained in all these areas of research, which made the creation of E-Logbook possible. It also introduces the design and results presented in the remainder of this dissertation.

2.1 E-Logbook Component Requirements

The following components were identified for E-Logbook in all levels of implementation:

- *I&T Log Components*: Logbooks needed to successfully carry out the I&T mission.
- *Design Components*: Components needed to create an interactive electronic database, and the relationship between them.
- *Development Platform*: The hardware and software needed to successfully implement the research at hand and provide the necessary tools to create the experimental setup.

The following sections of this Chapter will describe the requirements defined for each of these components, which provided the backbone of E-Logbook's design and implementation.

2.2 I&T Log Components

The I&T Logs required for E-Logbook were initially compiled by this author and the LAT I&T Manager based on the research described in Sections 1.1.2 and 1.2 of Chapter 1 and GLAST LAT I&T requirements [61]. In some cases, special reports for certain I&T Logs were also defined. As a result, a requirements document was created [62].

As user feedback was obtained during the development phase, new report formats were identified and implemented, resulting in the final list of required I&T Logs and Reports that will be shown here. For clarity, the tables showing the required fields for each I&T Log can be found in Appendix A:

- **User Log** (Table 22, Appendix A): The User Log is one of the most important I&T Logs defined for E-Logbook. It maintains the accurate account of all users currently active in the I&T facility. An operator cannot enter data in E-Logbook if a record with his information has not been previously entered in the User Log.
- **Shift Log** (Table 23, Appendix A): In intensive projects such as space systems I&T, there are typically several operator shifts per day. The Shift Log, also known as the “Daily Log”, maintains a record of all the activities performed during a shift in the I&T facility, such as ground software testing or FHW installation. It also assists in the communication of information from one shift to the next (shift hand-off process). The Shift Log contains a list of the personnel that worked on the shift (shifters), a summary/plan of the shift, and the list of activities performed in the shift classified by subsystem. Customized to the LAT I&T effort, the Shift Log is also required to list the ground test output information: the so-called test runs, and to link each run to the corresponding Run Report generated from the data stored by the ground test software in E-Logbook.
- **Run Report** (included in Table 23, Appendix A): The Run Report is a report of the data generated by each test run by the ground software on the EM, CU and FHW of the space system. The Run Report must be accessible for each run of the Shift Log’s test run list.
- **Mate/Demate (M/D) Log** (Table 24, Appendix A): The M/D Log is the I&T Logbook required to maintain a record of every time a piece of FHW is connected (mated) or disconnected (demated) either to other FHW or to EGSE. The M/D Log is responsible for keeping track of the current connection installation state of

the space system, as well as of the additional equipment connected to it for testing. Each mate/demate typically involves several steps that need to be completed, and documented as completed (signed off) by both the operator and the QA that supervises the operation. For each connector in the M/D Log, a M/D Report that shows its entire mate/demate history is required. Figure 22 shows a mate torque step for one of the LAT TKR modules.

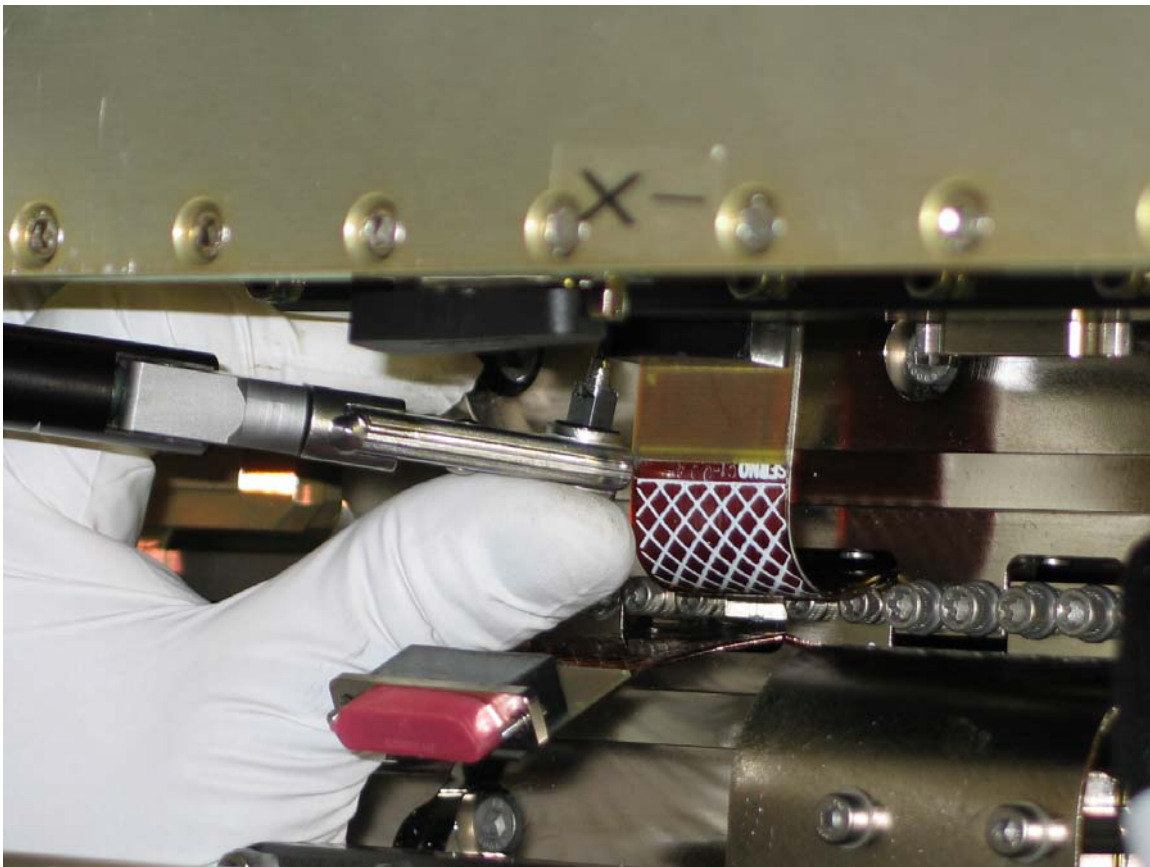


Figure 22: Connector mate for one of the LAT TKR modules

- **Unit M/D Report:** A special report obtained from the M/D Log that contains the latest mate/demate state of each connector in a certain FHW component.



Figure 23: First CAL Module Installation into the LAT

- ***Flight Hardware (FHW) Log*** (Table 25, Appendix A): The FHW Log is an I&T history of the space system hardware. It therefore maintains the current configuration of the space system in production, as well as tracking all the changes in installation made throughout the component’s history, such as if a FHW component had defects and needed to be removed. Each FHW component that is integrated and tested has a record in the FHW Log. In order to create a record in the FHW Log, the different installation stages that need to be completed, like flatness verification or torques applied, must be recorded. A minimum set of required information is entered and signed off by both the operator and the QA

supervising each installation stage. Figure 23 shows the installation of the first CAL module into the LAT GRID. The first 2 TKR modules already integrated into the GRID can also be seen underneath in the Figure.

- ***FHW Matrix Report*** (Table 26, Appendix A): The FHW Matrix Report was created based on user feedback, as a customization of the FHW Log. It is a snapshot of the current FHW installation state, displayed in grid format for easy association with the physical grid. Figure 42 shows a snapshot of the final implementation of the FHW Matrix Report, which can be compared to the real LAT system configuration shown in Figure 5 and Figure 9.
- ***Flight Software (FSW) Log*** (Table 28, Appendix A): The FSW Component Installation Log is the software counter part of the FHW Log: It is required to maintain a history of all the installations, revisions and patches made to any piece of FSW installed in the space system for use during the mission.
- ***Material Mix Record (MMR) Log*** (Table 27, Appendix A): Due to space environmental effects, any kind of material used for FHW bonding or coating needs to be controlled and tested to ensure its durability throughout the lifetime of the mission. In the case of adhesives that require material mixing, the control process involves the testing of the mix hardness after a wait period that is typically one week. The MMR Log must therefore maintain a history of material mixes performed for bonding or coating of FHW during the satellite integration. Taking advantage of the electronic capabilities of E-Logbook, each time a mix is created, a Material Mix Record number (MMR#) must be automatically generated, so it can be referenced in the appropriate integration procedure and FHW Log record to aid in determining where the material is used. Figure 24 shows the coating of a FHW connector using a material mix previously controlled with the MMR Log.



Figure 24: Coating of one of the LAT FHW Connectors using a mix recorded in the MMR Log

- ***Electronic Ground Support Equipment (EGSE) Log (Table 29, Appendix A):*** The EGSE Log is required to maintain a history of EGSE hardware and software equipment that is used to perform tests on the FHW. The EGSE equipment is typically configured in different setups on carts that are moved, as needed, around the I&T facility. These configurations or setups need to be controlled and validated throughout the production phase to ensure the safety of the FHW to which they are connected [18]. The EGSE components included in an EGSE configuration can be EGSE hardware and software, but also FHW, FSW or even other EGSE setups. Most EGSE component validations have an expiration date that needs to be monitored. Each time an EGSE component validation is about to expire, the entire EGSE setup needs to be revalidated and a new record with the updated validation date must be recorded. Figure 25 shows one of the EGSE setups in the LAT I&T facility.

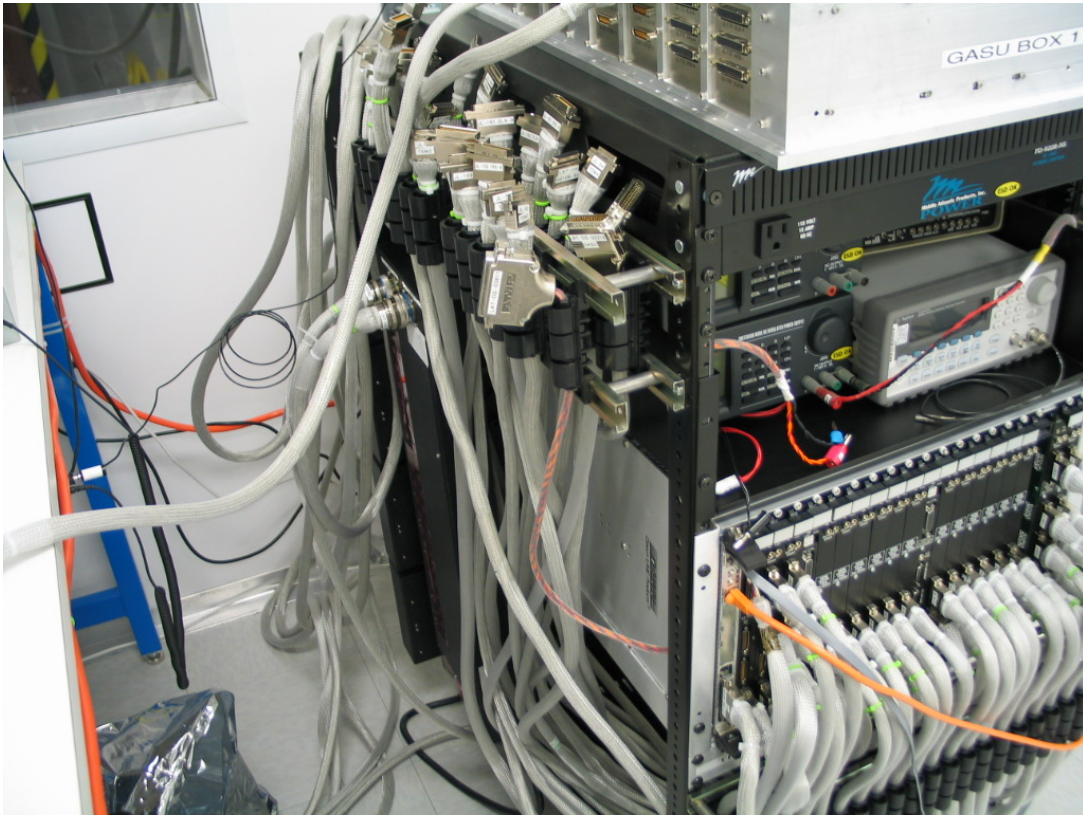


Figure 25: EGSE setup in the LAT I&T Facility

- **Configuration Report** (Table 31, Appendix A): A Configuration Report is required for each I&T Log that contains FHW or FSW. It is used to verify the instrument's configuration status prior to major test events. For E-Logbook, a Configuration Report was created for the FHW, FSW and EGSE Logs.
- **Configuration Request Log** (Table 30, Appendix A): The Configuration Request Log is required to keep track of which Configuration Reports were generated prior to major test events. It contains the name of the operator that created a report via the Configuration Log or the FHW Matrix Log, the date and time the report was created and the report type (FHW, FSW, EGSE or FHW Matrix).

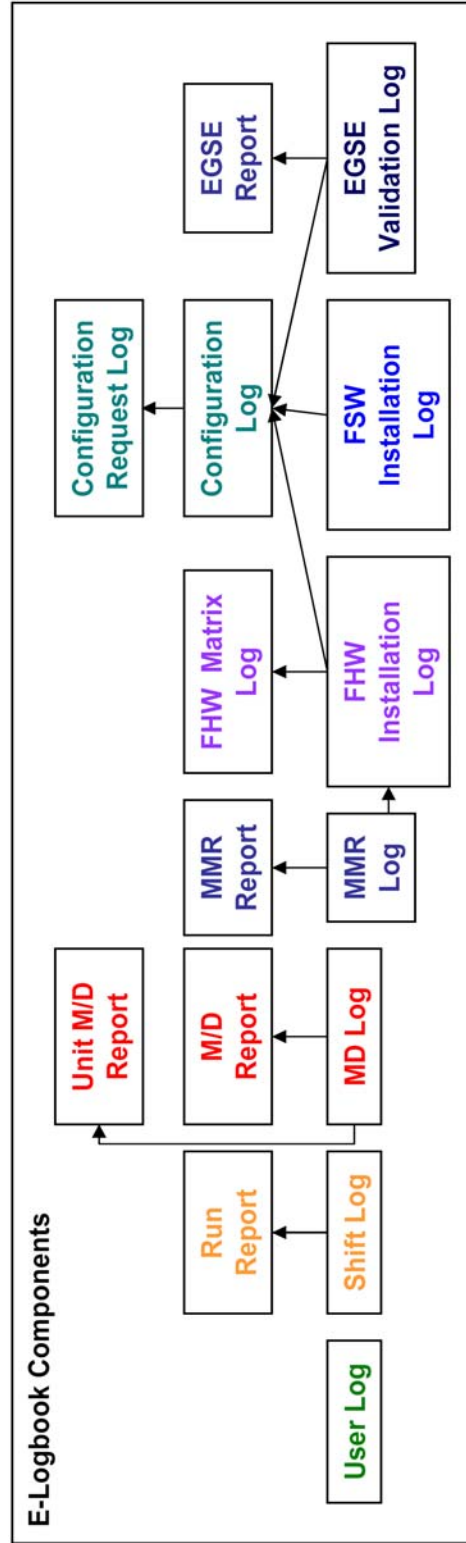


Figure 26: E-Logbook's I&T Log Configuration

Figure 26 shows the final configuration of I&T Logs and Reports that were defined for E-Logbook. The arrows indicate the relationships between the different Logs. For example, the FHW Log populates the FHW Matrix Report as well as the Configuration Report. The Configuration Report request information is stored in the Configuration Request Log.

2.3 Design Components

One of the main goals of this research is to create an optimized interactive electronic tool that maintains the required logbooks defined for space systems I&T. As seen in Section 1.1.5, a good course of action to carry out this effort is separating the design effort of computational and communication elements of the interactive tool.

In E-Logbook, the computational and communication components are identified as follows:

- **Computational Component:** The electronic relational database schema or structure of tables, indexes and keys that holds the data.
- **Communication Component:** The set of graphical user interfaces (GUI) and input/output (I/O) devices (mouse, keyboard, bar code reader) that communicate with the user, as well as the underlying human-computer interface (HCI) system that provides the connectivity mechanism with the underlying database schema.

The computational element is the fundamental structure that must be safely maintained throughout the implementation effort. It can be noted that E-Logbook cannot survive without this component, while there are multiple communication elements, such as the DBMS itself. Nevertheless, the success of E-Logbook relies on the creation of a HCI-GUI system that is the best communication element to the database schema. Without success on this effort, the integrity of the data in the database schema does nothing for the applicability of E-Logbook in the industry.

The separation of E-Logbook into a computational and communication element can also benefit greatly from the use of server-client architecture as introduced in Section 1.1.3. The database schema, which resides on the server, is accessed through the HCI-GUI system installed in multiple client workstations.

This architecture minimizes the database maintenance effort, since there is only one database in one server. On the other hand, it maximizes the connectivity maintenance, as the incorruptibility of multiple and synchronous I/O client activities needs to be enforced. DBMS such as MySQL were developed precisely to provide the necessary mechanisms to maintain this effort.

Figure 17 therefore becomes Figure 27 in the context of E-Logbook. The database schema resides in the server, which is then manipulated by the HCI-GUI system accessed by the different client workstations.

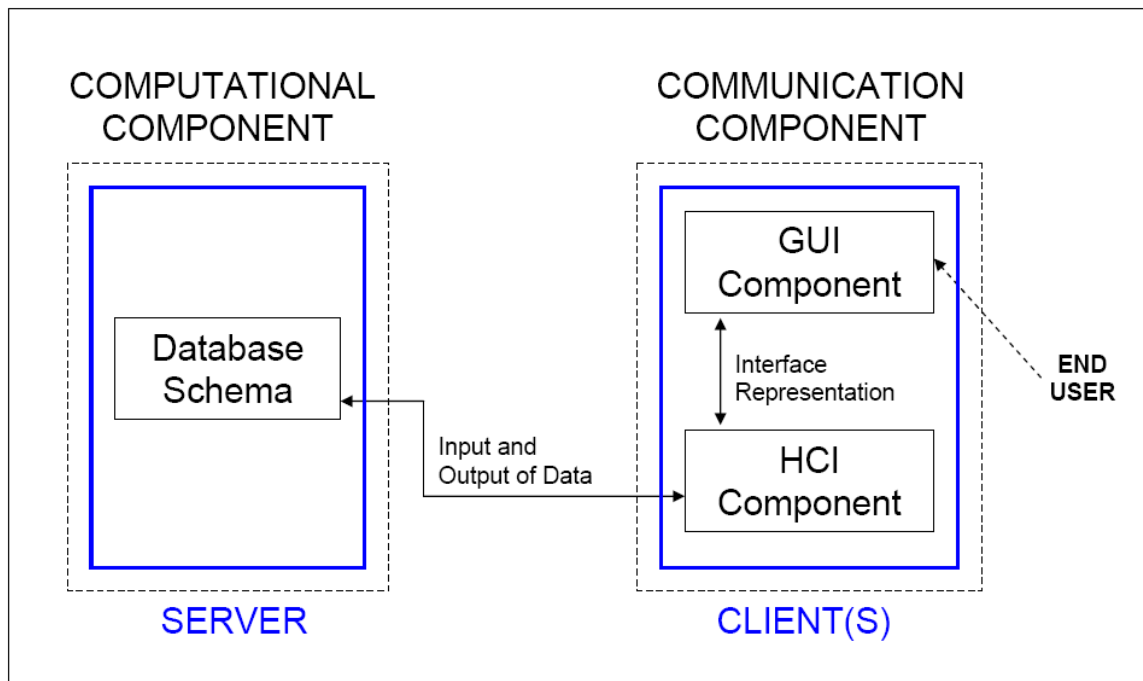


Figure 27: E-Logbook's interactive component system

2.4 Development Platform Components

The choice of E-Logbook's design tools was based on two main premises:

- *A platform independent tool:* The requirement of using platform independent tools to create a platform independent system was introduced to ensure that operating system constraints would not jeopardize the successful implementation of E-Logbook in the industry.
- *Use of open source software:* In the spirit of openness to the scientific community the selection of E-Logbook's development tools was constrained to the use of software following open source agreements [63] [64], or with license agreements that would allow the creation of an open source tool.

The following sections describe the final set of hardware and software components that was used for the development and implementation of E-Logbook based on these requirements.

2.4.1 Hardware Components

The hardware used to develop E-Logbook consisted of 2 Windows XP based machines (a workstation and a laptop for remote testing) with accessibility to a Linux server via SSH and Win-32 tools. This hardware used all the networking capabilities provided by SLAC.

The development of E-Logbook was also only possible thanks to its application to I&T operations of the FHW, FSW EGSE and MGSE built for the GLAST LAT project.

2.4.2 Software Components

In order to design an electronic database interactive system there are at least two software components needed: A DBMS, and the programming language that creates the HCI-GUI system. Consequently, the selection of E-Logbook's development software followed the

architecture described in Figure 28. Additional tools needed are also shown in this figure in different colors, depending on their role in the implementation effort.

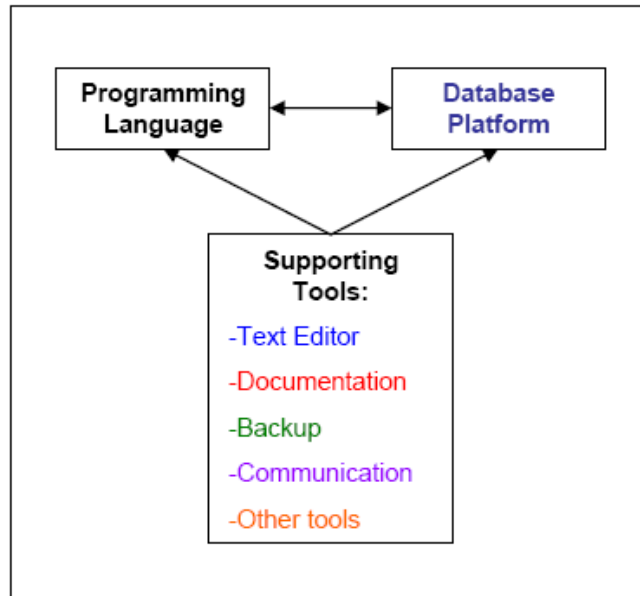


Figure 28: Programming tools architecture schema used in the development of E-Logbook

The following sections describe the complete set of development software that was selected based on this architecture and used in the course of implementation of E-Logbook.

2.4.2.1 Python

The selected programming language was Python, due to its open source characteristics and the fact that SLAC had previous experience with this language [65] [66] [67].

“Python is a simple but powerful object-orientated language. Its simplicity makes it easy to learn, but its power means that large and complex applications can be created. Its interpreted nature means that Python programmers are very productive because there is no edit/compile/link/run development cycle.” [68]

The Python implementation is portable: it runs on many brands of UNIX, on Windows, OS/2, Mac, Amiga, and many other platforms.” [68]

“The Python implementation is copyrighted” [65] [...] ”but freely usable and distributable, even for commercial use.” [68]

2.4.2.2 MySQL

Winner of the 2004 Developer.com product of the year award, the MySQL open source platform was selected as the database framework to create the structure of E-Logbook:

“The MySQL database server is the world's most popular open source database. Over six million installations use MySQL to power high-volume Web sites and other critical business systems — including industry-leaders like The Associated Press, Yahoo, NASA, Sabre Holdings and Suzuki.

MySQL is an attractive alternative to higher-cost, more complex database technology. Its award-winning speed, scalability and reliability make it the right choice for corporate IT departments, Web developers and packaged software vendors.” [69]

With its multiple design tools such as the control center, query browser or administrator, MySQL enabled the refinement of the database structure for each release. It also helped monitoring the data as well as the creation of the backup system required to meet the data survivability requirement.

2.4.2.3 Trolltech Qt Designer

Several available tools were investigated in order to optimize the creation of E-Logbook’s system of GUIs. For example, the standard Python interface with the Tk GUI toolkit is called Tkinter [70]. Nevertheless, Qt Designer was finally selected for this task

since it was the best program in providing “what you see is what you get” (“*WYSIWYG*”)¹⁰ design tools:

“Qt is a cross-platform C++ application framework developers can use to write single-source applications that run natively on Windows, Linux, UNIX, and Mac OS X and embedded Linux. Qt has been used to build thousands of successful commercial applications worldwide, and is the basis of the open source KDE desktop environment.” [72]

Qt was especially helpful in the initial design of the visual aspects of the GUI. Once the different window components and their position were defined, the implementation of its usability was primarily performed via code development in Python [43].

2.4.2.4 Other tools

Third party tools were also needed to complement MySQL, Python and Qt in the development, testing and documentation of E-Logbook. These tools are the following:

- *PyQt*: Due to the fact that Qt is a C++ application, a set of Python bindings was needed in order to use this application. These bindings were provided by Riverbank, an independent software developer company that specializes in open source software technologies [73].
- *SciTe*: Scintilla’s SciTe text editor has become a generally useful editor with facilities for building and running programs [74]. SciTe recognizes the Python language and adapts the script text to this programming language for easy edit.
- *Doxygen*: Doxygen is a documentation system for among others Python [75]. It generates an on-line documentation browser (in HTML) and/or an off-line reference

¹⁰ The term describes a user interface that allows the user to view something very similar to the end result while the document or image is being created. It implies the ability to

manual (a .CHM file in Windows). It also offers support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and UNIX man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code. It can also be configured to extract the code structure from undocumented source files. This is very useful to quickly find one's way in large source distributions. Finally, it helps to visualize the relations between the various elements by means of dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically. Doxygen is copyrighted and can be used, copied, modified and distributed under the terms of the GNU General Public License.

- *CVS*: To ensure version control and documentation E-Logbook resorted to CVS. CVS is robust and open source [76]. It records the history of source files and helps track changes and old versions when needed. It also allows for branching from a certain version so that multiple developers can work on a product without stepping over each other's code.
- *WinCVS*: WinCVS is a GUI front-end for CVS written in C++ and distributed under GNU General Public License agreement [77]. It provides easy interface with the CVS system outside the command line world.
- *WinMerge*: WinMerge is an open source visual text file differencing and merging tool for Win32 platforms. It is highly useful for determining what has changed between project versions, and then merging changes between versions [78].
- *JIRA*: JIRA is a web based issue tracking and project management system. It helps create, track and manage the creation of issues related to bugs, improvements or new features of a product that emerge during its development and implementation.

modify the layout of a document without having to type or remember names of layout commands [71].

“Tracking and managing the issues and bugs that emerge during a project is a critically important task, but one that few teams do effectively.” [79]

SLAC uses JIRA in many of its projects, including the GLAST LAT I&T project.

Based on Figure 28, E-Logbook’s final design software architecture is shown in Figure 29.

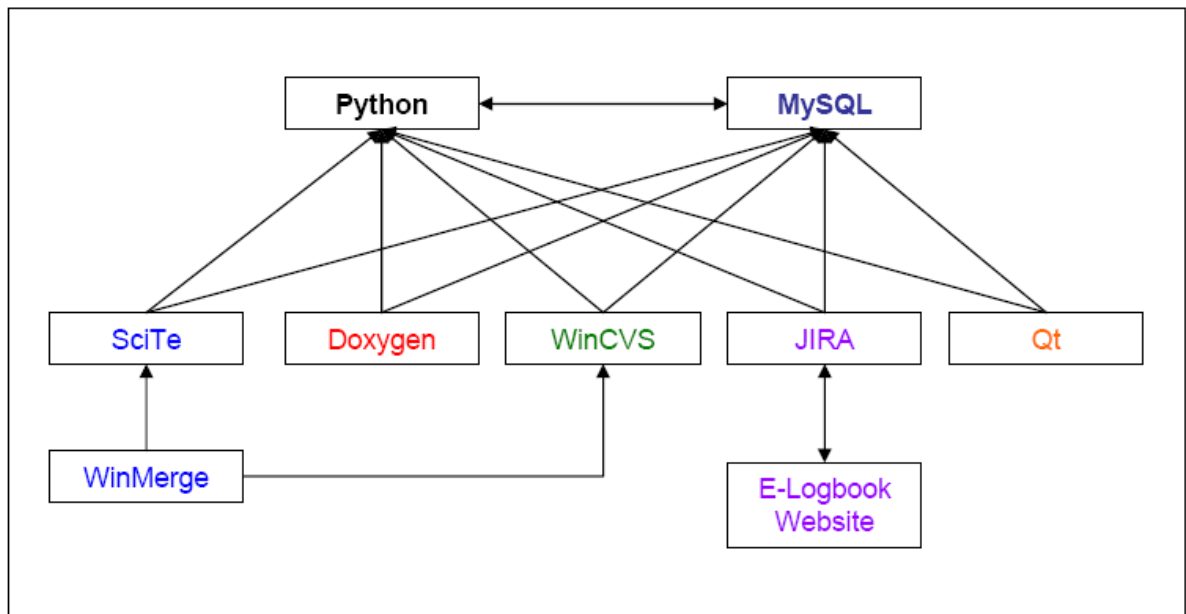


Figure 29: E-Logbook’s Development Software Architecture

2.5 E-Logbook Component Implementation

The following Chapters of this thesis work will describe the component implementation based on the requirements defined here for E-Logbook’s I&T Log components, design components, and development platform.

Chapter 3 focuses on the design of the electronic database component of E-Logbook based on I&T Logs field requirements compiled on Appendix A. Chapter 4 describes the design and optimization of the HCI-GUI system that was developed as E-Logbook’s

communication element. Chapter 5 shows the E-Logbook's code structure and experimental setup on the LAT I&T facility created based on the results obtained on its development platform. Finally, the results of this combined effort will be analyzed in Chapter 6. Chapter 7 concludes this thesis work and also shows ideas for future work.

Chapter 3

3 E-Logbook Electronic Database System

In Chapter 1 a database schema was defined as the set of tables and underlying connections that define the database structure and hold the data. This Chapter is devoted to the description of the database schema designed for E-Logbook during the course of LAT I&T. It explains the decisions that determined the creation of the initial design, as well as how the different tables were optimized using the concept of relational databases and user feedback. The final part of the Chapter will show the different statistical and performance results obtained regarding E-Logbook's database behavior during its experimental usage for the LAT I&T effort.

3.1 Database Requirements

As described in Chapter 2, initial database field requirements were defined for E-Logbook in reference [62] based on the standard I&T Logs used in the industry customized for the LAT I&T effort. Further requirements were defined based on user feedback during the LAT I&T effort at SLAC. Finally, additional requirements were established based on E-Logbook's compatibility with ground software used for environmental testing at NRL.

The overall database schema requirements defined for E-Logbook were:

- *Database components:* The database schema will hold the required information for the different components as established in the requirements documents [62] and from user feedback during operations.

- *Database Oracle replica:* A replica in the Oracle DBMS of the MySQL E-Logbook database will be provided, for usage with other ground support software in the scope of the GLAST project at SLAC.
- *Database compatibility with other I&T support tools:* Compatibility with other I&T support tools that interface with the database schema will be assured for every release of E-Logbook, during the entire course of LAT I&T at SLAC and environmental testing at NRL.
- *Database robustness:* The database must be robust against data loss or corruption. This applies to existing records as well as to new information entered or output from the database. A backup mechanism must be devised to ensure that data is not lost during critical events such as facility power outage, fire, flood or earthquake.
- *Database access protection:* The database must be protected against unauthorized user access. Furthermore, the information already entered into the database will not be modified by the user under any circumstance.
- *Database user and timestamp recording:* User identification and date and time (combined in what becomes a *timestamp*) will be recorded as required for management purposes.
- *QA supervision:* User sign-off will be ratified with QA approval sign-off recording (user identification and timestamp) as required.
- *Database unit system convention:* All measurements entered in the database will be recorded in the project's official unit system; in the case of GLAST, the International System of Units.
- *Database comment recording:* The database must ensure the possibility of adding user comments to any log in order to record unforeseen additional information. The comment creator and the timestamp of creation must be recorded as well.

3.2 Database Design

The design of E-Logbook's database schema followed the four phases of database design described in Figure 14 of Chapter 1: requirements analysis, conceptual design, logical design and physical design.

The four phases were addressed with the goal of minimizing maintenance due to error. This would ensure that no major changes that could jeopardize the integrity of the existing data would be required. It would also minimize the time invested in every release to work out compatibility issues.

It was of equal critical importance to create a schema that could grow with time and that would be flexible to changes, as well being robust and reliable. This would ensure successful implementation within the strict time constraints of the overall LAT I&T schedule.

3.2.1 Requirements Analysis and Conceptual Design

E-Logbook's requirements analysis and resulting design concept are based on five concepts: database compatibility with interfacing DBMS, the use of global tables and table partition where necessary, optimizing the table update process, requirements implementation and database schema optimization.

3.2.1.1 Database Compatibility Issues

The design of the tables dedicated to the Shift Log in E-Logbook was performed in synchronization with the design of the Oracle database version used by other subsystems in the scope of the LAT project, which in turn was a customized implementation of the BaBar Shift Log, as explained in the prior art section of this thesis. This led to having to ensure compatibility between the Oracle and MySQL DBMS for the affected tables.

While the issues derived from this constraint do not pertain to this research, the design of the tables dedicated to the Shift Log was somewhat constrained by the design on the Oracle side, and could not be optimized. The tables affected by this issue were the Shift Log tables and the User Log table, as defined in the following section.

GLAST Shift Logbook
Swing Shift for Fri, Feb 17, 2006, 17:00:00

[GLAST Home](#) [Help](#) [Shift Index](#) [List Runs](#)

[Previous Shift](#) no existing next shift

Shift Takers:

INT Engineer: Roger Williams
INT Elec. Tech.: Phuc Hoang

General Note:

Plan

- Continue data runs per NCR 829 Disposition per lat-aids-01921
- Finish Burn archive DVDs for AIDS-01900, verify DVDs open uner windows, then delete from burn directory

INT Activities

[01:03] PH: Perform lat-aids-01921 step 90. make-up run #4. e2e 4-3 for 15 minutes at 10kHz. run id 600001160 - 600001161. passed.

[01:38] PH: Perform lat-aids-01921 step 90. make-up run #5. e2e 4-3 for 15 minutes at 10kHz. run id 600001162 - 600001163, FAILED.

Runs:

Run Report	Status	E2E ID	Serial No. (TEM ID)	Test Script Name	Test Output	SVAC Reports	Events / Errors	Duration (s)	Root Files	Start (GMT)	Particle	Orientation
600001168	PASSED	4/3	(TkrFMA, FM104.0) (TkrFM3, FM106.9)	end2end	Report Dir	digi recon config	413522/33	901	digi recon merit	2006-02-18 02:01:26	Cosmics	Vertical

Figure 30: GLAST LAT web based Shift Log

For example, one of the most heavily used tables in E-Logbook held the software test run information. This table, called *elogreport* (see the following section), underwent multiple changes and additions throughout the lifetime of E-Logbook, as more capabilities were needed by the data analysis (SVAC) subsystem (see Figure 11). These changes needed to be synchronized with the Oracle side to ensure all the data created in E-Logbook was

fully updated in the Oracle database version. Figure 30¹¹ shows the subsequent web based Shift Log application developed by the SVAC subsystem from the original BaBar Shift Log (see Figure 21). This application obtains the shift and run data entered in E-Logbook by the operators and the run test software (see the corresponding GUIs developed for E-Logbook in Chapter 4, Figure 40 and Figure 41) and expands the run test information with the analysis files.

It is worth noting here that even though this problem was exclusive to the LAT I&T effort, similar tables holding test output data are expected to be needed for other projects, such that the lessons learnt in this case study are not lost.

3.2.1.2 Global tables and table partition

As defined in the previous Chapter, E-Logbook required the implementation of 11 distinctive logs: User Log, Shift Log, M/D Log, FHW Installation Log, FSW Installation Log, MMR Log, EGSE Log, Configuration Report and Configuration Request Log, FHW Matrix Log and Questionnaire.

It was straightforward to envision a separate and distinctive table for each of these Logs, where the entire information exclusive to them was held. Nevertheless, in some cases instead of one table a set of tables was needed to specify a log. This happened for example in the case when the number of fields was high, and putting them all together in one table was simply not optimal in order to run database queries. In other cases, the fields were entered at different points in time. All of the records in the different tables related to a specific activity were connected with foreign key fields, where the record primary key of one of the tables would be referenced in foreign key fields created in the other tables in the set. Furthermore, during the experimental use of E-Logbook for the LAT I&T, new fields for some of the logs were identified and requested by the users,

¹¹ Adapted from Reference [80] to summarize content.

which resulted in an implementation effort to plan for this event and ensure compatibility and non-corruptibility within the overall database schema.

Additionally, a set of fields actively and indistinctively used by all of the logs was identified and the resulting so-called global tables and connecting foreign key fields were created. For example, instead of having to enter a connector's name each time the user created a mate or demate record, a table holding the connectors' names was created, so that the user would enter a connector's name once, and select it from a list from that moment on. This would facilitate the use of the connector not only in the M/D Log, but also in other related Logs such as the EGSE Log.

A final table dedicated to reporting the database release history was identified in order to maintain the database.

Consequently, three different types of tables were defined for E-Logbook's database schema:

- *Component tables*: Tables that compile exclusive information related to the different Logs, sometimes split into sets of tables to optimize the design.
- *Global tables*: Tables that group the fields used by all the Logs.
- *Maintenance table*: A table that maintains the database itself for release purposes.

3.2.1.3 Table update

A third conceptual hurdle in the design of the database schema was the record update process. Most DBMS allow modifying or "updating" existing records, which can consist of editing populated fields or adding information to empty fields on an existing record. Nevertheless, modifying existing records is responsible for most of the data loss or corruption in DBMS. The best course of action is to enforce, wherever possible, that all fields in a record are filled at once, the time the record is created.

In the case of E-Logbook, another reason to minimize record updates is compatibility issues with the Oracle side, as described in the previous section. The synchronization process does not ensure identical primary keys (typically record row numbers) for any given record. Therefore, if the information on a record is modified, developing a mechanism to ensure its update on the Oracle side is cumbersome. Solutions such as recreation of the whole table when the table record length is relatively small, or creation of additional flag fields, are needed.

Consequently, special precautions were taken in the design of E-Logbook to ensure that all fields were filled in at record creation wherever possible. In the case of the Shift Log tables, no record update either to edit or add fields was allowed. In the case of the user table, quite small in size (100 records expected at the time of project completion), the table was altogether recreated during each synchronization, so that identical primary keys were enforced and updates were therefore possible.

In the case of the Shift Log activity tables, the activity note could not be modified under any circumstance. Instead, if the note needed to be corrected or additional information was needed, a new record indicating the correction was created.

A final example is the case of the Hardness Test Result information of the MMR Log. Once a material mix is prepared, the operator must wait for a specified amount of time (typically 7 days) to verify that the mix satisfies hardness requirements. After this time has passed the operator must fill in the hardness test result fields of the corresponding MMR created at the time of the mix. Instead of adding the new information to fields of an existing record, a whole new table called *eloghardtest* (see next section) was created. The resulting new record was linked to the original MMR using a foreign key field.

In general, a major part of the implementation effort was dedicated to ensuring that once the record was created, all the necessary information was entered and that no user modifications were available to an existing record. Modifications to an existing record were only allowed if performed automatically by the program, such as through the population of the foreign key fields. Additionally, an extensive system of pop-up

windows was developed to warn the user of missing information while creating the record (more details in the next Chapter).

3.2.1.4 Requirements implementation process

For each of E-Logbook's 11 Logs, Reference [62] created an initial set of requirements that included purpose, capabilities, and fields involved. Even though it is not as intuitive as in the case of the HCI-GUI system described in the next Chapter, E-Logbook's database schema was also affected very much by user feedback.

As the different Logs were used, the operators detected fields that were missing, and so the database schema evolved to include new fields, and even tables, as needed. This process took place for almost every Log, and the final design differs in some cases quite dramatically from the original requirements defined in Reference [62]. The final list of fields implemented for each Log can be found in Appendix A.

A great example is the case of the FHW Log. Table 3 [62] is the original database field requirements that were defined for this Log. Table 25 in Appendix A shows on the other hand the final list of fields implemented. There were 4 fields added: *Comments*, *Flatness Value*, *Grounding Tool* and *Grounding Tool Expiration*. There were also two fields removed: label tape *QA ID* and *QA Timestamp*. Finally, the Torque value field type was changed from value to array values or *tuple*, to include a list of more than one torque values, and additional information on the torque. Consequently, the tuple recorded a list foreign keys pointing to a brand new table, *elogtorque*¹², which contained the necessary fields: *Torque Value*, *Torque Tool* and *Torque Tool Expiration*.

¹² Incidentally, the M/D Log required the same modification, and so the *elogtorque* table was classified as global table (see next section) and also used by this Log (see global tables section).

Component Installation Log

Unit Description	<text>			
Component R/D	<text>			
	Operator ID/Password	Operator Date Time	QA ID/Password	QA Date Time
Operator:	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Date Time: (MM/DD/YY HH:MM:SS)	<datetime>			
Authorizing Document	<text>			
Part Number:	<text>			
Serial Number:	<text>			
Weight (kg):	<value>			
	Operator ID/Password	Operator Date Time	QA ID/Password	QA Date Time
Flatness (per ICD):				
Verification:	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Torque				
Torque Value:	<value>			
Apply Torque:	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Torque Stripe MMR# (autogenerated from Material Mix Record):	<value>			
Witness Stripe	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Grounding				
Installation:	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Limit (per ICD/Spec):	<value>			
Measured Resistance:	<value>			
Label Tape Witness:	<ID/Password>	<datetime>	<ID/Password>	<datetime>
Label Tape (Lot #):	<value>			
Label Tape Expiration:	<date>			

Table 3: Original field requirements for the FHW Log

Another example is the EGSE Log. Table 4 [62] shows the original database field requirements and Table 29 in Appendix A the final list of fields implemented. In this case, a whole new table containing EGSE Component fields was created in order to include information on each Component that was part of a certain EGSE setup. Furthermore, a new algorithm¹³ that would determine the global validation expiration date from the earliest expiration date of the EGSE Component list was created. Finally, the capability of recording EGSE Components whose validation did not expire, and

¹³ To review this algorithm go to *updateValidDate()* in the script *assignEGSEImpl.py* located at *Code/egse/* of the attached DVD.

therefore did not need to be included in this algorithm (a N/A field with values 0 or 1), was implemented.

EGSE Log		
Operator ID	<ID/Password>	
Date/Time	<value>	
EGSE Name or ID#	<text>	
Serial #	<text>	
Hardware or software change	<H or S>	
Description of the change	<text>	
	ID/Password	ID/Password
Operator and QA verification	<ID/Password>	<ID/Password>
EGSE Validation	<ID/Password>	<ID/Password>

Table 4: Original field requirements for the FHW Log

3.2.1.5 Database optimization

A final stage in the design of E-Logbook’s database schema was its usage optimization. This involved not only the creation of proper indexes, but also of a smart query system.

Database indexing optimizes data retrieval, in that if the table is queried by a certain field, the speed of retrieval will be higher if an index was created for the field in question. In this fashion indexes are exactly like their literary or filing cabinet counterparts; they help the DBMS system to identify the relevant information in the database before accessing the database itself.

A case example on the benefits of indexing is found in the elogreport table, which, due to the fact that it is populated electronically by the testing software each time a test run was performed on the LAT, holds the biggest amount of data in E-Logbook’s database schema. As the number of records in this table increased, an increase in the downtime during query retrieval from the table was also noticed; which in the worst case was of around 10 seconds.

This problem was easily fixed once it became clear that the field the database was being filtered by in the query (*StartTime*) did not have an index. Once an index for this field was created, the down time decreased to the order of milliseconds.

An example of database query optimization was the M/D Log, the second most heavily used Log in E-Logbook. As the number of connectors and mate/demate records increased, the query that performed the filtering to populate the connector list became increasingly sluggish, more than 18 seconds of down time at the worst, with the result that almost every user issued a claim.

A smart fix to this problem resulted from the discovery that one of the Qt Designer functions used in the code to populate the list (*findItem()*) was extremely slow, and that the table in question, *elogcomponent*, could be sorted at the time of query such that the use of this function was avoided. The result was that the down time decreased to an optimum third of a second. The algorithm used for the query was also implemented for the rest of the Logs¹⁴.

¹⁴ To review this algorithm go to *loadConnectorList()* and *loadComponentList()* in the script *converter.py* located at *Code/tools/* of the attached DVD.

3.2.2 Logical and Physical Design

This section describes the results of the final two phases of the design of E-Logbook's database schema: logical and physical design. It first shows the final list of tables implemented, classified by table category: component, global or maintenance. Finally it describes the logical connectivity created between them, according to the guidelines described in the previous section.

3.2.2.1 Component tables

The tables created for each of E-Logbook's components are as follows. The tables have been grouped by Log. The complete list of fields, indexes, foreign keys and default values defined for each table can be found for clarity in the script on Appendix B.

- *Shift Log:*

- *elogshiftsummary*¹⁵: Holds the summary information of an I&T shift, when it starts and ends, the type of shift (owl, day or swing), and brief summary of the tasks allocated to the shift. The primary key of this table is generated with the shift date and type. It is used to tie all the tables related to the shift log together via foreign keys.
- *elogshifftaker*: Holds pointers to the users that signed in the shift.
- *elogshiftactivity*, *elogshiftproblem*, *elogshiftother*: These tables hold information related to main activities, problems and other kind of entries made by the users. They hold the entry note as well as a pointer to the shift identifier and the user identifier.

¹⁵ The table naming convention used for E-Logbook in MySQL is lower case, name always starting with *elog*.

- *elogsubsysactivity*, *elogsubsysproblem*, *elogsubsysother*: Similar to the previous tables, but dedicated to the I&T subsystems. They hold the entry note and subsystem type as well as a pointer to the shift identifier and the user identifier.
- *elogreport*: Holds the information related to the tests performed on the FHW as part of the I&T procedures, and is directly populated by the run test software¹⁶. Each record holds the information for a run; each run is the output of each test.
- *LICOS_states*: Holds default information related to the tests performed by a ground test software program called LICOS¹⁵.
- *LICOS_activities*: Holds final values for each run related to the tests performed by LICOS¹⁵.
- *elogshiftschedule*: Available but not in use in E-Logbook.

- M/D Log:

- *elogmatedemate*: Holds the information related to mate or demate activities, such as procedure followed, if it is a flight or test mate or demate, and the pointers to the connectors and users involved.

- FHW Component Installation Log and Matrix Log:

- *eloginstallrecord*: Holds the information dedicated to a FHW component installation, such as weight, part number and serial number, flatness, torque applied, MMR of bonding material used, grounding information or label tape information.

- MMR Log:

- *elogmaterial*: Holds the name, manufacturer, lot and expiration number information of a material used in the MMR Log.
- *eloghardtest*: Holds the results of the hardness test performed on a material mix, which typically takes place several days after the mix is made.
- *elogmmr*: Holds the information of a material mix, with pointers to the users, different materials and the hardness test information.

- FSW Component Installation Log:

- *elogfswinstallation*: Holds the information dedicated to a FSW component installation, such as release type.

- EGSE Log:

- *elogegsevalidcomponent*: Holds the information related to the different EGSE components that become an EGSE setup for validation, such as component validation expiration and type.
- *elogegsevalidation*: Holds the EGSE setup information, type, validation date and document, validation expiration date and the list of pointers to the different EGSE component records created for the setup.

- Configuration Log and Request Log:

- *elogconfigrequest*: Holds the information of the user that requests the configuration record, as well as the configuration type and request timestamp.

¹⁶ Two different tools were developed by the LAT Online group at SLAC for this purpose, called LAT Test Executive (LATTE) and LAT Instrument Check-Out System (LICOS) [81].

- Questionnaire:

- *elogquestionnaire*: Holds the questionnaire results information (obsolete, see Chapter 6).

3.2.2.2 Global tables

Six global tables were defined for E-Logbook:

- *eloguser*: Holds the user information. Every user of E-Logbook needs to have an entry in this table in order to sign off in a log.
- *elogoperatorinfo*: Holds the user identification and timestamp of a user. Since most of the times a user signs off in a record QA verification is required, the table is optimized by the addition of the QA user identification and timestamp fields. If the QA approval is not needed, these fields are filled with the default values. The use of this table greatly optimizes the database schema. For example in the MD Log each record created in the corresponding *elogmatedemate* table needs six user and the corresponding six QA sign offs with their timestamps. By using the *elogoperatorinfo* table, instead of the 24 fields that would be needed to hold this information in the *elogmatedemate* table, only six fields that hold the foreign key pointers to the *elogoperatorinfo* table are created.
- *elogcomponent*: Every time a record is created in E-Logbook, it is dedicated to an activity performed on a hardware or software component. Each of these components is uniquely identified by a description and a reference designator or version number. Each component can be used in multiple logs. For example, a cable can be used in a mate but also can be part of an EGSE setup. The *elogcomponent* table was created to hold the information of the different components, FHW, FSW, EGSE, EGSE HW or EGSE SW. An additional field identifies this component type to sort them out in the different logs.

- *elogconnector*: In the initial versions of E-Logbook this table was created as a distinction between the FHW components (the only components so far used by the project) and flight and non-flight connectors. It was later understood that this distinction was not necessary if a field holding the component type was used, but by this time the MD Log had been extensively used. In fact there are 10 times more connectors currently recorded in E-Logbook than any of the other types combined (see Chapter 4 for more information on this and other experimental results). Instead a field called type was added to the *elogcomponent* table to hold other component types and the *elogconnector* table was exclusively dedicated for this type of component.
- *elogtorque*: Holds the torque information related to components and connectors integrated in the LAT. It is used by the FHW and M/D Logs.
- *elogcomment*: This table holds the different comments entered by a user. Again a foreign key points from this table to the original log to tie the information together.

Figure 31 shows how the global tables were used in the different E-Logbook I&T Logs as defined in the previous Chapter. It can be noticed that one of the most globally used tables was *elogoperatorinfo*, and so the optimization of the user sign-off combined with the QA sign-off became quite useful in the overall schema design.

3.2.2.3 Maintenance tables

Only one maintenance table was required for E-logbook, necessary to perform the database mirroring process:

elogparameters: Holds the version number, release date and CVS tag of every release made for E-Logbook. It is used to maintain compatibility between the E-Logbook under MySQL and the mirrored Oracle DBMS.

3.2.2.4 Final Schema Design

The final version of E-Logbook's global database schema is shown in Figure 32. The tables corresponding to each I&T Log are shown grouped by color. The arrows in the figure demonstrate the foreign key connections between the tables. The primary key values of the tables at the top were the referencing values used as foreign keys of the corresponding tables listed below.

As mentioned above, the detailed physical schema of each table can be found in Appendix B, which represents the final version of the script written in MySQL used to create E-logbook's database schema. A special detail that can be noted in the script is the need for at least one default entry for any table that is pointed to from another table via the use of foreign keys.

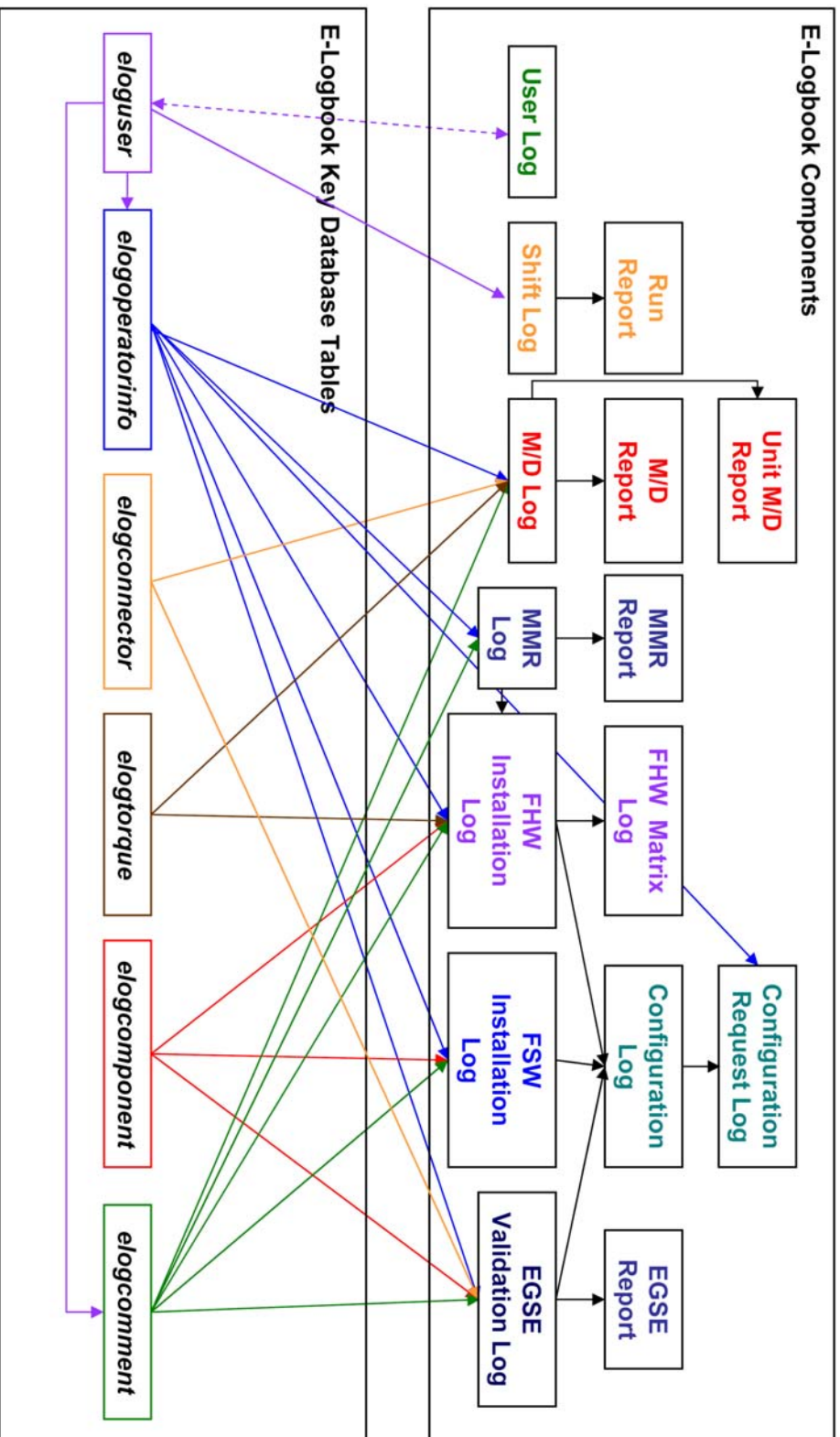


Figure 31: Relationships between E-Logbook's database schema global tables and I&T Logs

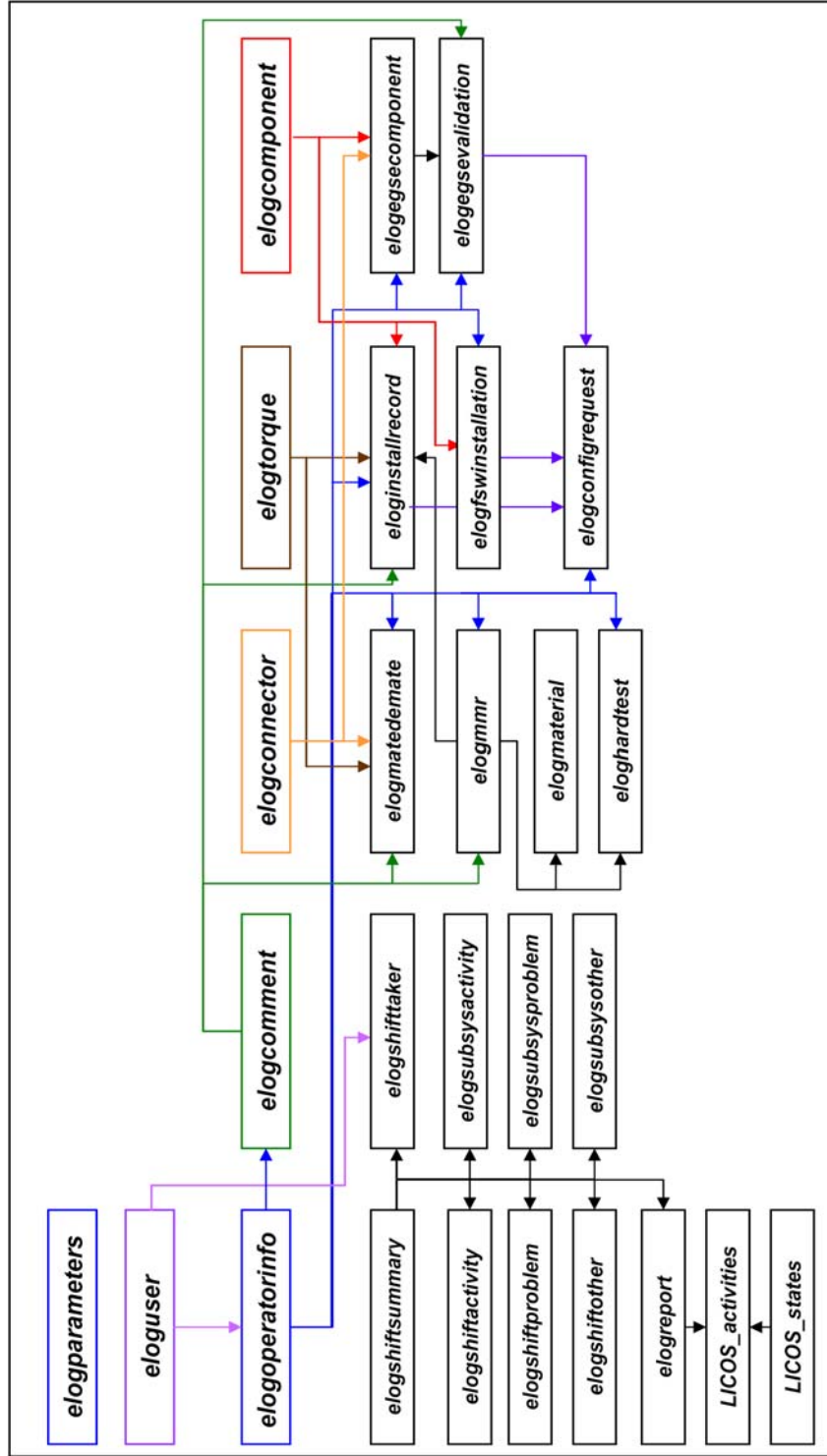


Figure 32: E-Logbook's Database Schema

3.3 Database Requirements implementation

Based on the concepts addressed in this Chapter the implementation of E-Logbook's database requirements was the following:

- *Database components:* Creation of database tables exclusively dedicated to holding the required information of each Log.
- *Database compatibility with other I&T support tools:* Restricting the design of the tables to be compatible with interfacing ground support tools, such as the Oracle database replica of the Shift Log tables.
- *Database robustness:* Enforcing a one time record insertion, with no updates other than the recording of foreign keys. Create a full database backup copy daily, which can reconstruct the database from scratch. Create a backup of the database with an efficient frequency range, both in the server inside the firewall of the I&T facility and the mirror server outside. Perform tests to exercise full data recovery from scratch from these copies.
- *Database access protection:* Create a user username and password for every user. The user information can only be modified with database administrative privileges. The users cannot modify or enter new data in the database without the use of this information.
- *Database user and timestamp recording:* Record the user identification and timestamp each time the user signs a record in any of the E-Logbook components.
- *Database unit system convention:* Save all measurements in International System units.
- *Database comment recording:* Create a table for comments linked to every log via foreign keys.

Chapter 4

4 Human-Computer Interactive (HCI) - Graphical User Interface (GUI) System

A proper interfacing system needs to be designed in order to communicate with and visualize the data stored in an electronic database schema. Although each of E-Logbook component interfacing tools has unique features, main design guidelines were defined for the overall system. The requirements' implementation was based on these guidelines and created the GUI structure as well as the global HCI system architecture. In some cases certain problems arose that required special solutions. The overall results regarding the design of E-Logbook's HCI-GUI system is shown here.

4.1 HCI-GUI System Requirements

The HCI requirements defined for E-Logbook are based on the research described in Section 1.1.4 and on LAT I&T requirements. The HCI is responsible of the connectivity and functionality between the user's input and the underlying MySQL database. The GUI system is the front end communication element that allows the user to input and output (I/O). Based on this concept the requirements for the combined HCI-GUI system were:

- *Database access via an ergonomic HCI-GUI system:* A HCI system must be designed and implemented to enhance the usability of the electronic database and provide with an underlying structure for a front end communication element, the

GUI system. The HCI-GUI system must be as user friendly as feasible within the overall project requirements. It must follow the philosophy contained in the five premises for easy to use interface design: reasonable constraints, intelligence, elegance, transparency and attention to detail.

- *User feedback:* The overall HCI-GUI system implementation must be strongly influenced by the user preferences obtained at all times through the lifetime of E-Logbook for the LAT I&T.
- *User training:* A method of training must be developed such that the user can learn how to use E-Logbook's HCI-GUI system on demand.
- *Data access and restrictions:* All of E-Logbook's existing data must be accessible, in a read-only format, through the HCI-GUI system. The no modification or corruption requirement must be enforced by this system.
- *Visual Format:* The different components' data will be visualized where required in two formats: record (a GUI that shows one entry in the database) and report (a GUI that shows a set of records grouped by a certain requirement or filter, as defined in Reference [62]. An example is a report GUI of all the records that correspond to a certain FHW component installation).
- *Print requirements:* Every report generated by the electronic database must be printable, emulating the original paper format, to ensure backward compatibility of the overall system.
- *Cut and paste requirements:* All GUI components must provide cut and paste capability to ensure interaction capability with other electronic tools.

4.2 HCI-GUI System Design

The analysis of the requirements for the HCI-GUI system yielded several decisions and guidelines that were followed throughout its design:

- *A Window GUI System:* One of the main design decisions described already in Chapter 2 was that the communication tool between the user and the data would be a stand alone GUI system; as opposed to a web based GUI or a command prompt based system. This decision was based on the assumption that the users would feel more comfortable using a window based system similar to popular tools such as the Microsoft Office family.
- *Familiar visual format:* Following the familiarity requirement, a second main design decision was to inherit the colors and styles of popular windows based systems. This helps the average user to feel familiar from the start with the new tool. Figure 34 is a snapshot of final design of E-Logbook's Main GUI is an example of this concept.
- *No sound policy:* It was also decided early in the design that no sounds would be implemented, since it was most likely that the systems would be muted in a restricted clean room environment.
- *Homogeneity:* Another major design decision was to unify the overall design of the different Logs, their GUI organization and flow. This would help the user to learn to use new Logs without needing extensive training.

4.2.1 HCI Design

As stated in Chapter 2 the HCI system is the connectivity infrastructure that interfaces the user input in the GUI system with the underlying database system.

Figure 33 shows the design path used to implement the requirements of each I&T Log. The first stage is the creation of the front end GUIs that will interact with the user;

typically a report and a record GUI (see next Section). This involves the selection of the widgets that are going to be used to perform a certain data I/O operation, and its distribution on the overall GUI layout. This stage is performed with the aid of the Qt Designer.

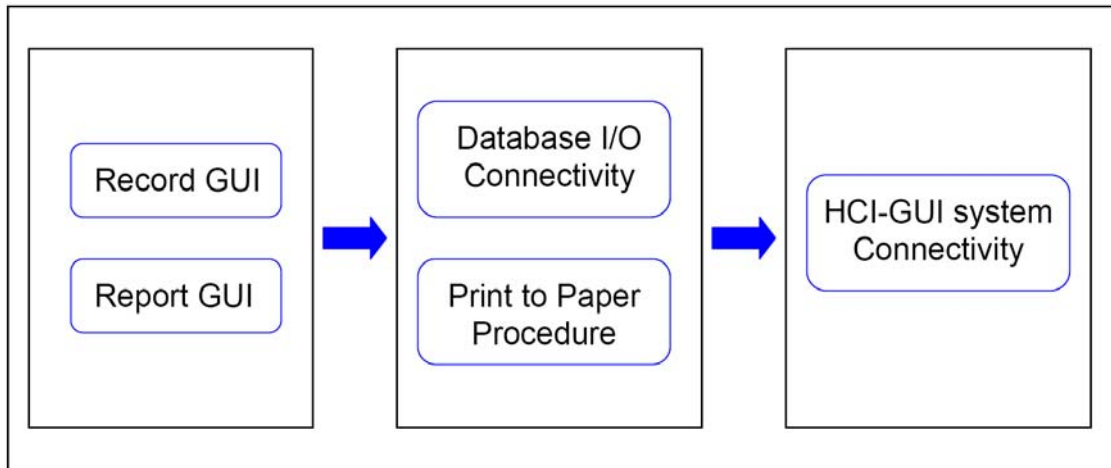


Figure 33: HCI-GUI system design path

The second stage is the heart of the implementation effort and involves the creation of the underlying code that will take care of connecting the user's data input and output requests to the underlying database schema. This stage is critical, since the code must ensure that the data is not lost or corrupted in the process of being saved, that no data is missing when a record is created, and that if the user commits any error when performing an entry he is properly warned with a comprehensive set of pop-up windows. Additionally, connectivity for data review must also be created, as well as print procedures wherever needed to ensure backwards compatibility with the paper based system. This stage is developed in Python language and by means of the PyQt binding program with the Qt Designer.

The final implementation stage involves the creation of the interfaces with the rest of the GUI system. As seen in Figure 15, Chapter 1, a sensible layout path for the user to navigate a GUI system is a tree structure. Figure 16 also describes how a main

application can be used as a central powerhouse to navigate the rest of the applications in the overall system.

The design of E-Logbook is based on these two concepts: a Main application performs the management of the global GUI system, which is subsequently navigated in a linear flow to each application, or I&T Log.

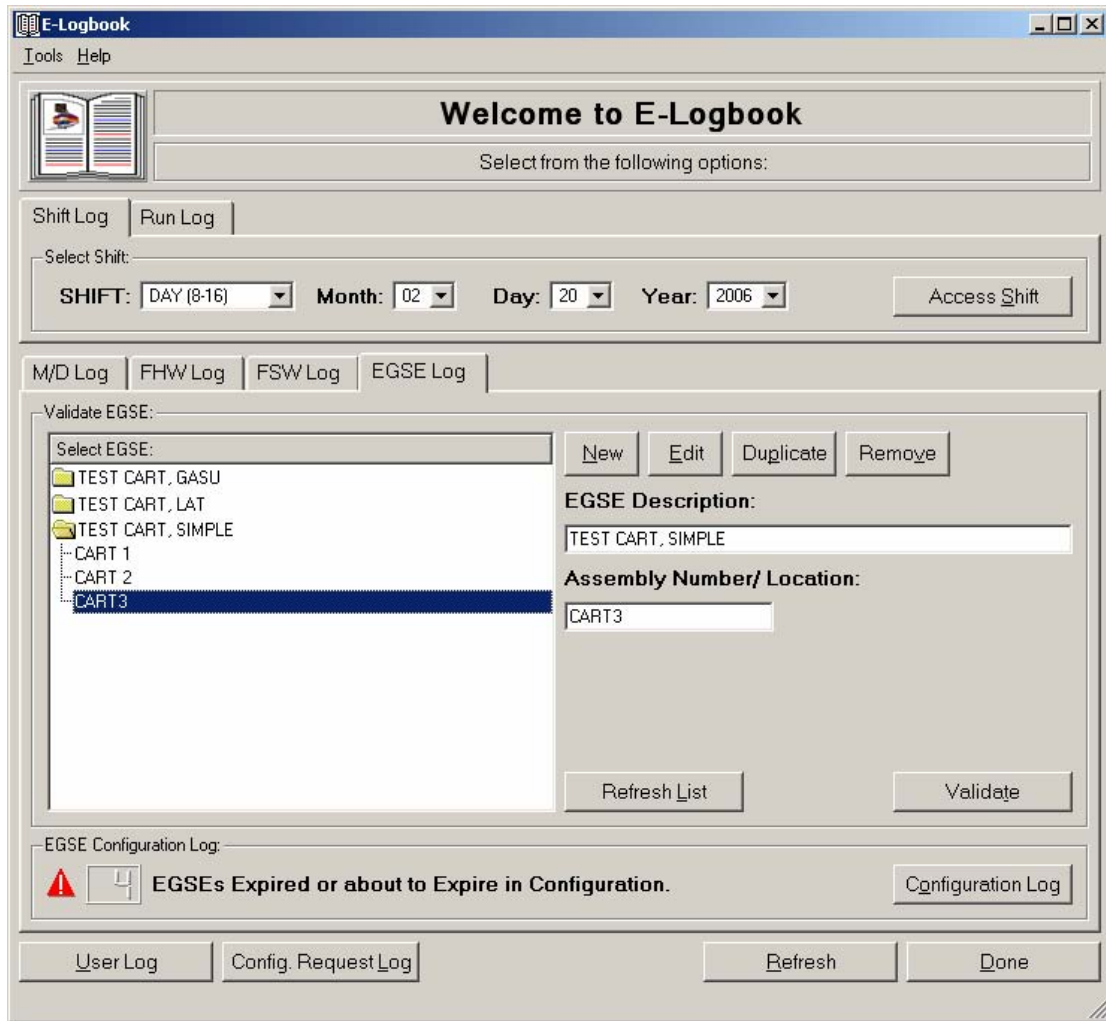


Figure 34: E-Logbook's Main window

Figure 34 shows a snapshot of E-Logbook's Main GUI. The global management buttons are located in the surrounding frame. First, global access buttons at the bottom right (the

Refresh and Done buttons, which reload and close the main application) and top right (resizing buttons and close button) corner of the window.

Second, a tool bar menu is located at the top left of the GUI. The Tools menu contains a Preferences dialog where several parameters of the overall GUI system can be user customized. The Help menu contains links to E-Logbook's online help (the user tutorial), the code documentation and the current release and copyright information.

On the bottom left are the buttons that provide access to the user related logs (the User Log and the Configuration Request Log). These Logs are needed globally and therefore have been placed in a prominent place.

Underneath the Welcome frame is a set of tabs that groups the time and test I&T Logs: The Shift Log and Run Log. Any shift can be accessed upon selection of the date and time of the shift. The time is distributed¹⁷ in three slots: Owl (0-8 hrs in local time), Day (8-16 hours in local time) and Swing (16-24 hrs in local time). Any Run Report can be accessed upon entry of the run ID. The run reports can be accessed in the Run Log tab provided a known Run ID.

The second and final set of tabs in the Main window groups E-Logbook's component I&T Logs. Each tab contains a list of the related components already entered in the database. To the right of the list and to the top the component maintenance buttons are located: New, Edit, Duplicate and Remove. Directly underneath line edit windows show the component selection (Description and R/D, Version or Assembly Number) that the user performs by navigating the list¹⁸. Once a component has been selected, the user clicks any of the report buttons that complete the frame, which provide access to the branch of the GUI system that corresponds to the selected I&T Log.

¹⁷ Like the Shift Log created for the BaBar experiment.

¹⁸ The location of line edit windows that show the selection to the right of a list is a convention used throughout the GUI system.

4.2.2 GUI Design

The design of the front end GUI system consisted mainly of identifying the optimum widgets to be used for every data I/O operation, as well as an optimum layout distribution. Beyond the individual GUI design, homogeneity of the system is achieved by using and locating similar widgets for similar operations for all the GUIs.

In the case of E-Logbook, two main types of GUIs were identified based on requirements:

- *Records*: GUIs that compile the information of I&T Log records. This is the only type of GUIs where the user enters data in E-Logbook.

Figure 35 shows the Edit M/D Record GUI. The record GUI basic layout consists of a line edit widgets where the user enters the information indicated by an attached label, timestamp widgets (disabled and automatically entered by the HCI system), combo boxes where the user selects from a fixed list of values (for example Flight or Test), and check boxes (for example N/A check boxes in the cases where a piece of information is not applicable or available).

The first frame is usually a header with the description of the component to which the shown record corresponds. The following frames correspond to the fields that are used to enter the record information. The top fields typically relate to general fields such as timestamp and authorization document. The rest of the fields are grouped by activity, for example the torque stage in Figure 35. The “...” buttons indicate where the user needs to access a secondary window to enter information, such as the username and password used to sign off stages of installation. A QA line edit is located wherever needed to the right of the operator’s sign off widget. The information flow ends in the global access buttons located at the bottom of the GUI. E-Logbook’s GUIs also use additional features such as the use of tabbing or *Alt+key* controls to navigate the widgets in the GUI or right clicking to cut and paste information.

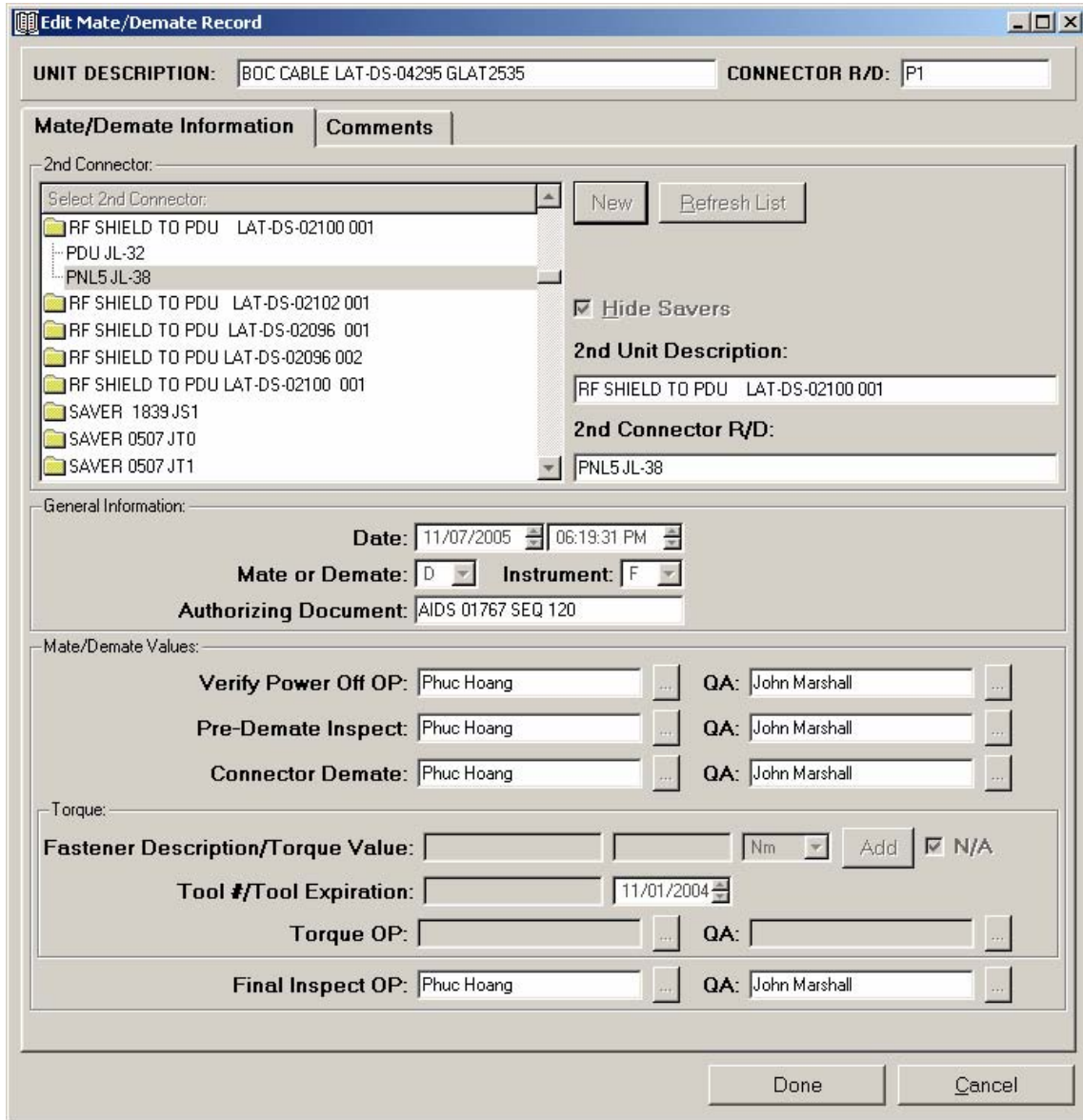


Figure 35: E-Logbook's Edit M/D Record GUI

The record GUI has an additional tab where the user can enter comments with additional or updated information related to a certain record. This is the only widget that can be used to update the information on a record, for example to indicate mistakes committed during the mate/demate procedure, or typos in the record entry itself. Global access buttons frame the GUI at the top right corner and bottom (the Save and Cancel button in

the bottom right corner; and in the upper right corner the Maximize, Minimize and Close buttons).

The record GUIs are also used to review existing records. In this case (as seen in Figure 35) the Save button is replaced by a Done button and no new information other than new comments can be entered, as the widgets have been enabled with read only access.

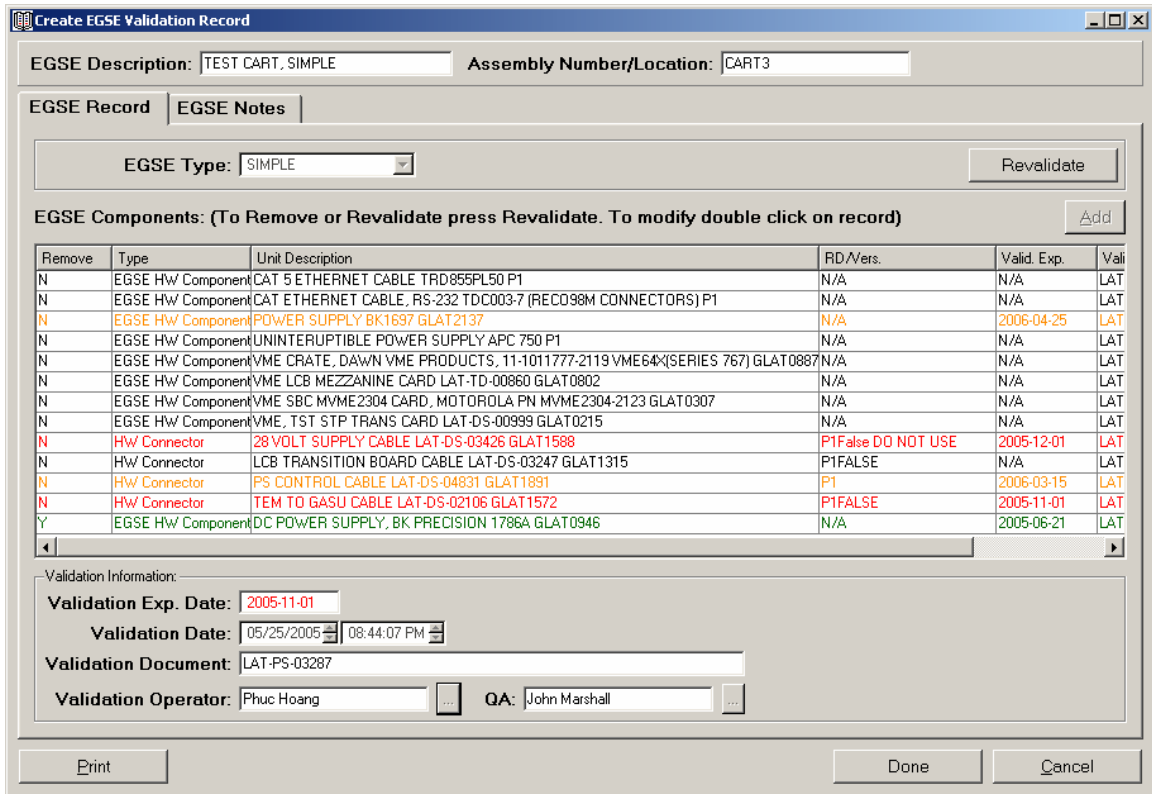


Figure 36: E-Logbook's EGSE Validation Record GUI

Figure 36 shows another example of a record GUI, the EGSE Validation Record GUI. It can be compared to the original paper-based record shown in Figure 20. After the component frame, it provides a list of the components that comprise the EGSE setup, showing the corresponding validation expiration, if applicable. The components already expired are colored in red, the records about to expire validation in orange, and the records removed from the EGSE setup in green. The global validation expiration date

calculated from the earliest component expiration and the rest of EGSE setup information is shown at the bottom of the GUI. This is the only record GUI where a print procedure was added (see Print button in the bottom left of the GUI) since each setup contained a lengthy list of components and, in that sense, can be considered a report GUI.

- *Reports:* Once records are created via the Record GUIs, the Report GUIs generate lists of records filtered by a certain value (generally a component or shift).

Figure 37 shows E-Logbook's Connector M/D Report GUI. The report GUI consists generally of a header that shows the selected component used to filter the records, a table with the record list and the global access buttons framing the GUI at the top and bottom (the Done button in the bottom right corner; and in the upper right corner the Maximize, Minimize and Close buttons) and the Print Button in the bottom left corner to print the report in paper based format. In some cases an Add button is incorporated to create a new record.

Figure 38 shows the Print statement generated by E-Logbook from the M/D Report GUI shown in Figure 37. Figure 39 shows the print statement generated by E-Logbook from the EGSE Report GUI. Both statements can be compared to the paper based reports shown in Figure 19 and Figure 20 of the Prior Art section of this thesis. The primary difference is that no information is missing in the report obtained from the GUI. Another evident contrast is how legible information is, compared to some of the values entered in handwriting.

M/D Report

UNIT DESCRIPTION: TEM/PSU-ASSY 01643 GLAT0356 **CONNECTOR/BRACKET R/D:** J1

Comments	Connector 2 Unit Description	Connector 2 R/D	Date	Authorizing Doc	Action	Instrument	Ver. Power Off OP	V
	CABLE 04831 GLAT1582	P1	2004-12-20 09:59:13	eng. direction dly run test for ps-03062	M	T	MM	EI
	CABLE 04831 GLAT1582	P1	2004-12-21 12:47:36	AIDS-EM2-F030	D	T	MM	EI
	CABLE 04831 GLAT1582	P1	2004-12-21 14:45:37	AIDS-EM2-F030	M	T	TL	D
	CABLE 04831 GLAT1582	P1	2004-12-22 14:18:01	AIDS 591	D	T	TL	BI
	CABLE 02095 GLAT1536	P1	2004-12-22 14:58:00	AIDS 591	M	T	TL	D
	CABLE 02095 GLAT1536	P1	2005-01-05 08:03:55	AIDS-0591	D	T	TL	BI
	CABLE 04831 GLAT1583	P1	2005-01-05 08:27:49	AIDS-0591	M	T	TL	BI
	CABLE 04831 GLAT1583	P1	2005-01-07 08:47:03	AIDS-0593	D	T	TL	BI
	CABLE 02095 GLAT1536	P1	2005-01-07 08:50:30	AIDS-0591	M	T	TL	BI
	CABLE 02095 GLAT1536	P1	2005-01-28 08:28:05	AIDS-0701	D	T	BH	JC
	CABLE 04831 GLAT1583	P1	2005-01-28 08:30:59	AIDS-0701	M	T	BH	JC
	CABLE 04831 GLAT1583	P1	2005-01-31 11:43:56	AIDS-0701	D	T	TL	BI
	CABLE 02095 GLAT1536	P1	2005-01-31 11:45:20	AIDS-0701	M	T	TL	BI
	CABLE 02095 GLAT1536	P1	2005-02-01 10:53:27	EM2	D	T	BH	EI
	CABLE 04831 GLAT1583	P1	2005-02-02 10:15:04	AIDS-00701	M	T	TI	RI

Figure 37: E-Logbook's M/D Report GUI with record information

In several cases the Report GUIs are much more elaborate. Figure 40 shows the Shift Report GUI. After the shift header the shift summary and shifter list is displayed. The following frame shows the subsystem activity list (the default subsystem can be customized in the Main window Preferences menu), organized by activity, problem and other notes. Finally, the test run information generated by the ground testing software¹⁹ is displayed.

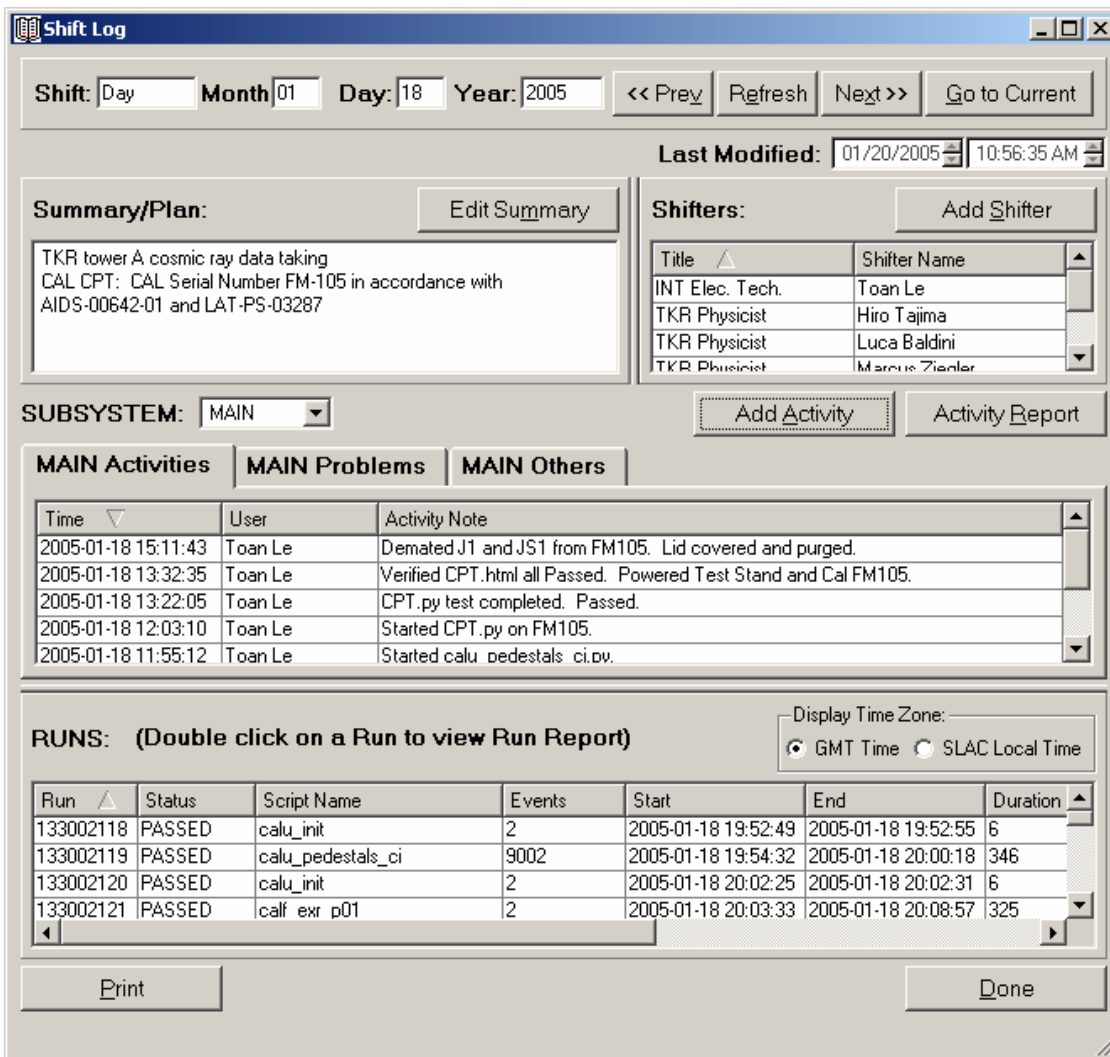


Figure 40: E-Logbook's Shift Report GUI

¹⁹ As described for the *elogreport* table in Chapter 3.

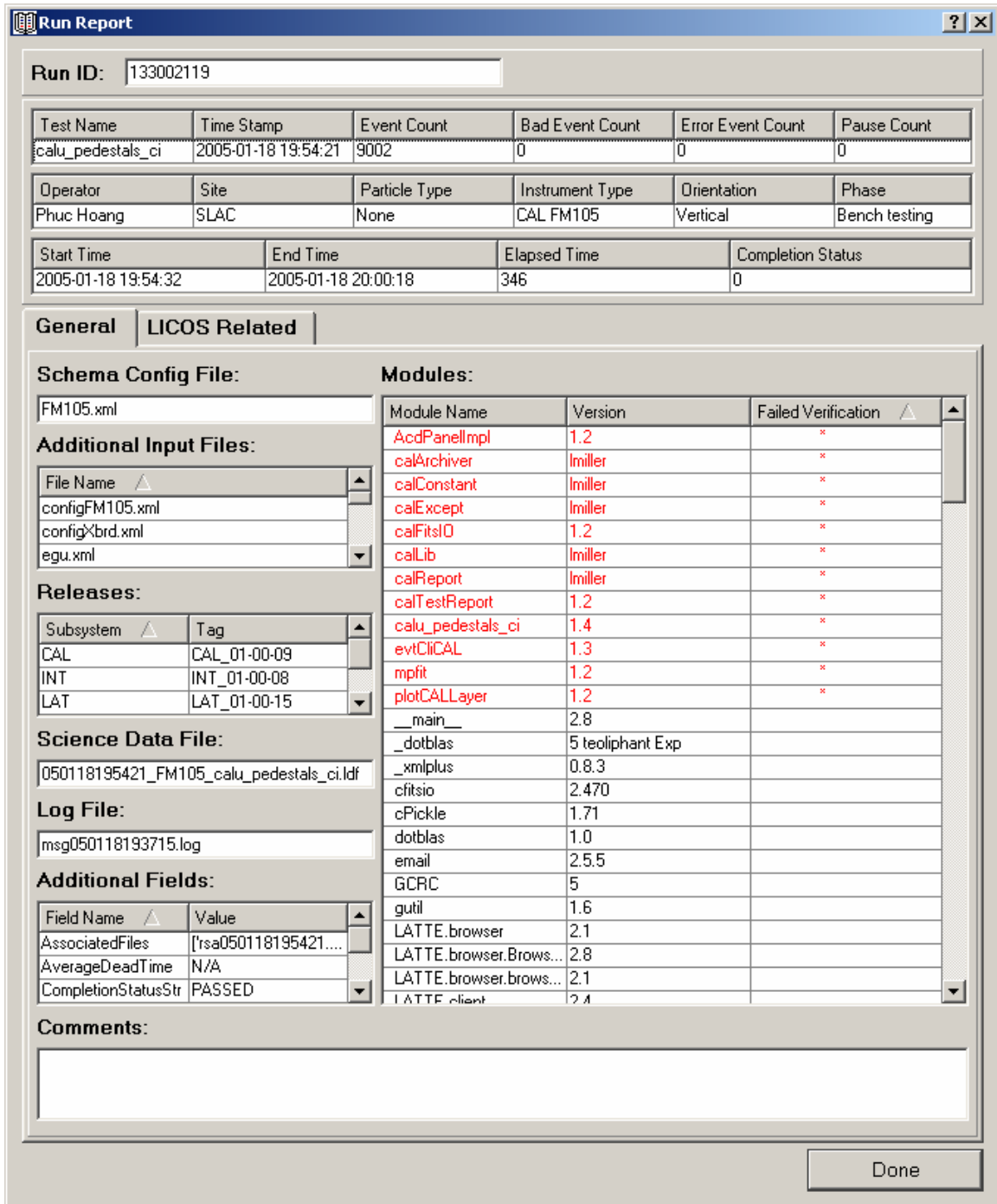


Figure 41: E-Logbook's Run Report GUI

If one of the activity entries is double clicked, the full note is displayed. In a similar fashion, if a test run is double clicked, the Run Report GUI with the selected run information is displayed. Figure 41 shows a snapshot of the Run Report GUI, with all the information of the run organized in tables and line edits. This GUI was used to monitor the tests performed by the ground software and identify which tests failed and needed to be re-run, and why.

Finally, Figure 42 shows the FHW Matrix Report GUI. This GUI is a customized visualization of the FHW Log through the use of a grid format alike to the LAT architecture. With this report the users can visualize the global current LAT I&T state of the LAT in a nutshell. Moreover, since the grid configuration is identical to the LAT physical layout (see Figure 4, Figure 5 and Figure 9) the report is extremely user-friendly. A combo box at the bottom right of the GUI allows the users to toggle between International and English units.

This report is also an example of how E-Logbook can be customized to the needs of a particular I&T project.

LAT FHW Component Installation Matrix			Date:	02/20/2006	02:02:11 PM	User:	Alicia Kavelaars								
Bay 12		ID	Wgt [Kg]	Bay 13		ID	Wgt [Kg]	Bay 14		ID	Wgt [Kg]	Bay 15		ID	Wgt [Kg]
CAL		110	85.729	CAL		108	82.772	CAL		113	86.119	CAL		114	83.121
CAL/TEM/TPS		02559-0008	87.997	CAL/TEM/TPS		02559-0007	86.183	CAL/TEM/TPS		02559-0011	89.176	CAL/TEM/TPS		02559-0012	86.636
EPU		GLAT2524	6.35	HIB		GLAT 2531	2.68	HIB		GLAT 2532	2.67	HIB		GLAT2533	2.68
TEM/TPS		GLAT 1838	7.32	TEM/TPS		GLAT 1836	7.42	TEM/TPS		GLAT 1842	7.35	TEM/TPS		GLAT 1843	7.42
TKR		6	31.026	TKR		4	30.3	TKR		10	32.749	TKR		11	31.842
Bay 8		ID	Wgt [Kg]	Bay 9		ID	Wgt [Kg]	Bay 10		ID	Wgt [Kg]	Bay 11		ID	Wgt [Kg]
CAL		107	92.079	CAL		106	85.82	CAL		111	85.198	CAL		112	84.368
CAL/TEM/TPS		02559-0006	88.088	CAL/TEM/TPS		02559-0005	92.896	CAL/TEM/TPS		02559-0009	88.451	CAL/TEM/TPS		02559-0010	88.269
PDU		GLAT 1898	11.32	GASU		GLAT2245	15.79	GASU		GLAT2245	15.79	SIU		11 GLAT2513	6.35
TEM/TPS		GLAT 1831	7.51	TEM/TPS		GLAT 1835	7.35	TEM/TPS		GLAT 1839	7.34	TEM/TPS		GLAT 1841	7.35
TKR		5	32.114	TKR		3	32.931	TKR		7	32.205	TKR		9	33.747
Bay 4		ID	Wgt [Kg]	Bay 5		ID	Wgt [Kg]	Bay 6		ID	Wgt [Kg]	Bay 7		ID	Wgt [Kg]
CAL		105	97.976	CAL		102	94.166	CAL		115	86.409	CAL		116	79.569
CAL/TEM/TPS		02559-0002	101.605	CAL/TEM/TPS		02559-0003	101.605	CAL/TEM/TPS		02559-0013	93.531	CAL/TEM/TPS		02559-0014	93.621
PDU		GLAT 1898	11.32	GASU		GLAT2245	15.79	GASU		GLAT2245	15.79	SIU		07 GLAT2527	6.38
TEM/TPS		GLAT 2491	7.31	TEM/TPS		GLAT 1833	7.35	TEM/TPS		GLAT 1844	89.176	TEM/TPS		GLAT 1847	7.36
TKR		B	35.38	TKR		1	34.201	TKR		12	31.933	TKR		13	31.933
Bay 0		ID	Wgt [Kg]	Bay 1		ID	Wgt [Kg]	Bay 2		ID	Wgt [Kg]	Bay 3		ID	Wgt [Kg]
CAL		104	97.976	CAL		103	97.976	CAL		117	86.341	CAL		118	86.137
CAL/TEM/TPS		02559-0001	87.09	CAL/TEM/TPS		02559-0004	101.605	CAL/TEM/TPS		02559-0015	93.531	CAL/TEM/TPS		02559-0016	93.712
EPU		GLAT 2490	7.33	HIB		GLAT 2529	2.68	HIB		GLAT 2530	2.67	EPU		GLAT2523	6.35
TEM/TPS		GLAT 1834	7.35	TEM/TPS		GLAT 1834	7.35	TEM/TPS		GLAT 1848	7.34	TEM/TPS		GLAT 1849	7.33
TKR		A	37.875	TKR		2	36.106	TKR		14	31.751	TKR		15	32.205

Wgt Units: **Kilograms**

Print Done

Figure 42: E-Logbook's FHW Matrix Report GUI

4.2.3 Problems and Solutions

Main design features and improvements arose from the user interaction with E-Logbook's HCI-GUI system. From among the multiple user requests (listed in Appendix C), several are worthy of particular mention due to their importance for the improvement of the overall system. Among all of these, only one (Shift Summary entry) could not be resolved due to design constraints:

- *Component unit entry*: Component entry is one of the most important aspects of E-Logbook. Before any record is created or any report visualized, the component unit that it relates to must exist in the database. The component units can be connectors that are mated or demated, FHW or FSW components installed in the LAT, or EGSE setups that are used to perform tests on the FHW and FSW components²⁰.

In E-Logbook, component unit entry was designed always using two fields: First of these is a description field that used an established LAT I&T convention that combines the component unit name, part number and serial number separated by dashes. A secondary field stores the information that distinguishes the specific item used in the I&T effort, for example each of the connector unit's R/Ds, each of the FSW components installed in the LAT unit, each of the versions available for a certain piece of FSW or each of the carts that hold a certain EGSE configured setup.

Each component unit is loaded in a folder structure in the component list of each of E-Logbook's component tabs (see Figure 34), grouping all of its secondary items as subfolders in the tree structure. This greatly helps the user organize and find information.

²⁰ Please note the distinction between a component unit, for example a CAL unit, defined here, and an E-Logbook component: each of the different I&T Logs that are part of E-Logbook, like the FHW Log.

A major effort on the part of the users of E-Logbook was to enter the names of all of the components of the LAT and EGSE equipment. An extreme example was the case of the M/D Log, where more than 2300 connectors were entered in the database.

In order to facilitate this task multiple features were added to E-Logbook. For example, a duplication feature that duplicates the name of a unit (adding a “(2)” at the end) and all its folder structure as added. An Edit feature was implemented also in order to allow the users to correct typos that occurred frequently. Finally a Remove feature would help the users get rid of components not used or incorrect. Following the requirement that no data would be lost in the database, this feature was implemented as a mechanism to flag and hide the component units. All of these features required heavy coding, since a warning dialog flag system needed to be in place to ensure that no identical component units were created, that component unit descriptions would not be swapped, or to issue an alert when component units could not be removed since they had already been used to create records in the database.

Another innovative solution to facilitate component unit entry was the use of a bar code reader. Since the users needed to tag each component unit, they suggested adding a bar code in these tags so that when the component unit was used for I&T, the tag would be read by the bar code reader into the database to avoid typing and typographical errors most of all.

- *Units system:* In Chapter 3 we saw that one of the project requirements was to record every unit in the International System. This led to multiple complaints by the users since they worked primarily in the English System of units. For example to estimate expected measurements, or when using tools (wrenches, scales ...) showing values in English units only.

The solution to this problem was to provide, where relevant, a combo box with every unit system requested for each task apart from the international system. For example, for

torque tasks the combo box would list N-m, ft-lb, in-lb and in-oz units. Once the user selected the unit system, entered the value, and saved the global record, the final value stored in the database would be its conversion to the International System of units.

The implementation of this solution was not entirely straightforward. Once a value had been entered in a certain unit, if the user toggled the unit system the original value could be rounded and therefore be lost due to recursive conversion operations. The solution to this problem was to store the value initially entered by the operator in a table that would also hold the conversion values to the rest of the unit systems. When the operator toggled the unit system, no further conversion operations were performed; instead, the correct value was picked from the table and displayed. Once the record was entered in the database, only the value in International units was saved.

This led to another problem since if the user originally entered the value in a unit system other than International, a recursive conversion was performed once the record was opened for review. Since the recursive conversion could not be avoided in this case, a solution was to store the value in the database with enough additional decimal points such that the original value would not be rounded upon conversion. The number of decimal places stored in the database was determined by adding three decimal spaces to the sensitivity of the tools used for each measurement operation. For example, if a torque in a FHW component installation was performed with a wrench with an error of thousands of an in-lb, the field used in the database had six decimal places, so that the value stored in the database had three additional decimal places in Nm. The application of this idea required the investigation of the sensitivity of each tool used for each operation in the process of LAT I&T. To view an example of the resulting algorithm, go to *Code/hardware/assignTorqueImpl.py* in the attached DVD.

- *Review colors:* A major decision in the design of E-Logbook was how to deal with the distinction between a record GUI window accessed for data entry and one accessed for data review.

Initially the design followed the default use of a widget's disabled mode, which in general grays it out and blocks all usage of the widget. With this design the GUIs became unclear and difficult to use. For instance, it was not obvious to the users that, once the widgets were disabled for data review, the cut and paste capability was still available. The solution was to change the convention so that for data review the GUI widgets would be left in their base color (generally white), but in read only mode. If the user would forget that he/she was reviewing a record, it would soon become apparent through the widget's read only mode. The disabled mode was left to indicate those widgets that at any point in time would not need data entry. For example, the date and time widgets, which are automatically populated by the underlying HCI system, were always disabled.

- *Preferences menu:* Another feature created upon user request to facilitate data entry was a preferences menu where the users could select several defaults. These defaults were stored in a configuration file so that they would persist locally in each workstation even if E-Logbook was closed.

Two items were implemented in the Preferences menu: First, the Shift Log default subsystem, which would make the Shift Log load the activities of a certain subsystem by default. This was useful in order to customize the workstations in the different areas of the clean room (see Chapter 5).

Second, the validation expiration limit (30 days by default). This value delimits the warning interval that flags EGSE setups whose validation is about to expire. The EGSE records are loaded in the report GUIs colored in black or yellow depending on this limit. Additionally, a counter in the EGSE tab of the Main Window indicates the number of records that have expired or are about to expire validation based on this limit (see the bottom portion of the EGSE Log tab shown in Figure 34).

- *Additional Reports:* Three report GUIs were added to E-Logbook based on user suggestions, which provided improved data visualization.

First, the FHW Matrix Report, already described in the previous section, was implemented. Second, by the time the LAT was 70% integrated, the users started to monitor the M/D state of the flight connector units. It became clear that a report filtered by subfolder, or connector unit R/D, was inefficient to visualize an overall connector unit M/D state. In some cases the users needed to open more than a dozen reports to this end. The solution was the creation of a connector unit M/D report, which, in a similar fashion to the Configuration Logs for the FHW, FSW and EGSE components, collected the current M/D state (or last record) of each R/D in the unit.

Finally, in a similar way the MMR Report was requested by the users to visualize in one snapshot the current state of the different MMRs, in particular their hardness test result. This report would facilitate keeping track of which mixes were apt to be used for LAT I&T activities.

- *Print procedure:* An exact copy of the original paper reports (as seen in Figure 38 for the M/D Report, and Figure 39 for the EGSE Report) was created to ensure backwards compatibility with the traditional paper logs. The implementation of these printed reports was difficult, due to the limited column space available in a fixed format. In the traditional paper logbooks this problem was “solved” by grossly writing past the column delimitations.

In those cases where the text length was too big to fit in the allocated space, a new algorithm was developed that would indent the text in the column in new lines. The row frame would then reformat according to the new number of lines.

Additionally, another algorithm was developed to dynamically create the printed reports given a default set of information. This ensured very little maintenance effort as well as the capacity of implementing new report’s print procedures in a fast and convenient way. To review these algorithms go to *Code/tools/logprinter.py* in the attached DVD.

- *Shift Summary Entry*: One of the widgets in the Edit Shift Record GUI (see Figure 40) is the shift summary. The purpose of the shift summary is to record a global summary of the activities performed in a shift.

Due to its nature, the shift summary did not become a read only widget once the information was initially entered, due to the fact that the user needed to have the possibility of expanding on the information entered in the summary as the shift proceeded. As a result, some of the users started to leave this widget open in their workstation, therefore making it impossible for the widget to refresh when new information was entered in other workstations. This led to the users overwriting information entered by other users every time they left the summary window open for a long period of time. The only solution to this problem so far is administrative: the only person allowed to enter information in the Shift summary is the Test Director. A solution to this problem is the creation of an additional field in the summary table of the database that changes value every time a user opens the summary window. The field can be consequently checked by the HCI system every time a user requests to open the window, and if the value indicates that the window is currently open, then access to the window will be denied and a pop-up window with the proper message is shown instead.

4.2.4 Final HCI-GUI System Structure

The final architecture of E-Logbook's HCI-GUI system is shown in Figure 43, adapted from a combination of Figure 15 and Figure 16.

E-Logbook's Main GUI becomes the inter-application communication tool, providing access to the rest of applications or I&T Logs in the HCI-GUI system. The Applications' Presentation Managers are I&T Log tabs in E-Logbook's Main GUI.

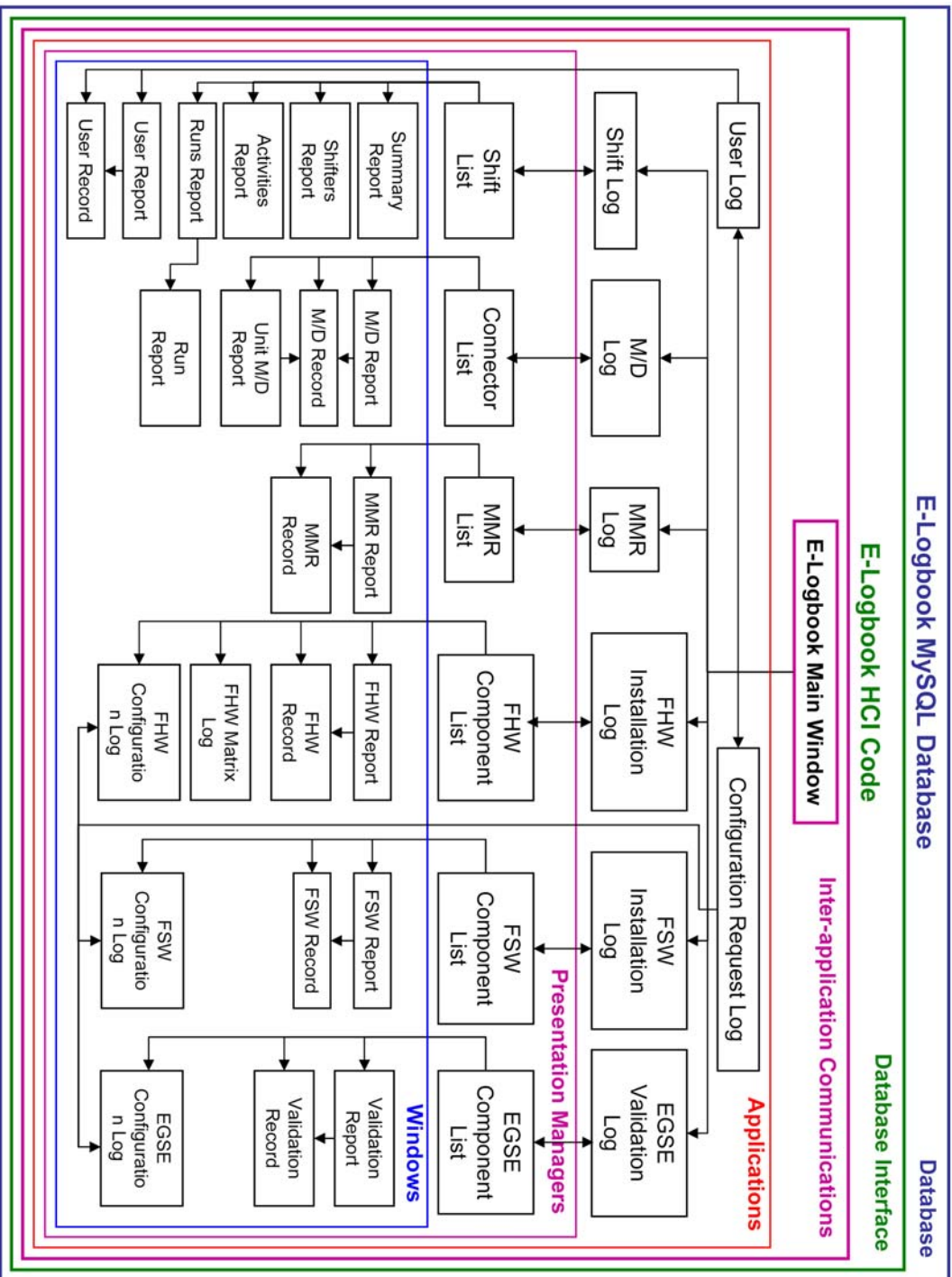


Figure 43: E-Logbook's GUI-HCI System Structure

Once a component, shift or run is selected in the I&T Log tab, a main – report – record or main – record linear flow is enforced in every Log or branch.

This rule was broken only in two cases where additional connectivity between I&T Logs was required. The first case connects the run report table in the Shift Report GUI to the Run Report GUI. By double clicking a row in this table the user can access the Run Report GUI to review the corresponding run. The second case connects the FHW Component Installation record GUI with the MMR record GUI, since the latter is part of one of the installation stages of the former.

The *Documentation/Tutorial* section on the attached DVD shows E-Logbook's tutorial on the use of its HCI-GUI system. The tutorial contains snapshots of every GUI created for E-Logbook as well as detailed walkthroughs on how to use them.

4.3 HCI-GUI System Requirements Implementation

Based on the concepts described in this Chapter, E-Logbook's HCI-GUI system requirements implementation was the following:

- *Database access via an ergonomic HCI-GUI system:* Designed and implemented a HCI-GUI system based on overall requirements and ergonomic design guidelines obtained from the research summarized in Chapter 1.
- *User feedback:* Improved design based on user feedback as described in the previous section.
- *User training:* Created a detailed tutorial with GUI to GUI walkthroughs for all of E-Logbook's components. Added a link to an online copy from the Main GUI. Additionally (seen in Chapter 5), scheduled tutorial sessions after each major release to inform the users of key changes or new features added.

- *Data access and restrictions:* Created an algorithm that ensures the widgets are read only once a record GUI is accessed for data review.
- *Visual Format:* Created a comprehensive and homogeneous system of report and record GUIs based on requirements for each I&T Log, navigated linearly from a Main GUI application.
- *Print requirements:* Created print to paper procedures for each report GUI plus the EGSE validation record GUI.
- *Cut and paste requirements:* Created cut and paste capability for every widget where applicable, both when used for data entry and data review.

Chapter 5

5 E-Logbook's Experimental Setup for the LAT I&T Project.

The previous Chapters have been dedicated to summarizing the background research, motivation, components and design that made the concept of E-Logbook possible. In this Chapter, the experimental setup of E-Logbook in the context of the LAT I&T project will be described.

This effort required not only the design of the hardware and software architecture for E-Logbook's use for the LAT I&T project, but also the inclusion of E-Logbook in the configuration control process established to ensure its successful completion.

Moreover, the development of E-Logbook was always constrained by the LAT I&T schedule: the I&T Logs included in E-Logbook needed to be implemented and tested before they were needed for I&T activities. An additional constraint defined for E-Logbook was the inclusion of the user's feedback early on in the implementation process. In this Chapter the final experimental setup created for E-Logbook's software and hardware architecture will be shown.

E-Logbook's design process within the LAT I&T timeline will be also addressed, introducing the iterative design loop that this researcher followed in order to include both LAT I&T configuration control and user feedback requirements in its development.

5.1 Experimental Requirements

The experimental requirements defined for E-Logbook's implementation in the LAT I&T project were:

- *Network architecture:* E-Logbook's software architecture will provide multiple and simultaneous access to the database schema from a network of computers located strategically within the LAT I&T facility.
- *Firewall access:* E-Logbook will reside, as the rest of the ground support equipment, within the security restrictions of the LAT I&T facilities, including the network firewalls.
- *Mirroring:* A mechanism to transfer or "mirror" the database through the LAT I&T facility to the outside world will be created, for read only user access and copied to the Oracle replica of the database. The users outside the firewall will be able to access the mirrored database for data review (read only) through local copies of E-Logbook's HCI-GUI system.
- *Backup:* A backup mechanism will be devised to ensure that no data is permanently lost during critical events such as fire, power or network failure. An optimum backup frequency will be defined.
- *Release test and installation plan:* Each of E-Logbook's releases will contain only those new features, improvements or bug fixes that have received prior approval of a Configuration Control Board (CCB) committee, as defined by NASA and military standards in the LAT I&T Configuration Management document [24]. Once the changes have been implemented the release will enter configuration control (no more modifications will be allowed) and it will go under a verification and validation (V&V) test procedure defined in the corresponding documentation, in the presence of a QA engineer. Once the release successfully completes the V&V process, it will be ready for official release and installation in the LAT I&T facility will be scheduled.

5.2 Software Architecture

In Chapter 2, the concept of a server-client architecture for the design of E-Logbook was introduced. Based on this idea and the requirements of the LAT I&T facility at SLAC introduced in the previous section, E-Logbook's software architecture was designed as seen in Figure 44.

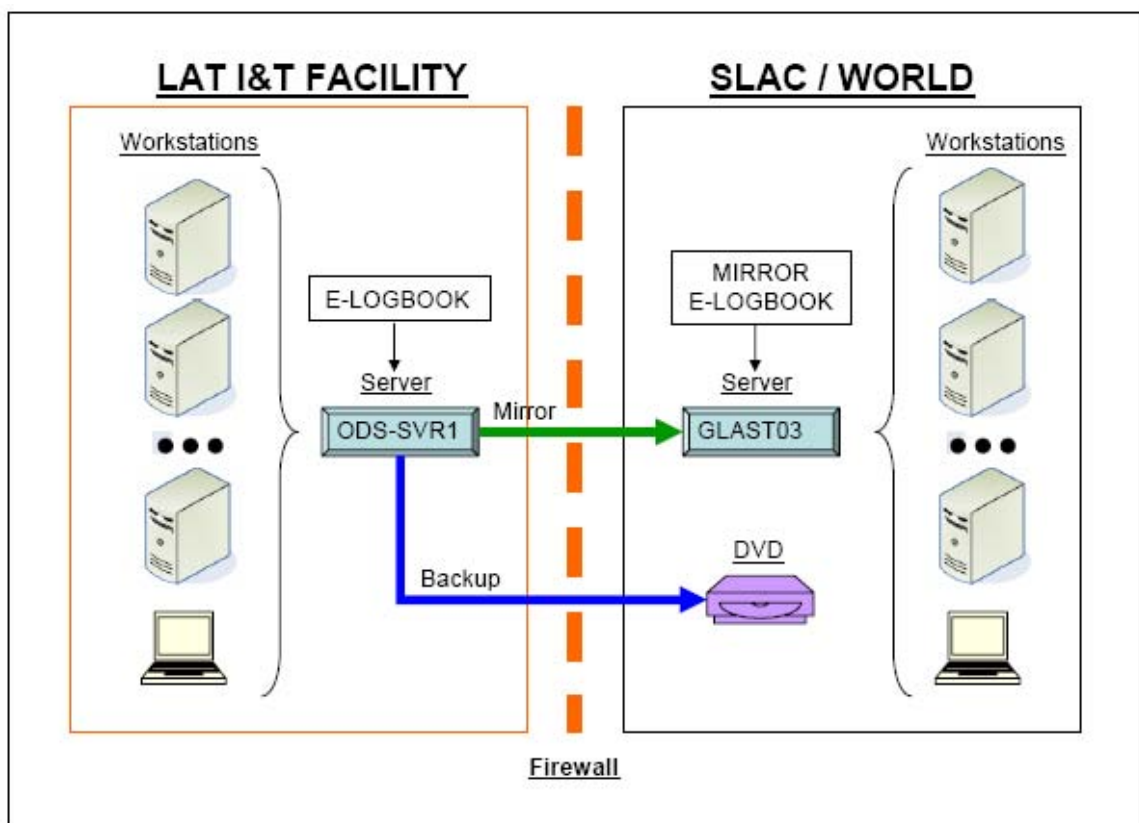


Figure 44: E-Logbook's software architecture for the LAT I&T Project

A centralized server, called ODS-SVR1, has a copy of E-Logbook and runs E-Logbook's MySQL database schema loaded from the *MySQLelogbook* module. The server is accessed by multiple clients, workstations in the LAT I&T facility. The connect dialog

loads a customized parameter list maintained in a script²¹ (including the server name) that allows the users to access the ODS-SVR1 server with the click of a button.

In an initial setup, each of the workstations had a copy of E-Logbook installed. It was soon noticed that this setup required a lot of maintenance, since every time a new release was available, every workstation needed to be upgraded with the new release installation package.

A solution to this problem was the use of only one copy of E-Logbook installed in the server. This copy was then accessed by as many workstations as needed via the proper setting of a shortcut icon. With this architecture the upgrade process was optimized and, as mentioned above, took an average of 20 minutes to complete, including the shut-down and upgrade of the database schema.

5.2.1 Database mirroring

The mirroring process transfers a copy of the database schema in ODS-SVR1, through the LAT I&T facility network firewall, to an outside UNIX server in the SLAC network, called GLAST03 (replaced in January 2006 by a new server called GLASTDB). The development of this process as well as its migration to the Oracle database counterpart²² (work beyond this research), was performed by Selim Tuvi, member of the LAT I&T Online group. The mirroring frequency was set to 10 minutes via a Windows XP scheduled task.

E-Logbook's data could be accessed outside the LAT I&T facility via the mirrored copy in GLAST03 (with authorized access to the SLAC network), provided that a copy of its package was installed locally. This access was restricted so that the users had read only permissions, to ensure that no data was corrupted.

²¹ See *ELogbook/dbconnect/dbpar.sample* in the attached DVD.

²² Included in the *ELogbook/dbsync* module of the attached DVD.

5.2.2 Database backup

Using MySQL tools and a Windows XP scheduled task, a backup copy of the database schema in ODS-SVR1 was created every day at 9pm²³. The copy was placed in a folder that was subsequently transferred outside the LAT I&T facility and stored in a DVD by the SLAC computing and security group (SCS). Additionally, the GLAST03 server was also backed up daily by the same group.

In the event of fatal failure a request of the latest DVD backup copy would be made to SCS, which would then retrieve the copy from tape and place it in a selected location. This copy could be then used to restore E-Logbook with data loss of up to a day, as determined by the defined backup frequency.

5.2.3 Mobile Computing Rack (MCR) architecture

Once the LAT successfully completes I&T at SLAC, it will be shipped to the environmental testing facility at NRL. E-Logbook needs to provide the same support at NRL as in the LAT I&T facility.

For this purpose all LAT I&T ground support software, including E-Logbook, was transferred to a mobile computing rack (MCR) in January 2006, based on the UNIX operating system.

E-Logbook needed little modifications in order to complete this transfer thanks to the platform independent requirement defined early on in its design. Nevertheless, improvements on font styles in Linux were needed, as well as the creation of running scripts, “*cron jobs*” to run the mirroring and backup processes (the Linux equivalent to the Windows XP scheduled tasks) and an RPM, equivalent to the Windows installation package. A new generation of E-Logbook with all these new features (E-Logbook 4.0.0

and on) was developed and is currently installed and used to run E-Logbook from the MCR at the LAT I&T facility.

Since the MCR also has a network firewall, an access point between the LAT I&T facility and the MCR firewalled networks was created for E-Logbook so that the users would be able to access it from the workstations in the LAT I&T facility until shipment to NRL. Additionally, several workstations of the LAT I&T facility were converted into Linux machines to be used within the MCR network firewall and E-Logbook was installed in them also with the corresponding shortcut setup.

The access, mirroring and backup architecture of E-Logbook at NRL goes beyond the scope of this thesis, but will be based on the one described here for the LAT I&T facility at SLAC. With some customization E-Logbook can be used in a similar fashion in any space systems I&T facility as the one at SLAC.

5.3 Hardware Architecture

E-Logbook's hardware setup was based as well in the LAT I&T facility requirements and the server/client architecture described in Chapter 2 and adopted for its software. Table 5 compiles the hardware (servers, workstations, and bar code reader) used for E-Logbook throughout the LAT I&T facility.

Room	E-Logbook Hardware
102	7 workstations (3 for testing)
103	4 workstations
104	1 workstation, 1 tap for laptop, 1 bar code reader
ACD Area	1 workstation, 1 tap for laptop
Visitor Area	2 taps for laptop

Table 5: Number of computers and location used for E-Logbook in the LAT I&T facility

²³ Included in the *ELogbook/dbbackup* module of the attached DVD.

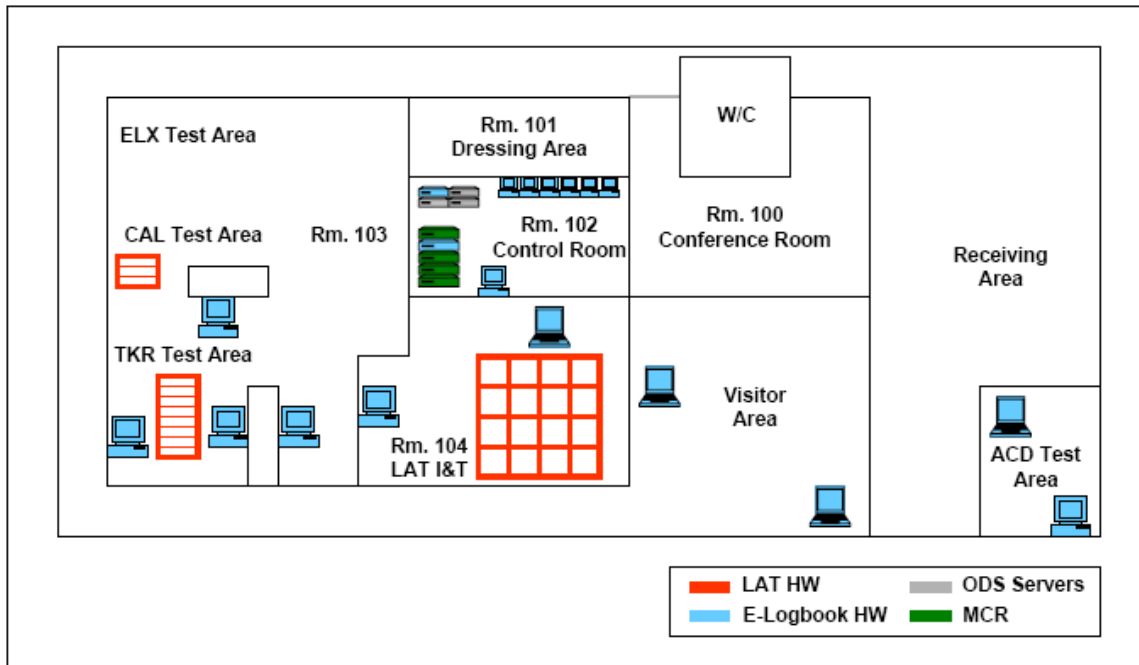


Figure 45: E-Logbook’s hardware architecture in the LAT I&T facility

Figure 45 shows the location of the hardware in the LAT I&T facility. The main bulk of the ground software testing hardware is located in the control room (Room 102). In this room, the ODS-SVR1 and the rest of the server machines, the MCR, and multiple testing workstations with double displays were located (see Figure 46). Shortcut icons to E-Logbook were created in each of these workstations. Beyond Room 102, E-Logbook’s client hardware architecture could not make use of solutions such as hand held devices or laptops in a wireless network, due to the stringent LAT I&T facility security requirements. Instead, shortcut icons to E-Logbook were created in those workstations closest to mechanical I&T locations. Additionally, firewalled network taps were requested and made available for laptop use in those locations where a workstation was not available, such as the Visitor or Receiving areas shown in Figure 45.

Apart from these computers the only other piece of hardware used for E-Logbook was the bar code reader mentioned in Chapter 4.



Figure 46: The LAT I&T facility control room (Room 102, Bldg. 33, SLAC)

5.4 Requirements Implementation Process

E-Logbook's implementation process within the scope of the LAT I&T project followed a design loop shown in Figure 47 based on the iterative incremental design approach described in Section 1.1.5 and this researcher's knowledge on feedback control.

The starting point in the design process was always the definition of requirements. In the initial months of development, these came from the requirements document [62]. Once use of the system for LAT I&T activities started, new requirements were created based on feedback from the users and improvements devised by this researcher.

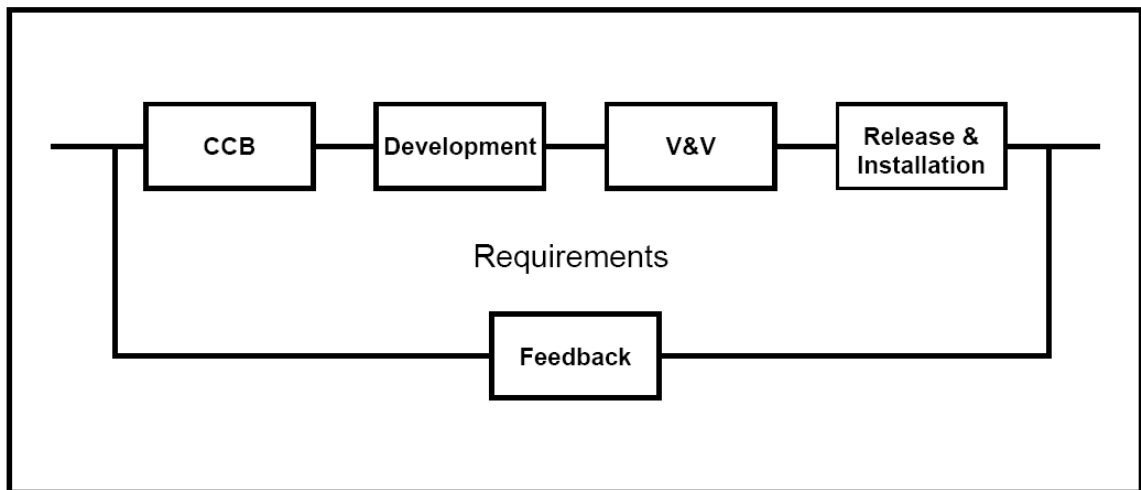


Figure 47: E-Logbook’s requirements implementation loop

In order to document the new requirements, E-Logbook benefited from the JIRA online issue tracking system introduced in Chapter 2. In JIRA, users and this researcher documented requests and ideas for new features, improvements or bug fixes (Feedback block in Figure 47). The issues were subsequently discussed and approved weekly by the CCB committee (CCB block in Figure 47), based on this researcher’s feedback and the overall project progress. Once the new requirements derived from the JIRA issues were approved, the resulting new code was developed and tested until a high confidence level for its stability was achieved (Development block in Figure 47).

A test procedure and report document was created to perform the V&V test of E-Logbook, approved by the CCB and published in the GLAST configuration controlled document database [82]. If modifications on the test procedure were needed, a new version of this document was created, approved and published. Only then was a V&V test scheduled (V&V block in Figure 47) with a QA engineer, and upon successful completion (and proper sign off of the corresponding documentation by the QA engineering present in the V&V test and the I&T Manager), the new version of E-Logbook was officially released and scheduled for installation (Release & Installation block in Figure 47) in the LAT I&T facility.

E-Logbook's installation in the LAT I&T facility also needed to comply with official configuration control requirements. The installation was properly scheduled and announced in JIRA. During installation, E-Logbook was shut down to ensure no data corruption or loss. The upgrade scripts were run in MySQL both in the LAT I&T facility and the mirror servers, and the new release executable package was run replacing the existing release in the LAT I&T facility. Once the optimum software and hardware architecture had been created in the LAT I&T facility (see next sections), E-Logbook's new release installation took an average of 20 minutes.

Appendix C shows the complete list of JIRA issues recorded for each E-Logbook release. The first table includes the JIRAS that were implemented ("fixed"), organized by release number. The table also shows the issue number (ELB-X by convention), the resolution (*fixed*, "cannot reproduce", "won't fix") and a short description. Among the 120 issues reported for E-Logbook, more than 80% correspond to the research described in this dissertation. The rest are related to the mirroring and transfer to Oracle efforts, and were addressed by other developers of the LAT I&T Online group.

Figure 48 shows E-Logbook's release history correlated to the LAT I&T progress by month and general area of activity. The release tag follows the convention X.X.X, where the first X corresponds to a major change in the program, for example addition of new features. The second X corresponds to improvements in major changes with no addition of new features, the final X to bug fixes. E-Logbook 1.0.0 and 2.0.0 were experimental releases that did not follow the complete requirements implementation process described here and therefore were not installed in the LAT I&T facility.

The most intense development periods took place in the winter of 2004, when initial requirements were implemented, the summer of 2005 (one of the most intense periods for the LAT I&T project) and in winter of 2005, when the preparations for the migration of E-Logbook to the MCR took place.

At this writing, the MCR connected to the LAT I&T facility network at SLAC is running the latest version of E-Logbook implemented for the LAT I&T project, E-Logbook 4.1.0.

For each of the 15 releases from 3.0.0 on, E-Logbook underwent the implementation loop described in this section. For each of these iterations, E-Logbook never failed the V&V test. A critical bug, defined as a bug in the code that led to data loss after E-Logbook was V&Ved and installed in the LAT I&T facility, was found only once (documented in JIRA issue ELB-27, Appendix C). As a consequence, only one of the four bug fix releases (E-Logbook 3.2.1) was critical. The overall reliability of the system will be discussed in more detail in Chapter 6.

5.4.1 E-Logbook's Code Structure

Initially, E-Logbook consisted of a set of scripts in a folder. It soon became clear that a more organized structure was needed. Using Python's modularity, the scripts were grouped in several folders that corresponded to either one of E-Logbook's Logs (the hardware module for the FHW Log, the software module for the FSW Log, etc.) or a global function in the program (setup, backup, connect, tools).

More than 45000 lines of code were written by this researcher in order to provide E-Logbook with all its functionality. A full copy of the resulting scripts can be found under *ELogbook Code* in the attached DVD.

Among all the different modules in E-Logbook's code structure, it is worth noting the *MySQLelogbook* folder, which contains the *MySQL* database schema of each E-Logbook release. Two kinds of scripts can be found, *elogbookXXX.sql* and *upgradeElogbookXXX.sql*, where *XXX* represents the release number. The first kind of scripts creates the database schema from scratch.

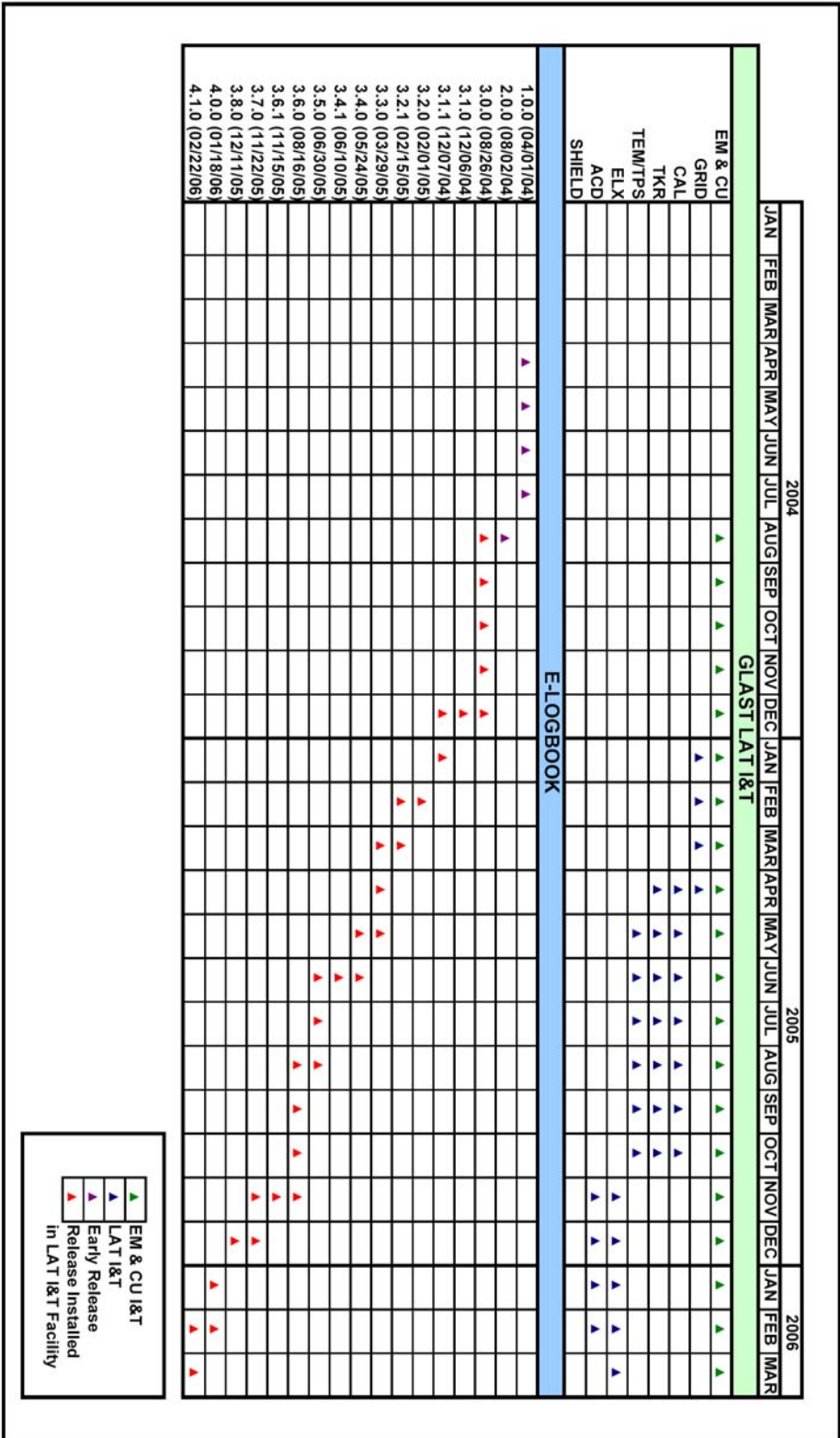


Figure 48: E-Logbook's release history correlated to LAT I&T progress

The second kind upgrades an existing database schema to a new release. This ensures forward compatibility as long as the scripts are run sequentially from one release to the next. If no changes in the database schema were necessary in the new release, this script contains only an entry in the *elogparameters* table to record the new release date and version information. Every time E-Logbook underwent a V&V process and was officially released, this second kind of script was used to upgrade the database in both the LAT I&T facility and the mirrored servers (see section 5.2.1).

For the reader interested in reviewing the functions and algorithms created for E-Logbook, a full copy of E-Logbook's code can be found under *ELogbook Code* in the attached DVD. Additionally, full documentation on the code generated with *Doxygen* is located as well in the attached DVD under *Documentation/ELogRef.chm* and in html format at *Documentation/ELogRefHtml//index.html*.

5.4.2 E-Logbook's Website

In order to publish E-Logbook's release progress, this researcher developed a website for E-Logbook based in html format. In this website the users were informed of the release history, all the project documentation, contact information, and links to the JIRA issue tracking system and other relevant websites in the global GLAST LAT network.

A download site was also created containing every E-Logbook release, the corresponding release notes linked to JIRA as well as links to the corresponding documentation.

Figure 49 shows a snapshot of E-Logbook's website's main page which can be accessed online at²⁴:

<http://www-glast.slac.stanford.edu/IntegrationTest/ONLINE/ELogbook/index.htm>

²⁴ A full version of E-Logbook's website can also be found on the attached DVD under *Website*.

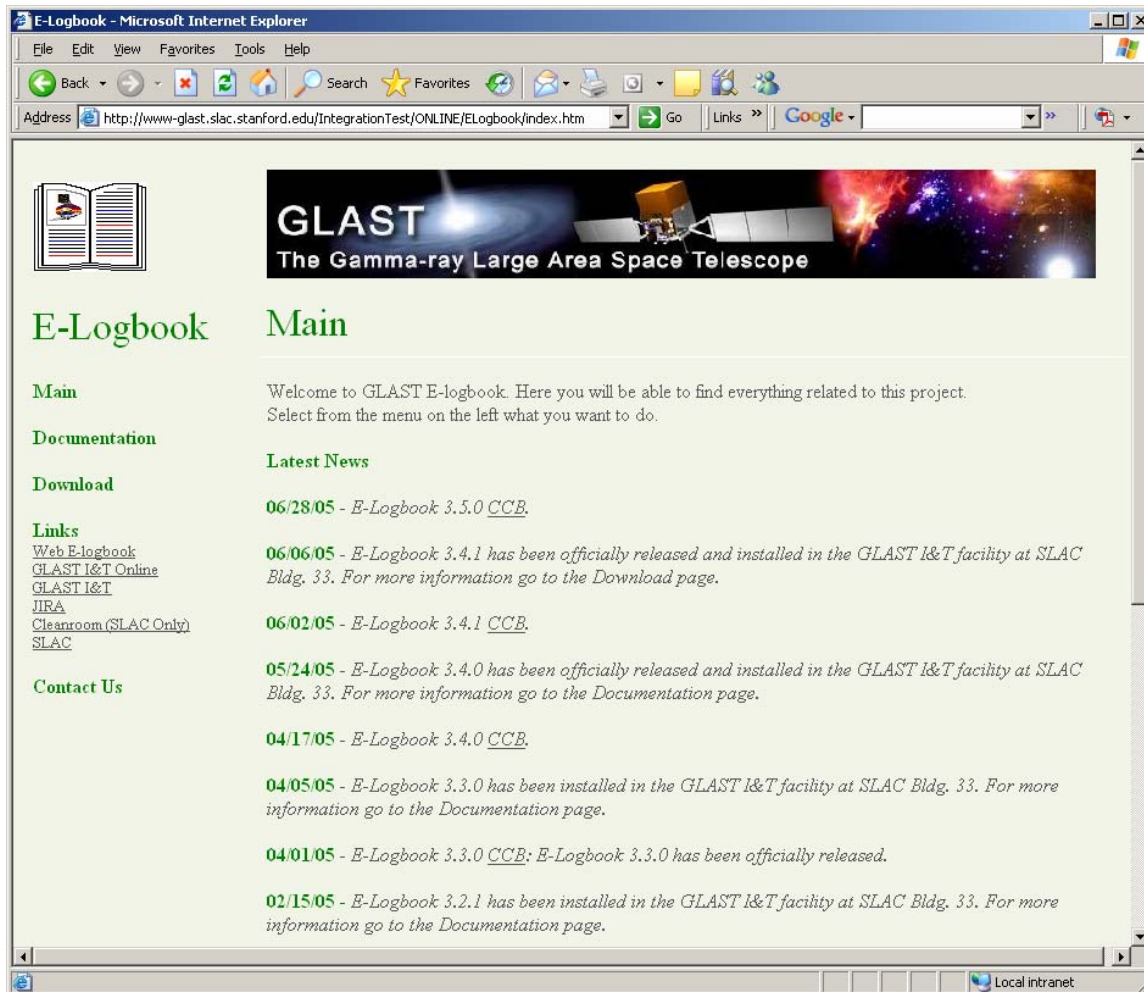


Figure 49: E-Logbook's website's main page

5.5 Experimental Requirements Implementation

Based on the research described in this Chapter, the implementation of the requirements for E-Logbook's experimental setup for the LAT I&T project can be summarized as follows:

- *Network architecture:* Created a server/client database/HCI-GUI system architecture to provide authorized access to E-Logbook from within the LAT I&T facility and the MCR networks. Created global user access via the mirrored server (with read only permissions).
- *Firewall access:* Created E-Logbook's software and hardware architecture compatible with the LAT I&T facility and MCR network firewalls.
- *Mirroring:* Created a mechanism to mirror the database schema from the dedicated MySQL servers in the LAT I&T facility (ODS-SVR1) and the MCR (glast-mrb01) to a secure server (GLAST03 becoming GLASTDB) in the SLAC network. Scheduled a task (or cron job in the MCR) to perform the mirroring process every 10 minutes.
- *Product backup:* Created a daily backup mechanism to store a copy of E-Logbook's database outside the LAT I&T facility. Backed up the mirror server (GLAST03).
- *Product release installation and test plan:* Followed release control loop described in Figure 47. Documented changes as well as CCB approval for each release via the JIRA online issue tracking system (see Appendix C). Created a test procedure and report document [82] maintained and used for every new release V&V test.

Chapter 6

6 E-Logbook Analysis and Results

The main issue that determines the success of this thesis work is the question of whether electronic technology can supersede the use of paper for space system I&T logging activities, and whether E-Logbook is the right solution to achieve this.

A reliable design of E-Logbook's database system based on I&T Log requirements is of no value if its HCI system is inefficient. The success of E-Logbook is therefore ultimately determined by the level of satisfaction of the users who made use of it for the LAT I&T project.

In order to analyze the results of the implementation of E-Logbook for the LAT I&T effort, two different kinds of analysis were performed:

- *Database Performance Analysis:* An analysis of the electronic database system performance in terms of the amount of data created versus the amount of data corrupted and/or lost due to deficiencies in the implementation of the I&T Log requirements and its maintenance.
- *HCI-GUI System Performance Analysis:* An analysis of the performance of the HCI system created to interact with the electronic database.

The combination of both analyses ultimately determines whether E-Logbook can replace paper for the documentation of space systems I&T operations.

6.1 Database Performance Analysis

In order to study the reliability of E-Logbook's database system, data stored in E-Logbook for the LAT I&T project from August 26th, 2004 until February 24th, 2006 was analyzed.

Property	Value
# Months of Operation	19
# Seconds of Operation	21888000
Size in Server (MB)	550
Size in dumped copy (MB)	279
Total # Tables	29
Total # Indexes	143
Total # Records	41750
Total # Records entered by the Users	21771

Table 6: E-Logbook database statistics for the LAT I&T project as of 02/24/06

Table 6 shows E-Logbook's database global statistics. Through the 19 months of operation of E-Logbook for the LAT I&T project, more than 41000 records were created in the 29 tables that comprise its database schema. The total number of records entered by the users via E-Logbook's HCI-GUI system is 21771, which corresponds to 52.15% of the total number of records in the system.

The rest of the data stored in E-Logbook corresponds to the test data electronically generated by the ground testing software and stored in *elogreport* and LICOS related tables.

The performance analysis shown in this section makes use of the average size per record values obtained for each table by means of the DBMS reporting tools. The current size of the database schema in the MySQL server is 549.63 MB, while the backup copy currently has a size of 279.21 MB.²⁵

²⁵ The discrepancy in size is caused by the default space stored in the DBMS for each table given its number and types of fields. This can be seen more clearly in Table 7, an

Table	Size (KB)	# Records	Avg. Size (KB) / Record
eloguser	16.38	59	0.28
elogshiftsummary	196.61	749	0.23
elogshifftaker	131.07	1539	0.08
elogshiftactivity	196.61	821	0.23
elogshiftproblem	81.92	142	0.52
elogshiftother	16.38	4	4.10
elogsubsysactivity	1589.25	4883	0.31
elogsubsysproblem	65.54	135	0.59
elogsubsysother	81.92	67	1.06
elogreport	545931.26	19928	22.91
LICOS_activities	16.38	42	0.39
LICOS_states	16.38	9	1.82
elogshiftschedule	16.38	0	0.00
elogmatedemate	212.99	1570	0.12
eloginstallrecord	49.15	85	0.83
elogmmr	49.15	64	0.96
elogmaterial	16.38	88	0.19
eloghardtest	16.38	64	0.26
elogfswinstallation	16.38	0	0.00
elogegsevalidation	16.38	14	1.17
elogegsevalidcomponent	16.38	182	0.09
elogconnector	196.61	2424	0.09
elogcomponent	16.38	156	0.11
elogoperatorinfo	524.29	7852	0.07
elogtorque	81.92	698	0.11
elogconfigrequest	16.38	109	0.15
elogcomments	16.38	46	0.36
elogquestionnaire	16.38	3	5.46
elogparameters	16.38	17	0.96
TOTAL	549634.05	41750	
TOTAL from User Entry	3670.02	21771	

Table 7: E-Logbook database schema statistics for the LAT I&T project as of 02/24/06, on a table by table basis

expansion of the database schema statistics on a table by table basis. For example, the *elogfswinstallation* table is allocated a default 16.38 KB of space in the server, even though it is empty.

Table 7 shows that most of the data corresponds to the *elogreport* table, which, as explained in Chapters 2 and 3, is with the LICOS related tables (*LICOS_activities* and *LICOS_states*) electronically generated by the ground testing software. With 545.96 MB these tables hold 99.33% of the total data stored in E-Logbook.

The fact that 52.15% of the records in the database schema store less than 4MB or 1% of the overall data in the system indicates how the design of E-Logbook's database schema maximizes the benefits of using a relational database model: instead of creating one table to store great amounts of data, E-logbook's database schema minimizes the amount of data stored by the use of the same information multiple times, creating multiple records of small size which are therefore faster to store and retrieve. Figure 50 also demonstrates the idea of displaying the correlation of the number of records created in each I&T Log with the number of records dedicated to each I&T Log in the database schema²⁶.

Optimizing the database system is not only achieved by minimizing the size of data stored, but also by ensuring that it is not lost or corrupted. The success in this effort is determined both by the reliability of the DBMS to store and manage data and by the design of a bug free HCI-GUI system in its data I/O operation.

Based on this premise the following possible sources of data corruption and/or loss were investigated and identified for E-Logbook:

- *A malfunction of the DBMS, either on the server or during the backup process:* While this kind of error is not in E-Logbook's control, it can jeopardize the users' trust in the reliability of electronic database technology. For example, corruption of the backup copy could cause failure of the project given a critical error of the system.

²⁶ It can be noted in the Figure that at this writing, FSW has not yet been installed in the LAT.

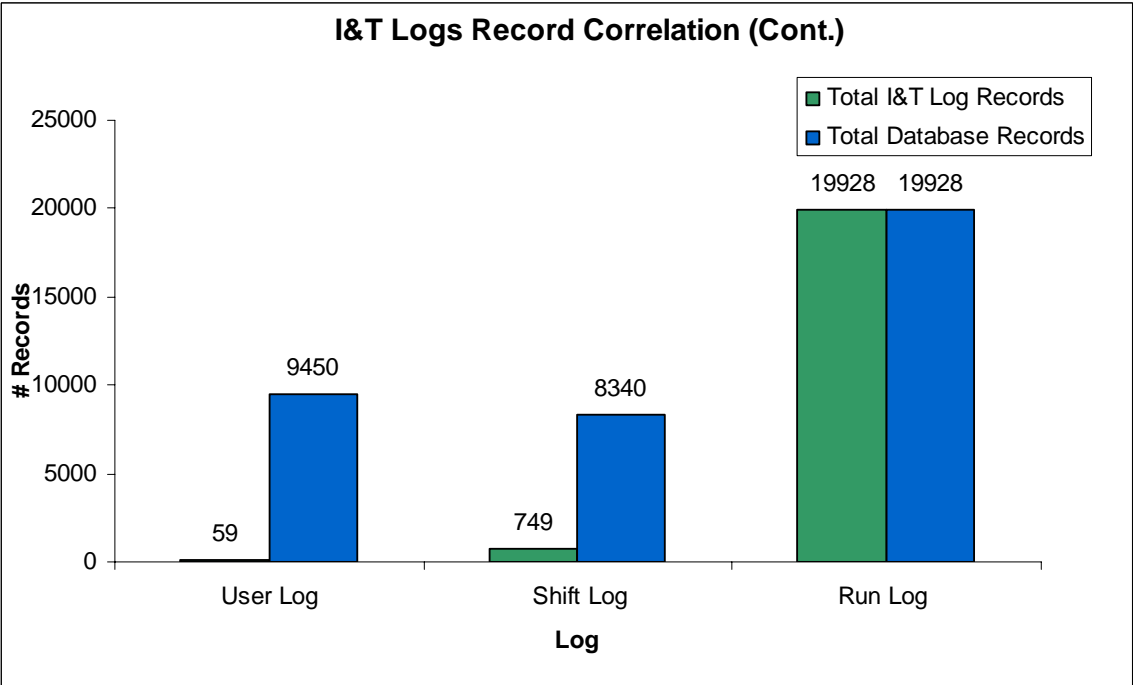
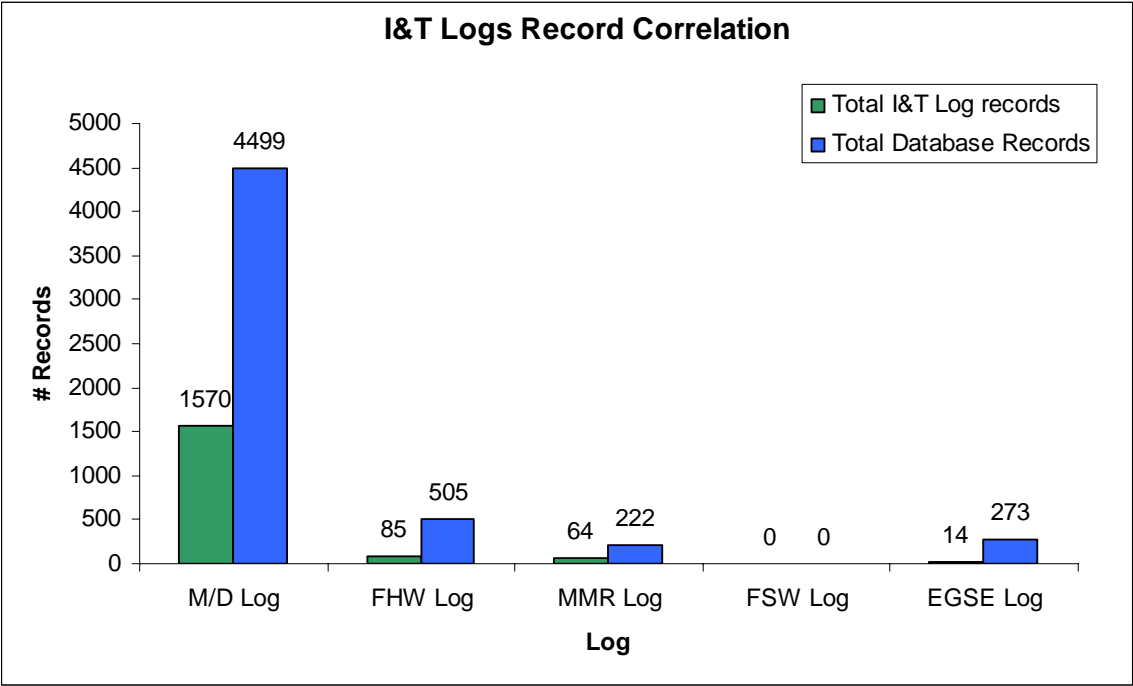


Figure 50: Correlation between I&T Log records created and number of record entries in the database schema

- *A bug introduced in the mirroring process:* This kind of error would only cause data loss in the mirrored server, and therefore not permanent data loss. Nevertheless, the system needed to be reliable for those users accessing the data stored in E-Logbook from outside of the LAT I&T facility.
- *A bug introduced in the transfer to Oracle process:* This kind of error would cause corruption of the data visualized by other LAT subsystems using the Oracle database, and therefore avoiding it is critical. Nevertheless, the data would never be lost since an original copy of the data still resided in the mirrored server.
- *A bug in the HCI algorithms that causes data loss in the process of transfer to the database schema in the MySQL server:* This is a very sensitive kind of error, since it could be caused by any of the multiple data I/O operations performed by the HCI system in E-Logbook. A reliable release process was critical to ensure that this kind of error is avoided.
- *A usability problem of the GUI system:* Data loss due to an unforeseen use of the GUI system, for example data overwrite due to users working on the same record in different workstations.

This researcher frequently monitored the DBMS to ensure that the mirrored server and Oracle copy of the database was working properly. Moreover, backup copies of E-Logbook were frequently reloaded in the development platform to verify the incorruptibility of the data and perform tests on the system.

In order to estimate data corruption or loss due to a usability problem of the HCI-GUI system, one of the questions asked to the users in the questionnaire (analyzed further in the following section) was the amount of times they had lost data in E-Logbook.

Data loss source	# Records	Size (Avg. KB)	System Error v. User Entry # Records	System Error v. Total # Records	System Error v. User Entry Database Size	System Error v. Database Size	System Error v. Seconds of Operation
Data corrupted/lost on MySQL DBMS	0	0	0.00%	0.00%	0.00%	0.0000%	0.00000%
Data corrupted/lost on backup	0	0	0.00%	0.00%	0.00%	0.0000%	0.00000%
Data corrupted/lost on mirroring	0	0	0.00%	0.00%	0.00%	0.0000%	0.00000%
Data corrupted/lost on sync. w/ Oracle	0	0	0.00%	0.00%	0.00%	0.0000%	0.00000%
Data loss - HCI related	6	0.11	0.03%	0.01%	0.02%	0.0001%	0.00003%
Data loss - GUI related	8	0.12	0.04%	0.02%	0.03%	0.0002%	0.00004%
Total data loss HCI-GUI system related (multiplied by safety margin of 2)	28	0.12	0.13%	0.07%	0.09%	0.0006%	0.00013%

Table 8: Data loss statistics for E-Logbook on the LAT I&T project

Table 8 lists the data loss due to all the factors listed above. The data loss statistics are also shown in terms of database size and seconds of operation. In order to estimate the system error in terms of size, the number of records lost was multiplied by the table's average record size obtained from the DBMS reporting tools as shown in Table 7. The total number of seconds of operation was estimated as 21888000 (19 months multiplied by an average of 20 days per month, 16 hours per day, and 3600 seconds per hour of operation), as indicated as well in Table 6.

In terms of data loss, the results in Table 8 show the successful management of the data by the DBMS, mirroring and transfer to Oracle processes. Even though there were several cases when the mirroring task went down, no data was lost on the long run due to this process. The same result applies to the process of transfer to Oracle.

On the very sensitive case of HCI malfunction, there was one case in which records were lost due to a bug introduced in the implementation of E-Logbook, in the 3.2.0 release on 02/01/05. This bug was described in Chapter 5 and documented in JIRA issue ELB-27 in Appendix C. It caused the loss of 3 records in the *elogmaterial* and *eloghardtest* tables of the MMR Log for a total of 6 records. The data in these records was reloaded once and the bug was fixed and E-Logbook 3.2.1 was released and installed in the LAT I&T facility.

A total loss of 8 records was reported for the GUI system, caused by the problem of shift summary/plan user overwrite described in Section 4.2.3. These records needed to be rewritten and it is safe to assume that some information recorded originally was lost. The impact of the loss of these records in the overall scope of the LAT I&T project was minimal.

In order to assess the impact of this loss of information on the overall performance of E-Logbook, the total number of records lost (14 multiplied by a safety margin of 2 for a total loss of 28 records) was correlated with the total amount of data stored. The loss rate was obtained in terms of different parameters: number of records, database size and seconds of operation, as shown in Table 8. Furthermore, estimations were obtained for the data entered by the users only. The final results yield a 0.07% loss rate of the total number of records or 0.13% of the records entered by the users. A 0.0006% loss rate of the total data size is also obtained; or 0.09% of the data stored by the users. Finally, a 0.00013% loss rate of the data stored was estimated per second of operation. These results were non critical since the data was recovered and the error cause was fixed.

Furthermore, a major result obtained in this analysis demonstrate that, throughout the 19 months of operation of E-Logbook in the LAT I&T facility, no data was corrupted as a result of operational problems of the DBMS or bugs introduced in the code of the HCI-GUI system.

6.2 Human-Computer Interface (HCI) – Graphical User Interface (GUI) System Performance Analysis

In order to evaluate the performance of E-Logbook's HCI-GUI system, as well as its implementation progress, a questionnaire based on research shown in Reference [83] was created.

The questionnaire, included in Appendix D, consisted of an initial section where statistics on the user completing the questionnaire were collected, an initial and final section with critical questions that would determine the user's overall impressions on the system, and a central section dedicated to identifying E-Logbook's usability problems based on several criteria.

The questionnaire was circulated two times, in November 2005 (identified by QNov05 in this study) and at its final implementation stage, in February 2006 (identified by QFeb06 in this study).

Table 9 shows the statistics of the set of users that were subject to E-Logbook's usability questionnaire both in November 2005 and February 2006. The users or "subjects" have been grouped by I&T specialization. The 13 subjects that used E-Logbook most heavily, labeled N1 to N13, have been placed first.

An evaluation Table 9 shows that the set of subjects exposed to E-Logbook in this experiment generally exhibits broad experience in the field of space systems I&T, with an average of 14 years of experience for the 13 subjects that most heavily used E-Logbook. The table also shows the correlation of the subject's expertise with the specific I&T Logs he/she used. For example, the Shift Log was used by most of the subjects, while the M/D Log and FHW Log were used mainly by Mechanical Technicians, and the EGSE Log by

Electrical Technicians. The Test Conductors used the Shift Log only. The I&T Managers used all of the I&T Logs to monitor overall I&T progress.

Subject	Years of Experience	Past Relevant Experience	Job Specialization	I&T Logs used	Data Input/Output
1	6	NASA, Rockwell	I&T Software Engineer	Shift Log	Input
2	17	Litton, Northrop, NASA	I&T Electrical Technician	Shift Log, EGSE Log	Input
3	2	SLAC	I&T Electrical Technician	Shift Log, EGSE Log	Input
4	27	SSL, Ford Aerospace	I&T Mechanical Technician	M/D Log, MMR Log, Shift Log, FHW Log	Input
5	12	SSL	I&T Mechanical Technician	Shift Log, M/D Log, FHW Log	Input
6	9	SSL	I&T Mechanical Engineer	Shift Log, M/D Log, FHW Log, MMR Log	Input
7	10	SSL	I&T Mechanical Technician	M/D Log, FHW Log	Input
8	0	-	I&T Test Conductor	Shift Log	Input
9	31	US Air Force, Ford Aerospace, Teledyne, SSL	I&T Test Conductor	Shift Log	Input
10	9	SSL	I&T QA	M/D Log, FHW Log	Input
11	28	Air Force, Lockheed Martin, Kastor Solutions, Boeing	I&T QA	M/D Log, MMR Log	Input
12	13	Space Systems Loral (SSL), Orbital Network Eng.	Test Director	Shift Log	Input
13	18	Lockheed Martin Missiles & Space	I&T Mechanical Technician	M/D Log	Input
14			I&T Electrical Technician	Shift Log, EGSE Log	Input
15			I&T Facility Manager	Shift Log	Output
16			I&T Manager	All	Output

Table 9: Subject statistics

The set of users that most heavily used E-Logbook for data input in the LAT I&T facility were asked to fill in the three sections of the questionnaire (see Appendix D). I&T managers were asked to fill in the initial and final sections only, since their usage of E-Logbook for data input was less intensive. Analysis of the results obtained in both questionnaires follows.

6.2.1 Main Results

In the view of the users completing the questionnaire, E-Logbook was most helpful in its capability to keep data in a central location and in an organized way. It greatly helped the process of shift hand-over and the maintenance of an accurate I&T history. It also eliminated problems such as keeping track of paper trails, illegible handwriting, and missing or misplaced information. Another advantage that the users perceived with regard to the usage of E-Logbook was that it set the minimum amount of data to be recorded in an I&T operation. The LAT I&T managing team was especially pleased about the way E-Logbook enforced discipline in the logging tasks. Finally, the users were greatly satisfied with the fact that they could retrieve necessary information instantly at any given point in time.

Several problems were also reported by the users. For example, the M/D Report GUI slow down problem addressed in Section 4.2.3, or the need of a MMR Report, both implemented and fixed in December 2005 based on the analysis of the results of QNov05. Second, the fact that the users were displeased with the record multiple sign-off process, a requirement that ensures the traceability of I&T operations. As a result, this researcher suggests investigating new ways of reducing the user's workload further for future work.

Another source of discontent was the fact that, as expected, the users sometimes wanted to be able to modify entries, which once again was at odds with the requirement to

prevent tampering with data in the system, and one of the main advantages of this technology over paper. Nevertheless, as a result of this response, this researcher reviewed the design of E-Logbook and, wherever possible, created additional capabilities to modify entries in a manner consistent with requirements. An example is the case of the new features introduced for component entry in each I&T Log, as described in Section 4.2.3. This also greatly helped in the resolution of a serious problem in the LAT I&T project: the lack of a component naming convention that caused multiple sources of confusion among users.

Another important input from the users was their desire to expand the operation of E-Logbook to the subsystems in charge of building the FHW and FSW. This would allow the component information to be preloaded by the subsystems prior to shipment to the I&T facility, therefore reducing the I&T personnel workload greatly and also ensuring that a default naming convention was established throughout the project.

Finally, one of the QA engineers also suggested an improvement on the usability of E-Logbook based on the idea of interfacing it with other I&T tools, such as work orders or assembly drawings. This idea opens a whole new implementation level that this researcher suggests as future work.

Beyond these initial results, several factors were taken into account in order to ascertain the user's perspective on the success of this research:

- *Whether the user prefers E-Logbook over paper.*
- *Whether the user needs to use printed copies of the data stored in E-Logbook.*
- *Whether E-Logbook performs to a level of satisfaction.*
- *Whether the user recommends the implementation of E-Logbook in industry.*

Subject	Better than paper to organize / input / output Data	# Times Data Lost	# Times Print to Paper	Recommend use in Industry QNov05	Overall Level of Satisfaction QNov05	Recommend use in Industry QFeb06	Overall Level of Satisfaction QFeb06
N1	Definitely / Somewhat / Definitely	0	0	Definitely, it's a beautiful application	4	Yes	4
N2	Yes/Yes/Yes	0	0	Yes	3	Yes	4
N3	Yes/Yes/Above all	0	10	Yes	3	Yes	4
N4	Yes/Yes/Yes	3	0	Absolutely!	4	Yes	4
N5	Yes/Yes/Yes	3	0	Yes	3	Yes	4
N6	Better /Better/Better	0	0	Yes	3	Yes	4
N7	Yes/Somewhat /Yes	2	0	Needs more updates	2	Yes	3
N8	Yes/Yes/Yes	0	0	Yes	3		3
N9	Yes/Yes/Yes	0	0	Yes	4	Yes, I love it.	3
N10	Yes/Yes/Yes	0	0	No		Yes	4
N11	Good/Good/Good	0	0	Yes	4	Yes by far	4
N12	Yes/Yes/Yes	0	5	Yes	3	Yes	4
N13		3	0			No	1
N14				Yes	4	Yes	4
N15				Yes	4	Yes	4
N16				Yes	4	Yes	4
TOT	100%/92% /100%	11	15	90%	3.33	94%	3.63

Very Satisfactory	4
Satisfactory	3
Neutral	2
Unsatisfactory	1
Very Unsatisfactory	0
Not Used	
Not Completed	

Table 10: E-Logbook main usability results

Table 10 shows the main results obtained in answering these questions in QNov05 and QFeb06.

In November 2005, 2 out of 10 subjects believe that E-Logbook is somewhat better than paper to input data. Considering that one of the major obstacles for the success of E-Logbook was whether the users would not prefer having to input information in the less flexible computer format, this result is quite satisfactory.

In the 19 months that E-Logbook was used in the LAT I&T facility, only 2 out of the 16 subjects of the questionnaire needed to print to paper E-Logbook's data. N3 used this option to attach EGSE setup records to the EGSE setup procedure document. N12 used it to attach Shift Log information to the run test procedure documentation. This result expresses the lack of need of paper on a regular basis for I&T activities, as opposed to the initial findings of this researcher regarding the belief that paper is crucial to undertake space systems I&T operations. Incidentally, this result also demonstrates the users' confidence in E-Logbook's reliability.

In November 2005, 12 out of the 13 users that most heavily used E-Logbook filled in the questionnaire. Among these 12 users, 10 were satisfied with E-Logbook and recommended its use to the industry, 1 was unsure and 1, a QA engineer, was not satisfied.

After the implementation of changes on the system based on their recommendations in November 2005, 100% of this initial set of users recommended the use of E-Logbook in the industry in February 2006. The user that did not fill in the questionnaire in November 2005 did not. This only stresses once more the fact that user feedback is crucial in the successful implementation of any human-computer interactive technology, regardless of the reliability of this technology. Analysis of the causes of dissatisfaction of this user will be shown in the following section.

The final result on the overall satisfaction level of the users subject to E-Logbook's usability questionnaire for the LAT I&T project indicate that 15 out of 16 users, or 94%, were satisfied and recommend its use to the industry. The significance of this result is of

course limited by the small set of users that have been so far exposed to E-Logbook, but must be considered relevant nevertheless due to their average 14 years of experience in the industry.

While a 94% overall user confidence level is a good result for a technology that did not exist in the past, this researcher believes that there is still much room for improvement, and so proposes new avenues of implementation in the following Chapter.

6.2.2 Usability Criteria Results

The central portion of E-Logbook's usability questionnaire was complimented by the I&T engineers and technicians that most heavily used E-Logbook during the 19 months of LAT I&T effort at SLAC. The criteria by which these users evaluated the design of E-Logbook were adapted from Reference [83] to the needs of this research, and are defined as follows:

- *Visual Clarity*: Whether the information displayed on the screen is clear, well organized, unambiguous and easy to read.
- *Consistency and Compatibility*: Whether the way the system looks and works is consistent and compatible at all times.
- *Explicitness*: Whether the way the system works is clear to the user.
- *Appropriate Functionality*: Whether the system meets the needs and requirements of users when carrying out tasks.
- *Informative Feedback and User Guidance*: Whether the users are given clear, informative feedback on where are they in the system. Whether relevant support is provided both on the computer and in a hard copy document.

- *Error Prevention and Correction*: Whether the system is designed to minimize the possibility of user error.
- *Flexibility and Control*: Whether the interface is sufficiently flexible in structure to suit the needs and requirements of all users, and allow them to feel in control of the system.
- *System Usability Problems*: Whether the users had any problems while using the system.
- *E-Logbook Feedback*: Whether the user felt involved in the design process of E-Logbook.

Each of these criteria is evaluated in the questionnaire by means of a table with questions to be answered checking one of three or four options: always, most of the time, some of the time, never; or no problems, minor problems and major problems. An additional column for comments concludes the tables. Additional space for comments is found underneath the tables, followed by a rating box that the subject uses to rate the system's overall performance with regards to section criteria: very satisfactory, satisfactory, neutral, unsatisfactory or very unsatisfactory.

In order to analyze the results of E-Logbook on each usability criterion, the subjects' answers have been compiled in the tables shown in Appendix E. Each table correlates the users' answers (columns) with the questions asked (rows). The results are evaluated using scales that assign unit increments from worst to best ratings, starting at 0. For example, never, some of the time, most of the time and always are valued as 0, 1, 2 and 3, respectively. An additional column and row show each the question's and user's average ratings. To help the analysis, the cells have been color coded based on the rating scale. The best and worst question and user average ratings have been colored in green and red respectively in each table. The global users' and questions' average ratings obtained in the table are shown at its bottom.

In order to compare the results of QNov05 and QFeb06, the global users' average rating based on each criterion have been compiled in additional tables shown in the following sections. For QFeb06, the average ratings for the same set of subjects that complimented QNov05 (identified as "Initial Set") are also shown. The values are displayed in percentages. The relative improvement has been calculated using the following relative increase equation:

$$\text{Relative Improvement} = \frac{QFeb06 - QNov05}{QNov05} \quad (\text{Eq. 1})$$

In the following analysis, the column results were used to evaluate not only a certain user's level of satisfaction, but also to combine the users' impressions on the system based on their job specialization. The row results were used to evaluate E-Logbook in terms of the questions asked for each evaluation criterion.

6.2.2.1 Visual Clarity

One of E-Logbook’s main design decisions was the implementation of GUIs that do not overwhelm the user with too much information, and employ highlights only to show critical information and errors. This generates problems for first-time users who did not use the tutorial, but reduces the user’s workload in the long run.

In E-Logbook, color was used only to indicate critical states: to distinguish mate and demate records (displayed in green) in the M/D Log, and to indicate validation expiration states (displayed in green, orange and red) in the EGSE Log.

The results for QNov05 (Table 33, Appendix E) on visual clarity indicated that some of the users would have liked the use of more color highlighting in the system. After QNov05, this researcher still believed that the use of an uncluttered system would perform better on the long run, and little changes were implemented on this front before QFeb06. On the other hand, the layout of the M/D Record GUI and the Main GUI were revised to optimize its clarity of use.

In QFeb06 (Table 34, Appendix E) the system shows an improvement based on visual clarity, which proves the benefits of minimizing the user’s visual workload when operating the GUIs. Table 11 summarizes the results obtained on QFeb05 and QFeb06 as well as the relative improvement between questionnaires. As mentioned before, the user average satisfaction rating in QFeb06 is also shown for the initial set of heavy users that filled in QNov05 (N1-N12) to correlate their results.

Visual Clarity	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	77.32%	87.91%	90.48%
Relative Improvement		13.70%	17.02%

Table 11: Visual Clarity average statistics comparison between QNov05 and QFeb06

6.2.2.2 Consistency and Compatibility

The results of QFeb05 (Table 35, Appendix E) on consistency and compatibility indicate that the users wanted to be able to visualize results in E-Logbook in the English system of units as a default. A direct result of the analysis was idea of creating a weight tab in the Preferences menu of the Main GUI to customize the user's weight selection. Worth noting is the fact that the reason why this capability was not implemented earlier on is that one of the requirements for the LAT I&T project was that all units be stored and displayed using the International System, as explained in Section 3.1. This researcher is currently in the process of obtaining the right permissions to proceed to implement this feature, which she believes can greatly improve the usage of E-Logbook.

In QFeb06 (Table 36, Appendix E and Table 12), there was overall a sensible improvement in E-Logbook based in this property, with seven perfect user score ratings. Q5 yielded again the worst ratings, as expected since the resolution of a customized weight preferences menu is still under implementation. An interesting result was that both Test Conductors had problems with the consistency and compatibility of E-Logbook, inconsistent with the fact that they only used one of the I&T Logs (the Shift Log) and were therefore not exposed to the rest of the system.

Consistency and Compatibility	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	69.97%	88.60%	90.12%
Relative Improvement		26.62%	28.80%

Table 12: Consistency and Compatibility average statistics comparison between QNov05 and QFeb06

6.2.2.3 Explicitness

In E-Logbook, no changes on the system affect other parts of the system other than the I&T Log the changes correspond to, with the exception of the customizations on the Preferences Menu. This is a result of the tree structure followed in the design, as shown in Figure 15 and Figure 43. As a consequence, the results on QNov05 for explicitness (Q5 in particular, Table 37, Appendix E) demonstrate that either the users were not aware of this fact, or they did not acquire the proper training.

Combined with the User Guidance results in Section 6.2.2.5, this shows that the users were not aware of the existence of a tutorial. However, this researcher had drawn attention to the existence of a tutorial in the first interviews she had with the users and the LAT I&T facility Test Director. As a result of QNov05, this researcher frequently reminded the users of the existence of both a hard copy and online version of the tutorial, as well as their location. On the other hand, the overall high ratings in explicitness demonstrate that E-Logbook was easy to learn, since it can be done without the use of a tutorial.

Based on explicitness, in QFeb06 (Table 38, Appendix E and Table 13) there is once more a sensible improvement on E-Logbook. In words of N9, “it is unclear if the improvement on the results is a consequence of the system organization and/or structure, or due to the extensive use of the system. On the same token, nothing is clear without a first explanation. Bottom line the users are navigating through the menus with great ease”.

Explicitness	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	75.15%	89.74%	93.33%
Relative Improvement		19.42%	24.19%

Table 13: Explicitness average statistics comparison between QNov05 and QFeb06

6.2.2.4 Appropriate Functionality

Appropriate functionality is one of the most important factors that determine the success of E-Logbook, since it indicates if the requirements defined and implemented throughout its development effort were satisfactory.

Before QNov05 (Table 39, Appendix E) E-Logbook's implementation effort had focused primarily on ensuring that all requirements had been identified, although effort to make it usable had also been underway.

After analyzing the results of QNov05 on appropriate functionality, when it became clear that E-Logbook was stable in terms of requirements, the work focus shifted to improving the usability of the system. The need of new reports was identified (as described in Chapter 4), for example the MMR Report and the Unit M/D Report. The resolution of the slow down problem in the M/D Log was carried out. Direct access to the M/D Record from the Main GUI was implemented. As a consequence, the overall ratings in QFeb06 (Table 40, Appendix E and Table 14) improved.

Appropriate Functionality	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	82.88%	89.23%	92.22%
Relative Improvement		7.66%	11.27%

Table 14: Appropriate Functionality average statistics comparison between QNov05 and QFeb06

6.2.2.5 Informative Feedback and User Guidance

As already mentioned above, the results on QFeb05 (Table 41, Appendix E) based on this criterion were surprising since half of the users were not aware of the existence of a tutorial, 6 of them did not know of online access to the tutorial and 7 did not know of the existence of a hard copy of the tutorial. As a consequence, the users were clearly informed of the existence of a hard copy of the tutorial in the LAT I&T facility, and of how to access it online from the Help menu in E-Logbook.

By the time QFeb06 took place (Table 42, Appendix E and Table 15), all of the users acknowledged the existence of a tutorial both online and in hard copy, but only 10 of the subjects used the hard copy and 4 the online version. This result only ratifies the typical human behavior of skipping the instructions of a product. Another interesting result on this front and frequently obtained in the questionnaire was the user’s belief that they “do not make mistakes”, caused probably by the “no error” culture rooted in the aerospace industry. 4 out of the 10 subjects were not fully satisfied with the instructions on the hard copy tutorial. As a result of QFeb06, the tutorial was revised²⁷.

Informative Feedback and User Guidance	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	77.31%	87.77%	90.14%
Relative Improvement		13.52%	16.59%

Table 15: Informative Feedback and User Guidance statistics comparison between QNov05 and QFeb06

²⁷ The final version of the tutorial can be found on the attached DVD under *Documentation/Tutorial*.

6.2.2.6 Error Prevention and Correction

One of the clear problems exposed in the analysis of QNov05 (Table 43, Appendix E) was the users' frustration with the lack of flexibility to correct errors in E-Logbook. This problem was also aggravated by the lack of a standard component naming convention for the LAT hardware, which, even though it was not caused by E-Logbook, definitely made the use of the system more difficult. In some cases the inability to correct errors was an I&T requirement, in order to ensure that no data in the system was tampered. In the case of the Shift Log, there was the constraint that no records were modified once they were transferred to Oracle, as explained in Chapter 5.

While there was no possibility of changing the no edit requirement for the installation records or the Shift Log, some progress could be made in the maintenance of the system components in E-Logbook. As a result of QNov05, this researcher developed a detailed system to help create, edit, rename, duplicate and remove components in E-Logbook, as described in Section 4.2.3. The process of component entry and edit greatly improve, as seen in the results on QFeb06 (Table 44, Appendix E and Table 16). Nevertheless, some users still maintain that they do not commit errors, while others are still dissatisfied (N8) that neither the activity entries nor the component installation records can be corrected. The users also state once more that a naming convention is needed, and that if there was a way to have the components preloaded by the subsystems responsible for building the components their workload will be greatly reduced. This idea is proposed as future work.

Error Prevention and Correction	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	70.39%	84.83%	87.04%
Relative Improvement		20.52%	23.66%

Table 16: Error Prevention and Correction average statistics comparison between QNov05 and QFeb06

6.2.2.7 Flexibility and Control

The results on this criterion in QNov05 (Table 45, Appendix E) can be analyzed in a similar fashion as for Error Prevention and Correction, since it indicates even more clearly that the users wanted to have more flexibility to modify entries.

However, one of the main requirements created for E-Logbook was to ensure the incorruptibility of the data, and this included modifications of the data performed by the users. In addition, mechanisms such as comments in each record were implemented in E-Logbook precisely to provide the users a place to add or expand on the information of a process. Another problem also captured by this criterion was the need for a default weight preferences menu, as already mentioned before.

Once the new mechanisms to facilitate the entry of component names were implemented, the users' feeling of flexibility and control on E-Logbook greatly improved, as shown in QFeb06 (Table 46, Appendix E and Table 17). More ways of improving the user's flexibility and control of E-Logbook are still being investigated, as described in the following Chapter.

Flexibility and Control	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	59.03%	82.95%	87.64%
Relative Improvement		40.52%	48.47%

Table 17: Flexibility and Control average statistics comparison between QNov05 and QFeb06

6.2.2.8 System Usability Problems

The questions asked in this section were focused on estimating the level of discomfort for the different usability problems that could be identified in E-Logbook. As expected by the analysis of the previous sections, in QNov05 (Table 47, Appendix E) the subjects were concerned about three aspects of the design of the system:

- *An inflexible, rigid structure:* The impossibility of entering information in more than one way.
- *System response times too slow:* Caused largely by the slow-down problem in the M/D Log.
- *Having to be too careful to avoid errors:* Due to the I&T requirement of no data modification.

One of the users (N13) gave overall low ratings to E-Logbook in QFeb06 (Table 48, Appendix E and Table 18) claiming that “it takes too long to enter the data, too much information is needed, and the connector structure is too confusing”, and that it is difficult to understand “which fields are mandatory and which fields require a correct answer”. The impression of this researcher is that, given the extensive experience of the subject in space system I&T, he was not comfortable changing the way he was used to perform logging activities. Figure 51 shows how N13 (identified as MM in the report) is used to fill reports on paper. N13 groups the information for the R/Ds using arrows to expand answers from the first R/D to the last. The record is concise and clear, but practices like this are what the I&T managers are trying to avoid with the implementation of an electronic logging system.

The capability of creating a record from the previous one is not recommended since it is prone to error: If the widgets are empty, the system can warn the user of missing entries, and the users can verify the accuracy of the data. If the widgets are preloaded with previous record information, the system cannot verify that new data has been entered and users can forget to update information on widgets that are already filled.

The claim that too much information is required in the system was made by more subjects, regardless, and as a result this researcher recommends new ways to minimize the process of data entry in E-Logbook. An example is the use of time-saving technology like finger print readers, as described in the section of this thesis on future work.

System Usability Problems	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	83.37%	87.61%	90.28%
Relative Improvement		5.08%	8.28%

Table 18: System Usability average statistics comparison between QNov05 and QFeb06

CONNECTOR MATE / DEMATE

TEM

UNIT DESCRIPTION:

Connector(s) Connector Reference Designator	Authorized by Procedure & para or NCR	Date MD/Y	Mate or De-mate M or D	Flight or Test F or T	Verify Power Off Emp. ID#	Pre-mate Inspect		ESD Bleed and Connector Mate		Final Inspect Fasteners Torqued, Witness Signae applied	
						*Emp. ID#	QA	*Emp. ID#	QA	*Emp. ID#	QA
JT7	LAT-ND-3875	8/5/04	M	T	TL	TL	TL	TL	TL	TL	TL
JT7	3875	8/5/04	D	T	TL	NA	TL	TL	TL	TL	TL
J51	3875	8/5/04	M	T	TL	NA	TL	TL	TL	TL	TL
J51	3287	8/5/04	M	T	BA	BA	BA	BA	BA	BA	BA
J40	LAT-ND-EM238	8/6/04	D	T	NA	NA	NA	NA	NA	NA	NA
JT1			M	T							
JT2											
JT3											
JT4											
JT5											
JT6											
JT7											

*Personnel that is Mate/Demate certified.

Connector /Bracket R/D: JT5-J51

Figure 51: Second sample of paper M/D Report used initially for the LAT I&T Project

6.2.2.9 E-Logbook Feedback

One of the main goals of this thesis work was to include the user's feedback throughout the design and implementation process. Therefore, a final section in E-Logbook's Usability Questionnaire was created in order to verify how integrated the users felt in the process of implementation of the system.

In QNov05 (Table 49, Appendix E) Q1 and Q2 results are somewhat contradictory, since the users seem not to feel greatly involved in the design, but on the other hand believe their recommendations are satisfactorily implemented from one release to the next. The reason for these results may be that the users would have liked to be more involved in the design, and as a result of QNov05 requests for their feedback increased. The users felt overall that E-Logbook improved from one release to the next, and that this researcher was available for feedback and problem solving.

In QFeb06 (Table 50, Appendix E and Table 19) the users feel more involved in the design of E-Logbook overall.

E-Logbook Feedback	QNov05	QFeb06	QFeb06 (N1-N12)
User Satisfaction Average Rating	84.85%	91.88%	93.29%
Relative Improvement		8.29%	9.95%

Table 19: E-Logbook Feedback average statistics

6.2.2.10 Summary of Criteria Results

The results on E-Logbook's based on the user's overall rating of each criteria at the bottom of each section on the questionnaire is shown in Table 20 and Table 21.

Apart from the problem of dissatisfaction of the user N13, already discussed in the system usability problems section, worth noting is the case of the user N8. This user is the only case where the ratings, even though slightly, decrease. The user, a Test Conductor, is above all dissatisfied with the fact that he can't modify entries in the Shift Log. This user has limited experience in the aerospace industry since he only started working in this field in October 2005, for GLAST. Therefore, his ratings have to be taken cautiously: It is safe to assume that he is not fully aware of the need of discipline to keep track and prevent data modification for space system I&T operations.

The overall results are consistent with the average ratings obtained from each subject's answers to the questions in the criteria tables, summarized for clarity in Figure 52. The results also show the clear improvement performed on the usability of E-Logbook as a result of QNov05. This is even more noticeable for the initial set of subjects, whose recommendations were addressed.

The biggest improvement concerns the flexibility and control criteria, based on the implementation of component edit capabilities and additional reports. The error prevention and correction is also improved for the same reasons. The best mark is obtained for E-Logbook feedback, a quite satisfactory result that proves the applicability of the user's feedback in the implementation process. These results also demonstrate how this practice can only improve the overall design of a new product or technology.

Criteria	User												Question Rating			
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCCond.	TCCond.	QA	QA		TDir.		
Visual Clarity	1	2	3	4	4	5	6	7	8	9	10	11	12	3.27		
	4	3	4	4	4	3	3	3	3	3	2	4	3	3.27		
	4	3	4	4	4	3	2	3	3	3	3	4	3	3.09		
Consistency and Compatibility	4	3	4	4	4	3	2	3	4	2	2	4	3	3.09		
	4	3	3	4	4	3	2	3	4	2	2	4	3	3.64		
	4	3	4	4	4	3	2	3	4	2	2	4	3	2.67		
Appropriate Functionality	4	4	4	4	4	4	3	3	4	3	3	4	4	3.64		
	4	4	4	4	4	4	3	3	4	3	3	4	3	2.67		
	4	2	3	2	3	3	3	2	2	2	2	4	3	2.91		
Informative Feedback and User Guidance	3	3	3	4	4	3	3	2	2	3	3	3	3	2.91		
	4	3	4	4	3	3	2	2	3	3	1	4	3	2.91		
	4	3	4	4	3	3	2	2	3	3	3	4	3	2.91		
Error Prevention and Correction	4	3	4	4	4	4	3	2	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
Flexibility and Control	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
E-Logbook Feedback	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
	4	3	4	4	4	4	3	4	2	3	3	4	4	3.45		
User Rating	3.88	3.00	3.67	3.63	3.29	2.63	2.75	2.83	2.75	2.71	4.00	3.25	3.25	3.45		
	Average Rating												3.20	3.15	79.95%	78.79%
													3.20	3.15	79.95%	78.79%

Table 20: Overall Criteria Rating Results on QNov05

Very satisfactory	4
Satisfactory	3
Neutral	2
Unsatisfactory	1
Very unsatisfactory	0
Not used	
Not completed	

Criteria	User													Question Rating
	Prog.	E Tech.	E Tech.	M Tech.	M Tech.	M Tech.	M Tech.	M Tech.	T Cond.	T Cond.	T Cond.	QA	T Dir.	
Visual Clarity	1	2	3	4	5	6	7	8	9	10	11	12	13	3.69
Consistency and Compatibility	4	4	3	4	4	4	3	4	3	4	4	4	3	3.62
Explicitness	4	4	4	4	4	4	3	3	3	4	4	4	2	3.54
Appropriate Functionality	4	4	4	4	4	4	3	3	3	4	4	4	2	3.62
Informative Feedback and User Guidance	4	4	3	3	4	4	3	3	3	4	4	4	3	3.54
Error Prevention and Correction	4	4	4	4	4	4	3	1	3	4	4	4	1	3.38
Flexibility and Control	4	4	3	3	4	4	3	2	3	4	4	4	1	3.31
E-Logbook Feedback	4	4	4	4	4	4	3	3	3	4	4	4	4	3.75
User Rating	4.00	4.00	3.63	3.63	4.00	4.00	3.00	2.75	3.00	4.00	4.00	4.00	2.00	
Average Rating										3.54	3.56	88.46%	88.88%	
Average Rating Initial Set										3.67	3.71	91.67%	92.73%	

Table 21: Overall Criteria Rating Results on QFeb06

4	Very satisfactory
3	Satisfactory
2	Neutral
1	Unsatisfactory
0	Very unsatisfactory
	Not used
	Not completed

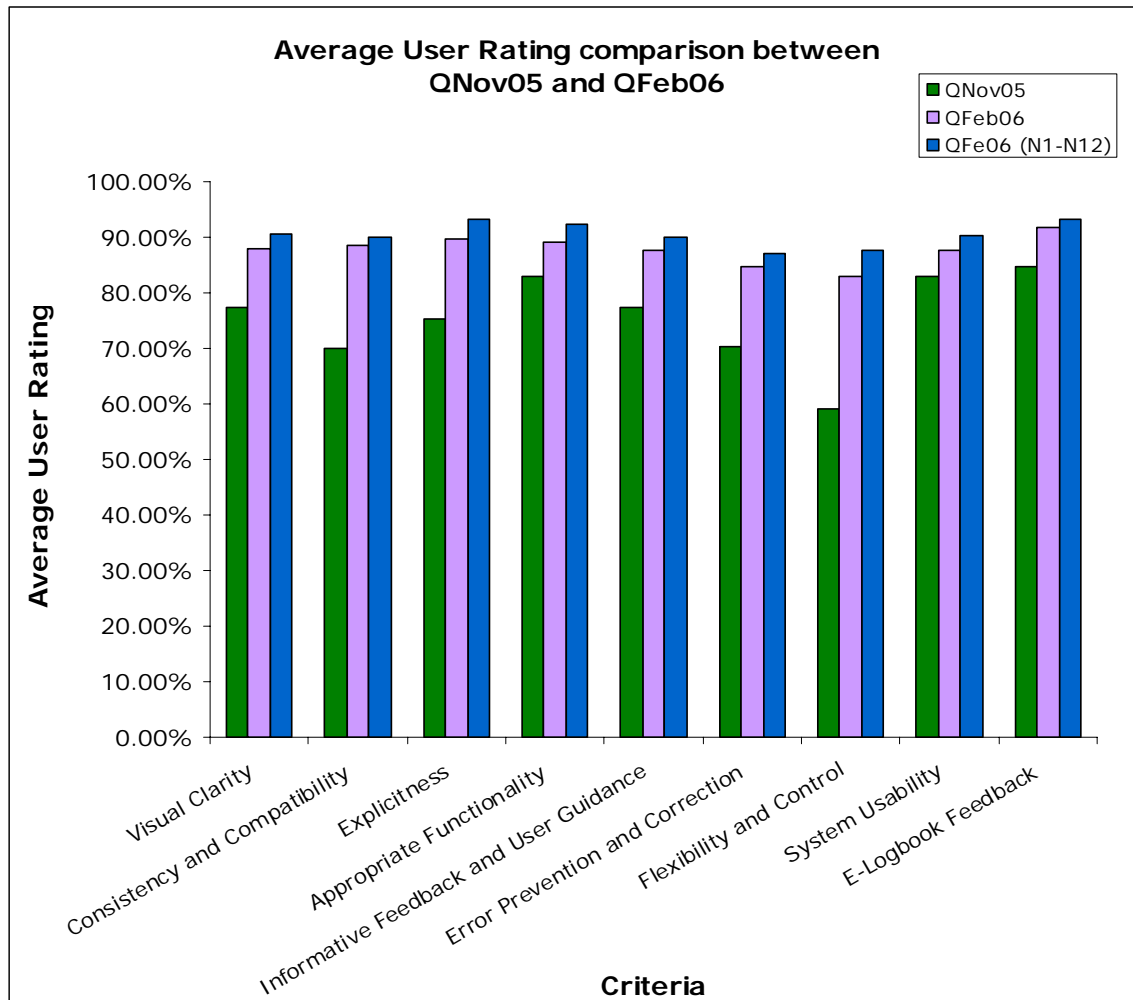


Figure 52: Correlation of overall question ratings after for QNov05, QFeb06 and the initial set of subjects in QFeb06

Overall, Figure 52 shows that the level of satisfaction of the subjects on E-Logbook based on this comprehensive set of criteria is on or above the 90% mark. Once again, while this result is quite satisfactory, there is still a lot of room for improvement of E-Logbook beyond these results, as will be described in the following Chapter.

Chapter 7

7 Conclusions and Future Work

This research shows that it is technically feasible to replace the use of paper logs for space system I&T operations. The new resulting technology, E-Logbook, provides capabilities that greatly enhance the management of information and enforce discipline in the documentation of I&T operations. Moreover, the information contained in such technology is instantly retrievable, and helps monitor the performance and overall progress of the I&T project in ways that are not even imaginable in paper. If a new report format is needed, this technology can provide it quickly with minimal modifications.

The design of E-Logbook focused on maximizing the benefits of electronic database technology, as well as the level of satisfaction of the users employing it. In order to do so, a stringent and extensive set of requirements was defined at every level of implementation in order to ensure that the final system would perform to be best of its abilities.

The identification of the requirements for each I&T logging activity was crucial, not only because such technology did not exist in the past, but also to optimize the storage of information and to create the right reporting mechanisms that would be valuable to the users.

Special attention was devoted to the design of a flexible database schema that could maximize the benefits of the relational model and easily adapt to the advent of new requirements. Furthermore, this design was carefully detached from customizations for its

use for the GLAST LAT I&T project. Only the tables that held ground test output information, *elogreport*, *LICOS_states* and *LICOS_activities*, were constrained to its use for this project. This translates in the HCI-GUI system into only the Run Report and LAT Matrix Report GUIs created for its exclusive use for the LAT I&T.

E-Logbook benefits not only from the capabilities of electronic database technology, but also from the inclusion of user feedback early in the design process through the means of an iterative design loop. With user feedback, the design of E-Logbook has improved and become more solid, and problems in its application for space systems I&T operations have been identified.

In the 19 months of operation of E-Logbook for the LAT I&T project, E-Logbook has been highly reliable, with 0% of data being corrupted and only a 0.0006% non-critical loss of the total amount of data stored in its database schema. Its design has matured to the extent that more than 94% of the experienced users exposed to it are satisfied with its performance and recommend its use in the industry.

Nevertheless, there is still a lot of room for improvement. First, the inclusion of additional capabilities can the amount of time spent in the process of entering information. Second, the retrieval of information can be improved through the creation of additional reporting mechanisms. Third, the interface of E-Logbook with other I&T electronic applications can be created.

Future work proposed by this researcher can be summarized in the following points:

- *Optimize user input with the use of personal identification devices:* Investigate the usefulness of personal identification hardware devices, such as fingerprint readers or card readers, to reduce the user's workload in the process of record sign-off.
- *Add a spell check engine:* Implement a spell check engine that help the users reduce the number of typographical errors when entering information in E-Logbook.

- *Design a new component classification system that matches the I&T installation level:* To help the user find a component or process faster, reorganize the component entry in the *elogcomponent* and *elogconnector* tables so that it matches the project's integration hierarchy (for example CAL into GRID, GRID into LAT). This also opens the implementation of a complete new set of reports that list records not only by component but also by hierarchy level.
- *Design a dynamic search engine:* Create a search engine that dynamically generates reports based on the user's preferences and field selection.
- *Create E-Logbook interfaces to other relevant space system I&T electronic tools:* Investigate which other electronic tools (assembly drawings, work orders, test procedures) can be linked to E-Logbook in order to improve the overall management of the I&T operations.

The new technology that E-Logbook embodies can easily be adapted for use not only in other space system I&T projects, but also in any kind of industry that needs to document the process of integration and test of mechanical and electrical parts. Examples include planes, submarines, medical devices, power plants, and high energy detectors (linear accelerators, synchrotron accelerators). E-Logbook's applications are innumerable, and even more important, its design is ready for use in all these areas with minimal modifications.

New technologies in today's computer world can only succeed if they are devoted to maximizing the satisfaction of the human being that operates them. Regardless of their efficiency and reliability, this will be the determining factor of their success. The research described in this thesis hopes to have achieved this goal and, ultimately, to have created a useful tool.

APPENDIX A: E-Logbook's I&T Logs Requirements

This appendix compiles for reference the database fields and resulting GUIs not shown already in the dissertation for each of E-Logbook's I&T Documentation Components.

A.1 User Log

Field	Type
Name	<text>
Initials	<text> Max. 3 characters
Group	<text>
Job Title	<text, from preselected list>
Status	<Active, Inactive>
Username	<text>
Password	<text> Encrypted

Table 22: User Log Database Fields

The screenshot shows a window titled "Edit User" with the following fields and values:

- Status: Active (dropdown menu)
- Name: Eliazar Ortiz (text box)
- Initials: EO (text box)
- Group: I&T (text box)
- Job Title: All (dropdown menu)
- New User Name: ortize (text box)
- Enter New Password: xxxxxxxx (password box)
- Re-enter New Password: xxxxxxxx (password box)

At the bottom of the window are "Save" and "Cancel" buttons.

Figure 53: User Record GUI

The image shows a graphical user interface window titled "User Log". It contains a table with five columns: Status, Name, Initials, Group, and Job Title. The table lists 15 users, with the last one, Reggie Rogers, marked as INACTIVE. Below the table are three buttons: Print, Add User, and Done.

Status	Name	Initials	Group	Job Title
ACTIVE	Phuc Hoang	PH	I&T	INT Elec. Tech.
ACTIVE	Ric Claus	RC	ONLINE	All
ACTIVE	Roger Williams	RW	TKR	TKR Engineer
ACTIVE	Selim Tuvi	ST	Online	All
ACTIVE	Su Dong	SD	ELEX	INT Physicist
ACTIVE	Takuya Kawamoto	TK2	TKR	TKR Physicist
ACTIVE	Toan Le	TL	I&T	INT Elec. Tech.
ACTIVE	Tom Borden		I&T	INT Engineer
ACTIVE	Tom Nieland	TN	I&T	INT Mech. Tech.
ACTIVE	Tsunefumi Mizuno	TM	TKR	TKR Phycisist
ACTIVE	Tsunefumi Mizuno	TM	TKR	TKR Physicist
ACTIVE	Tune Kamae	TK	I&T	INT Physicist
ACTIVE	Xin Chen		I&T	INT Physicist
INACTIVE	Reggie Rogers	RR	I&T	INT Physicist

Buttons: Print, Add User, Done

Figure 54: User Report GUI

A.2 Shift Log

Shift Log			
Field	Type		
Shift ID	<date+type>		
Last Update	<timestamp>		
Summary/Plan:			
Summary/Plan	<text>		
Shifter:			
Shifter Title	<text, from preselected list>		
Shifter/User ID	<ID/Password>		
For MAIN, ACD, CAL, INT, TKR:			
	Activity	Problem	Other
Operator ID	<ID/Password>	<ID/Password>	<ID/Password>
Date/Time	<timestamp>	<timestamp>	<timestamp>
Note	<text>	<text>	<text>
Last Update	<timestamp>	<timestamp>	<timestamp>
Run Report:			
Fields created on by LAT Test Executive (LATTE) Run Control test output, displayed in report format.	<text tuple>		

Table 23: Shift Log Database Fields

A.3 M/D Log

M/D Log	
Field	Type
Date/Time	<timestamp>
Connector R/D	<text>
Connector Description	<text>
Action	<Mate (M), Demate (D)>
Authorizing Document	<text>
Type	<Flight (F), Test (T)>
Comments	<text tuple>
Operator Sign Off:	
Verify Power Off Operator	<ID/Password> <timestamp>
Verify Power Off QA	<ID/Password> <timestamp>
Pre-mate/demate Inspect OP	<ID/Password> <timestamp>
Pre-mate/demate Inspect QA	<ID/Password> <timestamp>
ESD Bleed/Mate & Mate, or Demate OP	<ID/Password> <timestamp>
ESD Bleed/Mate & Mate, or Demate QA	<ID/Password> <timestamp>
Torque OP (if applicable)	<ID/Password> <timestamp>
Torque QA (if applicable)	<ID/Password> <timestamp>
Final Inspect OP	<ID/Password> <timestamp>
Final Inspect QA	<ID/Password> <timestamp>
Torque (if applicable):	
Torque Value	<float, N/A>
Torque Tool	<text>
Torque Tool Expiration	<timestamp>

Table 24: M/D Log Database Fields

A.4 FHW Log

FHW Component Installation Log		
Field	Type	
Part Info:		
Date/Time	<timestamp>	
OP	<ID/Password>	<timestamp>
QA	<ID/Password>	<timestamp>
Component R/D	<text>	
Component Description	<text>	
Authorizing Document	<text>	
Part Number	<text>	
Serial Number	<text>	
Weight (Kg)	<float>	
Comments	<text tuple>	
Flatness (if applicable):		
Value	<float, N/A>	
Verify OP	<ID/Password>	<timestamp>
Verify QA	<ID/Password>	<timestamp>
Torques (if applicable):		
Values, Tools, Tool Exps	<text tuple>	
Apply Torque OP	<ID/Password>	<timestamp>
Apply Torque QA	<ID/Password>	<timestamp>
MMR (if applicable):		
MMR #	<float, N/A>	
Witness OP	<ID/Password>	<timestamp>
Witness QA	<ID/Password>	<timestamp>
Grounding (if applicable):		
Limit Per #	<float, N/A>	
Resistance (must be smaller than limit per)	<float>	
Grounding Tool	<text>	
Grounding Tool Expiration	<timestamp>	
Grounding OP	<ID/Password>	<timestamp>
Grounding QA	<ID/Password>	<timestamp>
Label Tape (if applicable):		
Label Tape	<text, N/A>	
Label Tape Expiration	<timestamp>	
Label Tape OP	<ID/Password>	<timestamp>

Table 25: FHW Log Database Fields

FHW Matrix Log	
Field	Type
OP	<ID/Password>
Date/Time	<timestamp>
FHW Serial Number	<text tuple in grid format grouped by bay #>
FHW Weight	<int tuple in grid format grouped by bay #>

Table 26: FHW Matrix Log fields retrieved from FHW Log.

FHW Component Installation Record

Unit Description: LAT TOWER ASSEMBLY LAT-DS-02560 001
 Component R/D: TKR0
 ACTION: INSTALL

Part | Flatness | Torque | MMR | Grounding | Label Tape

Date: 04/06/2005 10:48:57 AM
 Authorizing Doc.: AIDS 332
 Part Number: LAT-DS-00722
 Serial Number: A
 Weight: 37.875 Kilograms
 Operator: QA:
 Record Creation: Eliazar Ortiz Doug Bartholomew

Notes: Add

Date	User	Comment

Last Modified: 08/26/2005 11:53:47 AM

Done Cancel

Figure 55: FHW Record GUI

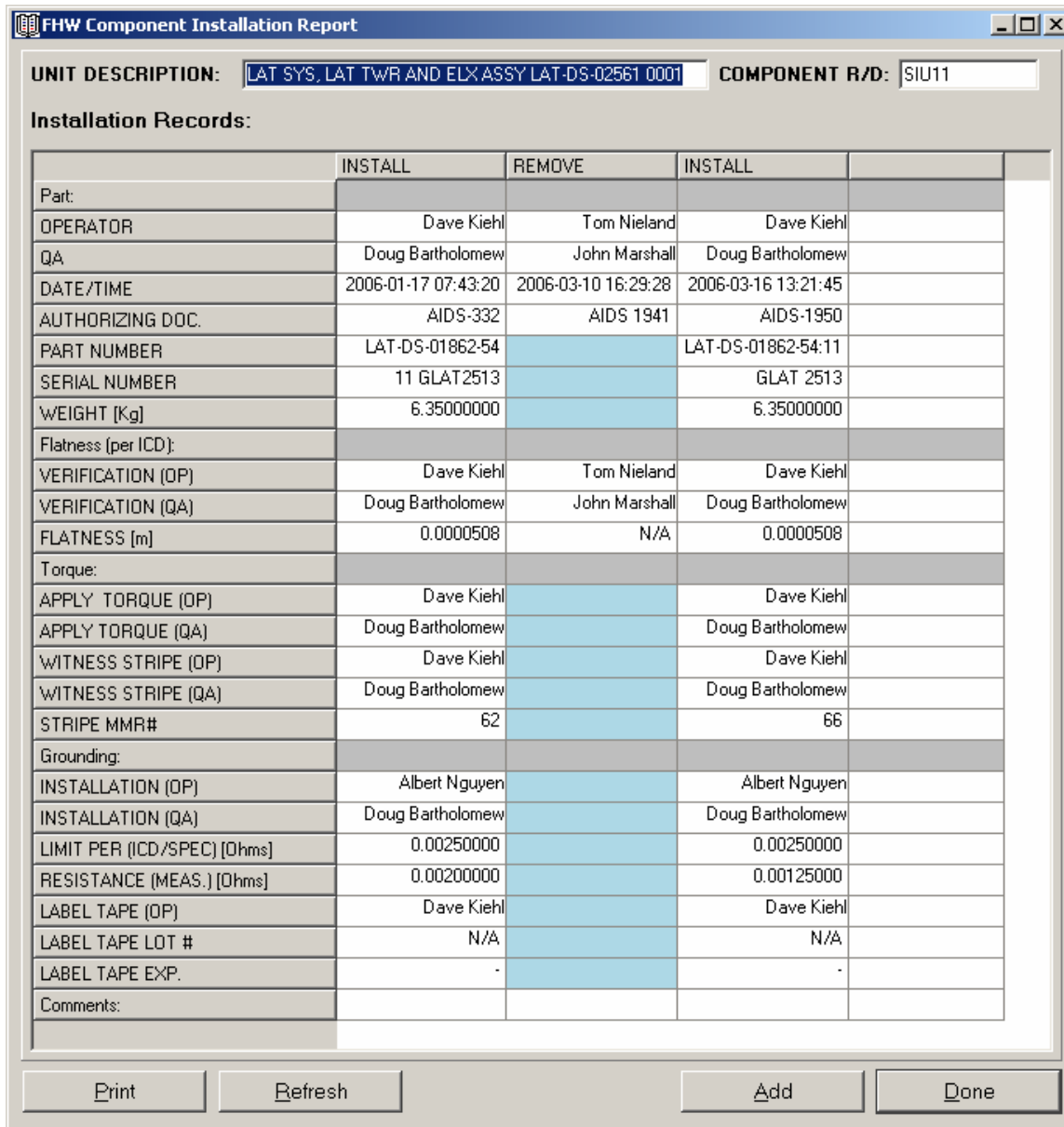


Figure 56: FHW Report GUI

A.5 MMR Log

MMR Log			
Field	Type		
MMR #	<int, autogenerated>		
Date/Time	<timestamp>		
Type	<Flight (F), Test (T)>		
OP	<ID/Password>	<timestamp>	
QA	<ID/Password>	<timestamp>	
Authorizing Document	<text>		
Sequence Number	<text>		
Remarks	<text>		
Comments	<text tuple>		
Materials (if applicable):			
Name/ID	<text>	<text>	<text>
Manufacturer	<text>	<text>	<text>
Lot	<text>	<text>	<text>
Expiration	<timestamp>	<timestamp>	<timestamp>
PBW (if applicable):			
Resin/Hardener Nomenclature	<text>		
Resin PBW	<float>		
Hardener PBW	<float>		
Filler PBW (max)	<float>		
Comments (Temperature, Humidity)	<text>		
Hardener PBW	<Yes (Y), No (N)>		
Weights (if applicable):			
Resin/Hardener (Kg)	<float>		
Filler (Kg)	<float>		
Other (Kg)	<float>		
Total (Kg)	<float>		
Mix Times (if applicable):			
Minimum Time	<timestamp>		
Start Time	<timestamp>		
Total Time	<timestamp>		
Hardness Test Results:			
Date/Time	<timestamp>		
Result	<Pass (P), Fail (F)>		
Minimum Hardness	<float>		
Average Hardness	<float>		
Durometer Type	<text>		
Hardness Test OP	<ID/Password>	<timestamp>	
Hardness Test QA	<ID/Password>	<timestamp>	

Table 27: MMR Log Database Fields

MMR

MMR#: 5

Authorizing Document: AIDS 470 Seq. No.: 50 Remarks: SERIAL #1 HEAT PIPE BOND 12-08-04

Materials | **Hardness Test Results** | MMR Notes

Flight or Test: F Date: 02/09/2005 02:03:27 PM Operator: Tom Nieland QA: Doug Bartholomew

Resin/Hardener: Name/ID: EP30AN-1Z0100 Manufacturer: MASTER BOND Lot: MFG 8-9-2005 Expiration: 02/12/2005 N/A

Filler: Name/ID: 602584 Manufacturer: POTTERS, INDUSTRIES Lot: 2828 Expiration: 01/01/2006 N/A

Other: Name/ID: Manufacturer: Lot: Expiration: 01/01/2005 N/A

PBW: Resin/Hardener Nomen: Resin PBW: 10.000 Hardener PBW: 1.000 Filler PBW (Max): 0.050 PBW Comments (Temp., Humid.): Degass. Req.?: N

Weight: Resin Wgt.: 100.0 Hardener Wgt.: 10.0 Filler Wgt.: 0.55 Other Wgt./Vol.: 0.0 TOTAL Wgt.: 110.55 Weights Units: Gr

Mix: Start Time: 10:45:00 AM Min. Time: 00:48:00 Total Time: 00:48:00

Last Modified: 06/13/2005 11:16:36 AM

Print Done Cancel

Figure 57: MMR Record GUI

MMR Report

MMR #	Date	OP/QA	HT Result	HT Date	HT OP/QA	Resin/Harden.	Filler	Other	In
2	2004-12-08 13:50:02	TN/DNB	P	2004-12-08 14:10:35	/	EP30AN-1Z0100	602584		F
3	2004-12-08 14:14:20	DK/ATK	P	2004-12-08 14:19:03	/	9309			F
4	2004-12-08 14:32:12	DK/DNB	P	2004-12-08 14:34:15	/	TEST			F
5	2005-02-09 14:03:27	TN/DNB	P	2005-06-13 11:16:36	DK/DNB	EP30AN-1Z0100	602584		F
6	2005-02-09 14:35:04	TN/DNB	P	2005-06-13 11:15:54	DK/DNB	EP30AN-1Z0100	602584		F
7	2005-02-25 13:56:09	DK/DNB	P	2005-06-13 11:14:31	DK/DNB	TRA-BOND			F
8	2005-04-06 11:20:45	EQ/DNB	P	2005-04-19 07:42:10	DK/DNB			EA 9394	F
9	2005-05-24 12:16:37	MM/DNB	P	2005-05-27 08:11:33	DK/DNB			BA-2151	F
10	2005-05-27 08:23:26	DK/DNB	P	2005-05-27 08:26:27	DK/DNB			BA-2151	F
11	2005-06-07 16:54:08	TN/	P	2005-06-07 20:48:01	TN/JM	BIPAX BA-2151			F
12	2005-06-08 11:10:55	DK/DNB	P	2005-06-13 11:13:09	DK/DNB			EA 9394	F
13	2005-06-09 22:20:23	LJM/	P	2005-06-13 11:09:58	DK/DNB	BA-2151			F
14	2005-06-21 20:44:44	LJM/JM	F	2005-06-21 21:14:44	LJM/JM			HYSOL 9394 SEM PACK	F
15	2005-07-01 22:06:56	TN/JM	P	2005-07-05 14:34:19	EQ/DNB			BIPAX BA-2151	F
16	2005-07-26 09:49:26	EQ/DNB	P	2005-08-02 07:58:16	DK/DNB			BIPAX BA-2151	F
17	2005-08-03 14:46:06	EQ/	P	2005-08-16 07:58:27	DK/DNB			EA 9394	F

Print Refresh Add Done

Figure 58: MMR Report GUI

A.6 FSW Log

FSW Component Installation Log	
Field	Type
Date/Time	<timestamp>
OP	<ID/Password> <timestamp>
QA	<ID/Password> <timestamp>
FSW Description	<text>
FSW Package	<text>
FSW Version Number	<text>
Authorizing Document	<text>
Installation Type	<New Version, Patch, Removal>
Comments	<text tuple>

Table 28: FSW Log Database Fields

Create FSW Component Installation Record

FSW Record | FSW Installation Notes

Select FSW Component:

FSW Component	Package
PIG_FLYING	PIG
TEST_CMX	CMX

Date: 06/10/2005 02:25:05 PM

Component: PIG_FLYING

Package: PIG

Version: V6-0-3

Action: NEW VERSION

Authorizing Document: _____

Operator: _____

QA: _____

Save Cancel

Figure 59: FSW Record GUI

A.7 EGSE Log

Field	Type
Validation Date/Time	<timestamp>
EGSE Description	<text>
EGSE Assembly Number	<text>
OP	<ID/Password> <timestamp>
QA	<ID/Password> <timestamp>
Setup Type	<SIMPLE, GASU, LAT, WORKSTATION, SERVER, HYBRID>
Global Validation Expiration (earliest of EGSE component expiration dates)	<timestamp, N/A>
Authorizing Document	<text>
Comments	<text tuple>
EGSE Components:	
Date/Time	<timestamp>
OP	<ID/Password> <timestamp>
Remove	<Yes (Y), No (N)>
EGSE Component Description	<text>
EGSE Component R/D, Version or Assy. #	<text>
EGSE Component Type	<HW Connector, FHW, FSW, EGSE Assy., EGSE HW, EGSE SW>
EGSE Component Validation Expiration	<timestamp, N/A>
EGSE Component Validation Document	<text>

Table 29: EGSE Log Database Fields

The screenshot shows a window titled "EGSE Validation Report". At the top, there are two input fields: "EGSE DESCRIPTION: TEST CART, LAT" and "EGSE ASSEMBLY NUMBER: CART 1". Below these is a table with the following data:

EGSE Type	Validation Exp.	Validation Date	Validation Doc	OP	QA	NOTES
HYBRID	2006-02-01	2005-05-02 19:13:57	LAT-PS-03276	Phuc Hoang	John Marshall	YES
HYBRID	2005-06-01	2005-05-12 12:26:39	LAT-PS-03276	Alicia Kavelaars	Alicia Kavelaars	
HYBRID	2005-06-01	2005-05-12 12:53:51	LAT-PS-03276	Alicia Kavelaars	Alicia Kavelaars	
HYBRID	N/A	2005-05-12 12:55:10	LAT-PS-03276	Alicia Kavelaars	Alicia Kavelaars	
HYBRID	2005-05-17	2005-05-12 12:56:19	LAT-PS-03276	Alicia Kavelaars	Alicia Kavelaars	

At the bottom of the window, there are four buttons: "Print", "Refresh", "Add", and "Done".

Figure 60: EGSE Report GUI

A.8 Configuration Log

Configuration Request Log	
Field	Type
OP	<ID/Password>
Date/Time	<timestamp>
Type	<FHW, FSW, EGSE or FHW Matrix>

Table 30: Configuration Request Log Database Fields

Configuration Log	
Field	Type
Type	<FHW, FSW, EGSE>
Component Record Tuple (latest record per FHW, FSW or EGSE component currently installed/configured in the LAT)	<record fields for FHW, FSW or EGSE>

Table 31: Configuration Report Format

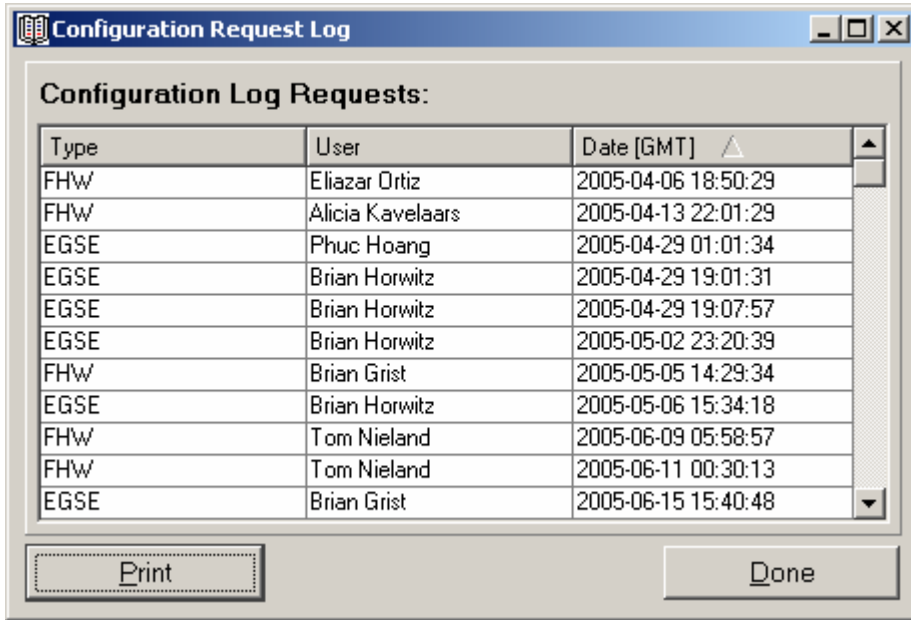


Figure 61: Configuration Request Report GUI

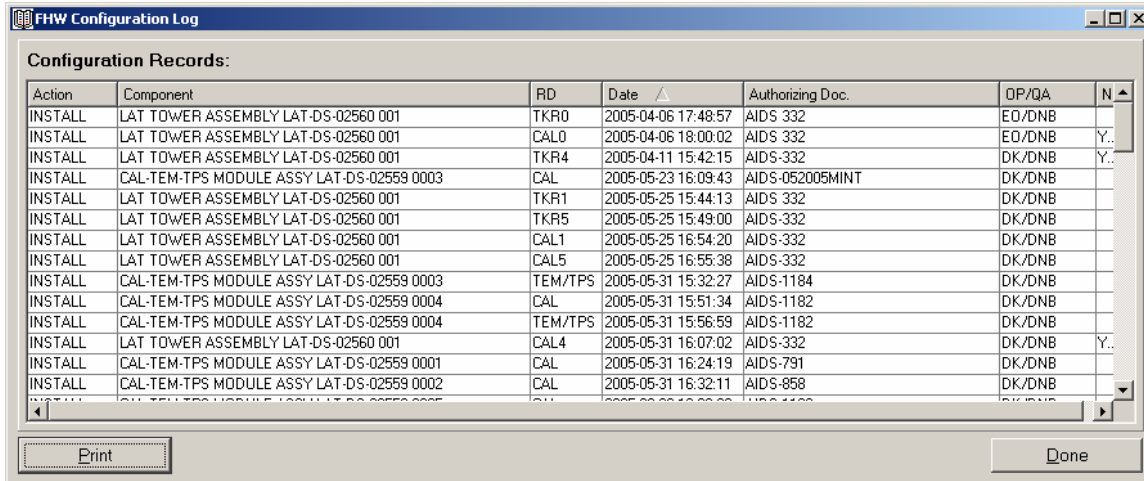


Figure 62: FHW Configuration Report GUI

APPENDIX B: E-Logbook's Database Schema

The following database schema is written in MySQL and creates from scratch a complete and empty table schema of E-Logbook a MySQL server, customized for the LAT I&T group:

```
--
-- Table structure for table `elogparameters`
--

CREATE TABLE elogparameters (
  ParameterID int(11) NOT NULL auto_increment,
  Date date NOT NULL default '0000-00-00',
  ELogVersion varchar(40) NOT NULL default '',
  Tag varchar(32),
  Notes longtext NOT NULL,
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ParameterID),
  KEY ELog_Version (ELogVersion),
  KEY sync (Sync)
) TYPE=InnoDB;

--
-- Dumping data for table `elogparameters`
--

INSERT INTO elogparameters VALUES (1,'2004-04-01','1.0.0','P01-00-00','Version 1.0.0','0');
INSERT INTO elogparameters VALUES (2,'2004-08-02','2.0.0','P02-00-00','Version 2.0.0','0');
INSERT INTO elogparameters VALUES (3,'2004-10-01','3.0.0','P03-00-00','Version 3.0.0','0');
INSERT INTO elogparameters VALUES (4,'2004-12-06','3.1.0','P03-01-00','Version 3.1.0','0');
INSERT INTO elogparameters VALUES (5,'2004-12-07','3.1.1','P03-01-01','Version 3.1.1','0');
INSERT INTO elogparameters VALUES (6,'2005-02-01','3.2.0','P03-02-00','Version 3.2.0','0');
INSERT INTO elogparameters VALUES (7,'2005-02-15','3.2.1','P03-02-01','Version 3.2.1','0');
INSERT INTO elogparameters VALUES (8,'2005-03-29','3.3.0','P03-03-00','Version 3.3.0','0');
INSERT INTO elogparameters VALUES (9,'2005-05-24','3.4.0','P03-04-00','Version 3.4.0','0');
INSERT INTO elogparameters VALUES (10,'2005-06-10','3.4.1','P03-04-01','Version 3.4.1','0');
```

```

INSERT INTO elogparameters VALUES (11,'2005-06-30','3.5.0','P03-05-00','Version 3.5.0','0');
INSERT INTO elogparameters VALUES (12,'2005-08-16','3.6.0','P03-06-00','Version 3.6.0','0');
INSERT INTO elogparameters VALUES (13,'2005-11-15','3.6.1','P03-06-01','Version 3.6.1','0');
INSERT INTO elogparameters VALUES (14,'2005-11-22','3.7.0','P03-07-00','Version 3.7.0','0');
INSERT INTO elogparameters VALUES (15,'2005-12-12','3.8.0','P03-08-00','Version 3.8.0','0');
INSERT INTO elogparameters VALUES (16,'2005-12-22','4.0.0','P04-00-00','Version 4.0.0','0');

```

```

--
-- Table structure for table `eloguser`
--

```

```

CREATE TABLE eloguser (
  UserID int(11) NOT NULL auto_increment,
  Status char(1) NOT NULL default 'A',
  Name varchar(40) NOT NULL default '',
  UserGroup varchar(10) NOT NULL default '',
  Initials char(3) NOT NULL default '',
  UserName varchar(20) NOT NULL default '',
  PWD varchar(32) NOT NULL default '',
  Quality varchar(40) NOT NULL default '',
  PRIMARY KEY (UserID)
) TYPE=InnoDB;

```

```

--
-- Dumping data for table `eloguser`
--

```

```

INSERT INTO `eloguser` VALUES (1, 'I', 'Default Record', 'N/A', 'N/A', '', '', 'INT Physicist');

```

```

--
-- Table structure for table `elogshiftsummary`
--

```

```

CREATE TABLE elogshiftsummary (
  ShiftID int(11) NOT NULL,
  StartTime datetime NOT NULL default '2003-01-01 00:00:00',
  EndTime datetime NOT NULL default '2003-01-01 00:00:00',
  ShiftType varchar(5) NOT NULL default '',
  ShiftNote longtext NOT NULL,
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ShiftID),
  KEY eLogShiftSummary_StartTime (StartTime),
  KEY eLogShiftSummary_EndTime (EndTime),
  KEY sync (Sync)
)

```



```

) TYPE=InnoDB;

--
-- Table structure for table `elogreport`
--

CREATE TABLE elogreport (
  TimeStamp datetime NOT NULL default '0000-00-00 00:00:00',
  RunID int(11) NOT NULL default '0',
  TestName varchar(100) NOT NULL default '',
  Operator varchar(100) NOT NULL default '',
  OperatorID int(11) NOT NULL default '0',
  EventCount int(11) NOT NULL default '0',
  BadEventCount int(11) NOT NULL default '0',
  PauseCount int(11) NOT NULL default '0',
  StartTime datetime NOT NULL default '0000-00-00 00:00:00',
  ElapsedTime decimal(10,0) NOT NULL default '0',
  EndTime datetime NOT NULL default '0000-00-00 00:00:00',
  SchemaConfigFile varchar(100) NOT NULL default '',
  AdditionalInputFiles longtext NOT NULL,
  Release longtext,
  ModulesFailedVerification longtext NOT NULL,
  VersionData longtext NOT NULL,
  CompletionStatus int(11) NOT NULL default '0',
  ArchiveFile varchar(100) NOT NULL default '',
  ErrorArchive varchar(100) NOT NULL default '',
  LogFile varchar(100) NOT NULL default '',
  FitsFile varchar(100) NOT NULL default '',
  Site varchar(100) NOT NULL default '',
  ParticleType varchar(100) NOT NULL default '',
  InstrumentType varchar(100) NOT NULL default '',
  Orientation varchar(100) NOT NULL default '',
  Phase varchar(100) NOT NULL default '',
  Comments longtext NOT NULL,
  AdditionFields longtext NOT NULL,
  ErrorEventCount int(11) NOT NULL default '0',
  PRIMARY KEY (RunID),
  KEY eLogReport_InstrumentType (InstrumentType),
  KEY eLogReport_Orientation (Orientation),
  KEY eLogReport_FitsFile (FitsFile),
  KEY eLogReport_Site (Site),
  KEY eLogReport_ParticleType (ParticleType),
  KEY eLogReport_StartTime (StartTime)
) TYPE=InnoDB;

--
-- Table structure for LICOS_states
-- Enumerates the possible processing states of the LICOS system
--

CREATE TABLE LICOS_states (
  State_PK INTEGER NOT NULL auto_increment,

```

```

StateName VARCHAR(64),
DescShort BLOB,
CONSTRAINT licos_states_PK PRIMARY KEY(state_PK)
)TYPE=InnoDB;

--
-- Dumping data for table `LICOS_states`
--

INSERT INTO LICOS_states (stateName, descShort) VALUES
( 'ActivityCreated', 'Activity is created but not started' ),
( 'CommandingStarted', 'Activity has started commanding but
commanding is not complete' ),
( 'CommandingComplete', 'Activity has completed command uploads but
data has not appeared' ),
( 'NoData', 'No data received' ),
( 'DataStarted', 'Activity data has started appearing, but run is
not complete' ),
( 'DataComplete', 'Activity data has completed, run is ready for
analysis' ),
( 'NoAnalysis', 'Analysis not started yet' ),
( 'AnalysisStarted', 'Run has started analysis step, but run is not
complete' ),
( 'AnalysisComplete', 'Run has completed Analysis' )
;

--
-- Table structure for LICOS_activities
-- Allows linkage between different LICOS runs and asynchronous
updates
--

CREATE TABLE LICOS_activities (
RunID_FK          INTEGER NOT NULL,
LdfFileName       TEXT,
ScriptOutput      BLOB,
DataStart datetime NOT NULL default '0000-00-00 00:00:00',
DataStop datetime NOT NULL default '0000-00-00 00:00:00',
JobCompleteList  BLOB,
SuiteIdentifier   Integer,
INDEX (SuiteIdentifier),
CommandingState_FK  INTEGER NOT NULL,
DataState_FK       INTEGER NOT NULL,
AnalysisState_FK   INTEGER NOT NULL,
CONSTRAINT licos_activities_PK PRIMARY KEY(RunID_FK),
CONSTRAINT commanding_fk FOREIGN KEY(CommandingState_FK)
REFERENCES LICOS_states(State_PK),
CONSTRAINT data_fk FOREIGN KEY(DataState_FK)
REFERENCES LICOS_states(State_PK),
CONSTRAINT analysis_fk FOREIGN KEY(AnalysisState_FK)
REFERENCES LICOS_states(State_PK),
CONSTRAINT runidentifier_fk FOREIGN KEY(RunID_FK)

```

```

        REFERENCES elogreport(RunID)
    ) TYPE=InnoDB;

--
-- Table structure for table `elogshiftactivity`
--

CREATE TABLE elogshiftactivity (
  ActivityID int(11) NOT NULL auto_increment,
  ShiftID int(11) default NULL,
  Time datetime NOT NULL default '2003-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  ActivityNote longtext NOT NULL,
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ActivityID),
  KEY ShiftID (ShiftID),
  KEY eLogShiftActivity_Time (Time),
  KEY eLogShiftActivity_UserID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1617` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_1618` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

--
-- Table structure for table `elogshiftother`
--

CREATE TABLE elogshiftother (
  ShiftOtherID int(11) NOT NULL auto_increment,
  ShiftID int(11) default NULL,
  Time datetime NOT NULL default '2003-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  OtherNote longtext NOT NULL,
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ShiftOtherID),
  KEY ShiftID (ShiftID),
  KEY eLogShiftOther_Time (Time),
  KEY eLogShiftOther_UserID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1623` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_1624` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

--
-- Table structure for table `elogshiftproblem`
--

```

```

CREATE TABLE elogshiftproblem (
  ShiftProblemID int(11) NOT NULL auto_increment,
  ShiftID int(11) default NULL,
  Time datetime NOT NULL default '2003-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  ProblemNote longtext NOT NULL,
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ShiftProblemID),
  KEY ShiftID (ShiftID),
  KEY eLogShiftProblem_Time (Time),
  KEY eLogShiftProblem_UserID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1620` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_1621` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogsubsysactivity`
--

```

```

CREATE TABLE elogsubsysactivity (
  SubSysActivityID int(11) NOT NULL auto_increment,
  ShiftID int(11) NOT NULL,
  Time datetime NOT NULL default '2003-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  ActivityNote longtext NOT NULL,
  SubSys varchar(10) NOT NULL default '',
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (SubSysActivityID),
  KEY eLogSubSysActivity_Time (Time),
  KEY ShiftID (ShiftID),
  KEY User_ID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1626` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_1627` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogsubsysother`
--

```

```

CREATE TABLE elogsubsysother (
  SubSysOtherID int(11) NOT NULL auto_increment,
  ShiftID int(11) NOT NULL default '0',
  Time datetime NOT NULL default '2003-01-01 00:00:00',

```

```

UserID int(11) NOT NULL default '1',
OtherNote longtext NOT NULL,
SubSys varchar(10) NOT NULL default '',
LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
Sync tinyint(1) NOT NULL default '0',
PRIMARY KEY (SubSysOtherID),
KEY eLogSubSysOther_Time (Time),
KEY ShiftID (ShiftID),
KEY User_ID (UserID),
KEY sync (Sync),
CONSTRAINT `0_1632` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
CONSTRAINT `0_1633` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogsubsysproblem`
--

```

```

CREATE TABLE elogsubsysproblem (
  SubSysProblemID int(11) NOT NULL auto_increment,
  ShiftID int(11) NOT NULL default '0',
  Time datetime NOT NULL default '2003-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  ProblemNote longtext NOT NULL,
  SubSys varchar(10) NOT NULL default '',
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (SubSysProblemID),
  KEY eLogSubSysProblem_Time (Time),
  KEY ShiftID (ShiftID),
  KEY User_ID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1629` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_1630` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogshiftschedule`
--

```

```

CREATE TABLE elogshiftschedule (
  ShiftScheduleID int(11) NOT NULL auto_increment,
  ShiftID int(11) NOT NULL default '0',
  SubSys varchar(20) NOT NULL default '',
  PRIMARY KEY (ShiftScheduleID)
) TYPE=InnoDB;

```

```

--

```

```

-- Table structure for table `elogshifttaker`
--

CREATE TABLE elogshifttaker (
  ShiftTakerID int(11) NOT NULL auto_increment,
  ShiftID int(11) NOT NULL default '0',
  SubSys varchar(20) NOT NULL default '',
  UserID int(11) NOT NULL default '0',
  LastUpdate datetime NOT NULL default '2003-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ShiftTakerID),
  KEY eLogShiftTaker_ShiftID (ShiftID),
  KEY eLogShiftTaker_UserID (UserID),
  KEY ShiftID (ShiftID),
  KEY UserID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_49` FOREIGN KEY (`ShiftID`) REFERENCES
`elogshiftsummary` (`ShiftID`),
  CONSTRAINT `0_50` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

--
-- Operator Info table to fill in pointers
--
-- Table structure for table `elogoperatorinfo`
--

CREATE TABLE elogoperatorinfo (
  InstallOpID int(11) NOT NULL auto_increment,
  Operation varchar(40) default NULL,
  OperatorID int(11) NOT NULL default '1',
  OperatorIDts datetime NOT NULL default '2004-01-01 00:00:00',
  OperatorQA int(11) NOT NULL default '1',
  OperatorQAts datetime NOT NULL default '2004-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (InstallOpID),
  KEY Operator_ID (OperatorID),
  KEY Operator_QA (OperatorQA),
  KEY sync (Sync),
  CONSTRAINT `0_1592` FOREIGN KEY (`OperatorID`) REFERENCES `eloguser`
(`UserID`),
  CONSTRAINT `0_1593` FOREIGN KEY (`OperatorQA`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

--
-- Dumping data for table `elogoperatorinfo`
--

INSERT INTO elogoperatorinfo VALUES (0,'Default',1,'2004-01-01
00:00:00',1,'2004-01-01 00:00:00', '0');

```

```

--
-- Mate/Demate tables
--
-- Table structure for table `elogconnector`
--

CREATE TABLE elogconnector (
  ConnectorID int(11) NOT NULL auto_increment,
  ConnectorRD varchar(20) NOT NULL default '',
  Description varchar(100) NOT NULL default '',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ConnectorID),
  KEY sync (Sync)
) TYPE=InnoDB;

--
-- Dumping data for table `elogconnector`
--

INSERT INTO elogconnector VALUES (0,'','Default', '0');

--
-- Table structure for table `elogmatedemate`
--

CREATE TABLE elogmatedemate (
  MateDeMateID int(11) NOT NULL auto_increment,
  Connector1ID int(11) NOT NULL default '1',
  Connector2ID int(11) NOT NULL default '1',
  AuthorizedByDoc varchar(100) NOT NULL default '',
  Date datetime NOT NULL default '2004-01-01 00:00:00',
  Action char(1) NOT NULL default '',
  Instrument char(1) NOT NULL default '',
  VerifyPowerOffPtr int(11) NOT NULL default '1',
  PreMateInspectPtr int(11) NOT NULL default '1',
  EsdBleedMatePtr int(11) NOT NULL default '1',
  TorquePtr int(11) NOT NULL default '1',
  Torque int(11),
  FinalInspectPtr int(11) NOT NULL default '1',
  LastUpdate datetime NOT NULL default '2004-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (MateDeMateID),
  KEY Connector1_ID (Connector1ID),
  KEY Connector2_ID (Connector2ID),
  KEY VerifyPowerOff_Ptr (VerifyPowerOffPtr),
  KEY PreMateInspect_Ptr (PreMateInspectPtr),
  KEY EsdBleedMate_Ptr (EsdBleedMatePtr),
  KEY Torque_Val (Torque),
  KEY TorqueOp_Ptr (TorquePtr),
  KEY FinalInspect_Ptr (FinalInspectPtr),
  KEY DateValue (Date),

```

```

    KEY LastUpdateValue (LastUpdate),
    KEY sync (Sync),
    CONSTRAINT `0_1610` FOREIGN KEY (`Connector1ID`) REFERENCES
`elogconnector` (`ConnectorID`),
    CONSTRAINT `0_1611` FOREIGN KEY (`Connector2ID`) REFERENCES
`elogconnector` (`ConnectorID`),
    CONSTRAINT `0_1612` FOREIGN KEY (`VerifyPowerOffPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
    CONSTRAINT `0_1613` FOREIGN KEY (`PreMateInspectPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
    CONSTRAINT `0_1614` FOREIGN KEY (`EsdBleedMatePtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
    CONSTRAINT `0_1615` FOREIGN KEY (`FinalInspectPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
    CONSTRAINT `0_1616` FOREIGN KEY (`TorquePtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

```

```

--
-- MMR tables
--
-- Table structure for table `elogmaterial`
--

```

```

CREATE TABLE elogmaterial (
  MaterialID int(11) NOT NULL auto_increment,
  MaterialName varchar(40) NOT NULL default '',
  Manufacturer varchar(40) NOT NULL default '',
  Lot varchar(40) NOT NULL default '',
  Exp date NOT NULL default '2004-01-01',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (MaterialID),
  KEY sync (Sync)
) TYPE=InnoDB;

```

```

--
-- Dumping data for table `elogmaterial`
--

```

```

INSERT INTO elogmaterial VALUES (0,'','Default','','2004-01-01','0');

```

```

--
-- Table structure for table `elohardtest`
--

```

```

CREATE TABLE elohardtest (
  HardTestID int(11) NOT NULL auto_increment,
  TestResult char(1) NOT NULL default 'P',
  Date datetime NOT NULL default '2004-01-01 00:00:00',
  MinHardness decimal(8,3) NOT NULL default '',
  AvgHardness decimal(8,3) NOT NULL default '',

```



```

Durometer varchar(40) NOT NULL default '',
HardTestOperatorPtr int(11) NOT NULL default '1',
Sync tinyint(1) NOT NULL default '0',
PRIMARY KEY (HardTestID),
KEY HardTestOperator_Ptr (HardTestOperatorPtr),
KEY DateValue (Date),
KEY sync (Sync),
CONSTRAINT `0_1740` FOREIGN KEY (`HardTestOperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

```

```

INSERT INTO eloghardtest VALUES (0,'P','2004-11-01 00:00:00', 0.0, 0.0,
'', '1','0');

```

```

--
-- Table structure for table `elogmmr`
--

```

```

CREATE TABLE elogmmr (
  MMRID int(11) NOT NULL auto_increment,
  Instrument char(1) NOT NULL default '',
  Date datetime NOT NULL default '2004-11-01 00:00:00',
  Doc varchar(40) NOT NULL,
  SeqNo varchar(10) NOT NULL,
  Remarks varchar(100) NOT NULL,
  OperatorPtr int(11) NOT NULL default 1,
  MatResinPtr int(11) NOT NULL default 1,
  MatFillerPtr int(11) NOT NULL default 1,
  MatOtherPtr int(11) NOT NULL default 1,
  PBWNomen varchar(40) NOT NULL default '',
  ResinPBW decimal(8,3) default NULL,
  HardPBW decimal(8,3) default NULL,
  FillerPBW decimal(8,3) default NULL,
  CommentsPBW varchar(100) NOT NULL default '',
  DegassReq char(1) NOT NULL default '',
  ResinWgt decimal(15,9) default NULL,
  HardWgt decimal(15,9) default NULL,
  FillerWgt decimal(15,9) default NULL,
  OtherWgt decimal(15,9) default NULL,
  TotalWgt decimal(15,9) default NULL,
  MixStartTime time NOT NULL default '00:00:00',
  MixMinTime time NOT NULL default '00:00:00',
  MixTotTime time NOT NULL default '00:00:00',
  HardTestPtr int(11) NOT NULL default 1,
  LastUpdate datetime NOT NULL default '2004-11-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (MMRID),
  KEY Resin_Ptr (MatResinPtr),
  KEY Filler_Ptr (MatFillerPtr),
  KEY Other_Ptr (MatOtherPtr),
  KEY Operator_Ptr (OperatorPtr),
  KEY HardTest_Ptr (HardTestPtr),

```

```

KEY DateValue (Date),
KEY LastUpdateValue (LastUpdate),
KEY sync (Sync),
CONSTRAINT `0_1731` FOREIGN KEY (`MatResinPtr`) REFERENCES
`elogmaterial` (`MaterialID`),
CONSTRAINT `0_1732` FOREIGN KEY (`MatFillerPtr`) REFERENCES
`elogmaterial` (`MaterialID`),
CONSTRAINT `0_1733` FOREIGN KEY (`MatOtherPtr`) REFERENCES
`elogmaterial` (`MaterialID`),
CONSTRAINT `0_1734` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
CONSTRAINT `0_1735` FOREIGN KEY (`HardTestPtr`) REFERENCES
`eloghardtest` (`HardTestID`)
) TYPE=InnoDB;

```

```

--
-- Dumping data for table `elogmmr`
--

```

```

INSERT INTO elogmmr VALUES (0, '', '2004-11-01 00:00:00',
'', '', '', '1', '1', '1', '1', '', 0.0, 0.0, 0.0,
'', '', 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
'00:00:00', '00:00:00', '00:00:00', 1, '2004-11-01 00:00:00', '0');

```

```

--
-- Install Record tables
--
-- Table structure for table `elogcomponent`
--

```

```

CREATE TABLE elogcomponent (
  ComponentID int(11) NOT NULL auto_increment,
  ComponentRD varchar(20) NOT NULL default '',
  Description varchar(100) NOT NULL default '',
  Type varchar(10),
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ComponentID),
  KEY Type (Type),
  KEY sync (Sync)
) TYPE=InnoDB;

```

```

--
-- Dumping data for table `elogcomponent`
--

```

```

INSERT INTO elogcomponent VALUES (0, '', 'Default', 'ALL', '0');

```

```

--
-- Table structure for table `eloginstallrecord`
--

```

```

CREATE TABLE eloginstallrecord (
  InstallRecordID int(11) NOT NULL auto_increment,

```

```

ComponentID int(11) NOT NULL default '1',
Action varchar(8) NOT NULL default '',
OperatorPtr int(11) NOT NULL default '1',
Part varchar(20) default NULL,
SN varchar(20) default NULL,
Date datetime NOT NULL default '2004-01-01 00:00:00',
Doc varchar(40) default NULL,
Wgt decimal(12,6) default NULL,
FlatVerPtr int(11) NOT NULL default '1',
Flatness decimal(12,7) default NULL,
CheckMMR tinyint(1) NOT NULL default '0',
Torque longtext,
CheckTorque tinyint(1) NOT NULL default '0',
AppTorquePtr int(11) NOT NULL default '1',
TorqueMMR int(11) NOT NULL default '1',
WitnessStripePtr int(11) NOT NULL default '1',
LimitPer decimal(12,8) default NULL,
Resistance decimal(12,8) default NULL,
GrnTool varchar(40),
GrnToolExp date NOT NULL default '2004-01-01',
CheckGrn tinyint(1) NOT NULL default '0',
InstallationPtr int(11) NOT NULL default '1',
LabelTape varchar(11) default NULL,
LabelTapeExp date NOT NULL default '0000-00-00',
LabelTapePtr int(11) NOT NULL default '1',
LastUpdate datetime NOT NULL default '2004-01-01 00:00:00',
Sync tinyint(1) NOT NULL default '0',
PRIMARY KEY (InstallRecordID),
KEY Component_ID (ComponentID),
KEY Operator_Ptr (OperatorPtr),
KEY FlatVer_Ptr (FlatVerPtr),
KEY AppTorque_Ptr (AppTorquePtr),
KEY WitnessStripe_Ptr (WitnessStripePtr),
KEY Installation_Ptr (InstallationPtr),
KEY LabelTape_Ptr (LabelTapePtr),
KEY DateValue (Date),
KEY LastUpdateValue (LastUpdate),
KEY Torque_MMR (TorqueMMR),
KEY sync (Sync),
CONSTRAINT `0_1602` FOREIGN KEY (`ComponentID`) REFERENCES
`elogcomponent` (`ComponentID`),
CONSTRAINT `0_1603` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
CONSTRAINT `0_1604` FOREIGN KEY (`FlatVerPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
CONSTRAINT `0_1605` FOREIGN KEY (`AppTorquePtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
CONSTRAINT `0_1606` FOREIGN KEY (`WitnessStripePtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
CONSTRAINT `0_1607` FOREIGN KEY (`InstallationPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),

```

```

    CONSTRAINT `0_1608` FOREIGN KEY (`LabelTapePtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`),
    CONSTRAINT `0_1609` FOREIGN KEY (`TorqueMMR`) REFERENCES `elogmmr`
(`MMRID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogtorque`
--

```

```

CREATE TABLE elogtorque (
  TorqueRecordID int(11) NOT NULL auto_increment,
  Type varchar(40),
  Torque decimal(12,6),
  Tool varchar(80),
  ToolExp date NOT NULL default '2004-11-01',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (TorqueRecordID),
  KEY Tool (Tool),
  KEY sync (Sync)
) TYPE=InnoDB;

```

```

--
-- Comment table
--
-- Table structure for table `elogcomments`
--

```

```

CREATE TABLE elogcomments (
  CommentID int(11) NOT NULL auto_increment,
  ParentTable varchar(4) NOT NULL,
  ParentRecordID int(11) NOT NULL,
  Time datetime NOT NULL default '2004-01-01 00:00:00',
  UserID int(11) NOT NULL default '1',
  Note longtext NOT NULL,
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (CommentID),
  KEY Parent_Table (ParentTable),
  KEY Parent_RecordID (ParentRecordID),
  KEY TimeValue (Time),
  KEY User_ID (UserID),
  KEY sync (Sync),
  CONSTRAINT `0_1700` FOREIGN KEY (`UserID`) REFERENCES `eloguser`
(`UserID`)
) TYPE=InnoDB;

```

```

--
-- EGSE tables
--
-- Table structure for table `elogegsevalidation`
--

```

```

CREATE TABLE elogegsevalidation (
  EGSEValidRecordID int(11) NOT NULL auto_increment,
  EGSEID int(11) NOT NULL default '1',
  Type varchar(80),
  OperatorPtr int(11) NOT NULL default 1,
  Date datetime NOT NULL default '2004-11-01 00:00:00',
  ValidDate date NOT NULL default '2004-11-01',
  Doc varchar(40) NOT NULL,
  RunID varchar(100),
  ComponentList longtext,
  LastUpdate datetime NOT NULL default '2004-11-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (EGSEValidRecordID),
  KEY EGSE_ID (EGSEID),
  KEY Run_ID (RunID),
  KEY DateValue (Date),
  KEY ValidDateValue (ValidDate),
  KEY Operator_Ptr (OperatorPtr),
  KEY LastUpdateValue (LastUpdate),
  KEY sync (Sync),
  CONSTRAINT `0_1750` FOREIGN KEY (`EGSEID`) REFERENCES
`elogcomponent` (`ComponentID`),
  CONSTRAINT `0_1751` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

```

```

--
-- Table structure for table `elogegsevalidcomponent`
--

```

```

CREATE TABLE elogegsevalidcomponent (
  EGSEValidComponentID int(11) NOT NULL auto_increment,
  Type varchar(80),
  CompRecordID int(11) NOT NULL,
  Date datetime NOT NULL default '2005-01-01 00:00:00',
  OperatorPtr int(11) NOT NULL default 1,
  ValidDate date NOT NULL default '2005-01-01',
  Doc varchar(40) NOT NULL,
  LastUpdate datetime NOT NULL default '2005-01-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (EGSEValidComponentID),
  KEY CompRecord_ID (CompRecordID),
  KEY DateValue (Date),
  KEY Operator_Ptr (OperatorPtr),
  KEY ValidDateValue (ValidDate),
  KEY LastUpdateValue (LastUpdate),
  KEY sync (Sync),
  CONSTRAINT `0_1760` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

```

```

--
-- FSW tables
--
-- Table structure for table `elogfswinstallation`
--

CREATE TABLE elogfswinstallation (
  FSWInstallRecordID int(11) NOT NULL auto_increment,
  FSWID int(11) NOT NULL default '1',
  Action varchar(4),
  Date datetime NOT NULL default '2004-11-01 00:00:00',
  Doc varchar(40) NOT NULL,
  OperatorPtr int(11) NOT NULL default 1,
  LastUpdate datetime NOT NULL default '2004-11-01 00:00:00',
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY(FSWInstallRecordID),
  KEY FSW_ID (FSWID),
  KEY DateValue (Date),
  KEY Operator_Ptr (OperatorPtr),
  KEY LastUpdateValue (LastUpdate),
  KEY sync (Sync),
  CONSTRAINT `0_1770` FOREIGN KEY (`FSWID`) REFERENCES `elogcomponent`
(`ComponentID`),
  CONSTRAINT `0_1771` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

--
-- Table structure for table `elogconfigrequest`
--

CREATE TABLE elogconfigrequest (
  ConfigRequestID int(11) NOT NULL auto_increment,
  OperatorPtr int(11) NOT NULL default '1',
  Type varchar(12),
  Sync tinyint(1) NOT NULL default '0',
  PRIMARY KEY (ConfigRequestID),
  KEY Operator_Ptr (OperatorPtr),
  KEY Type (Type),
  KEY sync (Sync),
  CONSTRAINT `0_1780` FOREIGN KEY (`OperatorPtr`) REFERENCES
`elogoperatorinfo` (`InstallOpID`)
) TYPE=InnoDB;

--
-- Table structure for table `elogquestionnaire`
--

CREATE TABLE elogquestionnaire (
  QuestID int(11) NOT NULL auto_increment,
  OperatorPtr int(11) NOT NULL default '1',
  PaperResults longtext,

```

```
ElogResults longtext,  
Sync tinyint(1) NOT NULL default '0',  
PRIMARY KEY (QuestID),  
KEY Operator_Ptr (OperatorPtr),  
KEY sync (Sync),  
CONSTRAINT `0_1790` FOREIGN KEY (`OperatorPtr`) REFERENCES  
`elogoperatorinfo` (`InstallOpID`)  
) TYPE=InnoDB;
```


APPENDIX C: JIRA Issues




Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.0.0	ELB-21	Fixed	User Table does not Update in the mirror database		1/19/05	Selim Tuvi
3.1.0	ELB-10	Fixed	runs do not show up on web pages		11/24/04	Alicia Kavelaars
3.1.0	ELB-9	Fixed	Elogbook integration is not working		11/19/04	Alicia Kavelaars
3.1.0	ELB-7	Won't Fix	Implementation of a dynamic allocation of shift intervals.		10/27/04	Alicia Kavelaars
3.1.0	ELB-4	Fixed	User passwords should be stored in encrypted form in the database		10/1/04	Alicia Kavelaars
3.1.0	ELB-3	Fixed	Failure accessing E_logbook		9/30/04	Selim Tuvi
3.1.0	ELB-2	Fixed	Duplicate user entries with different job titles causes problems		9/21/04	Alicia Kavelaars
3.1.0	ELB-1	Fixed	Adding multiple shifters to a category overwrite the previous entry		9/21/04	Alicia Kavelaars
3.1.1	ELB-17	Fixed	'elogreport' table 'release' column size needs to be changed to longtext		12/8/04	Alicia Kavelaars
3.2.0	ELB-19	Fixed	Simplify the location of a single run in the run log at the bottom of the Edit Shift Window		1/10/05	Alicia Kavelaars
3.2.0	ELB-16	Fixed	Run report does not list the schema that failed verification		12/8/04	Alicia Kavelaars
3.2.0	ELB-14	Fixed	Completion status should be shown as a string value		12/6/04	Alicia Kavelaars
3.2.0	ELB-13	Fixed	Module list in run report detail should support sorting		12/6/04	Alicia Kavelaars
3.2.1	ELB-31	Fixed	Reimplement comment table as view to be able to sort		2/16/05	Alicia Kavelaars
3.2.1	ELB-27	Fixed	MMR does not save material and test result information		2/9/05	Alicia Kavelaars
3.2.1	ELB-26	Fixed	.py files generated from .ui files not version controlled		2/7/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.2.1	ELB-24	Fixed	Remove FITS File and rename Archive File		2/2/05	Alicia Kavelaars
3.2.1	ELB-22	Fixed	Runs appear in reverse order on the Web e-logbook		1/20/05	Xin Chen
3.2.1	ELB-15	Fixed	Only the first two shift takers are showing up		12/6/04	Xin Chen
3.2.1	ELB-6	Fixed	Shift log display improvements		10/20/04	Xin Chen
3.3.0	ELB-48	Fixed	New activities should not be added to old shifts		4/4/05	Alicia Kavelaars
3.3.0	ELB-46	Fixed	Prevent selection of invalid date to enter the shift log		3/29/05	Alicia Kavelaars
3.3.0	ELB-44	Won't Fix	Display times in local time, too		3/26/05	Charlotte Hee
3.3.0	ELB-42	Fixed	Shift summary does not appear in the online version of logbook		3/24/05	Alicia Kavelaars
3.3.0	ELB-40	Fixed	How to tell the ELogbook version		3/22/05	Alicia Kavelaars
3.3.0	ELB-39	Fixed	When the runs are refreshed the run date format reverts back to GMT time		3/22/05	Alicia Kavelaars
3.3.0	ELB-38	Fixed	Change Password Length on entry from 8 to 32 characters		3/18/05	Alicia Kavelaars
3.3.0	ELB-36	Fixed	In runReport dialog listRuns, listRuns2, listRuns3 filter is blank by default	 	2/22/05	Alicia Kavelaars
3.3.0	ELB-34	Fixed	Unrecognized initials appear in the shift log		2/18/05	Alicia Kavelaars
3.3.0	ELB-33	Fixed	Online Report column mislabeled		2/18/05	Xin Chen
3.3.0	ELB-32	Fixed	Web Logbook response is extremely slow		2/18/05	Xin Chen
3.3.0	ELB-30	Fixed	To be able To put N/A for expiration date of mix materials that do not expire		2/15/05	Alicia Kavelaars
3.3.0	ELB-28	Fixed	When mirroring E-Logbook should update the shiftsummary note		2/10/05	Selim Tuvi
3.3.0	ELB-25	Won't Fix	Dialog boxes do not hold enough text.		2/3/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.3.0	ELB-20	Fixed	Can't tell if it is a bug, Users can log in and put data in the wrong shift		1/13/05	Alicia Kavelaars
3.3.0	ELB-18	Fixed	changes to instrument type		1/6/05	Selim Tuvi
3.3.0	ELB-8	Fixed	Shift handover procedure		10/27/04	Alicia Kavelaars
3.4.0	ELB-68	Fixed	Provide clipboard copy and tooltip capability for the listviews in the Run Report window		5/12/05	Alicia Kavelaars
3.4.0	ELB-66	Fixed	Click enter after entering password exists user validation window		5/11/05	Alicia Kavelaars
3.4.0	ELB-65	Fixed	Web links in E-logbook		5/9/05	Alicia Kavelaars
3.4.0	ELB-64	Fixed	Add link to shift E-log page in the list runs output		5/7/05	Alicia Kavelaars
3.4.0	ELB-63	Fixed	elog egse log data does not appear without re-start of application		4/29/05	Alicia Kavelaars
3.4.0	ELB-61	Cannot Reproduce	Entry lost if shift synchronization GUI pops		4/28/05	Alicia Kavelaars
3.4.0	ELB-60	Fixed	Windows can be minimized and maximized. Reports are independent.		4/19/05	Alicia Kavelaars
3.4.0	ELB-59	Fixed	Print Statements for Configuration Logs and Activity Reports		4/19/05	Alicia Kavelaars
3.4.0	ELB-58	Fixed	New does not open Report for every component. False components are not displayed		4/19/05	Alicia Kavelaars
3.4.0	ELB-57	Fixed	FHW Install Log needs new Torque design and added N/A feature		4/19/05	Alicia Kavelaars
3.4.0	ELB-56	Fixed	Mate/Demate Log needs Torque fields		4/19/05	Alicia Kavelaars
3.4.0	ELB-55	Fixed	Create a Global Add Activity in Shift Log with selectable subsystem and type		4/19/05	Alicia Kavelaars
3.4.0	ELB-54	Fixed	F5 Refreshes Shift Log and Component Reports		4/19/05	Alicia Kavelaars
3.4.0	ELB-53	Fixed	Preferences Menu		4/19/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.4.0	ELB-52	Fixed	'online' link in suiteSummary has the wrong link	↓	4/18/05	Xin Chen
3.4.0	ELB-51	Fixed	Test Output column is mislabeled when a test report does not exist	↓	4/11/05	Xin Chen
3.4.0	ELB-50	Fixed	F5 refreshes shift in the Shift Log window	↑	4/8/05	Alicia Kavelaars
3.4.0	ELB-43	Fixed	Get rid of the activity, problem, other tabs	↓	3/26/05	Alicia Kavelaars
3.4.0	ELB-41	Fixed	Create a Preferences Menu where Shift Log Subsystem can be saved	↑	3/24/05	Alicia Kavelaars
3.4.0	ELB-37	Fixed	New types of error events	↑	3/17/05	Xin Chen
3.4.0	ELB-23	Fixed	FITS file column is not being populated in web e-log	↑	1/20/05	Xin Chen
3.4.0	ELB-5	Fixed	Detect and provide hyperlinks to URLs in shift logs	↓	10/15/04	Alicia Kavelaars
3.4.1	ELB-74	Fixed	Increase size of Wgt fields to avoid error	↑	6/2/05	Alicia Kavelaars
3.4.1	ELB-73	Fixed	Add N/A to Torque and Grounding stages on FHW Installation Log	↑	6/1/05	Alicia Kavelaars
3.4.1	ELB-72	Fixed	When you paste info into the Summary Area it does not persist.	↑	6/1/05	Alicia Kavelaars
3.4.1	ELB-69	Fixed	the bay position should be searchable	↑	5/23/05	Xin Chen
3.5.0	ELB-80	Fixed	E-Logbook Questionnaire	↑	6/27/05	Alicia Kavelaars
3.5.0	ELB-79	Fixed	Reports cannot be closed while creating a record.	↑	6/27/05	Alicia Kavelaars
3.5.0	ELB-78	Fixed	Minor Bugs on Context menu and MMR Print procedures	↓	6/23/05	Alicia Kavelaars
3.5.0	ELB-77	Fixed	Edit Capability with QA approval for Components in Main Window	↑	6/16/05	Alicia Kavelaars
3.5.0	ELB-76	Fixed	Mate action is not passed when cancelling an entry	↓	6/8/05	Alicia Kavelaars
3.5.0	ELB-75	Fixed	Clear Torque Information on First Demate (If Applicable)	↑	6/7/05	Alicia Kavelaars
3.5.0	ELB-71	Fixed	LAT component installation summary report	↑	5/27/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.5.0	ELB-70	Fixed	Serial No. column shows "FM102???"	↑	5/24/05	Xin Chen
3.5.0	ELB-67	Fixed	Ability to have more than one Run Report detail window	↑	5/12/05	Alicia Kavelaars
3.5.0	ELB-62	Fixed	Search facility	↑	4/29/05	Alicia Kavelaars
3.5.0	ELB-45	Fixed	TimeStamp or StartTime to sort runs into Shifts?	↑	3/29/05	Alicia Kavelaars
3.6.0	ELB-92	Fixed	Ensure that OP and QA are entered in the MMR Log	↓	8/15/05	Alicia Kavelaars
3.6.0	ELB-91	Fixed	TEST Recovery of E-Logbook from fatal crash in the cleanroom through backup system	↑	8/11/05	Alicia Kavelaars
3.6.0	ELB-88	Fixed	Create duplicates of components for edit to a new item	↑	8/3/05	Alicia Kavelaars
3.6.0	ELB-87	Fixed	Change required to MySQL to Oracle synchronization due to Python upgrade	↑	7/29/05	Selim Tuvi
3.6.0	ELB-85	Cannot Reproduce	CAL test reports are not showing up in Test Output column	↑	7/21/05	Charlotte Hee
3.6.0	ELB-84	Fixed	Double clicking on run 135003983 brings up the hour glass, but no Run Report	↑	7/20/05	Alicia Kavelaars
3.6.0	ELB-83	Fixed	Add a symbol to the single line log displays to indicate there are more lines to the message	↑	7/20/05	Alicia Kavelaars
3.6.0	ELB-82	Duplicate	E-logbook still truncates passwords	↑	7/20/05	Alicia Kavelaars
3.6.0	ELB-81	Won't Fix	Runs missing from elogshiftsummary data table in Oracle	↑	7/5/05	Selim Tuvi
3.6.0	ELB-49	Fixed	RUNS part of e-logbook does not appear on the web.	↑	4/4/05	Alicia Kavelaars
3.6.1	ELB-93	Cannot Reproduce	The eloguser table in Oracle needs to be resynchronized with the stand-alone e-logbook version	↓	10/26/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
3.6.1	ELB-90	Fixed	data entry error in component installation record		8/11/05	Alicia Kavelaars
3.7.0	ELB-102	Fixed	E-Logbook 3.70 ready for RPM (some fixes in SPEC file are needed)		11/23/05	Paul Kunz
3.7.0	ELB-101	Fixed	FHW Matrix Log needs editing to load ELX Assy info.		11/20/05	Alicia Kavelaars
3.7.0	ELB-100	Fixed	Fix slow down of Mate/Demate Log		11/17/05	Alicia Kavelaars
3.7.0	ELB-98	Fixed	Torque Value field needs two additional categories added. 1) NOT REQUIRED (NR) 2) NOT COMPLETED		11/17/05	Alicia Kavelaars
3.7.0	ELB-97	Fixed	Add a checkbox to hide savers in Mate/Demate Log (to reduce slowdown)		11/16/05	Alicia Kavelaars
3.7.0	ELB-96	Fixed	Mate/Demate Torque Information needs to have N/A for new components (do not require screw with Torque)		11/16/05	Alicia Kavelaars
3.7.0	ELB-95	Fixed	Preferences Dialog		11/16/05	Alicia Kavelaars
3.7.0	ELB-94	Fixed	Linux Fixes		11/10/05	Alicia Kavelaars
3.8.0	ELB-104	Fixed	Create a MD report by Component showing current state of all its R/D		12/8/05	Alicia Kavelaars
4.0.0	ELB-106	Fixed	Exception when requesting run report for a run that has the version collection turned off		12/15/05	Alicia Kavelaars
4.0.0	ELB-105	Fixed	Add genui.py for Unix setup and add schema tables for new fields dedicated to LICOS		12/8/05	Alicia Kavelaars
4.0.0	ELB-89	Fixed	Need web based e-logbook to properly associate analysis runs with appropriate bay		8/3/05	Charlotte Hee
4.1.0	ELB-120	Fixed	cut-and-paste to e-logbook on mobile rack doesn't work		2/15/06	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
4.1.0	ELB-119	Fixed	Remove the foreign key constraint between LICOS_activities table and elogreport	↓	2/14/06	Alicia Kavelaars
4.1.0	ELB-117	Fixed	Module version data is grouped by SIU, EPUX in some cases on LICOS. Update run report accordingly	↑	2/9/06	Alicia Kavelaars
4.1.0	ELB-116	Fixed	Creation of bogus shiftIDs on shifter sign up in unsynchronized activity entries.	↑	2/9/06	Alicia Kavelaars
4.1.0	ELB-115	Fixed	Add LICOS related fields in Run Report	↑	2/6/06	Alicia Kavelaars
4.1.0	ELB-112	Fixed	Clear hard coded PYTHONPATH in .csh startup script	↑	1/31/06	Alicia Kavelaars
4.1.0	ELB-111	Fixed	dumpElogbook should have MySQL create output files in /tmp/glast rather than /tmp	↑	1/30/06	Selim Tuvi
4.1.0	ELB-110	Fixed	Move hard coded database and user references to the .cnf file in the mirroring code	↓	1/25/06	Selim Tuvi
4.1.0	ELB-109	Fixed	Change hard coded elogbook database name to selected name in connect dialog	↑	1/10/06	Alicia Kavelaars
4.1.0	ELB-108	Fixed	Create a MMR Report to facilitate MMR review	↑	1/10/06	Alicia Kavelaars
4.1.0	ELB-107	Fixed	Create a MMR Report showing current Hardness Test Results of MMR	↑	12/16/05	Alicia Kavelaars
4.1.0	ELB-47	Fixed	How to capture database name if other than "elogbook"	↑	3/30/05	Alicia Kavelaars
	ELB-118	UNRES.	Version table should be parsed differently in the summary info page for LICOS runs	↑	2/10/06	Charlotte Hee
	ELB-114	UNRES.	Create a configuration file parameter on startup	↑	2/1/06	Alicia Kavelaars
	ELB-113	UNRES.	Create an E-Logbook Search Engine	↑	1/31/06	Alicia Kavelaars
	ELB-103	UNRES.	Documentation of E-Logbook's scripts	↑	11/30/05	Alicia Kavelaars

Release Version	Issue #	Resolution	Summary	Priority	Created	Assignee
	ELB-99	UNRES.	Discrepancy between standalone elogbook and web elogbook	↑	11/17/05	Charlotte Hee
	ELB-86	UNRES.	Add a "Number of entries" field to the List Runs filter	↑	7/21/05	Charlotte Hee
	ELB-35	UNRES.	elogparameters table is not automatically updated during a new install	↑	2/22/05	Alicia Kavelaars

Table 32: E-Logbook JIRA issues

APPENDIX D: Usability Questionnaire

The following is the questionnaire used to gather the data analyzed in Chapter 6 on the usability of E-Logbook. To save pages the spaces left between questions have been removed.

GENERAL QUESTIONS

Please answer the following questions:

1. What is your role in the LAT I&T team? (e.g. Mechanical Technician, Test Director)
2. Which of E-Logbook's Logs do you mostly use?
3. Do you use E-Logbook mostly for data input or data output?
4. Are there any parts of E-Logbook that you find confusing or difficult to fully understand? What about after reading the tutorial?
5. Are there any aspects of the system that you find particularly irritating although they do not cause major problems?
6. What are the most common mistakes you make when using E-Logbook?
7. What is E-Logbook most helpful with?

SECTION 1. VISUAL CLARITY

The information displayed on the screen should be clear, well organized, unambiguous and easy to read.

Please check and answer the following questions:

VISUAL CLARITY	Always	Most of the time	Some of the time	Never	Comments
1. Is important information highlighted on the screen? (e.g. cursor position, instructions, errors)					
2. Does information appear to be organized logically on the screen? (e.g. menus organized by probable sequence of selection, or alphabetically)					
3. Are different types of information clearly separated from each other on the screen? (e.g. instructions, control options, data displays)					
4. Does the use of color help to make the displays clear?					
5. Is the information on the screen easy to see and read?					
6. Do screens appear uncluttered?					
7. Is it easy to find required information on a screen?					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of visual clarity?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 2. CONSISTENCY AND COMPATIBILITY

The way the system looks and works should be consistent and compatible at all times.

Please check and answer the following questions:

CONSISTENCY AND COMPATIBILITY	Always	Most of the time	Some of the time	Never	Comments
1. Are there different colors used consistently throughout the system? (e.g. errors always highlighted in the same color)					
2. Is the same type of information (e.g. instructions, menus, messages, titles) displayed: (a) in the same location on the screen? ----- (b) in the same layout?					
3. Is the method of entering information consistent throughout the system?					
4. Are the abbreviations used understandable? (e. g. MMR, EGSE)					
5. Is the information presented and analyzed in the units with which the users normally work? (e.g. kilos, pounds)					
6. Does the system work in the way the user thinks it should work?					
7. Does the sequence of activities required to complete a task follow what the user would expect?					
8. Are the graphical displays compatible with the user's view of what they represent?					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of visual consistency and compatibility?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 3. INFORMATIVE FEEDBACK AND USER GUIDANCE

Users should be given clear, informative feedback on where are they in the system. Relevant support should be provided both on the computer and in a hard copy document. Please check and answer the following questions:

INFORMATIVE FEEDBACK AND USER GUIDANCE	Always	Most of the time	Some of the time	Never	Comments
1. Is it clear what actions the user can take at any stage?					
2. Are the instructions and messages displayed by the system concise and positive?					
3. Do error messages explain clearly: (a) where the errors are? ----- (b) what the errors are? ----- (c) why they have occurred?					
4. Were there any tutorials on the usage of the system? ----- Were they useful enough? ----- Were they frequent enough?					
5. Is there some form of hard-copy guide to the system (e.g. user guide or manual)? ----- (a) If so, does it provide an in-depth description of the system? ----- (b) Is it easy to find the required section?					
6. Is there some form of online guide to the system? ----- Did you use it?					

Are there any comments (good or bad) you wish to add regarding to the above issues?
Overall, how would you rate the system in terms of informative feedback and user guidance?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 4. EXPLICITNESS

The way the system works should be clear to the user.

Please check and answer the following questions:

EXPLICITNESS	Always	Most of the time	Some of the time	Never	Comments
1. Is it clear what the user needs to do in order to complete a task?					
2. Is it clear what part of the system the user is in?					
3. Is it clear what the different parts of the system do?					
4. Is it clear why the system is organized and structured as it is?					
5. Is it clear how, where and why changes on one part of the system affect other parts of the system?					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of explicitness?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 5. APPROPRIATE FUNCTIONALITY

The system should meet the needs and requirements of users when carrying out tasks.

Please check and answer the following questions:

APPROPRIATE FUNCTIONALITY	Always	Most of the time	Some of the time	Never	Comments
1. In general, is it clear what the system is doing?					
2. Is the way in which information is presented appropriate for the tasks?					
3. Does each screen contain all the information which the user feels is relevant for the task?					
4. Are the users provided with all the options and tutorial help which they feel are necessary at any particular stage in a task?					
5. Where task sequences are particularly long, are they broken into appropriate subsequences? (e.g. data input in a Log)					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of appropriate functionality?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 6. FLEXIBILITY AND CONTROL

The interface should be sufficiently flexible in structure to suit the needs and requirements of all users, and to allow them to feel in control of the system.

Please check and answer the following questions:

FLEXIBILITY AND CONTROL	Always	Most of the time	Some of the time	Never	Comments
1. In menu-based systems, is it easy to return to the main menu from any part of the system?					
2. Do users have control over the order in which they request information, or carry out a series of activities?					
3. Does the system pre-fill repeated information on the screen, where possible? (e.g. to save the user having to enter the same information several times)					
4. Can the users tailor certain aspects of the interface for their own preferences or needs? (e.g. parameters)					
5. Can the user access a particular screen in a sequence of screens directly? (e.g. where a list or table covers several screens)					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of flexibility and control?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 7. ERROR PREVENTION AND CORRECTION

The system should be designed to minimize the possibility of user error.

Please check and answer the following questions:

ERROR PREVENTION AND CORRECTION	Always	Most of the time	Some of the time	Never	Comments
1. In general, is the system free from errors and malfunctions?					
2. Does the system clearly and promptly inform the user when it detects an error?					
3. Is the system protected against common trivial errors?					
4. Is it easy for the user to correct errors?					
5. Does the system ensure that the user corrects all detected errors before the input is processed?					
6. Does the system prevent users from taking actions which they are not authorized to take? (e.g. by requiring passwords)					
7. When system errors occur, can the user access all necessary diagnostic information to resolve the problem? (e.g. where and what the fault is, what is required to solve it)					
8. Does the system inform the user when the amount of information entered exceeds the available space?					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of error prevention and correction?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

SECTION 8. SYSTEM USABILITY PROBLEMS

When using the system, did you experience problems with any of the following:

SYSTEM USABILITY PROBLEMS	No problems	Minor problems	Major problems	Comments
1. Working out how to use the system				
2. Finding the information you want				
3. Poor system documentation				
4. Knowing what to do next				
5. Understanding how the information in the screen relates to what you are doing				
6. Too many colors on the screen				
7. An inflexible, rigid structure				
8. Losing track of where you are in the system or of what you are doing or have done				
9. Having to remember too much information while carrying out a task				
10. System response times are too quick for you to understand what is going on.				
11. System response times are too slow.				
12. Unexpected actions by the system.				
13. An input device which is difficult or awkward to use.				
14. Knowing where or how to input information.				
15. Having to spend too much time inputting information.				
16. Having to be too careful in order to avoid errors.				
17. Having to spend too much time correcting errors.				
18. Having to carry out the same type of activity in different ways in the system.				

SECTION 9. E-LOGBOOK FEEDBACK

Please check and answer the following questions:

E-LOGBOOK FEEDBACK	Always	Most of the time	Some of the time	Never	Comments
1. Did you feel involved in the design and improvement of E-Logbook?					
2. During the course of LAT I&T, were the recommendations you gave to improve E-Logbook implemented satisfactorily in the next release?					
3. Did E-Logbook improve from one release to the next?					
4. Was the developer (Alicia) available for feedback and problem solving?					

Are there any comments (good or bad) you wish to add regarding to the above issues?

Overall, how would you rate the system in terms of error prevention and correction?

Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

FINAL QUESTIONS

1. Overall, do you think that E-Logbook improves the way to maintain I&T Logs?
2. What are the best aspects of E-Logbook?
3. Overall, do you think that E-Logbook is better than paper to:
 - a. Organize data?
 - b. Input data?
 - c. Retrieve data?
4. Please list aspects that could help improve E-Logbook (please focus on the Logs you used most):
 - a. Overall:
 - b. Shift Log:
 - c. Mate/Demate Log:
 - d. FHW Log:
 - e. MMR Log:
 - f. FSW Log:
 - g. EGSE Log:
5. Were you overall satisfied with E-Logbook?

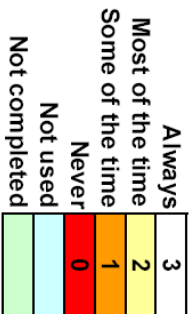
Very satisfactory	Satisfactory	Neutral	Unsatisfactory	Unsatisfactory

6. Would you recommend its use in the industry?

APPENDIX E: Questionnaire Results

Visual Clarity	User												Question Rating	
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA		TDir.
1. Is important information highlighted on the screen? (e.g. cursor position, instructions, errors)	2	2	3	3	2	2	3	3	3	0	1	3	2	2.09
2. Does information appear to be organized logically on the screen? (e.g. menus organized by probable sequence of selection, or alphabetically)	2	2	3	3	2	2	3	3	3	2	2	3	2	2.36
3. Are different types of information clearly separated from each other on the screen? (e.g. instructions, control options, data displays)	3	2	3	3	2	2	3	3	2	2	3	3	2	2.45
4. Does the use of color help to make the displays clear?	3	3	3	3	2	1	2	3	3	1	3	3	1	2.20
5. Is the information on the screen easy to see and read?	3	3	3	3	2	2	2	2	2	1	3	3	3	2.45
6. Do screens appear uncluttered?	2	3	3	3	2	2	2	3	3	1	3	3	3	2.30
7. Is it easy to find required information on a screen?	3	3	2	3	2	2	3	3	3	1	3	3	1	2.27
User Rating	2.57	2.57	2.80	3.00	2.00	1.86	2.57	N/A	1.71	1.43	3.00	2.00	2.32	2.31
Average Rating													77.32%	76.84%

Table 33: Visual Clarity Results on QNov05



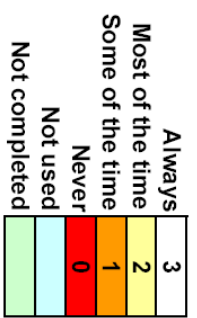
Prog.	User													MTEch.	TDir.	QA	QA	MTEch.	Question Rating			
	E Tech.	E Tech.	M Tech.	M Tech.	M Tech.	M Tech.	M Tech.	M Tech.	T Cond.	T Cond.	T Cond.	T Cond.	QA							QA		
1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	10	11	12	13	2.69		
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	1	2.69	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2.69	
3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2.69	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	1	2.62	
3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	3	2.69	
3	3	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2.54	
3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	3	1	2.54	
User Rating	3.00	2.29	3.00	3.00	3.00	3.00	3.00	3.00	3.00	2.71	2.29	2.29	2.29	3.00	2.00	3.00	2.64	3.00	3.00	1.71		
Average Rating											2.64	2.71	2.71	2.71	2.64	3.00	2.64	2.71	2.71	2.64	1.71	
Average Rating Initial Set											87.91%	90.48%	90.48%	90.48%	87.91%	87.91%	87.91%	87.91%	90.48%	90.48%	87.91%	

Table 34: Visual Clarity Results on QFeb06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

Consistency and Compatibility	User												Question Rating
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	
1. Are there different colors used consistently throughout the system? (e.g. errors always highlighted in the same color)	1	2	3	4	5	6	7	8	9	10	11	12	2.36
2. Is the same type of information (e.g. instructions, menus, messages, titles) displayed: (a) in the same location on the screen? (b) in the same layout?	3	3	3	3	3	2	1		0	3	2	3	2.50
3. Is the method of entering information consistent throughout the system?	3	3	3	2	3	2	3		1	2	3	3	2.50
4. Are the abbreviations used understandable? (e.g. MMR, EGSE)		3		2	3	2	3		2	2	3	3	2.30
5. Is the information presented and analyzed in the units with which the users normally work? (e.g. kilos, pounds)		1	3	3	2	2	0		2	1	3	3	1.82
6. Does the system work in the way the user thinks it should work?	3	2	3	3	2	1	2		1	1	2	2	2.00
7. Does the sequence of activities required to complete a task follow what the user would expect?	3	2	3	3	2	2	2			1	2	2	2.20
8. Are the graphical displays compatible with the user's view of what they represent?	3	2	2	3	2	2	2		1	2	2	2	2.09
User Rating	3.00	2.44	2.88	2.78	2.56	1.89	1.86	N/A	1.13	1.67	2.56	2.44	
Average Rating	2.10	2.25	69.97%	74.92%									

Table 35: Consistency and Compatibility Results on QNov05



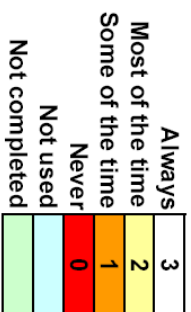
Consistency and Compatibility		User													Question Rating		
		Prog.	ETech.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	TCond.	QA	QA		TDir.	MTech.
1	3	2	3	3	3	3	3	3	3	2	2	3	3	3	3	3	2.77
3	3	3	2	3	3	3	3	3	3	3	1	2	3	3	3	2	2.62
3	3	3	3	3	3	3	3	3	3	3	1	2	3	3	2	2.69	
3	3	3	3	3	3	3	3	3	3	3	1	2	3	3	3	2.77	
3	2	3	3	3	3	3	3	3	3	3	1	2	3	3	3	2.69	
3	2	3	3	3	3	3	3	3	3	3	0	2	3	3	1	2.46	
3	3	3	2	3	3	3	3	3	3	2	2	2	3	3	1	2.54	
3	3	3	3	3	3	3	3	3	3	2	2	2	3	3	2	2.69	
3	3	3	2	3	3	3	3	3	3	2	2	2	3	3	2	2.67	
User Rating		3.00	2.78	2.56	3.00	3.00	3.00	3.00	3.00	2.44	1.56	2.11	3.00	3.00	3.00	2.11	
Average Rating													2.66	2.66	2.66	88.60%	88.51%
Average Rating Initial Set													2.70	2.70	2.70	90.12%	90.04%

Table 36: Consistency and Compatibility Results on QFeb06

3	Always
2	Most of the time
1	Some of the time
0	Never
	Not used
	Not completed

Explicitness	User												Question Rating
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	
1. Is it clear what the user needs to do in order to complete a task?	1	2	3	4	5	6	7	8	9	10	11	12	2.09
2. Is it clear what part of the system the user is in?	3	2		2	2	1	2	3	2	2	2	3	2.55
3. Is it clear what the different parts of the system do?	3	3		3	2	1	3	3	2	2	3	2	2.36
4. Is it clear why the system is organized and structured as it is?	3	2		2	3	1	3	3	1	2	2	2	2.27
5. Is it clear how, where and why changes on one part of the system affect other parts of the system?	3	2		2	3	1	1	3	2	1	2	2	2.00
User Rating	3.00	2.20	N/A	2.40	2.60	1.00	2.40	3.00	1.80	1.80	2.40	2.20	
Average Rating									2.25	2.25	75.15%	75.15%	

Table 37: Explicitness Results on QNov05



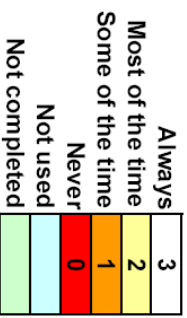
Explicitness	User													Question Rating		
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	TDir.		MTech.	
1. Is it clear what the user needs to do in order to complete a task?	1	2	3	3	3	3	3	3	3	8	9	10	11	12	13	2.77
2. Is it clear what part of the system the user is in?	3	3	3	3	3	3	3	2	2	3	3	3	3	3	2	2.69
3. Is it clear what the different parts of the system do?	3	3	3	3	3	3	3	2	2	3	1	3	2	3	2	2.62
4. Is it clear why the system is organized and structured as it is?	3	3	3	3	3	3	3	3	3	3	2	3	3	3	2	2.85
5. Is it clear how, where and why changes on one part of the system affect other parts of the system?	3	3	3	3	3	3	3	2	2	2	2	3	3	3	0	2.54
User Rating	3.00	3.00	3.00	3.00	3.00	3.00	3.00	2.60	2.40	2.40	1.80	3.00	2.80	3.00	1.40	
Average Rating											2.69	2.69	89.74%	89.74%		
Average Rating Initial Set											2.80	2.80	93.33%	93.33%		

Table 38: Explicitness Results on QFeb06

3	Always
2	Most of the time
1	Some of the time
0	Never
	Not used
	Not completed

Appropriate Functionality	User												Question Rating
	Prog. 1	ETech. 2	ETech. 3	MTech. 4	MTech. 5	MTech. 6	MTech. 7	MTech. 8	TCond. 9	QA 10	QA 11	TDir. 12	
1. In general, is it clear what the system is doing?	3	3		3	3	2	2	3	2	1	3	3	2.55
2. Is the way in which information is presented appropriate for the tasks?	3	3		2	3	2	2	3	2	2	3	3	2.55
3. Does each screen contain all the information which the user feels is relevant for the task?	3	3		3	3	2	2	3	2	1	3	3	2.55
4. Are the users provided with all the options and tutorial help which they feel are necessary at any particular stage in a task?	1	3			3	2	2	3	0	2	3	3	2.20
5. Where task sequences are particularly long, are they broken into appropriate subsequences? (e.g. data input in a Log)	3	3		3	3	2	2	3	2	1	3	3	2.55
User Rating	2.60	3.00	N/A	2.75	3.00	2.00	2.00	3.00	1.60	1.40	3.00	3.00	
Average Rating	2.49	2.48	82.88%	2.48	82.88%	82.55%	82.55%						

Table 39: Appropriate Functionality Results on QNov05



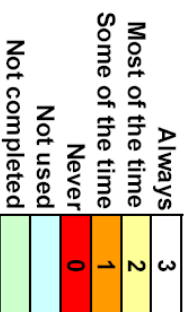
Appropriate Functionality	User													Question Rating			
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	TDir.		MTech.		
1. In general, is it clear what the system is doing?	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2.85
2. Is the way in which information is presented appropriate for the tasks?	3	3	3	3	3	3	3	2	2	3	3	3	3	3	2	2	2.69
3. Does each screen contain all the information which the user feels is relevant for the task?	3	3	3	3	3	3	3	2	3	1	3	3	3	3	2	2	2.69
4. Are the users provided with all the options and tutorial help which they feel are necessary at any particular stage in a task?	3	3	3	3	3	3	3	3	2	3	3	3	3	3	1	1	2.70
5. Where task sequences are particularly long, are they broken into appropriate subsequences? (e.g. data input in a Log)	3	3	2	3	3	3	3	2	2	2	3	3	3	3	1	1	2.50
User Rating	3.00	3.00	2.80	3.00	3.00	3.00	3.00	2.25	2.40	1.75	3.00	3.00	3.00	3.00	1.60		
									Average Rating		2.68	2.69	89.23%	89.54%			
									Average Rating Initial Set		2.77	2.79	92.22%	92.95%			

Table 40: Appropriate Functionality Results on QFeb06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

Informative Feedback and User Guidance	User												Question Rating
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	
1. Is it clear what actions the user can take at any stage?	1	2	3	4	5	6	7	8	9	10	11	12	2.08
2. Are the instructions and messages displayed by the system concise and positive?	3	3	3	2	2	2	2	3	2	1	3	2	2.33
3. Do error messages explain clearly:	3												
(a) where the errors are?	3												2.27
(b) what the errors are?	3												2.27
(c) why they have occurred?	3												2.18
4. Were there any tutorials on the usage of the system?		2	3		3	2					3	3	2.67
Were they useful enough?		2	3		3	2					3	3	2.67
Were they frequent enough?		2	3		3	1					3	3	2.50
5. Is there some form of hard-copy guide to the system (e.g. user guide or manual)?		3	2		3	3	0					3	2.33
(a) If so, does it provide an in-depth description of the system?		2	2		3	3	0					3	2.17
(b) Is it easy to find the required section?		3	2		3	3	0					3	2.33
6. Is there some form of online guide to the system?		3	3		3	3	3					3	3.00
(a) Did you use it?		3	3		3	1						3	2.60
User Rating	3.00	2.50	2.46	2.00	2.62	2.08	1.33	3.00	1.20	1.80	3.00	2.85	
Average Rating	2.32	2.42	77.31%	80.54%									

Table 41: Informative Feedback and User Guidance Results on QNov05



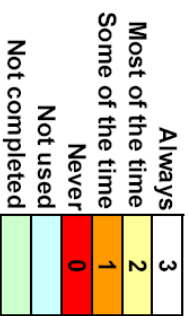
User																						
Prog.	ETech.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	TDDir.	QA	QA	QA	QA	QA	QA	Question Rating				
1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.77				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.75				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.60				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.60				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.60				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.70				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.70				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.70				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.55				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.55				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.55				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.80				
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.00				
User Rating													3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	1.78	
Average Rating													2.63	2.68	2.74	2.70	2.63	2.68	2.74	2.70	2.63	89.37%
Average Rating Initial Set													2.70	2.74	2.74	2.70	2.74	2.74	2.74	2.70	91.36%	

Table 42: Informative Feedback and User Guidance Results on QFeb06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

Error Prevention and Correction	User												Question Rating
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	
1. In general, is the system free from errors and malfunctions?	1	2	3	4	5	6	7	8	9	10	11	12	2.00
2. Does the system clearly and promptly inform the user when it detects an error?	2	2	3	2	2	2	3	3	3	2	3	3	2.50
3. Is the system protected against common trivial errors?	2	2	2	3	2	2	2	2	2	2	3	2	2.17
4. Is it easy for the user to correct errors?	3	1	1	3	1	2	1	0	2	1	2	2	1.55
5. Does the system ensure that the user corrects all detected errors before the input is processed?	2	2	3	3	2	1	1	0	2	2	2	3	1.92
6. Does the system prevent users from taking actions which they are not authorized to take? (e.g. by requiring passwords)	2	3	3	3	3	3	3	1	3	3	2	3	2.67
7. When system errors occur, can the user access all necessary/ diagnostic information to resolve the problem? (e.g. where and what the fault is, what is required to solve it)	3	2	1	2	2	2	2	1	1	2	2	2	1.82
8. Does the system inform the user when the amount of information entered exceeds the available space?	3	2	3	3	3	3	3	1	1	2	2	3	2.29
User Rating													2.38
Average Rating													1.38
													2.00
													2.11
													1.86
													2.43
													2.50
													70.39%
													70.41%

Table 43: Error Prevention and Correction Results on QNov05



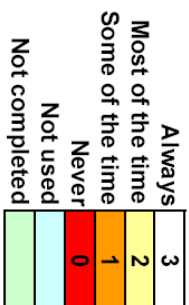
User																				
Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	TDDir.	MTEch.	Question Rating						
1	2	3	4	5	6	7	8	9	10	11	12	13								
3	3	2	3	3	2	2	1	2	3	2	3	2		2.38						
3	3	3		3	3		1	2	3	3	3	2		2.64						
3	3	3	2	3	3	3	2	3	3	2	3	1		2.62						
3	3	3		3	3	1	0	1	3	3	3	1		2.25						
3	3	3	3	3	3	2	0	2	3	3	3	3		2.62						
3	3	3	3	3	3	3	2	3	3	3	3	3		2.92						
3		2			3	3	0	1	3	3	3	1		2.20						
3		3			3		1		3	3	3	1		2.50						
User Rating													3.00	3.00	2.75	2.75	3.00	2.75	3.00	1.75
Average Rating													2.54	2.52	84.83%	83.85%				
Average rating Initial Set													2.61	2.60	87.04%	86.57%				

Table 44: Error Prevention and Correction Results on QFeb06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

Flexibility and Control	User												Question Rating
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	
1. In menu-based systems, is it easy to return to the main menu from any part of the system?	1	2	3	4	5	6	7	8	9	10	11	12	2.33
2. Do users have control over the order in which they request information, or carry out a series of activities?	3	2	3	1	2	0	0	3	2	2	3	3	2.00
3. Does the system pre-fill repeated information on the screen, where possible? (e.g. to save the user having to enter the same information several times)	3	2	3	2	1	2		0	1	1	3	1	1.73
4. Can the users tailor certain aspects of the interface for their own preferences or needs? (e.g. parameters)	2	2	2	1	2	0	0	0	0	0	3	0	1.00
5. Can the user access a particular screen in a sequence of screens directly? (e.g. where a list or table covers several screens)	3		3		2	2	1	2	2	0	3	1	1.90
User Rating	2.80	2.00	2.80	1.50	1.80	1.00	0.75	1.60	1.40	1.00	3.00	1.60	
Average Rating	1.77	1.79	59.03%	59.74%									

Table 45: Flexibility and Control Results on QNov05



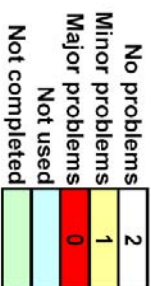
		User													Question Rating	
Prog.	E.Tech.	E.Tech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	TCond.	QA	QA	TDir.	MTech.	Question Rating
1	2	3	4	5	6	7	8	9	10	11	12	13				
3	3	3		3	3	3	2	2	3	3	3	1				2.67
3	3	2	3	3	1	1	1	3	3	3	3	1				2.31
3	3	2	3	3	3	2	2		3	3	3	1				2.58
3	3	3	3	3	3		2	1	3	3	3	0				2.50
3	3	3	3	3	3	1	1	2	3	3	3	1				2.46
User Rating		3.00	3.00	3.00	2.60	1.75	1.60	2.00	3.00	3.00	3.00	0.80				
		Average Rating											83.46%			
		Average Rating Initial Set											88.48%			

Table 46: Flexibility and Control Results on QFeB06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

System Usability Problems	User												Question Rating	
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA		TDir.
1. Working out how to use the system	2	2	1	1	1	1	1	1	2	1	2	1	1	1.33
2. Finding the information you want	2	2	2	2	1	1	1	2	2	1	2	2	1	1.67
3. Poor system documentation	2	2	2	2	2	1	2	2	1	1	2	2	2	1.75
4. Knowing what to do next	2	2	2	1	1	1	1	2	2	2	2	2	1	1.67
5. Understanding how the information in the screen relates to what you are doing	2	2	2	1	2	1	1	2	2	2	2	2	2	1.83
6. Too many colors on the screen	2	2	2	2	2	2	2	2	2	2	2	2	2	2.00
7. An inflexible, rigid structure	2	1	2	2	2	1	0	1	2	1	2	2	2	1.50
8. Losing track of where you are in the system or of what you are doing or have done	2	1	2	2	2	1	1	1	2	2	2	2	2	1.75
9. Having to remember too much information while carrying out a task.	2	1	2	2	2	1	2	2	2	2	2	2	2	1.82
10. System response times are too quick for you to understand what is going on.	2	2	2	2	2	2	2	2	2	2	2	2	2	2.00
11. System response times are too slow.	1	0	2	2	1	1	1	1	2	0	2	2	1	1.25
12. Unexpected actions by the system.	1	2	2	1	1	1	2	2	2	1	2	2	2	1.58
13. An input device which is difficult or awkward to use.	2	2	2	2	2	2	2	2	2	2	2	2	2	2.00
14. Knowing where or how to input information.	2	2	2	2	2	1	1	2	2	2	2	2	2	1.83
15. Having to spend too much time inputting information.	2	2	2	2	1	0	1	1	2	0	2	2	1	1.33
16. Having to be too careful in order to avoid errors.	1	1	2	2	1	1	0	0	2	1	1	2	2	1.25
17. Having to spend too much time correcting errors.	2	2	1	2	2	1	1	1	2	1	2	2	2	1.60
18. Having to carry out the same type of activity in different ways in the system.	2	2	2	2	2	1	2	2	2	1	2	2	2	1.83
User Rating	1.82	1.65	1.89	1.78	1.61	1.11	1.44	1.71	1.83	1.50	1.94	1.72	1.83	
Average Rating	1.67	1.67	1.87	1.83	1.50	83.37%	83.34%	1.67	1.83	1.50	1.94	1.72	1.83	

Table 47: System Usability Results on QNov05



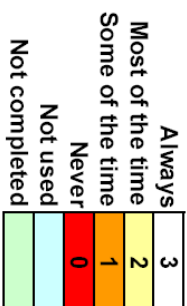
Prog.	User											Question Rating	
	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	TDir.		MTech.
1	2	3	4	5	6	7	8	9	10	11	12	13	1.77
2	2	2	2	2	2	1	2	1	2	2	2	1	1.62
2	2	1	2	2	2	2	1	1	2	2	2	2	1.92
2	2	2	2	2	2	2	2	2	2	2	2	2	1.85
2	2	2	2	2	2	2	2	2	2	2	2	2	1.92
2	2	1	2	2	2	2	1	2	2	2	2	2	1.85
2	2	2	2	2	2	2	2	2	2	2	2	0	1.77
2	2	2	2	2	2	1	1	1	2	2	2	1	1.69
2	2	2	2	2	2	1	0	2	2	2	2	0	1.62
2	2	2	2	2	2	2	1	2	2	2	2	2	1.92
2	2	1	2	2	2	2	0	2	2	2	2	2	1.77
2	2	2	2	2	2	2	2	1	2	2	2	2	1.92
2	2	2	2	2	2	2	2	1	2	2	2	2	1.92
2	2	2	2	2	2	1	1	1	2	2	2	1	1.69
2	2	1	2	2	2	1	0	2	0	2	2	0	1.38
2	2	1	2	2	2	1	0	1	2	2	2	0	1.46
2	2	2	2	2	2	1	0	2	2	2	2	0	1.62
2	2	2	2	2	2	2	1	2	2	2	2	1	1.85
2.00	2.00	1.72	2.00	2.00	2.00	1.56	0.89	1.61	1.89	2.00	2.00	1.11	
User Rating													Average Rating
													1.75
													87.61%
Average Rating Initial Set													1.81
													90.28%

Table 48: System Usability Results on QFeb06

No problems	2
Minor problems	1
Major problems	0
Not used	
Not completed	

		User												Question Rating		
	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	TDir.		
E-Logbook Feedback	1. Did you feel involved in the design and improvement of E-Logbook?	1	2	3	4	5	6	7	8	9	10	11	12	3	1.70	
	2. During the course of LAT 1&T, were the recommendations you gave to improve E-Logbook implemented satisfactorily in the next release?	3	2	2	3	3	3	3			2			3	2.67	
	3. Did E-Logbook improve from one release to the next?	3	3	3	3	3	3	3			3			3	2.90	
	4. Was the developer (Alicia) available for feedback and problem solving?	3	3	3	3	3	3	3			3			3	3.00	
User Rating	2.50	2.50	2.25	2.50	3.00	2.75	2.75	1.50	3.00	2.25	N/A	3.00				
Average Rating																
		2.55	2.57	84.85%	85.56%											

Table 49: E-Logbook Feedback Results on QNov05



User														Question	
E-Logbook Feedback	Prog.	ETech.	ETech.	MTech.	MTech.	MTech.	MTech.	MTech.	TCond.	TCond.	QA	QA	TDir.	MTech.	Rating
1. Did you feel involved in the design and improvement of E-Logbook?	1	2	3	3	3	3	3	3	1	2	3	2	3	2	2.38
2. During the course of LAT I&T, were the recommendations you gave to improve E-Logbook implemented satisfactorily in the next release?	3	3	0	3	3	3	3	3	3	3	3	3	3	1	2.75
3. Did E-Logbook improve from one release to the next?	3	3	3	3	3	3	3	3	3	2	3	3	3	3	2.92
4. Was the developer (Alicia) available for feedback and problem solving?	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3.00
User Rating	3.00	3.00	2.25	2.75	3.00	3.00	3.00	2.50	2.33	3.00	2.75	3.00	3.00	2.25	
Average Rating															
Average Rating Initial Set															
2.76 91.88% 92.15%															
2.80 93.29% 93.69%															

Table 50: E-Logbook Feedback Results on QFeb06

Always	3
Most of the time	2
Some of the time	1
Never	0
Not used	
Not completed	

Bibliography

- [1] Wolfram Research Science World, <http://scienceworld.wolfram.com/physics/ElectronVolt.html>

- [2] Nolan, P.L.; Bertsch, D.L.; Fichtel, C.E.; Hartman, R.C.; Hofstadter, R.; Hughes, E.B.; Hunter, S.D.; Kanbach, G.; Kniffen, D.A.; Lin, Y.C.; Mattox, J.R.; Mayer-Hasselwander, H.A.; Michelson, P.F.; von Montigny, C.; Pinkau, K.; Rothermel, H.; Schneid, E.; Sommer, M.; Sreekumar, P.; Thompson, D.J., “*Performance of the EGRET astronomical gamma ray telescope*”, Nuclear Science, IEEE Transactions; Volume 39, Issue 4, Page(s):993 – 996; August 1992.

- [3] W.B. Atwood, W.B. et al., Nucl. Instr. Meth. A342(1994)302.

- [4] Michelson, P.F; “*GLAST, a detector for high-energy gamma rays*”, Proceedings SPIE; Vol.2806; Ramsey and Parnell eds. (1996)31.

- [5] Bloom, E.D.; Godfrey, G. Ritz, S., scientific editors; Atwood, W.B.; Michelson, P.F. et al.; “*GLAST Collaboration, Proposal to the DoE for the Gamma-ray Large Area Space Telescope*”; SLAC-R-522; 1998.

- [6] Michelson, P.F. et al.; “*GLAST Large Area Telescope Flight Investigation*”, Response to NASA A0 99-OSS-03; 1999.

- [7] Johnson, R.P., “*The GLAST LAT science instrument*”; Nuclear Science Symposium Conference Record, IEEE; Page(s): 1604- 1608 Vol.3; October 2003.

- [8] Yoshida, S. et al.; “*Performance of large-area silicon strip sensors for GLAST*”, Nuclear Science, IEEE Transactions; Volume 49, Issue 3, Part 2, Page(s):1017 – 1021; June 2002.

- [9] Yoshida, S. et al.; “*Performance of large area silicon strip sensors for GLAST*”; Nuclear Science Symposium Conference Record, IEEE; Page(s): 180 - 184 vol.1; November 2001.
- [10] Thompson, D.J. et al.; “*Gamma-ray Large-Area Space Telescope (GLAST) balloon flight engineering model: overview*”; Nuclear Science, IEEE Transactions; Volume 49, Issue 4, Part 1, Page(s):1898 – 1903; August 2002.
- [11] Burnett, T.H. et al.; “*Gamma-ray Large-Area Space Telescope (GLAST) balloon flight data handling overview*”; Nuclear Science, IEEE Transactions; Volume 49, Issue 4, Part 1, Page(s):1904 – 1908; August 2002.
- [12] Thompson, D.J. et al., “*Gamma ray Large Area Space Telescope Balloon Flight Engineering Model: overview*”; Nuclear Science Symposium Conference Record, IEEE; Volume 1, Page(s): 487 - 491 vol.1; November 2001.
- [13] Vinson, J.; “*Hubble Space Telescope Project – Goddard. Vehicle Electrical System Facility (VEST) Integration and Test Test Plan*”; J&T/VTPL-0002, NASA Goddard, January 1990.
- [14] Clough, S.A.; “*Typical GSFC Electrical I&T*”, presentation for the GLAST Project Office, NASA Goddard, April 2004.
- [15] Bloom, E.; Fouts, K.; Grist, B.; “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Integration and Test Subsystem Test Plan*”; LAT-MD-01376-02, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; April 2005.
- [16] Nordby, M. et al.; “*LAT Integration Sequence*”; LAT-MD-00676-04, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; July 2005.

- [17] Hascall, P.; “*LAT Performance & Operations Plan*”; LAT-MD-02730-02, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; April 2005.
- [18] Claus, R.; Horwitz, B.; Wai, L.; “*LAT I&T EGSE Plan*”; LAT-MD-01533-01, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; December 2004.
- [19] US FED-STD-209, “*Airborne Particulate Cleanliness Classes in Cleanrooms and Clean Zones*”; IHS website, Rev. 2002; http://global.ihs.com/doc_detail.cfm?currency_code=USD&customer_id=21254E4D260A&shopping_cart_id=2825585F284B30344C5B5D383E0A&rid=Z56A&input_doc_number=FED%2DSTD%2D209D&mid=W084&input_doc_number=FED%2DSTD%2D209D&country_code=US&lang_code=ENGL&item_s_key=00136753&item_key_date=070831&input_doc_number=FED%2DSTD%2D209D&input_doc_title=
- [20] Davis, W. (on behalf of Orbital Network Engineering, Inc.); “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Contamination Control Plan*”; LAT-MD-00404-03, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; April 2003.
- [21] Borden, T., Wai, L.; “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Integration and Test Subsystem Facility Plan*”; LAT-MD-01386-02, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; February 2005.
- [22] Canfield, J.C.; Wong, R.W.; “*Network Infrastructure and Configuration Control, GLAST Building 33*”; LAT-PS-05504-02, GLAST-LAT Documentation

- Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; August 2005
- [23] Oxford English Dictionary online, http://dictionary.oed.com/cgi/entry/50060363?nearest_to=demate.
- [24] Horn, D.; “*LAT I&T Configuration Management/Change Control Requirements*” LAT-MD-03492-01, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST>; SLAC; March 2004.
- [25] Codd, E.; “*A Relational Model of Data for Large Shared Data Banks*”, Communications of the ACM 13 (6), pp. 377-387; 1970.
- [26] Beaulieu, A.; “*Learning SQL*”; O’Reilly & Associates, Inc., August 2005.
- [27] King, T.; Reese, G.; Yarger, R. J.; “*Managing and Using MySQL*”; O’Reilly & Associates, Inc.; 2nd Edition, April 2002.
- [28] Patel, B.K.; “*Automated tools for database design and criteria for their selection for aerospace applications*”; Aerospace Applications Conference. IEEE Digest.; Page(s):13 pp.; February 1989.
- [29] Miller, R.J.; Ioannidis, Y.E.; Ramakrishnan, R.; “*Understanding schemas*”; Research Issues in Data Engineering: Interoperability in Multi-database Systems; Proceedings RIDE-IMS, Third International Workshop; Page(s):170 – 173; April 1993.
- [30] Castelli, D.; “*A strategy for reducing the effort for database schema maintenance*”; Software Maintenance and Reengineering; Proceedings of the Second Euromicro Conference; Page(s):29 – 35; March 1998.

- [31] Oertly, F.; Schiller, G.; “*Evolutionary database design*”; Data Engineering; Proceedings, Fifth International Conference; Page(s):618 – 624; February 1989.
- [32] SpaceRef.com website, <http://www.spaceref.com/news/viewpr.html?pid=12500>
- [33] SpaceToday.net website, <http://www.spacetoday.net/Summary/2590>
- [34] Myers, G.L.; Davis, M.E.; “*The human-computer interface requirements specification (HCIRS) as implemented within IBM SID engineering practices*”; Aerospace and Electronics Conference, NAECON 1989, Proceedings of the IEEE 1989 National; Page(s):734 - 740 vol.2; May 1989.
- [35] Meister, D.; “*The Nature of Human Error*”; Global Telecommunications Conference, 1989, and Exhibition. 'Communications Technology for the 1990s and Beyond' GLOBECOM '89, IEEE 27-30 Page(s):783 - 786 vol.2, November 1989.
- [36] Barker, P.; Lamont, C.; “*What are GUIs good for? Constructing and controlling multimedia information*”; IEE Colloquium; Page(s):6/1 - 6/6; January 1994.
- [37] U.S. Department of Labor, Bureau of Labor Statistics, <http://www.bls.gov/opub/ted/2000/Dec/wk1/art02.htm>
- [38] Goedicke, M.; Sucrow, B.E.; “*Towards a formal specification method for graphical user interfaces using modularized graph grammars*”; Software Specification and Design; Proceedings of the 8th International Workshop; Page(s):56 – 65; March 1996.
- [39] Sholl, H.A.; Ammar, R.A.; Weiss, W.S.; “*Using probabilistic finite state models to evaluate human-computer interfaces*”; Systems, Man, and Cybernetics, 'Decision Aiding for Complex Systems, Conference Proceedings; IEEE International Conference; Page(s):1145 - 1150 vol.2; October 1991.

- [40] Tanikoshi, K.; Arai, T.; Tani, M.; Yokoyama, T.; Tanifuji, S.; “*Design concept of a system for developing application-dependent user interface*”; Computers and Communications, Conference Proceedings., Tenth Annual International Phoenix Conference; Page(s):776 – 782; March 1991.
- [41] Kluger, L.; “*The Open Look graphical user interface and its toolkits*”; User Interface Management Systems, IEE Colloquium; Page(s):3/1 - 3/3; November 1989.
- [42] Trolltech, <http://www.trolltech.com/company/index.html>
- [43] Rempt, B.; “*GUI Programming with Python: Using the Qt Toolkit*”; Opendocs Publishers; 2002.
- [44] Bickford, P.; “*Interface Design, the Art of Developing Easy-To-Use Software*”; AP Professional, 1997.
- [45] Kaiser, M.K.; Proffitt, D.R.; “*Human-Display Interactions: Context-Specific Biases*”; Publication NASA-TM-89430; NASA; April 1987.
- [46] Rigden, C.; “*Now you see it, now you don't (graphical user interface design)*”; Computer, Volume 35, Issue 7, Page(s):104 – 105; July 2002.
- [47] Braim, S.P.; Hepworth, N.; “*A fully-featured human computer interface for intelligence and command and control applications*”; Information-Decision-Action Systems in Complex Organizations, International Conference; Page(s):149 – 152; April 1992.
- [48] Johnson, J.; “*GUI Bloopers, Don'ts and Do's for Software Developers and Web Designers*”; Morgan Kaufmann Publishers; 2000 by Academic Press.

- [49] “*Recommended Practice for Human-Computer Interfaces for Space System Operations*”; ANSI/AIAA R-023A-1995; American National Standard, American Institute of Aeronautics and Astronautics; August 1997.
- [50] EVIA, C.; “*Quality over quantity: a two-step model for reinforcing user feedback in transnational Web-based systems through participatory design*”; Professional Communication, IEEE Transactions; Volume 47, Issue 1, Page(s):71 – 74; March 2004.
- [51] Sexton, A.; Yeo, G.; Turner C.; Hockey S.; “*User Feedback: testing the LEADERS demonstrator application*”; Journal of the Society of Archivists; Volume 25, No. 2, 2004.
- [52] Gould, J.D.; “*How to design usable systems*”, in Handbook of Human-Computer Interaction; M. Helander, Editor; Elsevier Science Publishers, pp. 757–789; Amsterdam 1988.
- [53] Linger, R.C.; “*Cleanroom process model*”; Software, IEEE; Volume 11, Issue 2, Page(s):50 – 58; March 1994.
- [54] Beizer,B.; “*Cleanroom process model: a critical examination*”; Software, IEEE; Volume 14, Issue 2, Page(s):14 – 16; March/April 1997.
- [55] Vajde, R.; Rozman, I.; “*Tools for human-computer interface and dialogue management*”; Electro-technical Conference, 6th Mediterranean; Page(s):1141 - 1144 vol.2; May 1991.
- [56] PEP-II Home Page, <http://www.slac.stanford.edu/accel/pepii/home.html>
- [57] BaBar Collaboration Home Page, <http://www.slac.stanford.edu/BFROOT>
- [58] “*Connector Mate/Demate*”, presentation of the Western Region Manufacturing Technology Transfer Center, NASA JPL.

- [59] Microsoft website, 2005; <http://www.microsoft.com/resources/casestudies/CseStudy.asp?CaseStudyID=13593>
- [60] Wertz, J., Larson, W. (editors), “*Space Mission Analysis and Design*”; Space Technology Library, Microcosm Press and Kluwer Academic Publishers; 2001.
- [61] Bright, R.; “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Program Instrument Performance Verification Plan*”; LAT-MD-00408-04; GLAST-LAT Documentation Database; [https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST;SLAC](https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST;SLAC;); January 2005.
- [62] Fouts, K., Kavelaars, A.; “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Integration and Test (I&T) E-Logbook Implementation Plan*”; LAT-MD-04601-02; GLAST-LAT Documentation Database; [https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST;SLAC](https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST;SLAC;); January 2005.
Also located in attached DVD under *Documentation*.
- [63] GNU General Public License, <http://www.gnu.org/copyleft/gpl.html>
- [64] Open Source, <http://www.opensource.org/licenses/gpl-license.html>
- [65] Python, <http://python.org/doc/Copyright.html>
- [66] Beazley, D; “*Python Essential Reference*”; New Riders Publishing; 2nd Ed.; 2001.
- [67] Martelli, A.; “*Python in a Nutshell*”; O’Reilly & Associates, Inc.; 2003.
- [68] Python, <http://python.org/doc/Summary.html>
- [69] MySQL, <http://www.mysql.com>

- [70] Tkinter, <http://www.pythonware.com/library/tkinter/introduction/whats-tkinter.htm>
- [71] <http://en.wikipedia.org/wiki/WYSIWYG>
- [72] Trolltech, <http://www.trolltech.com/company/index.html>
- [73] River Bank Computing, PyQt, <http://www.riverbankcomputing.co.uk/pyqt/index.php>
- [74] Scintilla, <http://www.scintilla.org>
- [75] Doxygen, <http://www.stack.nl/~dimitri/doxygen>
- [76] <http://www.nongnu.org/cvs>
- [77] WinCVS, <http://www.wincvs.org>
- [78] WinMerge, <http://winmerge.sourceforge.net>
- [79] Atlassian, JIRA, <http://www.atlassian.com/software/jira>
- [80] GLAST web based Shift Log (password protected), Swing Shift for Friday, February 17 2006, <http://www.slac.stanford.edu/cgi-wrap/eLog.pl/show?shiftid=200602172&subsys=>
- [81] GLAST LAT I&T Online Group Website, <http://www-glast.slac.stanford.edu/IntegrationTest/ONLINE/Default.htm>
- [82] Kavelaars, A.; “*Gamma-ray Large Area Space Telescope (GLAST) Large Area Telescope (LAT) Integration and Test (I&T) E-Logbook Test Procedure and Report*”; LAT-MD-05664-06, GLAST-LAT Documentation Database; <https://oraweb.slac.stanford.edu:8080/pls/slacquery/DOCUMENTS.STARTUP?PROJECT=GLAST;SLAC;February2006>.
Also located in attached DVD under *Documentation*.

- [83] Ravden, S., Johnson, G.; *“Evaluating Usability Of Human-Computer Interfaces, a practical method”*; Ellis Horwood Ltd., Publishers; Distributed by John Wiley and Sons; 1989.
- [84] *“Style and Policy Manual For Theses and Dissertations”*; Revised Ed. Seattle: University of Washington Graduate School, 1994.
- [85] Hacker, D.; *“A Writer’s Reference”*; Bedford Books of St. Martin’s Press; 2nd Ed. Boston; 1992.