1998

The RMT (Recursive multi-threaded) tool: A computer aided software engineering tool for monitoring and predicting software development progress

Chungping Lin

Follow this and additional works at: http://scholarworks.lib.csusb.edu/etd-project

Part of the Data Storage Systems Commons

Recommended Citation


http://scholarworks.lib.csusb.edu/etd-project/1787

This Thesis is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.
THE RMT (RECURSIVE MULTI-THREADED) TOOL: A COMPUTER AIDED SOFTWARE ENGINEERING TOOL FOR MONITORING AND PREDICTING SOFTWARE DEVELOPMENT PROGRESS

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Chung-Ping Lin
December 1998
THE RMT (RECURSIVE MULTI-THREADED) TOOL: A COMPUTER AIDED SOFTWARE ENGINEERING TOOL FOR MONITORING AND PREDICTING SOFTWARE DEVELOPMENT PROGRESS

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Chung-Ping Lin
December 1998

Approved by:
Dr. Josephine Mendoza
Dr. Kerstin Voigt
ABSTRACT

The Recursive Multi-Threaded (RMT) Tool is an implementation of the Recursive Multi-Threaded software life-cycle model developed in the Masters' Thesis of Scott Simon [11]. The RMT Tool supports the object-oriented approach of software development and it monitors and predicts the progress of software during its development.

This project uses the Unified Modeling Language (UML), an object-oriented analysis design methodology, in the design of the RMT Tool. The design is independent of particular programming languages. For the implementation of the Tool, Java programming language, Java Database Connectivity (JDBC) Driver, and SQL databases are used. The Java programming language provides a platform independent environment, which allows the Tool to be accessed on the Internet. The JDBC driver defines a structured interface of the Tool to the Standard Query Language (SQL) database. The SQL database provides the storage space for the Tool.

The functions of the RMT Tool includes the project manager, the progress manager, the single project progress report, the multiple projects progress report, the project prediction, and personnel manager. The functionality of the Tool was fully validated by entering actual data from the
Algorithm 98 Project which is an on-going software project in the undergraduate software engineering course (CSCI455) and also actual data was entered for the software development of the Tool itself.
ACKNOWLEDGMENTS

First of all, I would like to thank my project advisor: Dr. Arturo I. Concepcion for his strong support in developing this project. His valuable guidance and suggestions contributed substantially to this project.

And thanks to Dr. Mendoza and Dr. Voigt, my project committee professors, giving their insightful comments and suggestion, pointing out the conceptual flaws during the development of this project, and making it better.

I would like to acknowledge Associated Studies, Incorporated (ASI) at California State University, San Bernardino, who helped support my project through an ASI grant.

Finally, a special note of appreciation to my family - in particular, my parents and my wife. Because of my parents’ encouragement and financial support, they made my education possible. Because of my wife’s love and prayers, I was able to put my all time and energy to complete this project.

Chung-Ping Lin
To Tiffany and Ariel, My family
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>CHAPTER ONE  INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Purpose of This Project</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Project Products</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER TWO  SOFTWARE REQUIREMENTS SPECIFICATION</td>
<td></td>
</tr>
<tr>
<td>2.1 Functionality of RMT Tool</td>
<td></td>
</tr>
<tr>
<td>2.1.1 Project Manager</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Progress Manager</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3 Single Project Progress Report</td>
<td>6</td>
</tr>
<tr>
<td>2.1.4 Multiple Project Progress Report</td>
<td>6</td>
</tr>
<tr>
<td>2.1.5 Project Prediction</td>
<td>7</td>
</tr>
<tr>
<td>2.1.6 Personnel Management</td>
<td>7</td>
</tr>
<tr>
<td>2.2 System Objective</td>
<td>9</td>
</tr>
<tr>
<td>2.3 System Analysis</td>
<td></td>
</tr>
<tr>
<td>2.3.1 Java</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2 Java Database Connectivity (JDBC)</td>
<td>11</td>
</tr>
<tr>
<td>2.3.3 SQL Database Engine</td>
<td>11</td>
</tr>
<tr>
<td>2.3.4 Personnel Software Process</td>
<td>12</td>
</tr>
<tr>
<td>2.3.5 Overall Design of RMT Tool</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Development, Operating, and Maintenance Environment</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1 Development Environment</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.4.1.1 Client Station</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1.2 Server Station</td>
<td>15</td>
</tr>
<tr>
<td>2.4.2 Operating Environment</td>
<td>15</td>
</tr>
<tr>
<td>2.4.2.1 Client Station</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2.2 Server Station</td>
<td>16</td>
</tr>
<tr>
<td>2.4.3 Maintenance Environment</td>
<td>16</td>
</tr>
<tr>
<td>2.5 External Interface</td>
<td></td>
</tr>
<tr>
<td>2.5.1 Authorization Page</td>
<td>17</td>
</tr>
<tr>
<td>2.5.2 Main Menu</td>
<td>18</td>
</tr>
<tr>
<td>2.5.3 Create New Project</td>
<td>19</td>
</tr>
<tr>
<td>2.5.4 Assign Number of Teams</td>
<td>20</td>
</tr>
<tr>
<td>2.5.5 Browse Current Project</td>
<td>21</td>
</tr>
<tr>
<td>2.5.6 Assign the Number of Iteration and</td>
<td>22</td>
</tr>
<tr>
<td>Percentages</td>
<td></td>
</tr>
<tr>
<td>2.5.7 Organization Tree</td>
<td>23</td>
</tr>
<tr>
<td>2.5.8 Graphical Thread</td>
<td>24</td>
</tr>
<tr>
<td>2.5.9 Documentation</td>
<td>25</td>
</tr>
<tr>
<td>2.5.10 Progress Update</td>
<td>26</td>
</tr>
<tr>
<td>2.5.11 Progress Report</td>
<td>27</td>
</tr>
<tr>
<td>2.5.12 Personnel Management</td>
<td>28</td>
</tr>
<tr>
<td>2.5.13 Maintenance Menu</td>
<td>29</td>
</tr>
<tr>
<td>2.5.14 Personnel Information Maintenance</td>
<td>30</td>
</tr>
<tr>
<td>2.5.15 Assign New Project</td>
<td>31</td>
</tr>
<tr>
<td>2.5.16 Change Password</td>
<td>32</td>
</tr>
<tr>
<td>2.5.17 Multi-Project Management</td>
<td>33</td>
</tr>
</tbody>
</table>
2.5.18 New Project Group ......... 34
2.5.19 Set Project Group ........ 35
2.5.20 Display Project Status ....... 36

2.6 Functional Requirement

2.6.1 Finite State Diagram Notation .... 37

2.6.2 State Diagram

2.6.2.1 Start Menu ............ 38
2.6.2.2 Project Leader Page ....... 39
2.6.2.3 Project Leader Browsing .... 40
2.6.2.4 Project Leader Browsing (Continued) ...... 41
2.6.2.5 Team Leader Browsing ....... 42
2.6.2.6 Programmer Browsing ...... 43
2.6.2.7 Personnel Management ....... 44
2.6.2.8 Team Leader Page ......... 45
2.6.2.9 Programmer Page .......... 46
2.6.2.10 Multiple Project Progress Report ......... 47

2.7 Performance Requirement

2.7.1 Human Engineering ........ 48
2.7.2 Progress Prediction .......... 48
2.7.3 Portability ............. 48
2.7.4 Modifiability .......... 49
2.7.5 Understandability .......... 49
2.7.6 Testability .......... 49
2.7.7 Reusability .................................. 49

CHAPTER THREE DESIGN

3.1 Class Diagram ............................... 50
3.2 Component Diagram ......................... 53
3.3 Collaboration Diagram ...................... 55
3.4 Sample Physical Design Tables ............. 60

CHAPTER FOUR SYSTEM VALIDATION

4.1 Unit Test ................................... 65
4.2 Integration Test ............................. 69
4.3 System Test ................................ 69

CHAPTER FIVE MAINTENANCE

5.1 mSQL Database Maintenance ................ 70
5.2 Java Programs ................................ 71

CHAPTER SIX FUTURE DIRECTIONS AND CONCLUSIONS .... 72

Appendix A Glossary ............................. 75
Appendix B Check List ........................... 77
Appendix C Sample of msql.conf ................ 81
Appendix D Sample of showcpm.msql ............ 82
Appendix E Directory List of Source Codes .... 83
Appendix F Sample Screens ..................... 85

REFERENCES .................................... 107
LIST OF TABLES

Table 2.1 Factors of Prediction ............... 7
Table 2.2 User Level and Access Privileges .... 8
Table 2.3 RMT Tool Objectives .................. 9
Table 3.1 Personnel Profile .................... 60
Table 3.2 Project Database ..................... 61
Table 3.3 Team Database ....................... 61
Table 3.4 Programmer Database ................. 62
Table 3.5 Multiple Project Database .......... 62
Table 3.6 Multiple Project Item Database ...... 62
Table 3.7 Thread Database ...................... 63
Table 3.8 Thread Log Database .................. 63
Table 4.1 Unit Test Results .................... 66
Table 4.2 Integration Test Result ............... 69
LIST OF FIGURES

Figure 1.1 Progress Computation .................................. 3
Figure 2.1 System Design of RMT Tool .............................. 14
Figure 3.1 Class Diagram ............................................ 52
Figure 3.2 Component Diagram ........................................ 54
Figure 3.3.1 Collaboration Diagram – Main Menu ................. 55
Figure 3.3.2 Collaboration Diagram – Personnel Management for Project Leader .................................. 56
Figure 3.3.3 Collaboration Diagram – Browsing Project By Project Leader ................................................. 57
Figure 3.3.4 Collaboration Diagram – Personnel Management for Team Leader ........................................ 57
Figure 3.3.5 Collaboration Diagram – Browsing Project By Team Leader ................................................. 58
Figure 3.3.6 Collaboration Diagram – for Programmer .......... 58
Figure 3.3.7 Collaboration Diagram – for Multi Project Management ......................................................... 59
CHAPTER ONE INTRODUCTION

1.1 Introduction

Recursive multi-threaded (RMT) software life cycle is an object-oriented software life cycle developed by Scott Simon in his Master's thesis[11]. It is based on the thread concept of computer programming where a thread is a light-weight process. All threads share memory information and work concurrently. A thread can also create its own child threads. When a child thread finished its job, it passes results back to its parents. At the end of the process the root thread will collect all results, which becomes the final result. The recursive concept makes threads reusable. Each thread represents a subgoal of the overall project. When the programmer achieves the goal, the thread will aim to finish the next subgoal until the programmer fulfills the requirements. In the current version of RMT, programmers are responsible for progress measuring. The architectural design is created by the project leader and team leaders. Programmers are given the amount of tasks that they need to perform. Therefore, as an assumption, programmers are in the best position to know exactly what their progress is by comparing their work with the design that was assigned to the programmer. The Personal Software Process (PSP)[5] needs to be implemented.
for collecting the programmers' progress information. Project managers and team leaders simply collect all measurements from programmers. Then, they compute progress through the weight of threads. By doing this, management can determine the overall progress of the project. The weight of a thread is the percentage of the overall project. The sum of the weights must be one. For example, for Project A, the project leader estimates to complete the project in three threads or iterations. Each iteration may weigh as 30%, 40%, and 30%. Also, the project leader decides that he/she needs to form two teams to accomplish the project. The two team leaders can also make their own respective iterations, too. Let us say both teams decide to finish the first iteration of the overall project in 2 iterations. The weights of the threads for Team A are 40% and 60%, and 30% and 70% for Team B. For each team, the programmers make their decision also. Let Team A consists of two programmers and both programmers will use two iterations to complete the first iteration of the team. The weight of their threads are 50% and 50% for Programmer A and 60% and 40% for Programmer B. Assume Programmer A completes 60% of his/her first iteration and Programmer B is 20% and Team B is 40% complete for the team overall. The current progress of the project is
30% \times (40\% \times (50\% \times 60\% + 60\% \times 20\% ) + 40\%) = 17\%, \text{ See Figure 1.1. In RMT, determining the progress of the project is made easier and current.}

![Diagram](image)

**Figure 1.1 Progress Computation**

**1.2 Purpose of This Project**

The purpose of this project is to implement the Recursive Multi-Threaded Software Life Cycle. Also, the project is to develop a software engineering tool and utilize Internet and Java technologies to provide an effective tracking method to monitor the software development process. The tool stores all information about software development progress. By compiling information, the tool calculates the most recent status of the individual project and gives the estimation of the date of
the project completion. In addition, the tool can store software engineering documents and artifacts for the purpose of maintenance. The software engineer can browse documents online for maintaining and storing information for the project development.

1.3 Project Products

The completed project will deliver the following products in the final phase:

**Implementation of Recursive Multi-Threaded Tool:** This is a software engineering tool that helps to manage the progress of the software development. The tool provides essential information to help project leaders to make proper decisions if the project is behind schedule.

**Project Documentation:** The project documentation provides the best available reference document for the specification, design, implementation, project plan and source lists.
CHAPTER TWO SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Functionality of RMT Tool

In this section, the functionality of RMT tool will be discussed. The functions of the RMT Tool include: project manager, progress manager, progress report (single and multiple), project prediction, and personnel management.

2.1.1 Project Manager

The project manager is a database, which contains information for all projects. Only the project leader has the right to create a new project. All other personnel has only browse access right to projects. If a user has more than one project in progress, the project manager may allow him/her to choose a project to review the progress. All project information should remain in the database for further reference. The project information cannot be removed from the project manager even if the project was terminated or finished.

2.1.2 Progress Manager

The progress manager provides the input function, which allows a user to change the status of the current progress. Also, through the progress manager, the user can modify the number of threads and assign new weights for threads. The progress manager keeps the most current information for the project database.
2.1.3 Single Project Progress Report

The progress report calculates the current progress of a single project. Through the report, the project leader and the team leaders will know the most current progress of the project. They will know if the project is ahead or behind schedule. Also, from the report, they can prevent further delays in the later stages and keep the damage at a minimum. The most current progress should be input by individual programmers through the progress manager. There are two ways to show the current progress: the progress will be presented as the thread bar chart, or the progress will be a text-based report.

2.1.4 Multiple Project Progress Report

In the real world of software engineering, the final product is a collection of projects. The multiple project progress report can determine which projects need the most amount of time. The group of projects that requires the longest time may form the critical path of when the multiple projects will finish. The report is a tree-type graphic. Each node represents a project. The critical path shows on the graph using colored lines on the screen. The nodes' filled up colors represent the progress of the project.
2.1.5 Project Prediction

The prediction function can give an estimate of when the project will be finished. It calculates the current progress and compares it with information from the project database. The result will provide an estimate of a calendar day that the project will be completed. Table 2.1 shows factors of the prediction, which is based on the statistics of a software engineering group in Hewlett-Packard Company[10], that are used in predicting termination date.

<table>
<thead>
<tr>
<th>Development Phase</th>
<th>Percentage of Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Specification and Planning</td>
<td>18%</td>
</tr>
<tr>
<td>Design</td>
<td>19%</td>
</tr>
<tr>
<td>Implementation</td>
<td>34%</td>
</tr>
<tr>
<td>Integration</td>
<td>29%</td>
</tr>
</tbody>
</table>

Table 2.1 Factors of Prediction

2.1.6 Personnel Management

Basically, the personnel database contains information for each project. The information includes name, the position, the login id, and the password. See Table 2.2 for the user level and access privileges.
<table>
<thead>
<tr>
<th>User Level</th>
<th>Access Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>1. Personnel information management</td>
</tr>
<tr>
<td></td>
<td>2. Project assignment</td>
</tr>
<tr>
<td></td>
<td>3. Password change</td>
</tr>
<tr>
<td>Team Leader</td>
<td>1. Project assignment</td>
</tr>
<tr>
<td></td>
<td>2. Password change</td>
</tr>
<tr>
<td>Programmer</td>
<td>1. Password change</td>
</tr>
</tbody>
</table>

Table 2.2 User Level And Access Privileges

Currently, the project schedule is defined by the project leader. It is determined through a check list of things to do which is based on the concepts of the Personal Software Process (PSP).
2.2 System Objective

Table 2.3 shows the RMT Tool functions, which were defined in section 2.1:

<table>
<thead>
<tr>
<th>Function</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project manager</td>
<td>New project are created by project manager; allows personnel to browse the project within their security clearance.</td>
</tr>
<tr>
<td>Progress manager</td>
<td>Provides an input function, which allows users to enter new progress or modify the current information of iterations and weights.</td>
</tr>
<tr>
<td>Single project progress report</td>
<td>Produces progress reports depending on the user security clearance; the progress report will be in bar chart or text format.</td>
</tr>
<tr>
<td>Multiple projects progress report</td>
<td>Produces a tree graphic and determines the critical path in a group of projects.</td>
</tr>
<tr>
<td>Project prediction</td>
<td>Predicts when the project will be delivered; it includes how many days remaining for completing the project and a calendar day of completion.</td>
</tr>
<tr>
<td>Personnel management</td>
<td>Contains information for all projects and personnel included with the project; the database is maintained by project leader; programmers are added from project by the project leader or team-leaders; all other personnel can change their personal password for security reason.</td>
</tr>
</tbody>
</table>

Table 2.3 RMT Tool Objectives
2.3 System Analysis

The RMT tool consists of several components: Java interface, Java Database Connectivity driver, and SQL database engine. In this section, each component will be described briefly. In the RMT tool, all information will be stored in SQL databases. The front end application will be written in Java programming language. All applications and applets will use JDBC gateway to access the databases.

2.3.1 Java

Java is the latest programming language developed by Sun Microsystem Inc.. The goal of Java is to provide a common system development kit (SDK) and a programming language that will run across platforms. Over the Internet, there are various systems providing services. Each platform has its own system development environment. Even in the Unix environment, different vendors have their specially designed Unix systems. They are not compatible with each other. A common language was not a major issue in the past because the operating environment was not complex. However, with the growth of Internet, a common SDK becomes more important. For this reason, Java was developed. Java is an object-oriented programming language. If we follow the object-oriented design, the software is easy to maintain. Also, because Java is independent of
platforms, it is not necessary to rewrite or recompile the same software for each system, which reduces the software development time and cost. This is the most important advantage of using Java. On the other hand, the Java application or the Java applets cannot stand alone, they need a Java viewer or a Java VM, which may slow down the performance of the software.[2,3,6,9]

2.3.2 Java Database Connectivity (JDBC)

The Java Database Connectivity (JDBC) defines a structured interface to Standard Query Language (SQL) database. The JDBC API provides Java developers with a consistent approach to accessing SQL databases that is comparable to existing database development techniques. The JDBC API includes classes for common SQL database constructs such as database connections, SQL statements, and result sets. JDBC database drivers can either be written entirely in Java or they can be implemented using native methods to bridge Java application to existing database access libraries.[7, 9]

2.3.3 SQL Database Engine

The Structured Query Language (SQL) is an industry standard language for database development. Most database products, such as Oracle, Sybase, DB2, FoxPro, and Windows NT SQL Server, supports SQL. Oracle is the most widely
used database in the world. It runs on virtually every kind of computer, from PCs and Macintoshes, to workstations, minicomputers, and mainframes. It functions virtually identical in all of these machines. Therefore, if the database is created on Oracle SQL server then it should not be difficult to port it to different platforms. The Mini-SQL server is a small SQL engine in the market. It provides a scalable database environment for small business. Currently, the Mini-SQL server is running only on Unix operating systems.[1,4,7,8]

2.3.4 Personal Software Process

The personal software process (PSP) is a self-improvement process designed to help the developer control, manage, and improve the development process.[5] It is a structured framework of forms, guidelines, and procedures for developing software. The RMT Tool supports PSP to assist project leaders, team leaders, and programmers measure accurately the progress of the project.

2.3.5 Overall Design of RMT Tool

The RMT Tool is an Internet-based DBMS application. It uses a web browser as the client interface. When a user inquires for information, the browser will send the query to the World Wide Web (WWW) server. After the WWW server receives the request, the query passes to a JDBC driver and
connects to an mSQL server. The JDBC result set will access the database and return the requested information to the user.

On the other hand, when a user wants to update the database, the process is the same as inquiry on the client side. The WWW server passes the updated information to the JDBC driver through a Java applet. Then the SQL statement updates the database. Figure 2.1 is the system design of RMT Tool.
Figure 2.1 System Design of RMT Tool

Browse Client
A web browser

World Wide Web Server

HTML Form
Java Applet

JDBC API

SQL Database Engine

Update Client
A web browser

World Wide Web Server

Java Applet

Result

Request
2.4 Development, Operating, and Maintenance Environment

2.4.1 Development Environment

Development of this product will make use of the following hardware and software:

2.4.1.1 Client Station

1. Sun Classic workstation and Intel based PC
2. Internet Connection
3. Sun Solaris 2.5 or later version.
4. Microsoft Windows 95/98
5. Sun Java Development Kit (JDK) 1.1.5
6. Sun JFC 1.0.2
7. Microsoft Visual Studio
8. Netscape Navigator 4.05 with Java enhanced
9. HTML 3.0
10. Structured Query Language

2.4.1.2 Server Station

1. Intel based PC server
2. Linux Kernel 2.0 or FreeBSD 2.2.x
3. MiniSQL 2.0.3
4. Apache Web Server
5. JDBC for mSQL 1.0.3

2.4.2 Operating Environment

Operation of this product will make use of the following hardware and software
2.4.2.1 Client Station

1. Any operating system that has Internet connectivity
2. A World Wide Web browser with Java VM

2.4.2.2 Server Station

1. Unix operating system.
2. Apache-Httpd
3. mSQL
4. JDBC for mSQL

2.4.3 Maintenance Environment

Maintenance of this product will be performed within the environment as specified above in section 2.4.1.
2.5 External Interface

2.5.1 Authorization Page

User id and password are two input fields. Pressing the [Submit] button will go to the main page. Pressing the [Reset] button will clear userid and password fields.
Recursive multi-threaded software life cycle is an object oriented software life cycle developed by Scott Simon in his Master's thesis. It is based on the thread concept of computer programming where a thread is a light-weight process. All threads share memory information and work concurrently. A thread can also create its own child threads. When a child thread finished their job, they pass results back to their parents. At the end of the process, the root thread will collect all results, which is the final result. The recursive concept makes threads reusable. Each thread represents a subgoal of the overall project. When the programmer achieves the goal, the thread will aim to next subgoal until the programmer fulfills the requirement. In RMT, programmers are responsible for the progress measuring. Because programmers directly implement the project, they are in the best position to know exactly what the progress is. Project leaders and team leaders simply collect all measurements from the programmer. Then, they compute all measurements with the weight of threads. By doing this, all related personnel in management can determine the overall progress of the project. The weight of a thread is the percentage of the overall project. The sum of weights must be one.

XXX... is the user's name who is currently login in. The [New Project] button is only available for project leaders. For the System Administrator, the [Personnel Mgmt] button is the only function available. [Multi-project Mgmt] button is only accessible by the project leader.
2.5.3 Create New Project

New Project

XXXXXXXX login as Project Leader

- Project Name: Project 1
- Number of iterations: 3
- Weight of each iteration: 30; 40; 30;
- Starting date: November 30, 1997

The [Submit] button will submit new project information to the RMT server. The [Reset] button clears the entry space to empty.
2.5.4 Assign Number of Teams

Pressing [Submit] button will send the number of teams to the server. The [Reset] button is going to clear the data entry.
2.5.5 Browse Current Project

The [Select] button confirms your selection and brings up an organization tree to the user. The program also checks if the iteration number is zero, the tool will pop up a window to ask the user to assign the number of iterations and the percentage for each iteration. If the number of teams is zero, another window (Section 2.5.4) will show up to ask the user to define how many teams are needed to accomplish the project.
2.5.6 Assign the Number of Iteration and Percentages

This window is similar to section 2.5.3. The project ID is assigned by the RMT Tool automatically. Number of iterations and percentages are fields that need input from a user. Pressing the [Submit] button will submit the data to the RMT server. Pressing the [Reset] button will clear these two fields. In the percentage field, data must be delimited by semicolon and the sum of data must equal to 100.
The first level is the root which depends on what is the position (project leader, team leader, or programmer) of the user. If the user is a project leader, the root node represents an overall project. The second level is the thread of the iteration. The third level is nodes of next level, which is either team leaders or programers. For the root and leaves, labels are names of nodes' owners. For the second level of the tree, nodes are labeled by thread IDs.
2.5.8 Graphical Thread

The upper left hand corner indicates the thread ID. Each block in the thread is an image map which may bring up an progress update menu or a documentation menu.
The document may be a copy of source code, design diagram, specifications, or manual. It is a file stored on the server.
2.5.10 Progress Update

Progress Update By Project Leader

Requirement Analysis

☐ Problem/System Definition
☐ System Environment
☐ Interface Definition
☐ Human Resource/Engineering Requirement

Submit  Reset

The progress update pages are check lists of things to do. For each positions (project leader, team leader, or programmer), there is a total of 18 different check lists. Each position has six pages which are requirement analysis, planning, analysis, design, implementation, and testing. The user just check the item after they complete the tasks. The RMT Tool converts the check list into a numerical representation of the progress. Appendix B shows the detail of the check list.
### Progress Report

<table>
<thead>
<tr>
<th>XXXXXXXX login as Project Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Name</strong>: Project1</td>
</tr>
<tr>
<td><strong>Current iteration</strong>: 1</td>
</tr>
<tr>
<td><strong>Overall Completion</strong>: 40%</td>
</tr>
<tr>
<td><strong>The project will be completed on XX/XX/XX</strong></td>
</tr>
</tbody>
</table>

This progress report is only available to project leaders. When the project leader clicks on the root node of the organization tree, this window will pop up on the screen.
2.5.12 Personnel Management

This is the main menu for the personnel manager. The [Maintenance] button brings up a maintenance main menu. The menu consist of add and update functions. The [Assign New Project] button brings the menu of assignments, which allows project leaders and team leaders to assign personnel. The [Change Password] button will pop up a password change window. Users must provide both the old and new passwords in order to change passwords. In this menu, project leaders can access all buttons. The [Maintenance] button will be disabled for team leaders and programmers. The programmer can only access [Change Password].
2.5.13 Maintenance Menu

Personnel Management

XXXXXXX login as Project Leader

Add
Update

The [Add] button allows project leaders and system administrator to set up a new personal account. The [Update] button allows the user to query and update the personnel information.
The position field is a pull down menu. The system administrator or project leader can select one of three positions from the preset selections. The password field is obscured by asterisks. Pressing [Submit] button will send data to the RMT server and stored in the database. Pressing [Reset] button will clear all fields and reset position selection to the project leader.
### Assign New Project

<table>
<thead>
<tr>
<th>Select a project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
</tr>
<tr>
<td>Project 1</td>
</tr>
<tr>
<td>Project 2</td>
</tr>
<tr>
<td>Project 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Select a name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny Lin</td>
</tr>
<tr>
<td>Sunny Lin</td>
</tr>
<tr>
<td>Lee Simpson</td>
</tr>
<tr>
<td>Stven Blaha</td>
</tr>
</tbody>
</table>

**Submit**  **RESET**

Both fields are pull down menus. Users can only select from these data sets. Pressing the [Submit] button will assign the project to a new personnel. The [Reset] button will clear all fields and reset to the first record in the data sets.
2.5.16 Change Password

<table>
<thead>
<tr>
<th>Change Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXX login as Project Leader</td>
</tr>
<tr>
<td>User ID: clin</td>
</tr>
<tr>
<td>Old Password: *****</td>
</tr>
<tr>
<td>New Password: *****</td>
</tr>
<tr>
<td>Re-type new Password: *****</td>
</tr>
</tbody>
</table>

Submit  RESET

Pressing the [Submit] button will allow the RMT Tool to change the password. The [Reset] button will clear all fields. The old password is the user's current password. In order to change the password, the new password and the re-type password must exactly match.
Only the project leader has the access privilege of this page. The [New Project Group] button will bring up the new project group screen, which allows the user to create the new group. The [Set Project Group] will bring up set project group screen, which allows project leaders to group projects. The [Project Status Report] button will bring up the tree graph which shows the critical path of multiple projects.
Pressing [Submit] will send group information to
the server. The RMT server will assign an unique ID
to be stored in the database. [Reset] will clear the
data field.
2.5.19 Set Project Group

Set Project Group

Group Name: Grandmother
Project Name: Grandmother
Completion Date: 1/1/1998

Submit  RESET

All data fields are pull-down menus. Users may select information from the data set.
The critical path will be indicated in the graph starting from the start node to the terminal node. All nodes will be filled with color proportionately to the current percentage of its project's progress.
2.6 Functional Requirement

2.6.1 Finite State Diagram Notation

- Start State
- Terminate State
- State
- Transition
Authorization Page allows users to input their ID and password for identify purpose. Project Leader Page, Team Leader Page, and Programmer Page are all main menus. For different access privileges, the menu looks different.
2.6.2.2 Project Leader Page

Project Leader Page is the start and terminate state for the Project Leader function. New Project Page creates a new project. Browse Current Project lists all current projects and can select one for review. Personnel Database is the main menu for personnel management. Multi-Project Manager main menu provides selection of management for multiple projects.
2.6.2.3 Project Leader Browsing

Project Leader Tree shows the overall structure of the project. Team Leader Tree shows structures for each of the teams. Programmer Node represents a programmer’s thread.
Project Leader Thread shows the overall progress in color bar. Progress Report reviews the current progress in text format. Documentation displays related document for references.
The Team Leader Thread shows the progress in color bar. Progress Report reviews the current progress in text format. Documentation displays related document for references.
2.6.2.6 Programmer Browsing

The Programmer Thread shows the progress in color bar. Documentation displays related document for references. Entering New Progress allows programmers to update their most current progress.
Personnel Maintenance allows project leaders to add new personnel information. Update Information allows the project or team leaders to update personnel information. Assign Project allows project managers and team leaders to assign tasks to a personnel. Change Password allows the users to change their own password.
Team Leader Main Page is the start and terminate state for the Team Leader function. Browse Current Project lists all current projects and can select one for review. Team Leader Tree shows the structures of the team. Personnel Database main menu is for personnel management.
Programmer Main Page is the start and terminate state for the Programmer function. Browse Current Project lists all current projects and can select one for review. Programmer Thread shows the progress in color bar. Personnel Database facilitates functions for personnel maintenance.
2.6.2.10 Multiple Project Progress Report

Multiple Project Report Manager allows project leaders to add new project group and set the project group. The status report provides a critical path analysis to the project leader.
2.7 Performance Requirement

2.7.1 Human Engineering

The system uses a Web browser as its interface. Any experienced Internet user could use this product without difficulty. On-line help should be sufficient to guide users through the system.

2.7.2 Progress Prediction

The RMT Tool must consider the difficulty of each iteration because the degree of difficulty for all iterations are not the same. The Tool assigns a degree factor to each phase in iteration [10], which are 18% for specification and planning, 19% for design, 34% for implementation, and 29% for integration, in order to calculate a precise number of percentage of the overall project, see Section 2.1.5.

2.7.3 Portability

The program must follow the standard Java and SQL such that the system can be ported from the current environment to another environment without difficulty.
2.7.4 **Modifiability**

The program must follow the object-oriented design. The RMT Tool will integrate more complex features as the system grows.

2.7.5 **Understandability**

Source code has been produced such that a programmer with intermediate knowledge in Java, HTML, and SQL can read and understand the code. In addition, documents will consist of: Software Requirement Specification, Preliminary Design, Detailed Design, Documented Source Code, and Maintenance Manual.

2.7.6 **Testability**

Each requirement in the Software Requirement Specification has been identified and tested in the final product. The unit test, integration test, and system test will be used.

2.7.7 **Reusability**

Because the system follows the object-oriented design, the object modules are easily reusable.
CHAPTER THREE DESIGN

The development of Recursive Multi-Threaded Tool follows the concept of the Recursive Multi-Threaded Software Life-Cycle. Also, the design follows the methodology of the Unified Modeling Language (UML). The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, making a "blueprint" for construction. By using UML, the core concept of RMT Tool can be visualized. The design is independent of a particular programming languages and development processes. The core concept can be expanded easily for the further development.

3.1 Class Diagram

A class diagram is a graph of class elements connected by their various static relationships. (Note that a 'class' diagram may also contain interfaces, packages, relationships, and even instances. Figure 3.1 shows classes and relationships in the design of the RMT Tool. The Personnel Class is a super class of the project leader, team leader, and programmer. This class provides basic personnel information for the different positions. All personnel are associated with the Project Class by
"work on" relationship. The Project Class is associated with the Thread Class by "iteration" relationship. The Threadlog Class records the thread activities. It is associated with Thread Class by "recursion" relationship. The Project Class is associated with the CPM (Critical Path Measurement) Class by the CPM_Item Class. The Drawing Class is responsible for drawing the tree and threads. This class is associated with project leader, team leader, or programmer by the "browsing" relationship.
Figure 3.1 Class Diagram
3.2 Component Diagram

A Component Diagram is a graph of components connected by dependency relationships. Components may also be connected to components by physical containment representing composition relationships. Figure 3.2 shows the different components and relationships in the design of the RMT Tool.

The RMT Tool consists of three major components: HTML programs, SQL databases, and graphic drawing programs. Each major component is the collection of sub-components. The HTML program provides the user interface for the RMT Tool, which includes Java applets. The SQL databases provide storage for the RMT TOOL, which include CPM, CPM_Item, Thread_log, Thread, personnel, project, team, and programmer. The Drawing programs draws tree graph and thread diagrams, which include Draw_Tree and Draw_Thread.
Figure 3.2: Component Diagram
3.3 Collaboration Diagram

A Collaboration Diagram shows an interaction organized around the objects in the interaction and their links to each other. Unlike a sequence diagram, a collaboration diagram shows the relationships among the object roles. On the other hand, a collaboration diagram does not show time as a separate dimension, so the sequence of messages and the concurrent threads must be determined using sequence numbers.

Figure 3.3.1 Collaboration Diagram - Main Menu
Figure 3.3 shows the Collaboration Diagrams of the RMT Tool. Figure 3.3.1 shows how the RMT Tool determines the user privilege and returns the proper main page.

Figure 3.3.2 shows the activities when the project leader presses the [Personnel Mgmt] button.

Figure 3.3.2  Collaboration Diagram - Personnel Management for Project Leader

Figure 3.3.3 shows the activities when the project leader creates a new project or browses a current project. Figure 3.3.4 shows the activities when the team leader performs personnel management.
Figure 3.3.3 Collaboration Diagram - Browsing Project by Project Leader

Figure 3.3.4 Collaboration Diagram - Personnel Manager for Team Leader
Figure 3.3.5  Collaboration Diagram - Browsing Project by Team Leader

Figure 3.3.5 shows the activities when a team leader works on the personnel management. Figure 3.3.6 shows the activities when a programmer browses a current project and changes the password. Figure 3.3.7 shows the activities of multi-project management.
Figure 3.3.7 Collaboration Diagram - for Multi Project Management
3.4 Sample Physical Design Tables

The following physical database table are the category detail entities for this project. Following relational design of database is for the purpose of demonstrating the capabilities of the RMT Tool to store and retrieve data. The Tool does not allow deletion, modification, or tables join operation of databases. Therefore, the relational design was not normalized. This means the relational design could have anomalies.

Table 3.1 Personnel Profile

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fname</td>
<td>char(15)</td>
<td>User’s first name</td>
</tr>
<tr>
<td>lname</td>
<td>char(15)</td>
<td>User’s last name</td>
</tr>
<tr>
<td>userid</td>
<td>char(10)</td>
<td>User’s login ID</td>
</tr>
<tr>
<td>password</td>
<td>char(10)</td>
<td>User’s login password</td>
</tr>
<tr>
<td>phone</td>
<td>char(12)</td>
<td>User’s contact phone number</td>
</tr>
<tr>
<td>email</td>
<td>char(40)</td>
<td>user’s email address</td>
</tr>
<tr>
<td>position</td>
<td>char(15)</td>
<td>User’s position(Project Leader, Team Leader, or Programmer)</td>
</tr>
</tbody>
</table>
Table 3.2 Project Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>project_name</td>
<td>char(15)</td>
<td>The name of project</td>
</tr>
<tr>
<td>project_id</td>
<td>char(3)</td>
<td>The ID of project, constructed by using first character of project name and the last two characters of project name</td>
</tr>
<tr>
<td>userid</td>
<td>char(10)</td>
<td>The Userid of project leader</td>
</tr>
<tr>
<td>iterations</td>
<td>int</td>
<td>Number of iterations needed for the project</td>
</tr>
<tr>
<td>current_iteration</td>
<td>int</td>
<td>Current iteration number</td>
</tr>
<tr>
<td>percentages</td>
<td>char(15)</td>
<td>Percentage information</td>
</tr>
<tr>
<td>progress</td>
<td>real</td>
<td>Overall progress of the project</td>
</tr>
<tr>
<td>team</td>
<td>int</td>
<td>Number of teams needed for the project</td>
</tr>
<tr>
<td>team_assigned</td>
<td>int</td>
<td>Number of teams that have been assigned with personnel</td>
</tr>
<tr>
<td>start_date</td>
<td>char(28)</td>
<td>The date when the project begin</td>
</tr>
</tbody>
</table>

Table 3.3 Team Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>teamid</td>
<td>char(10)</td>
<td>ID of the team constructed by project_id + project.current_iteration + 'T' + project.team_assigned</td>
</tr>
<tr>
<td>userid</td>
<td>char(10)</td>
<td>Userid of the team leader</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>Number of iteration needed for the team</td>
</tr>
<tr>
<td>current_iteration</td>
<td>int</td>
<td>Current iteration number of the team</td>
</tr>
<tr>
<td>Field Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>programmer_id</td>
<td>char(10)</td>
<td>ID of the programmer constructed by team_id + team.current_iteration + 'P' + team.team_assigned</td>
</tr>
<tr>
<td>iterations</td>
<td>int</td>
<td>Number of iterations needed for the programmer</td>
</tr>
<tr>
<td>current_iteration</td>
<td>int</td>
<td>Current iteration number</td>
</tr>
<tr>
<td>progress</td>
<td>real</td>
<td>Overall progress of the programmer</td>
</tr>
<tr>
<td>percentage</td>
<td>char(15)</td>
<td>Percentage information for the programmer</td>
</tr>
</tbody>
</table>

Table 3.5 Multiple Project Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpm_id</td>
<td>char(3)</td>
<td>ID of the group</td>
</tr>
<tr>
<td>cpm_name</td>
<td>char(15)</td>
<td>Name of the Group</td>
</tr>
</tbody>
</table>

Table 3.6 Multiple Project Item Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpm_id</td>
<td>char(3)</td>
<td>ID of the associated group</td>
</tr>
<tr>
<td>project_id</td>
<td>char(3)</td>
<td>ID of the involved project</td>
</tr>
</tbody>
</table>
item_type char(3) Type of the item. "CPM" represents the project group, and "PRJ" represents the project.

end_date char(28) Completion date of the project.

Table 3.7 Thread Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thread_id</td>
<td>char(10)</td>
<td>ID of the thread</td>
</tr>
<tr>
<td>userid</td>
<td>char(10)</td>
<td>Owner of the thread</td>
</tr>
<tr>
<td>weight</td>
<td>real</td>
<td>Weight of the thread</td>
</tr>
<tr>
<td>progress</td>
<td>real</td>
<td>Overall progress of the thread</td>
</tr>
</tbody>
</table>

Table 3.8 Thread Log Database

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thread_id</td>
<td>char(10)</td>
<td>ID of the thread</td>
</tr>
<tr>
<td>status</td>
<td>char(15)</td>
<td>Phase of the thread</td>
</tr>
<tr>
<td>item1</td>
<td>char(5)</td>
<td>First item of the check list</td>
</tr>
<tr>
<td>item2</td>
<td>char(5)</td>
<td>Second item of the check list</td>
</tr>
<tr>
<td>item3</td>
<td>char(5)</td>
<td>Third item of the check list</td>
</tr>
<tr>
<td>item4</td>
<td>char(5)</td>
<td>Fourth item of the check list</td>
</tr>
<tr>
<td>item5</td>
<td>char(5)</td>
<td>Fifth item of the check list</td>
</tr>
<tr>
<td>item6</td>
<td>char(5)</td>
<td>Sixth item of the check list</td>
</tr>
<tr>
<td>URL</td>
<td>char(255)</td>
<td>URL of the document</td>
</tr>
<tr>
<td>progress</td>
<td>real</td>
<td>Progress of the phase</td>
</tr>
</tbody>
</table>

The current implementation does not allow to delete records from databases. Once the data is inputted, the
information stays in the database forever. The information deletion can be added as a function of the future expansion.
CHAPTER FOUR  SYSTEM VALIDATION

The system validation is the process of conducting critical tests and the results being evaluated against design specifications and intended functionality. The purpose of system validation is to have assurance about the Recursive Multi-Threaded Tool software quality and functionality also. This guarantees system performance reliability.

A series of different schemes are undertaken to test the RMT Tool. In the section that follows, each test scheme is presented in detail.

4.1 Unit Test

Unit test is the initial step in the software testing process. The unit test focuses on the smallest functional unit of design. This smallest functional unit can be a single class, programs, internal procedures, or independent modules in an isolated test environment. In the RMT Tool, the unit test includes some of the following items to check each functional unit and ensure that all the command buttons work as expected.

- Check normal and abnormal program termination
- Verify the handling of all valid input data type
- Verify the handling of error conditions.

While this is an enormous workload, unit testing
generally yields the highest number of detected problems within all testing techniques. A loose and sloppy unit test will cost someone a great deal of time in later stages of integration testing, system testing, and use. The results of the unit test for the RMT Tool are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Forms</th>
<th>Tests Performed</th>
<th>Results</th>
</tr>
</thead>
</table>
| Authorization          | • Verify the handling of data type  
                         • Ensure Submit and Reset button work as expected                                                                                                        | Pass    |
| Main Menu              | • Check the header label appearance  
                         • Ensure all buttons work as expected                                                                                                               | Pass    |
| Create New Project     | • Verify the handling of data type  
                         • Check the header label appearance  
                         • Ensure Submit and Reset button work as expected                                                                                     | Pass    |
| Assign Number of Team  | • Verify the handling of data type  
                         • Check the project Id label appearance  
                         • Ensure Submit and Reset button work as expected                                                                                     | Pass    |
| Browse Current Project | • Check the Choice box appearance  
                         • Verify the handling of selection  
                         • Ensure Select button works as expected                                                                                                  | Pass    |

Table 4.1 Unit Test Results
| Assign Number of Iteration and Percentage | • Verify the handling of data type  
• Check the project id label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Organization | • Check nodes position and drawing  
• Check nodes labeling appearance  
• Ensure image map works as expected | Pass |
| Graphical Thread | • Check thread progress drawing  
• Check phases labeling appearance  
• Ensure image map works as expected | Pass |
| Documentation | • Check URL linkage  
• Check document panel appearance | Pass |
| Progress Update | • Verify the handling of data type  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Progress Report | • Check all labels appearance  
• Check predication of completion | Pass |
| Personnel Management | • Check button labels appearance  
• Ensure all buttons work as expected | Pass |
| Personnel Maintenance Management | • Check button labels appearance  
• Ensure all buttons work as expected | Pass |

Table 4.1 (Continued) Unit Test Results
| Personnel Maintenance | • Verify the handling of data type  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
|-----------------------|-----------------------------------------------------------------|------|
| Assign New Project    | • Check Choice box items appearance  
• Verify the handling selection  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Change Password       | • Verify the handling of data type  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Multi Project Management | • Check button labels appearance  
• Ensure all buttons work as expected | Pass |
| New Project Group     | • Verify the handling of data type  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Set Project Group     | • Verify the handling of data type  
• Check the header label appearance  
• Ensure Submit and Reset button work as expected | Pass |
| Status Report         | • Check nodes labeling and drawing  
• Check the critical path drawing | Pass |

Table 4.1 (Continued) Unit Test Results
4.2 Integration Test

The integration of the RMT Tool take individual units and link with other units of the overall RMT Tool and subsystems. The integrated system is the tested.

<table>
<thead>
<tr>
<th>Components Integrated</th>
<th>Test Performed</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>The RMT Tool forms and Reports</td>
<td>* test the proper linkage between all the forms and report programs</td>
<td>Pass</td>
</tr>
<tr>
<td>The RMT Tool forms and their associated classes</td>
<td>* test the proper linkage between classes and the RMT Tool forms</td>
<td>Pass</td>
</tr>
</tbody>
</table>

Table 4.2 Integration Test Result

4.3 System Test

After the Recursive Multi-Threaded Tool software is completely unit and integration tested, and meets all functional and performance requirements, the final step in the system validation process is the system test. Since the Algorithm 98 Project is an on-going project in the CSCI 455 (Software Engineering) course, it was used for the system testing. Data from the Algorithm 98 Project was inputed into the RMT Tool. The results from the system test provides actual information on how the RMT Tool functions. Feedback from the development teams of the next Algorithm project will help to determine which area needs to be improved and extended. These information will be considered in the next revision of the RMT Tool.
CHAPTER FIVE MAINTENANCE

The step of maintenance is an important process to keep the software up and running. Also, the revision of the RMT Tool is necessary to keep up with current technologies. For the maintenance of Recursive Multi-Threaded Tool, there are two components: the mSQL database and the Java programs. The methods of maintenance are discussed in the following section.

5.1 mSQL Database Maintenance

The mSQL database is stored in Unix server, which is in "/usr/local/msql/msqlodb" directory. In order to perform the maintenance process, the user must have access privileges to the mSQL server. There are two levels of privileges: administrator and user. To grant the access privileges, the mSQL administrator has to update a file called msql.conf. The administrator needs to append the user ids in the mSQL_User, Admin_User fields. Appendix C is shows a sample file of msql.conf.

Two commands are used to manipulate the mSQL server: msqladmin and msql. The msqladmin command is an administrative command which can shutdown or restart the server. The user has to be an administrator to use msqladmin. The msql command is used for database maintenance. The user types the command under the Unix
prompt as the following:

$masql rmt < showcpm.msql

The "showcpm.msql" is an SQL script file, which shows all entries in the multiple project database. Users may write their own SQL script for maintenance. Appendix D shows a sample SQL script.

5.2 Java Programs

The location of the java program is in the directory "/home/httpd/html/rmt". Also, in this directory, there is a "com" directory, which is the place where JDBC-mSQL drivers are stored. In addition, before compiling the program, "swing.jar" must be in the directory as well. To compile the Recursive Multi-Threaded Tool, developers need to type "javac rmt.java" and press "Enter" key, the java compiler will generate all necessary class files. Appendix E shows the directory list for all java programs. The source code of this project is available on the Internet with the following URL:

http://spider.csci.csusb.edu/sources.
CHAPTER SIX  FUTURE DIRECTIONS AND CONCLUSIONS

The Recursive Multi-Threaded Tool as presented in this project, is intended to show the concept of the Recursive Multi-Threaded software life cycle. The RMT Tool is a starting point of many additional and improved implementations of this software engineering tools. One may want to take the concepts presented in this project and extend them by:

- **Creating an Object Design Analyzer:** An object analyzer is a software agent, which provides computation complexity measurements of the objects to project leaders, team leaders, and programmers. The agent compiles all existing information and compares them with the current object. The result is a degree of the complexity of the object. The analyzer can also give a warning message that the objects are becoming too complex. This function can prevent the objects to grow exponentially in their complexity thus making implementation and maintenance very difficult.

- **Creating an Progress Calculator:** In this version of RMT Tool, the progress relies on the input of users. If the user makes a mistake of his/her measurements, the overall progress will be wrong. The intelligent agent can measure the progress based on the
programmer's past progress information. The result of this measurement should be accurate than the input from the personnel.

**Creating a Design Pattern Library:** The design pattern library allows the project leader to drag and drop the pre-designed object pattern to create the design of the object. Another goal of the pattern library is to include a code generator that generates the C++ or Java source according to the design pattern. Using the design pattern, RMT Tool can give an exact complexity measurement to the user. The user can then modify the design before the project goes into the implementation phase.

**Supporting the Programmer PSP:** The programmer PSP (Personal Software Process)[5] records all activities for each programmers. By reviewing the previous activities and designs and using statistical methods, the RMT Tool can predict the size of the object and the time needed to complete the task.

**Redesign of Relational Databases:** The relational databases should allow deletion, modification and tables join operation. The database should be normalized. Pointers to the projects, threads, and other information should automatically update for new
relational databases.

In conclusion, the Recursive Multi-Threaded Tool was developed to implement the Recursive Multi-Threaded software life cycle model. Java, Java Database Connectivity (JDBC), and Structure Query Language (SQL) queries were used to delivering fast, effective, and comprehensive software engineering management. In addition, Web technology provides a platform independent software. Users, who have a Web browser with JDK 1.1 support, can access the RMT Tool any where. RMT Tool provides a flexible computer-aided software engineering tool and an on-line means of monitoring of software progress.

Moreover, the experiences in implementing this project consist of using various technologies which provided valuable knowledge for myself.
<table>
<thead>
<tr>
<th>Appendix A</th>
<th>Glossary</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application programming interface is a set of program library which allows programmers to call and build applications.</td>
</tr>
<tr>
<td>Web Browser</td>
<td>A client program allows users viewing information on Internet.</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface allows Web server interfacing other application programs and extend the services.</td>
</tr>
<tr>
<td>Client</td>
<td>A program requests services.</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common object request broker architecture allows object applications to communicate with one another no matter where they are located or who has designed them.</td>
</tr>
<tr>
<td>Database Engine</td>
<td>A server program allows users accessing databases.</td>
</tr>
<tr>
<td>Internet</td>
<td>A global network connects millions of computers which interchanging their information.</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java database connectivity is a database API for Java.</td>
</tr>
<tr>
<td>ORB</td>
<td>Object request broker is the middleware</td>
</tr>
</tbody>
</table>
that establishes the client-server relationships between objects.

Progress manager A program module in RMT Tool calculates the progress of the project.

Project Leader A person leads the development of the project

Project Manager A program module in RMT Tool provides management tools for projects.

RMT Recursive multi-threaded is a software life cycle developed by Scott Simon.

SDK System development kid is an integrated environment which makes software development easier.

Server A program provides services.

SQL Structured query language is a standard language using in database development.

Team Leader A person is a head of the programmer team.

Java VM Virtual machine is a program interpreter which executes Java programs.

WWW An Internet service integrates FTP, Telnet, Gopher, HTTP protocols, which provides information over Internet.
Appendix B  Check List

- **Project Leader**
  - Requirement Analysis
    Problem/System Definition
    System Environment
    Interface Definition
    Human Resource/Engineering Requirement
  - Planning
    Producing SQA
    Producing SPMP
  - Analysis
    Use Case Diagram
    Sequence Diagram
    Class Diagram
    Component Diagram
    Interaction Diagram
    State Diagram
  - Design
    Architectural Design
• Implementation
  Distributing Object Modules to Team
  Collecting Object Modules from Team
• Testing
  (Wait for) Team level Unit Test
  System Wide Integration Test

• Team Leader
• Requirement Analysis
  Study and Understand SRS
  Specify Team Level Requirement
• Planning
  Study and Understand SQA
  Team Level Development Schedule Based on SPMP
  Organize Programmers Based on SPMP
• Analysis
  Decomposing Object
  Class Diagram
  Component Diagram
  Interaction Diagram
  State Diagram
• Design
  Number of Objects Completed
  Object Relation Definition
• Implementation
  Distributing Object Module to Programmer
  Collecting Object Modules from Programmer
• Testing
  Team level Unit Test Done
  Team Level Integration Test

• **Programmer**

  • Requirement Analysis
    Study and Understand SRS and SQA
    Specify Requirement for Object
  • Planning
    Programmer Level SQA
    Programmer Level Development Schedule based on SPMP
  • Analysis
    Modified Object
    Delete Object
    Added Object
    Reused Object
  • Design
    Object Pseudo Code
    Object Design -- Number of Completed:
• Implementation
  Modified Object
  Delete Object
  Added Object
  Reused Object
• Testing
  Object Unit Test
  Programmer Level Integration Test
Appendix C    Sample of msql.conf

#  # msql.conf - Configuration file for Mini SQL Version 2  #  #----------------------------------------------------------  #  # This file is an example configuration and may require modification to  # suit your needs or your site. The values given are the default values  # and will be used by the software if either this file is missing or a  # specific value is not specified.  #  #----------------------------------------------------------  #  # NOTE : The install directory and all files/directories below it _must_  # be owned by the user specified in the mSQL_User parameter (the  # user msql by default). If you change the mSQL_User value then  # you must 'chown -R' the install directory.  #  #  [general]  

Inst_Dir = /var/local/msql
mSQL_User = clin
Admin_User = clin
Pid_File = %I/msql2d.pid
TCP_Port = 1114
UNIX_Port = %I/msql2.sock

[system]

Msync_Timer = 30
Host_Lookup = True
Read_Only = False

[w3-msql]

Auth_Host = NULL
Footer = True
Force_Private = False
Appendix D  Sample of showcpm.msql

#
# showcpm.msql  - Display all data entries in com table
#
# This file is an example SQL script and may require modification to
# suit your needs or your database.
#
#
SELECT * FROM cpm
\p\g
Appendix E Directory List of Source Codes

- rmt.java main program of RMT Tool
- addProject.java Function of adding new project
- addThread.java Function of creating the new project thread
- browsing.java Function of selecting current project
- rmtTree.java Function of drawing organization trees
- ThreadProcess.java Function handles drawing graphical thread and image map
- plUpdate.java Function updates project thread
- tlUpdate.java Function updates team thread
- prUpdate.java Function updates programmer thread
- showDocument.java Function shows the related documents
- personnel.java Main menu of the Personnel Management
- maintenance.java Menu for personnel maintenance
  - addperson.java Function adds new person
  - update.java Function update personal information
- assignPrj.java Function allows project leaders and team leaders to make project
• changePwd.java  Function changes password.
• cpm.java  Main menu of Multi Project Management
• createCpm.java  Function creates the new project group
• setCpm.java  Function sets up the project group
• cpmMap.java  Function shows the critical path and tree graph
• windowsClosser.java  Function closes the window
Appendix F  Sample Screens

Figure F.1  Login menu
Recursive Multi-Threaded Tool

Ernest Joseph Moran login as Project Leader

Introduction:

Recursive multi-threaded (RMT) software life cycle is an object-oriented software life cycle developed by Scott Simon in his Master's thesis. It is based on the thread concept of computer programming where a thread is a lightweight process. All threads share memory information and work concurrently. A thread can also create its own child threads. When a child thread finished its job, it passes...
<table>
<thead>
<tr>
<th>Project Name:</th>
<th>Algorithm98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations:</td>
<td>2</td>
</tr>
<tr>
<td>Progress Weight:</td>
<td>70;30;</td>
</tr>
<tr>
<td>Start Date:</td>
<td>Mon Oct 12 10:01:42 PDT 1998</td>
</tr>
<tr>
<td>Submit</td>
<td>Reset</td>
</tr>
</tbody>
</table>

Figure F.3 Create New Project
Project ID: A98
Project Leader: jmoran
Teams:

Submit | Reset

Figure F.4 Team Assignment Menu
Figure F.5 Select A Current Project
Figure F.6 Organization Tree
The Thread of A982

Figure F.7 Graphical Thread
Progress Update by Project Leader

Testing

- Team level unit test Done
- Inter-Team Integration Test
- System wide integration test

Submit  Reset

Figure F.8  Progress Check List
Pseudo Code for ScatterCanvas

Figure F.9 Documentation Page
Progress Report for A98

Current Iteration: 2

Current Progress: 0.77316


Figure F.10 Progress Report
Figure F.11  Personnel Management Main Menu
Figure F.12 Personnel Maintenance Menu
Figure F.13  Add New Personnel
Figure F.14 Selection for Updating Personnel Information
Name: Ernest Joseph Moran

Position: Project Leader

Phone: (909)880-5330

EMail: jmoran@csusb.edu

Submit  Reset

Figure F.15  Update Personnel Information
Figure F.16 Project Assignment Menu
Figure F.17   Change Password Menu
Figure F.18  Multi Project Management
Figure F.19    Create New Project Group
Create New Project Group

Group Name: Grandmother

Project Name: Algorithm98

Completion Date: 6 / 1 / 2000

Submit Reset

Figure F.20 Setup Project Group
Select a project group

Group Name

Select

Figure F.21  Select A Project Group For Report
Figure F.22 Project Status Report
Bibliography